

Article

Formation Tracking Control and Obstacle Avoidance of Unicycle-Type Robots Guaranteeing Continuous Velocities

Jose Bernardo Martinez ¹, Hector M. Becerra ^{1,*}  and David Gomez-Gutierrez ^{2,3} 

¹ Centro de Investigación en Matemáticas (CIMAT), Jalisco S/N, Guanajuato, Gto. 36023, Mexico; jose.morales@cimat.mx

² Multi-agent Autonomous Systems Lab, Intel Labs, Intel Tecnología de México, Av. del Bosque 1001, Zapopan, Jalisco 45019, Mexico; david.gomez.g@ieee.org

³ Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Av. General Ramón Corona 2514, Zapopan, Jalisco 45201, Mexico

* Correspondence: hector.becerra@cimat.mx

Abstract: In this paper, we addressed the problem of controlling the position of a group of unicycle-type robots to follow in formation a time-varying reference avoiding obstacles when needed. We propose a kinematic control scheme that, unlike existing methods, is able to simultaneously solve the both tasks involved in the problem, effectively combining control laws devoted to achieve formation tracking and obstacle avoidance. The main contributions of the paper are twofold: first, the advantages of the proposed approach are not all integrated in existing schemes, ours is fully distributed since the formulation is based on consensus including the leader as part of the formation, scalable for a large number of robots, generic to define a desired formation, and it does not require a global coordinate system or a map of the environment. Second, to the authors' knowledge, it is the first time that a distributed formation tracking control is combined with obstacle avoidance to solve both tasks simultaneously using a hierarchical scheme, thus guaranteeing continuous robots velocities in spite of activation/deactivation of the obstacle avoidance task, and stability is proven even in the transition of tasks. The effectiveness of the approach is shown through simulations and experiments with real robots.

Keywords: formation control; consensus; multi-agent system; hierarchical task-based control; unicycle-type robots



Citation: Martinez, J.B.; Becerra, H.M.; Gomez-Gutierrez, D. Formation Tracking Control and Obstacle Avoidance of Unicycle-Type Robots Guaranteeing Continuous Velocities. *Sensors* **2021**, *21*, 4374. <https://doi.org/10.3390/s21134374>

Academic Editors: Carlos J. Pérez Del Pulgar and Riccardo Carotenuto

Received: 31 March 2021

Accepted: 22 June 2021

Published: 26 June 2021

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



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

1. Introduction

Wheeled mobile robots have been one of the most studied robotic platforms due to their applicability and motion capabilities. In the field of multi-agent systems (MAS), control schemes for collaborative wheeled robots have been developed for environmental sensing [1], task allocation for search and rescue [2], coverage control for precision agriculture [3] and object transportation [4], among others. For such applications, different control protocols have been reported in the literature, using centralized methods that depend on a global coordinate system as a first attempt, e.g., [5,6]. In these works, a virtual structure approach is used, where the complete formation is treated as a single rigid body and the motion of the formation is prespecified, i.e., given the trajectory of the formation centroid in global coordinates, the desired motion of each robot is obtained and an individual feedback controller corrects deviations.

Decentralized or distributed control methods are more robust to failures since only local interactions between agents of a network are required [7,8]. Consensus theory has become an essential tool to develop distributed control protocols [9]; the control laws based on consensus rely on the relative state of connected agents (neighbors) in a network. The problem of formation control aims to drive multiple agents to achieve prescribed constraints on their states [10]. In distributed approaches, the formation control is addressed

by using relative measurements without relying on global coordinates. Several works have addressed the formation control for a group of nonholonomic mobile robots as a consensus problem by introducing some constant deviations for the relative states. For instance, distributed control schemes to stabilize a group of robots to achieve the desired formation in both position and orientation simultaneously are proposed in [11,12]. A control scheme that generates circular motion around virtual positions to define the desired formation is proposed in [13].

The formation tracking problem for nonholonomic robots has also been addressed using distributed schemes following the same approach based on consensus [9,14]. The aim is that the group of robots moves synchronized by tracking a reference trajectory, usually time-varying [15]. The leader-following configuration is common in the formation tracking problem, where one of the agents acts as a leader [16] or a virtual leader can be defined [17]. Due to the nonholonomic motion constraints of unicycle-type robots, dynamic feedback linearization of the kinematic model has been used to propose nonsingular control laws for the robot position, for instance, in [18]. This kind of linearization has the issue that all the robots' linear velocities must be non-null. The forward motion of a formation has also been treated in [19] using a two-stage approach. Formation tracking of both position and orientation has been proposed to achieve asymptotic convergence in [20] or globally uniformly ultimately bounded stability in [21]. Other works, as [22,23], have proposed a state transformation to linearize the model of the robots and a distributed observer to estimate the leader's trajectory. Then, each follower tracks the estimated trajectory using an individual controller. All the previously referred works assume that the group of robots moves in an environment without obstacles and that collisions between agents do not occur.

The capability to deal with fixed obstacles and avoid collisions is essential during the motion of a formation of nonholonomic robots [24]. A distributed collision-free formation control scheme for nonholonomic robots based on potential fields is proposed in [25]. The same problem of formation and the heading consensus is addressed in [26] using a hybrid switching control law. In general, both collision and obstacle avoidance capabilities become more important when the whole formation moves tracking the desired trajectory imposed by a leader. In the work [27], a rotational and repulsive force-based approach is proposed, which is evaluated in V-shape and circular formations. The work [28] addresses formation tracking with multiple leaders, where the team of leaders and followers are fully independent, and only the followers have collision avoidance capability. A control law with prescribed transient and steady-state performance is proposed in [29]. The collision avoidance between the followers is introduced as inequality constraints in the prescribed performance controller. These kinds of constraints are typically used in model predictive control to avoid collisions, which has also been used in formation tracking [30], with the disadvantage of the computational cost to solve the optimization problem. Other works, such as [31], focus on maintaining a rigid formation, for instance, a regular polygon, and the formation is scaled to navigate in an environment with obstacles.

The formation tracking control with obstacle avoidance can be seen as a two-task problem. Most of the described literature addresses the problem by switching between two control laws, one dedicated to the formation task and the other committed to avoiding collisions/obstacles. In [32], a multi-region control scheme for formation tracking with obstacle avoidance and connectivity preservation is proposed. In that work, a different control law is applied depending on the detection of an obstacle within concentric regions around each agent. Therefore, only one task/objective is taken into account at each time instant. In robotics, it is known that the task-function approach [33] allows the design of control laws to deal with problems involving several contradictory objectives. Although the task-function approach has been mainly used for the control of robot manipulators [34], there are examples of its application for triangular formation control of 3D vehicles [35] and formation control of unicycle-type robots [36–38]. Nevertheless, all the aforementioned works that have used null-space projectors to solve the formation control of nonholonomic robots are centralized control schemes relying on global measurements. We have presented

preliminary results of a priority task-based distributed formation control (without tracking) to deal with obstacles for holonomic robots in the conference work [39].

In this paper, we propose to tackle the two-tasks problem of distributed formation tracking control and obstacle avoidance for unicycle-type robots, in particular differential-drive robots, in the framework of a hierarchical task-based control approach. Static feedback linearization is used to compute the robot velocities for each robot from the motion of a control point. The distributed formation control is addressed as a consensus problem of virtual agents, related to the real agents' position by desired displacements, such that if the virtual agents achieve consensus, the real agents reach the desired formation. Obstacle avoidance becomes the task with higher priority when an obstacle is within a given security distance. An obstacle can be another agent, such that collisions between agents are avoided, or a fixed object in the environment. Since a security radius is defined for the nearest obstacle to each agent, the method assumes that the obstacles are convex and it is able to deal with polygonal obstacles. The main contributions of this work are the following:

- The proposed control scheme is fully distributed since it is based on consensus, valid for connected communication topologies, scalable for a large number of robots, and generic to define the desired formation. The computation of the robots' velocities depends only on relative information of neighboring agents and local measurements to avoid obstacles. Neither a global coordinate system nor a map of the environment is required. The leader is able to track any smooth time-parameterized reference, also avoiding obstacles if needed. The leader is part of the formation, interacting with all the followers, with the advantage that the leader also adapts its motion if the followers perform avoidance actions. Existing approaches of formation tracking with obstacle avoidance are valid for specific formation shapes [27]; the leaders and followers are independent and only the followers can perform collision avoidance [28,29]; some methods focus on rigid-body formation by scaling it to avoid obstacles [31]; and some approaches are computationally costly [30].
- A task function is proposed for the distributed formation tracking, and it is combined with an obstacle avoidance task using a hierarchical scheme to solve both tasks simultaneously, which to the authors' knowledge has not been done before in a distributed scheme. The proposed scheme guarantees continuity of the robots' control inputs when both tasks transit to be activated or deactivated. The stability of both task function errors is proven even in the transition of tasks. Given that the control law's design is based on the kinematics of a control point for each agent, the proposed control scheme can be easily extended for formation tracking of holonomic agents as UAVs. Existing approaches that are based on the hierarchical task-based approach tackle only the formation control (without tracking) [39] in a centralized fashion [36–38]. Moreover, most of the formation schemes with obstacle avoidance are discontinuous switching approaches, e.g., [24,26,32], or require a careful tuning of the weight of the different terms of both tasks [25,27].
- The proposed approach is evaluated through realistic simulations using a dynamic simulator in environments with obstacles and in real experiments with differential-drive robots. Few works on distributed formation control of nonholonomic robots have reported results in real robots, e.g., [5,13,22], but none of them solve the formation tracking with obstacle avoidance.

As a summary, we highlight some advantages of the proposed scheme with respect to different existing approaches. One classical approach is the artificial potential-based approach, which is used in [25]. With respect to that work, our approach does not require to predefine the desired position or trajectory of the whole team of robots, nor to know a map of the environment to build the potential functions. Our control law is not a direct sum of controllers for each task that might cancel in the case of conflicting tasks, but a convex combination of control laws for each task is proposed in our control scheme. Another kind of approach is the formulation of a consensus problem to solve the formation problem as in [26]. We also formulate the formation control as a consensus problem, but we extend

the formulation to formation tracking where the group of robots is guided by a leader. In our case, heading consensus is naturally achieved because the group of nonholonomic robots follows a common trajectory. The main advantage of our approach is that we integrate the obstacle avoidance task using priorities and smooth transitions to avoid a hybrid switching control law, guaranteeing stability of both tasks and continuity of the robot velocities. A third kind of approach is the one based on optimization methods, as the model predictive control in [30]. Unlike that work, our approach relies on consensus using relative information between agents and is not needed to precompute the reference trajectory of each follower from the one of the leader. We provide a formal stability proof for both formation and obstacle avoidance tasks, which is not trivial in an optimization-based formulation. We also use feedback linearization of a control point, which has no singularity issues in comparison to a dynamic extension where the control law becomes singular when the robot has null translational velocity. Besides, the MPC-based approach implicitly has a computational cost to solve a quadratic programming problem in a preview horizon. In the same vein, works as [40], that uses distributed optimization for a split and merge strategy of a formation in dynamic environments, suffer from the computational burden. In that strategy, the robots compute locally optimal parameters for the formation to remain within the convex neighborhood of the robots via sequential convex programming. A low level local planner is used to generate the individual robot motion. With respect to that work, the novelty of our approach is that it directly provides the robot velocities without relying on a high level planning strategy. The control scheme presented in [41] is close to ours, however, that work addresses the rendezvous problem in environments with obstacles unlike the formation tracking that we tackle. Besides that important difference, the referred approach consists of a high level strategy that is executed relying on a low level control law and ours directly provides the robot velocities. Moreover, the novelty of our approach with respect to that method is the guarantee of continuity of the robot velocities, which is not the case of the referred work since sudden changes of orientation might occur in the robot trajectories. We consider the generation of smooth velocities important to improve the accuracy of the position control when the robots are modeled at kinematic level. Due to robot and actuator dynamics, discontinuous velocity commands cannot be exactly realized by a vehicle. Moreover, the transient to converge to the new command may lead to collisions. Sudden fast motions are not convenient in practice since the sensing capabilities of the robots might deteriorate, in our case the distance measurement might introduce noise for sudden motions.

The rest of the paper is structured as follows. Section 2 describes the model of a unicycle-type robot, and some considerations to control it. It also recalls some results on graph theory and the task-function approach. In Section 3, the problem of formation tracking for differential-drive robots with obstacle avoidance is formulated, and the proposed control scheme is detailed. Section 4 presents the performance of the proposed approach through simulations and real experiments. Finally, some concluding remarks and future work ideas are presented in Section 5.

2. Preliminaries

In this section, we will present the model considered for each agent of the network of robots and the basics on graph and consensus theory that are used in the control law's design.

2.1. Model of the Robots

Let $\mathcal{N} = \{R_1, R_2, \dots, R_N\}$ be a set of unicycle-type robots, in particular differential-drive robots (DDRs), moving on the xy -plane with pose $\mathbf{q}_i = [x_i, y_i, \theta_i]^T, i \in \{1, \dots, N\}$. The kinematic model for each robot, according to Figure 1, is given by:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & 0 \\ \sin(\theta_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad i \in \{1, \dots, N\}, \quad (1)$$

where $[x_i, y_i]$ are the coordinates of the position of the center of rotation of the i -th robot, θ_i the orientation with respect to the x -axis, v_i is the translational velocity and ω_i the angular velocity.

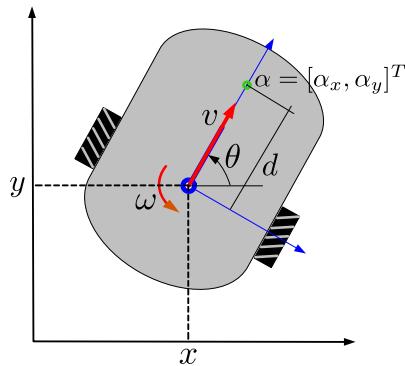


Figure 1. Scheme of a differential-drive mobile robot and definition of the control point α .

We are interested that the team of robots can achieve formation just for position, not orientation. It is known that is not possible to control the robot position $\mathbf{p}_i = [x_i, y_i]^T$, seen as output vector, using an static feedback linearization [42]. Then, we will consider α_i as control point, which is a point out of the wheels axis as in Figure 1. For simplicity, the point α_i is chosen shifted by a distance $d > 0$ along the longitudinal axis passing over point \mathbf{p}_i of the robot. The distance d is such that the point α_i is inside the robot periphery. The coordinates of the point α_i are given by

$$\alpha_i = \begin{bmatrix} \alpha_{x_i} \\ \alpha_{y_i} \end{bmatrix} = \begin{bmatrix} x_i + d \cos(\theta_i) \\ y_i + d \sin(\theta_i) \end{bmatrix}. \quad (2)$$

The time-derivative of the point α_i (2) is as follows

$$\begin{bmatrix} \dot{\alpha}_{x_i} \\ \dot{\alpha}_{y_i} \end{bmatrix} = \mathbf{M}_i(\theta_i) \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad (3)$$

where the decoupling matrix $\mathbf{M}_i(\theta_i)$ is given by

$$\mathbf{M}_i(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -d \sin(\theta_i) \\ \sin(\theta_i) & d \cos(\theta_i) \end{bmatrix}, \quad (4)$$

and for each robot R_i is not singular, since $\det(\mathbf{M}_i(\theta_i)) = d \neq 0$. Then, the control point coordinates can be driven as desired by the robot velocities and the orientation angle remains as zero-dynamics of the system (1). However, it is known that such dynamics is stable when the control point is driven to a constant position or tracks a time-varying reference [42].

2.2. Graph and Consensus Theory

In this section, we present some notations and preliminaries about graph and consensus theory that is extensively presented, for instance, in [7,9].

In a network of agents, consensus means to reach an agreement on a certain quantity of interest that depends on the state of all agents in the network. In consensus theory, a network of agents is typically modeled as a graph \mathcal{G} , which consists of a vertex set $\mathcal{V}(\mathcal{G})$ and an edge set $\mathcal{E}(\mathcal{G})$. An edge is denoted by ij and $j \sim i$ denotes that the vertex i and vertex j are neighbors. The neighbors of vertex i in a graph \mathcal{G} is denoted by $\mathcal{N}_i(\mathcal{G}) = \{j : ji \in \mathcal{E}(\mathcal{G})\}$.

The adjacency matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ of a graph with N vertices is a square matrix with entries a_{ij} corresponding to the weight of the edge ij ; when i is not adjacent to j then

$a_{ij} = 0$. The Laplacian matrix of \mathcal{G} is $\mathbf{L}(\mathcal{G}) = \Delta - A$ where $\Delta = \text{diag}(d_1, \dots, d_N)$ with $d_i = \sum_{j=1}^N a_{ij}$.

A graph \mathcal{G} is connected if there is a path between any two vertices, otherwise it is disconnected. If the graph \mathcal{G} is connected, then the Laplacian matrix has an eigenvalue $\lambda_1(\mathbf{L}) = 0$ with algebraic multiplicity one, associated to the eigenvector $\mathbf{1}_N = [1 \dots 1]^T$, i.e., $\ker(\mathbf{L}(\mathcal{G})) = \{\mathbf{x} : x_1 = \dots = x_N\}$. For undirected graphs the Laplacian matrix \mathbf{L} is positive semidefinite and symmetric.

In this work, we will consider a multi-agent system (MAS) composed of N differential-drive robots (DDRs) modeled as in the previous section and with an associated graph \mathcal{G} modeling the network's connectivity. Given the expression of the dynamics to be controlled (3), it can be treated as decoupled single-integrator dynamics expressed in general form as

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i, \quad i \in \{1, \dots, N\}, \quad (5)$$

where $\mathbf{x}_i, \mathbf{u}_i(t) \in \mathbb{R}^2$ are the state and auxiliary control input of agent i , respectively. The network dynamics for the agents (5) can be written in vector form as

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t), \quad (6)$$

with $\mathbf{x}(t) = [\mathbf{x}_1(t)^T, \dots, \mathbf{x}_N(t)^T]^T \in \mathbb{R}^{2N}$ and $\mathbf{u}(t) = [\mathbf{u}_1(t)^T, \dots, \mathbf{u}_N(t)^T]^T \in \mathbb{R}^{2N}$.

According to [7], the consensus error for agent i with respect to its neighbors is defined as

$$\mathbf{e}_{ci}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(\mathbf{x}_j(t) - \mathbf{x}_i(t)), \quad i \in \{1, \dots, N\}. \quad (7)$$

If this error is zero for all the agents, it is accomplished that $\mathbf{x}_i = \mathbf{x}_j$ for all $i, j \in \mathcal{E}(\mathcal{G})$, $i \neq j$. The individual consensus error (7) can be expressed in matrix form as

$$\mathbf{e}_c(t) = [\mathbf{e}_{c1}(t)^T, \dots, \mathbf{e}_{cN}(t)^T]^T = -(\mathbf{L} \otimes \mathbf{I}_2)\mathbf{x}(t) \in \mathbb{R}^{2N}, \quad (8)$$

where \otimes denotes the Kronecker product and \mathbf{I}_2 is the identity matrix of size 2.

2.3. Hierarchical Task-Based Control Approach

Let us consider one agent of the form (5) with state $\mathbf{q} \in \mathbb{R}^n$ and a differential mapping, denoted by $\mathcal{X}(\mathbf{q}) \in \mathbb{R}^m$, between a task space and the state space, which models a task to be performed by the agent. When addressing a regulation problem not a tracking one, it is desired that the task reaches a constant value \mathcal{X}^d by enforcing the convergence to zero of the error function

$$\mathbf{e}(\mathbf{q}) = \mathcal{X}(\mathbf{q}) - \mathcal{X}^d \in \mathbb{R}^m. \quad (9)$$

The time-derivative of (9) is given by

$$\dot{\mathbf{e}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (10)$$

being $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ a Jacobian matrix. In (10), $\dot{\mathbf{q}}$ can be seen as the vector of control inputs, and to find it out, the Moore–Penrose pseudoinverse of the Jacobian matrix can be used as follows

$$\dot{\mathbf{q}} = \mathbf{J}^+(\mathbf{q})\dot{\mathbf{e}}, \quad (11)$$

with $\mathbf{J}^+(\mathbf{q}) = \mathbf{J}^T(\mathbf{q})(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))^{-1} \in \mathbb{R}^{n \times m}$. The task space control approach is valid if the dimension of the task space is less than or equal to the dimension of the state space of the agent $m \leq n$ [43]. The desired dynamics of the task error function is as follows

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \quad (12)$$

with $\lambda > 0$, such that exponential convergence of the task function to its desired value is guaranteed by using the vector of control inputs

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}^+(\mathbf{q}) \mathbf{e}(\mathbf{q}). \quad (13)$$

Since we address the problem that a group of DDRs reaches a desired formation avoiding collisions/obstacles when needed, two tasks must be carried out simultaneously. The hierarchical task-based approach introduced in [33] allows two tasks to be solved as good as possible, in the sense that the solution of the task with higher priority is guaranteed and the solution of the task with lower priority is subject to the former [36,44,45].

The hierarchical task-space approach requires the formulation of some projectors. Following [44], the null space projector of the i -th task, considered the priority task, is computed as

$$\mathbf{N}_i = \mathbf{I}_n - \mathbf{J}_i^+(\mathbf{q}) \mathbf{J}_i(\mathbf{q}) \in \mathbb{R}^{n \times n}. \quad (14)$$

As an example, consider two tasks \mathcal{X}_1 and \mathcal{X}_2 , where \mathcal{X}_1 has the highest priority. According [36,44], the total control action is given by

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \mathbf{N}_1 \dot{\mathbf{q}}_2, \quad (15)$$

where $\dot{\mathbf{q}}_1 = -\lambda_1 \mathbf{J}_1^+(\mathbf{q}) \mathbf{e}_1(\mathbf{q}) \in \mathbb{R}^n$ is the input computed individually from \mathcal{X}_1 and $\dot{\mathbf{q}}_2 = -\lambda_2 \mathbf{J}_2^+(\mathbf{q}) \mathbf{e}_2(\mathbf{q}) \in \mathbb{R}^n$ is the input obtained from \mathcal{X}_2 .

It is common that initially a system only has to solve one task by using the control inputs vector (13), and in some moment the system requires to solve more than one task. Then, in the case of two tasks the control inputs vector must switch from (13) to (15) and eventually the system returns to solve only the initial task.

3. Problem Formulation and Proposed Solution

Let \mathcal{N} be a set of DDRs as described in Section 2.1. Consider that there exists a finite set of obstacles in the environment. The agents themselves can be obstacles for the others or there may be fixed obstacles in the environment.

The problem addressed in this work is to design a control law for each agent $\mathbf{u}_i = \gamma(\alpha_i, \mathcal{N}_i)$, $i = \{1, \dots, N\}$ to achieve:

- Asymptotic tracking of a predefined smooth trajectory $\mathbf{m}(t)$ by the leader, for instance, considered as the robot R_1

$$\lim_{t \rightarrow \infty} (\alpha_1(t) - \mathbf{m}(t)) = 0, \quad (16)$$

where $\alpha_1(t)$ is the position of the control point of the leader.

- Asymptotic convergence to a desired formation by the follower agents, i.e.,

$$\lim_{t \rightarrow \infty} a_{ij}(\alpha_i(t) - \alpha_j(t)) = \Delta_{ij} = \text{const.}, \quad i, j \in \{1, \dots, N\}, i \neq j, \quad (17)$$

where Δ_{ij} denotes desired constant offsets between agents to make that the desired formation has a fixed-predefined form.

- Obstacle avoidance between agents and with respect to any fixed object in the environment, always maintaining a safety distance σ , i.e., the condition

$$\|\alpha_i - \alpha_o^i\| > \sigma \quad (18)$$

must be accomplished, where α_o^i is the position of the nearest obstacle to the i -th agent.

To solve the enunciated problem, the following assumptions are considered:

- Each robot has omnidirectional capability to measure distance and only the nearest obstacle is considered to avoid it at a time.

- The obstacles are considered convex since a circumference that must not be trespassed is defined around the nearest point on an obstacle from each robot.
- The communication topology for the group of robots is considered connected.
- The robots velocities are available for their neighbors and they are communicated according to the communication topology.

The described problem requires to solve two tasks, the formation tracking and the obstacle avoidance. We will use the hierarchical task-based formulation to guarantee that both tasks are adequately accomplished. Most of the time, the formation tracking task is active, but both task must be solved simultaneously when obstacles are close to an agent. The formulation of each task is presented in the following sections.

3.1. Task of Obstacle Avoidance

Every robot R_i must avoid collisions with obstacles (either fixed obstacles or other agents). This task is activated when $\|\alpha_i - \alpha_o^i\| \leq \sigma$, such that the agents must always maintain a safety distance σ to the obstacles and has priority over the formation tracking. To meet this objective, the task $x_1^i = \rho(\alpha_i)$ is defined as the relative distance from the robot position α_i to the position α_o of the nearest obstacle as

$$\rho(\alpha_i) = \|\alpha_i - \alpha_o^i\| \in \mathbb{R}. \quad (19)$$

Given the assumption of sensing capabilities of the robots, the distance $\rho(\alpha_i)$ can be measured by each robot R_i and thus, the global position of the obstacle is not required. Thus, the following error function is defined

$$e_{o_i} = \rho(\alpha_i) - \sigma \in \mathbb{R}. \quad (20)$$

The dynamics of the error is given by:

$$\dot{e}_{o_i} = \mathbf{J}_{o_i}(\alpha_i)\dot{\alpha}_i, \quad (21)$$

where

$$\mathbf{J}_{o_i}(\alpha_i) = \frac{(\alpha_i - \alpha_o^i)^T}{\rho(\alpha_i)} \in \mathbb{R}^{1 \times 2}. \quad (22)$$

As desired dynamics for (21) we want to have

$$\dot{e}_{o_i} = -\lambda e_{o_i}, \quad (23)$$

with $\lambda > 0$. Solving for α_i , the feedback control law for obstacle avoidance is given by

$$\dot{\alpha}_i = -\lambda \mathbf{J}_{o_i}^+(\alpha_i) e_{o_i}, \quad (24)$$

where $\mathbf{J}_{o_i}^+(\alpha_i) = \mathbf{J}_{o_i}^T(\alpha_i)(\mathbf{J}_{o_i}(\alpha_i)\mathbf{J}_{o_i}^T(\alpha_i))^{-1}$.

3.2. Task of Agents' Formation

The formation control problem is addressed in this work as a consensus problem in order to achieve a distributed and scalable formulation. Consider a set of N DDRs connected through a communication network such that each one exchanges information with a set of neighbors. We will consider dynamics for each DDR as in (3)

$$\dot{\alpha}_i = \mathbf{M}_i(\theta_i)\mathbf{u}_i(t), \quad i \in \{1, \dots, N\}. \quad (25)$$

Agents' formation can be achieved by consensus of a virtual network [26]. We will specify the desired formation as a set of fixed translation vectors $\mathbf{z}_i \in \mathbb{R}^2$ with respect to an

arbitrary common reference frame, thus the position α_i of the i -th agent is related to the position α_{v_i} of the virtual agent by

$$\alpha_{v_i} = \alpha_i + \mathbf{z}_i, \in \mathbb{R}^2, \quad i \in \{1, \dots, N\}, \quad (26)$$

such that if the virtual agents reach the same value, i.e., $\alpha_{v_i}^* = \alpha_v^* \in \mathbb{R}^2, i \in \{1, \dots, N\}$, then the real agents will reach the desired formation given by the vectors \mathbf{z}_i . Notice that the network of virtual agents has the same Laplacian matrix than the original network and the virtual agent's dynamics is given by

$$\dot{\alpha}_{v_i} = \dot{\alpha}_i = \mathbf{M}_i(\theta_i)\mathbf{u}_i(t). \quad (27)$$

The consensus error for the virtual agent i with respect to its neighbors is

$$\mathbf{e}_{v_i} = \sum_{j \in \mathcal{N}_i} (a_{ij}(\alpha_{v_j} - \alpha_{v_i})) \in \mathbb{R}^2, \quad (28)$$

and the consensus error vector is

$$\mathbf{e}_v = [\mathbf{e}_{v_1}^T, \mathbf{e}_{v_2}^T, \dots, \mathbf{e}_{v_N}^T]^T = -(\mathbf{L} \otimes \mathbf{I}_2)\alpha_v \in \mathbb{R}^{2N}, \quad (29)$$

with $\alpha_v = [\alpha_{v_1}^T, \alpha_{v_2}^T, \dots, \alpha_{v_N}^T]^T$. Applying control inputs of the form $\mathbf{u}_i(t) = \mathbf{M}_i^{-1}(\theta_i)\mathbf{e}_{v_i}$ to the systems (25), the virtual agents will reach consensus and consequently the real agents will reach the desired formation specified by the vectors \mathbf{z}_i .

As consensus can be reached for the virtual agents' state, such that each virtual agent reaches the position α_v^* , this allows us to formulate a secondary task $\mathbf{x}_2^i = \alpha_{v_i}$ in the hierarchical task-based approach to solve the formation control with obstacle avoidance. Thus, the error of the consensus task (29) can be rewritten as

$$\mathbf{e}_v = \alpha_v - (\mathbf{1}_N \otimes \alpha_v^*) \in \mathbb{R}^{2N}. \quad (30)$$

Thus, the corresponding time-derivative of the consensus task is given by

$$\dot{\mathbf{e}}_v = \mathbf{J}_v(\alpha_v)\dot{\alpha}_v, \quad (31)$$

where $\mathbf{J}_v(\alpha_v) = \mathbf{I}_N \otimes \mathbf{J}_{v_i}(\alpha_{v_i})$. Since $\mathbf{J}_{v_i}(\alpha_{v_i}) = \mathbf{I}_2$, then the whole Jacobian matrix of the consensus task is $\mathbf{J}_v(\alpha_v) = \mathbf{I}_{2N} \in \mathbb{R}^{2N \times 2N}$. In the sequel, we will not denote the dependence of the Jacobians from α_v . In the following sections, we will use this formulation of consensus of virtual agents to propose a formation tracking control law for a group of DRRs.

3.2.1. Formation Tracking Control Law

Let us address the problem where a leader robot R_1 have to track a predefined time-varying trajectory $\mathbf{m}(t)$ and the followers must follow the leader in formation (equivalently the virtual agents follow the leader in consensus), first assuming that no avoidance actions are required. The following distributed control protocol is proposed to solve this problem:

$$\mathbf{u}_1 = \mathbf{M}_1^{-1}(\theta_1) \left[-\gamma(\alpha_1 - \mathbf{m}) + \dot{\mathbf{m}} - k \sum_{j \in \mathcal{N}_1} a_{1j}(\alpha_{v_1} - \alpha_{v_j}) \right], \quad (32)$$

$$\mathbf{u}_i = \mathbf{M}_i^{-1}(\theta_i) \left\{ \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij} [\dot{\alpha}_j - k(\alpha_{v_i} - \alpha_{v_j})] \right\}, \quad i \in \{2, \dots, N\}, \quad (33)$$

where $\mathbf{M}_i^{-1}(\theta_i)$ is the inverse of the decoupling matrix associated to each robot, $\dot{\mathbf{m}}$ is the velocity of the desired trajectory, γ is the tracking control gain and k is the formation control gain.

The vector of position error for the whole MAS, seen as a new task function error, is given by

$$\mathbf{e}_r = \boldsymbol{\alpha}_v - \mathbf{r} \in \mathbb{R}^{2N}, \quad (34)$$

where $\mathbf{r} = \left[(\mathbf{m} + \mathbf{z}_1)^T, \frac{1}{|\mathcal{N}_2|} \sum_{j \in \mathcal{N}_2} a_{2j} \boldsymbol{\alpha}_{v_j}^T, \frac{1}{|\mathcal{N}_3|} \sum_{j \in \mathcal{N}_3} a_{3j} \boldsymbol{\alpha}_{v_j}^T, \dots, \frac{1}{|\mathcal{N}_N|} \sum_{j \in \mathcal{N}_N} a_{Nj} \boldsymbol{\alpha}_{v_j}^T \right]^T$ and \mathbf{z}_1 is the relative constant displacement vector assigned to the leader in the formation. The dynamics of the error (34) are given by

$$\dot{\mathbf{e}}_r = \mathbf{J}_r \dot{\boldsymbol{\alpha}}_v - \dot{\mathbf{r}} \in \mathbb{R}^{2N}, \quad (35)$$

with $\dot{\mathbf{r}} = \left[\dot{\mathbf{m}}^T, \frac{1}{|\mathcal{N}_2|} \sum_{j \in \mathcal{N}_2} a_{2j} \dot{\boldsymbol{\alpha}}_{v_j}^T, \frac{1}{|\mathcal{N}_3|} \sum_{j \in \mathcal{N}_3} a_{3j} \dot{\boldsymbol{\alpha}}_{v_j}^T, \dots, \frac{1}{|\mathcal{N}_N|} \sum_{j \in \mathcal{N}_N} a_{Nj} \dot{\boldsymbol{\alpha}}_{v_j}^T \right]^T$ and $\mathbf{J}_r = \mathbf{I}_{2N}$, where for each agent $\mathbf{J}_{r_i} = \mathbf{I}_2$.

Since the velocities of the virtual agents depend directly on the control input and the decoupling matrix, we rewrite (35) as

$$\dot{\mathbf{e}}_r = \mathbf{M}(\boldsymbol{\theta}) \mathbf{u} - \dot{\mathbf{r}}, \quad (36)$$

with $\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_N^T]^T$ and $\mathbf{M}(\boldsymbol{\theta}) = \text{diag}(\mathbf{M}_1(\theta_1), \dots, \mathbf{M}_N(\theta_N))$.

Theorem 1. Consider an MAS of N unicycle-type robots with position dynamics of the control point as in (3), under a connected communication graph \mathcal{G} , and related to the virtual agents' position according to (26). The leader robot knows a continuously differentiable reference trajectory $\mathbf{m}(t)$. Then, for control parameters $\gamma > 0$ and $k > 0$ of the distributed protocol (32) and (33), each component of the error function \mathbf{e}_r (34) converges asymptotically to zero. Therefore, the leader's position converges to the desired trajectory and the followers converge to the specified formation around the leader.

Proof. First, we rewrite the control inputs (32) and (33) in terms of the errors as follows

- For the leader

$$\mathbf{u}_1 = \mathbf{M}_1^{-1}(\theta_1) \left[-\gamma \mathbf{e}_{r_1} + \dot{\mathbf{m}} - k \sum_{j \in \mathcal{N}_1} a_{1j} (\boldsymbol{\alpha}_{v_1} - \boldsymbol{\alpha}_{v_j}) \right].$$

- For the followers

$$\mathbf{u}_i = \mathbf{M}_i^{-1}(\theta_i) \left[\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij} \dot{\boldsymbol{\alpha}}_{v_j} - k \mathbf{e}_{r_i} \right] \quad i \in \{2, \dots, N\}.$$

Then, the control inputs in vector form are given by

$$\mathbf{u} = \mathbf{M}^{-1}(\boldsymbol{\theta}) [-\mathbf{K} \mathbf{e}_r + \dot{\mathbf{r}} - \mathbf{d}], \quad (37)$$

where $\mathbf{K} = \text{diag}(\gamma, \gamma, k, k, k, \dots, k) \in \mathbb{R}^{2N \times 2N}$ and $\mathbf{d} = [k \sum_{j \in \mathcal{N}_1} a_{1j} (\boldsymbol{\alpha}_{v_1} - \boldsymbol{\alpha}_{v_j})^T, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0}]^T \in \mathbb{R}^{2N}$.

By introducing the control input (37) in the dynamics of the system errors (36) to get the closed loop dynamics, it is obtained

$$\dot{\mathbf{e}}_r = -\mathbf{K} \mathbf{e}_r - \mathbf{d}. \quad (38)$$

By expanding (38) we obtain the error dynamics of the leader and the followers

$$\dot{\mathbf{e}}_{r_1} = -\gamma \mathbf{e}_{r_1} + k \sum_{j \in \mathcal{N}_1} a_{1j} (\boldsymbol{\alpha}_{v_j} - \boldsymbol{\alpha}_{v_1}), \quad (39)$$

$$\dot{\mathbf{e}}_{r_i} = -k \mathbf{e}_{r_i} \quad i \in \{2, \dots, N\}. \quad (40)$$

On the one hand, the second term of the leader's error dynamics (39) is the consensus error of the leader with respect to its neighbor followers, i.e., \mathbf{e}_{v_1} of (29). On the other hand, the dynamics of (40) that represents the consensus errors for the agents $\{2, \dots, N\}$ clearly have asymptotic stability to zero. Then, according (34) and (29), at steady state we have

$$\mathbf{e}_{r_i} = \mathbf{e}_{v_i} = \boldsymbol{\alpha}_{v_i} - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij}(\boldsymbol{\alpha}_{v_j}) = 0 \quad i \in \{2, \dots, N\}. \quad (41)$$

Since \mathbf{e}_{v_1} depends on the first row of the Laplacian matrix as given by (29) and that row is linearly dependent on the rest of rows of the matrix, $\mathbf{e}_{v_i} = 0$ for $i \in \{2, \dots, N\}$ implies that $\mathbf{e}_{v_1} = 0$. Therefore, the second term of (39) vanishes as the time elapses and also the tracking error \mathbf{e}_{r_1} converges asymptotically to zero. Then the virtual followers achieve consensus, and consequently, the real robots achieve the desired formation defined by the displacement vectors \mathbf{z}_i according (26), tracking the leader in formation. \square

The previous theorem demonstrates the stability of the multi-agent system in closed loop with the control protocol that solves the formation tracking task. In the next section, this result will be integrated into a protocol that also includes the obstacle avoidance task.

3.3. Hierarchical Combination of Formation Tracking and Obstacle Avoidance Tasks

The problem formulated in Section 3 can be addressed by following the task-based approach described in Section 2.3. When an obstacle is detected, an avoidance action must be performed with higher priority over the formation tracking, but taking into account both tasks simultaneously. However, the direct switching between the formation tracking control law ((32) and (33)) and a control law combining both tasks generates discontinuous robot velocities. To alleviate this issue, we propose the use of a smooth transition as presented in [45]. Then, the velocities for each robot that combines the obstacle avoidance task, denoted by subscript o , as primary task, and DDRs formation tracking, denoted by subscript r , as secondary task is given by

$$\mathbf{u}_i = \mathbf{M}_i^{-1}(\theta_i)(\dot{\boldsymbol{\alpha}}'_{o_i} + \dot{\boldsymbol{\alpha}}_{r|o_i}), \quad i \in \{1, \dots, N\}, \quad (42)$$

where

$$\begin{aligned} \dot{\boldsymbol{\alpha}}'_{o_i} &= \mathbf{J}_{o_i}^+ \dot{e}'_{o_i}, \\ \dot{\boldsymbol{\alpha}}_{r|o_i} &= (\mathbf{J}_{r_i} \mathbf{N}_{o_i})^+ (\dot{\mathbf{e}}_{r_i} + \dot{\mathbf{r}}_i - \mathbf{J}_{r_i} \mathbf{J}_{o_i}^+ \dot{e}'_{o_i}), \\ \dot{e}'_{o_i} &= h(t) \dot{e}_{o_i} + (1 - h(t)) \mathbf{J}_{o_i} \mathbf{J}_{r_i}^+ \dot{\mathbf{e}}_{r_i}, \end{aligned}$$

with $\mathbf{N}_{o_i} = \mathbf{I}_2 - \mathbf{J}_{o_i}^+ \mathbf{J}_{o_i} \in \mathbb{R}^{2 \times 2}$, \mathbf{J}_{o_i} as in (22) and $\mathbf{J}_{r_i} = \mathbf{I}_2$ as stated below of (35), \dot{e}_{o_i} as in (23) and $\dot{\mathbf{e}}_{r_i}$ as in (39) and (40). The transition function $h(t)$ is a smooth bounded function such that $0 \leq h(t) \leq 1$ and increases from 0 to 1 (activated) when an obstacle is detected within the safety distance σ . Its value is $h(t) = 1$ while the obstacle is within the safety distance and decreases from 1 to 0 (deactivated) when the obstacle leaves the safety distance. The duration of the transition function is fixed by the user and is the same value for both activation and deactivation. Thus, the use of the transition function yields continuity of the computed robots velocities. In the sequel we will denote $h(t) = h$.

In the following theorem, the stability of the control law for both obstacle avoidance and formation tracking tasks is proven. All the expressions in the theorem are presented in vector form, however, the approach keeps distributed as given by the individual control law (42) for each agent.

Theorem 2. Consider an MAS of N unicycle-type robots with position dynamics of the control point as in (3), under a connected communication graph \mathcal{G} , and related to the virtual agents' position according to (26). The leader robot knows a continuously differentiable reference trajectory $\mathbf{m}(t)$.

Then, for control parameters $\gamma > 0$, $k > 0$ and $\lambda > 0$ of the distributed protocol expressed in vector form

$$\mathbf{u} = \mathbf{M}^{-1}(\boldsymbol{\theta})(\dot{\alpha}'_o + \dot{\alpha}_{r|o}), \quad (43)$$

where $\dot{\alpha}'_o = \mathbf{J}_o^+ \dot{\mathbf{e}}'_o$, $\dot{\alpha}_{r|o} = (\mathbf{J}_r \mathbf{N}_o)^+(\dot{\mathbf{e}}_r + \dot{\mathbf{r}} - \mathbf{J}_r \mathbf{J}_o^+ \dot{\mathbf{e}}'_o)$, $\dot{\mathbf{e}}'_o = h \dot{\mathbf{e}}_o + (1-h) \mathbf{J}_o \mathbf{J}_r^+ \dot{\mathbf{e}}_r$, both the formation tracking error \mathbf{e}_r and the obstacle avoidance error \mathbf{e}_o convergence to zero asymptotically when both tasks are active, i.e., with $0 < h \leq 1$. The terms of (43) are as follows: $\dot{\mathbf{e}}_r$ given by (38), $\dot{\mathbf{e}}_o = -\lambda \mathbf{e}_o \in \mathbb{R}^N$, with $\mathbf{e}_o = [e_{o_1}, e_{o_2}, \dots, e_{o_N}]^T$, $(\mathbf{J}_r \mathbf{N}_o)^+ = \text{diag}((\mathbf{J}_{r_1} \mathbf{N}_{o_1})^+, \dots, (\mathbf{J}_{r_N} \mathbf{N}_{o_N})^+)$, $\mathbf{J}_o^+ = \text{diag}(\mathbf{J}_{o_1}^+, \dots, \mathbf{J}_{o_N}^+)$, $\mathbf{N}_o = \text{diag}(\mathbf{N}_{o_1}, \dots, \mathbf{N}_{o_N})$, where the individual entries have been defined below (42).

Proof. Let us consider the following extended error function

$$\mathbf{e}' = \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_o \end{bmatrix}. \quad (44)$$

Consider a Lyapunov function candidate given by

$$V = \frac{1}{2} \mathbf{e}'^T \mathbf{e}', \quad (45)$$

whose time-derivative is given by

$$\dot{V} = \mathbf{e}'^T \dot{\mathbf{e}}'. \quad (46)$$

Using the open-loop dynamics of $\dot{\mathbf{e}}_r$ and $\dot{\mathbf{e}}_o$ for the whole network to compute \dot{V} , we have the former given by (35). Expanding the obstacle task error for the N agents, we get

$$\dot{\mathbf{e}}_o = \mathbf{J}_o \dot{\alpha} = \text{diag}(\mathbf{J}_{o_1}, \dots, \mathbf{J}_{o_N}) \dot{\alpha}, \quad (47)$$

where $\mathbf{J}_o \in \mathbb{R}^{N \times nN}$. Now, we have

$$\dot{V} = [\mathbf{e}_r^T \quad \mathbf{e}_o^T] \left(\begin{bmatrix} \mathbf{J}_r \\ \mathbf{J}_o \end{bmatrix} \dot{\alpha}_v - \begin{bmatrix} \dot{\mathbf{r}} \\ \mathbf{0} \end{bmatrix} \right). \quad (48)$$

Considering that $\dot{\alpha}_v = \mathbf{M}(\boldsymbol{\theta})\mathbf{u}$ according to (27) and introducing the hierarchical task-based control protocol (43), then

$$\dot{V} = [\mathbf{e}_r^T \quad \mathbf{e}_o^T] \left(\Psi - \begin{bmatrix} \dot{\mathbf{r}} \\ \mathbf{0} \end{bmatrix} \right), \quad (49)$$

where

$$\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_r(\mathbf{J}_o^+ \dot{\mathbf{e}}'_o) + \mathbf{J}_r((\mathbf{J}_r \mathbf{N}_o)^+(\dot{\mathbf{e}}_r + \dot{\mathbf{r}} - \mathbf{J}_r \mathbf{J}_o^+ \dot{\mathbf{e}}'_o)) \\ \mathbf{J}_o(\mathbf{J}_o^+ \dot{\mathbf{e}}'_o) + \mathbf{J}_o((\mathbf{J}_r \mathbf{N}_o)^+(\dot{\mathbf{e}}_r + \dot{\mathbf{r}} - \mathbf{J}_r \mathbf{J}_o^+ \dot{\mathbf{e}}'_o)) \end{bmatrix}. \quad (50)$$

Expanding the expressions using $\dot{\mathbf{e}}'_o$ as in (43), we have

$$\begin{aligned} \Psi_1 &= \mathbf{J}_r \mathbf{J}_o^+ [h \dot{\mathbf{e}}_o + (1-h) \mathbf{J}_o \mathbf{J}_r^+ \dot{\mathbf{e}}_r] + \mathbf{J}_r (\mathbf{J}_r \mathbf{N}_o)^+ \{ \dot{\mathbf{e}}_r + \dot{\mathbf{r}} - \mathbf{J}_r \mathbf{J}_o^+ [h \dot{\mathbf{e}}_o + (1-h) \mathbf{J}_o \mathbf{J}_r^+ \dot{\mathbf{e}}_r] \}, \\ \Psi_2 &= \mathbf{J}_o \mathbf{J}_o^+ [h \dot{\mathbf{e}}_o + (1-h) \mathbf{J}_o \mathbf{J}_r^+ \dot{\mathbf{e}}_r] + \mathbf{J}_o (\mathbf{J}_r \mathbf{N}_o)^+ \{ \dot{\mathbf{e}}_r + \dot{\mathbf{r}} - \mathbf{J}_r \mathbf{J}_o^+ [h \dot{\mathbf{e}}_o + (1-h) \mathbf{J}_o \mathbf{J}_r^+ \dot{\mathbf{e}}_r] \}. \end{aligned}$$

Given the known properties $\mathbf{J}_o \mathbf{J}_o^+ = \mathbf{I}_N$, $\mathbf{J}_o \mathbf{N}_o = \mathbf{0}_{N \times 2N}$, $\mathbf{N}_o = \mathbf{N}_o^T$, and $\mathbf{N}_o \mathbf{N}_o = \mathbf{N}_o$, the following holds

$$\mathbf{J}_o (\mathbf{J}_r \mathbf{N}_o)^+ = \mathbf{J}_o \mathbf{N}_o^T (\mathbf{N}_o \mathbf{N}_o^T)^{-1} = \mathbf{J}_o \mathbf{N}_o (\mathbf{N}_o \mathbf{N}_o^T)^{-1} = \mathbf{0}_{N \times 2N}, \quad (51)$$

$$(\mathbf{J}_r \mathbf{N}_o)^+ = (\mathbf{J}_r \mathbf{N}_o)^T ((\mathbf{J}_r \mathbf{N}_o) (\mathbf{J}_r \mathbf{N}_o)^T)^{-1} = \mathbf{N}_o (\mathbf{N}_o \mathbf{N}_o^T)^{-1} = \mathbf{I}_{2N}. \quad (52)$$

Thus, Ψ_1 and Ψ_2 are simplified as follows

$$\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}}_r + \dot{\mathbf{r}} \\ (1-h) \mathbf{J}_o \dot{\mathbf{e}}_r + h \dot{\mathbf{e}}_o \end{bmatrix}. \quad (53)$$

Introducing the closed-loop dynamics (38) and $\dot{\mathbf{e}}_o = -\lambda \mathbf{e}_o$ in (53), the time-derivative of the Lyapunov function, is given by

$$\dot{V} = -[\mathbf{e}_r^T \quad \mathbf{e}_o^T] \left(\mathbf{P} \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_o \end{bmatrix} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right), \quad (54)$$

where

$$\mathbf{P} = \begin{bmatrix} (\mathbf{K} \otimes \mathbf{I}_2) & \mathbf{0}_{2N \times N} \\ (1-h)\mathbf{J}_o(\mathbf{K} \otimes \mathbf{I}_2) & \lambda h \mathbf{I}_N \end{bmatrix}. \quad (55)$$

The eigenvalues of matrix \mathbf{P} depend on the values $\lambda > 0$, $0 < h(t) \leq 1$ and the matrix \mathbf{K} , which is defined as $\mathbf{K} = \text{diag}(\gamma, \gamma, k, k, k, \dots, k) \in \mathbb{R}^{2N \times 2N}$, where $\gamma > 0$ is the tracking gain and $k > 0$ is the consensus gain. Therefore, the matrix \mathbf{P} is positive definite. As in the case without obstacle avoidance of the Theorem 1, \mathbf{d} only affects the dynamics of the tracking error \mathbf{e}_{r_1} and represents the consensus error of the leader with respect to its neighbors. Given the positive definiteness of the matrix \mathbf{P} , the rest of the consensus errors $\mathbf{e}_{r_i}, i \in \{2, \dots, N\}$ altogether with the evasion error \mathbf{e}_o have asymptotic stability to zero. As shown below (41), the effect of \mathbf{d} vanishes as the time elapses and the tracking error also converges asymptotically to zero. Consequently, the leader tracks the time-varying reference, the virtual followers reach consensus to the leader position (respectively the robots achieve the desired formation around the leader) and the robots perform obstacle avoidance when needed. \square

The previous theorem guarantees the stability of the multi-agent system in closed loop with the distributed control protocol that solves both the formation tracking and obstacle avoidance tasks and provides continuity of the computed robot velocities. Since we are modeling the robots at kinematic level, the generation of smooth velocities is important to improve the accuracy of the position control. It is well known that due to robot and actuator dynamics (masses and rotational inertias), velocity commands with discontinuous profile cannot be exactly realized by a vehicle.

From our point of view, the proposed approach has two limitations. (1) Connectivity of the communication model is very important to guarantee convergence of the consensus error; however, connectivity can be easily lost when the robots move in cluttered environments. Although our scheme does not consider an explicit handling of connectivity, the consensus control law favors the maintenance of it. Nevertheless, there are no theoretical guarantees that connectivity is preserved in the current formulation. We consider that the approach can be extended to include an additional task dedicated to contribute in the velocities' computation to move the group of robots preserving connectivity, such that the control law might be a combination of the solution of three tasks. (2) The proposed approach is considered as a sensor-based reactive navigation method that works properly not only for circular (strongly convex) obstacles but also for unknown convex polygonal obstacles of different forms. Nevertheless, our approach is not able to deal with non-convex obstacles. We consider that the proposed approach offers a good compromise between effectiveness and required information to solve the formation tracking problem in cluttered environments without the need of a high level global planner.

4. Evaluation of the Proposed Scheme

In this section, we present results of implementing the proposed distributed control law, first in the dynamic simulator Gazebo with a group of TurtleBots 3 Burger (Robotis Inc., Lake Forest, CA, USA) and then, with real robots using the Robot Operating System (ROS) in both cases. The simulations are presented for two different cases: a 4-agent system where the predefined trajectory to follow was a quadrifolium curve and a 10-agent system that had to follow a circular predefined trajectory. Results in an environment with unknown polygonal obstacles are also shown.

In the implementation, the distance d to define the control point (2) for all the robots was set to 10 cm. Besides, the transition function is $h(t) = \frac{1}{2} \left(1 - \cos \left(\pi \frac{(t-t_0)}{(t_f-t_0)} \right) \right)$, where

t_0 was set to the current time value ($t_0 = t$) at the instant that a DDR crossed the security distance of an obstacle, $t_f = t_0 + t_d$ with t_d the duration of the transition function that was set to 0.8 s for both activation and deactivation. The sampling time in the simulations was set to 25 ms. Since the formation tracking control law requires knowing the velocities of the neighboring robots in the formation, we used estimated values taken from the velocities computed by the control law in one sampling time before the current time. This aspect could be improved as future work by using, for instance, an estimation of the velocities given by an exact differentiator [46] of the robots positions, or a distributed observer as in [47], although in the last case only estimation of the leader's state would be needed.

4.1. A 4-Agent System

The results to implement the distributed control law (42) for a 4-agent system ($N = 4$) are presented in this section. We considered the network with undirected communication topology described by the graph of Figure 2a and the desired formation of Figure 2b. We set $a_{ij} = 1$ for connected agents and initial conditions $\mathbf{q}_x(0) = [0, -2, -3, -1]^T$, $\mathbf{q}_y(0) = [0, 1.8, -1, -0.8]^T$ and $\mathbf{q}_\theta(0) = [0, -0.6, 1.14, 0]^T$ for agents 1 to 4, respectively. The reference trajectory had a duration of $\tau = 200$ s and was given by $\mathbf{m}(t) = 4[\sin(4\pi t/\tau) \cos 2\pi t/\tau, \sin(4\pi t/\tau) \sin 2\pi t/\tau]^T$. There was a fixed obstacle in the environment, in particular a column in the position $\alpha_o = [0, -2]^T$, and the security distance for it and between robots was set to $\sigma = 0.6$ m. The displacement vectors to define the desired formation were $\mathbf{z}_1 = 1.5[\cos(3\pi/2), \sin(3\pi/2)]^T$, $\mathbf{z}_2 = 1.5[\cos(2\pi), \sin(2\pi)]^T$, $\mathbf{z}_3 = 1.5[\cos(\pi/2), \sin(\pi/2)]^T$ and $\mathbf{z}_4 = 1.5[\cos(\pi), \sin(\pi)]^T$, for agents R_1 to R_4 respectively, with respect to the formation center. The control gains were set to $\gamma = 1$, $\lambda = 0.8$ and $k = 0.1$ for tracking, evasion and consensus, respectively.

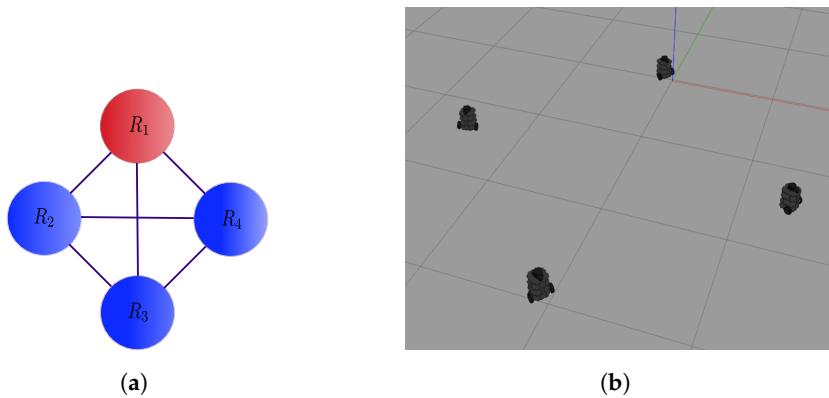


Figure 2. The 4-agent system. (a) Connected graph used in the simulations. (b) Specified formation in the dynamic simulator Gazebo.

The results of the formation tracking with obstacle avoidance for the 4-agent MAS are presented in Figures 3–5. A video of this experiment can be seen in the link https://drive.google.com/file/d/1SEDzs_H3UoK7Y2TqMTGHMZ5M3Q16p86q/view (accessed on 24 June 2021). Figure 3 shows snapshots of the motion of the robots in an environment with a fixed obstacle in the form of a column. The images on the left correspond to the initial position of the DDRs in an arbitrary configuration; the images at the center present an intermediate position where one of the DDRs was avoiding the obstacle, and the images on the right show that the desired formation was effectively reached despite that the four agents had to avoid the obstacle in some moment during their motion. This can be appreciated in Figure 4a, where the trajectory followed by each robot and the obstacle position with its security distance are depicted. Since the obstacle was at the center of the environment, the trajectories of the four agents passed close to the obstacle and the robots had to avoid it in different moments. Figure 4b shows the trajectories of the virtual agents (variables α_{v_i}) during the experiment. The initial virtual positions are marked with \times and the final positions with a circle, which appears as a single one around coordinates (0.3, 0.5).

This demonstrates that consensus of the virtual agents was finally achieved although not all the time these agents moved in consensus due to the transients introduced by the obstacle avoidance task. In Figure 5, it is shown that every robot activated the multitask control (42) at least one time during the experiment and the time interval that such control was active was different for each robot.

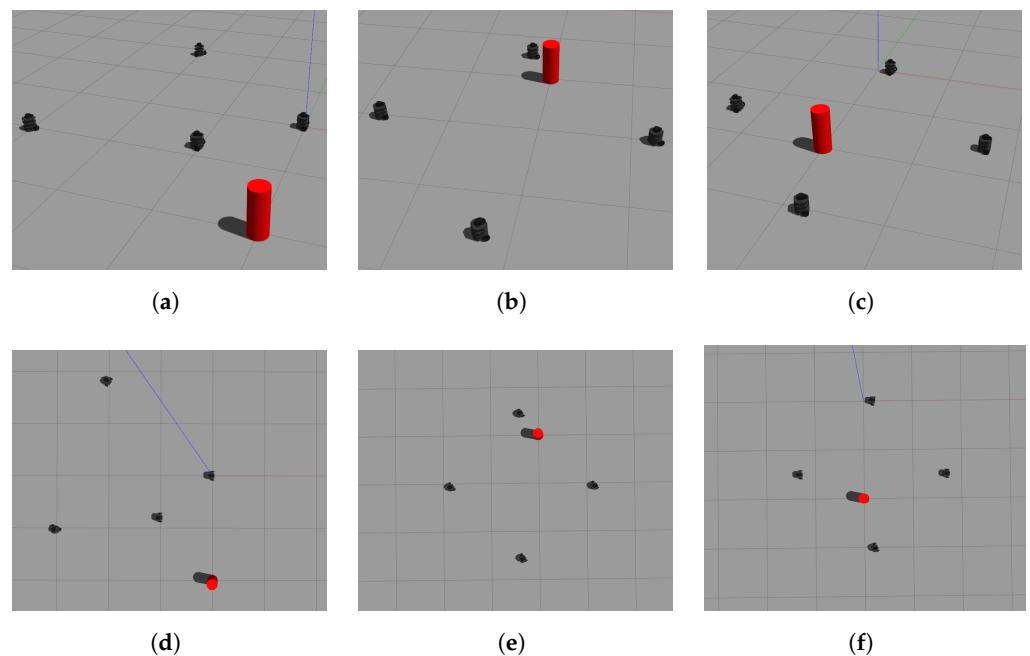


Figure 3. The 4-agent system. Snapshots of the DDRs motion in the simulation. (a) Initial position. (b) Intermediate position. (c) Final position. (d) Upper view of initial position. (e) Upper view of intermediate position. (f) Upper view of final position.

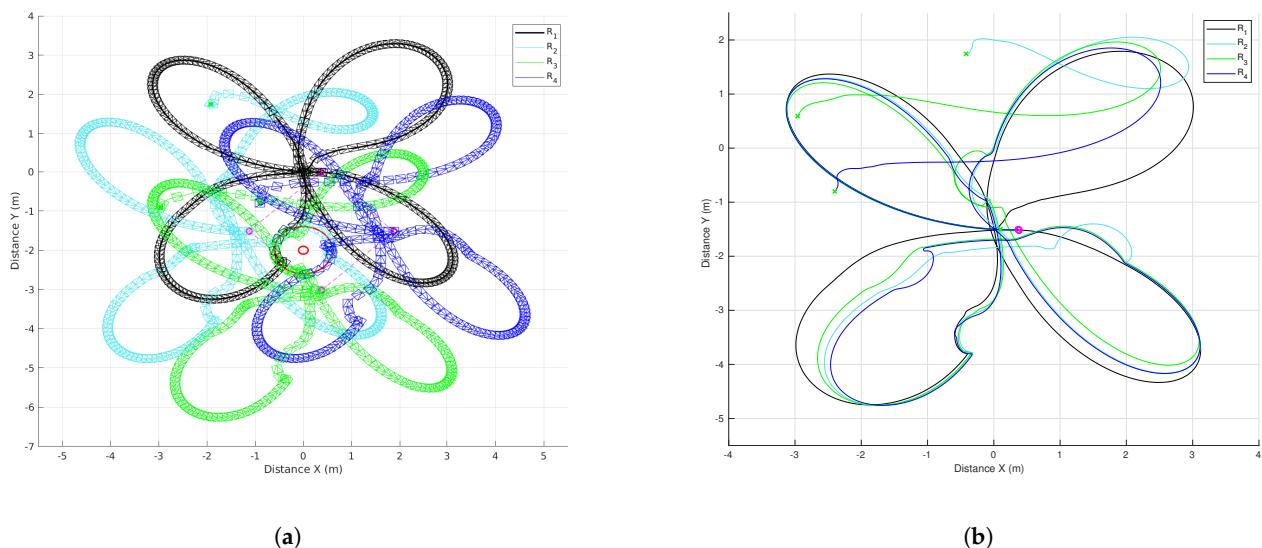


Figure 4. The 4-agent system. Trajectories of the DDRs. (a) Trajectories of the real agents. (b) Trajectories of the virtual agents.

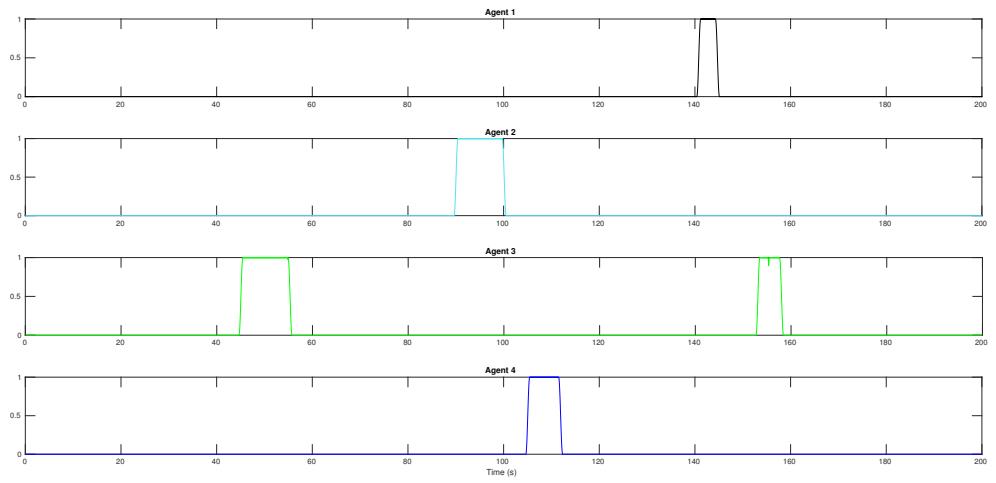


Figure 5. The 4-agent system. Task transition functions for each agent.

Figure 6a shows the evolution in time of the tracking error for the leader R_1 . The leader avoided the obstacle slightly after time 140 s, which was evident in the rapid but continuous change of the tracking error. In other times of the experiment, as around 50 s and 90 s, the effect can be seen of the evasions performed by the followers that affected the tracking error of the reference by the leader; after a transient the tracking error decreased toward zero. The consensus errors of the virtual MAS for coordinates x and y are shown in Figure 6b,c, respectively. Consensus was achieved in both coordinates in around 45 s for the first time, but after that, the agent R_3 performed an evasion motion and the consensus error increased for a while, to asymptotically decrease to zero around 90 s. Thus, this effect appeared always that one of the agents avoided an obstacle, due to the use of local information of the network.

The profiles of the control inputs (robot velocities) of each agent are shown in Figure 7a,b, for the linear and angular velocities, respectively. During the first seconds, some important changes were appreciated in the angular velocities due to the DDRs having to align their orientation to the one imposed by the reference trajectory of the leader. It can be seen that all the velocities were the same when the virtual agents are in consensus (the robots move in formation), for instance around 40 s and 85 s. As expected, the evasion actions yielded changes on the velocities evolution, mainly among the follower agents, for instance, the robot R_3 avoided the fixed obstacle around the time 45 s; it yielded an important change in the velocities of R_2 and R_4 and less for the leader R_1 since the tracking of the reference had a higher weight for the leader's dynamics ($\gamma > k$). Notice that the profiles of velocities were continuous all the time thanks to the use of the smooth transition function $h(t)$.

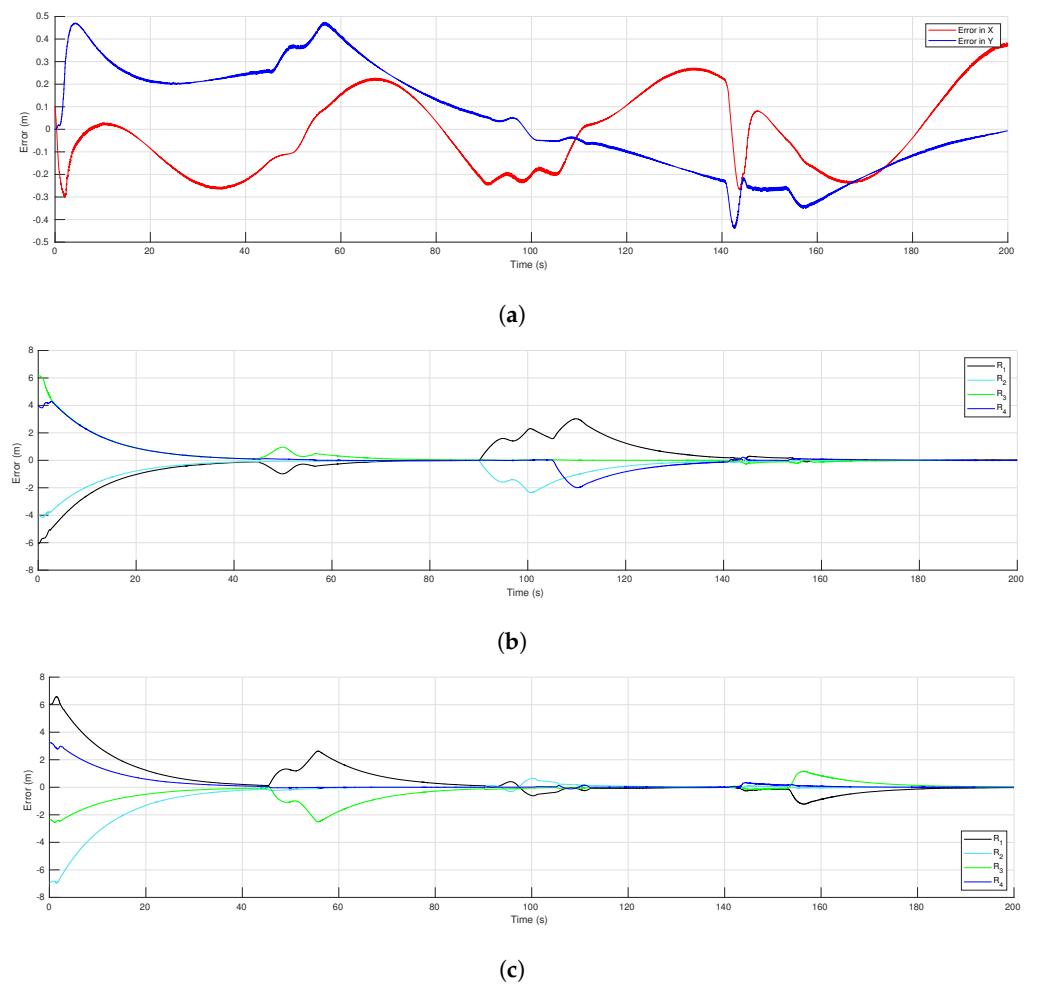


Figure 6. The 4-agent system. Error functions of the formation tracking. **(a)** Tracking errors of the leader. **(b)** Consensus errors in the x -axis. **(c)** Consensus errors in the y -axis.

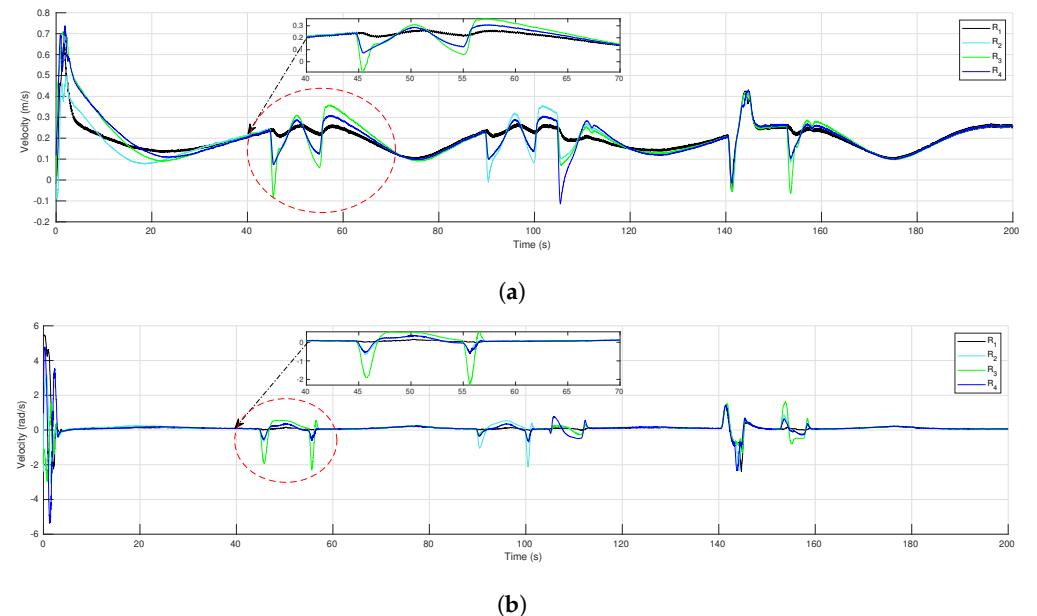


Figure 7. The 4-agent system. Control inputs (velocities) of the DDRs. **(a)** Linear velocities. **(b)** Angular velocities.

4.2. A 10-Agent System

As a second experiment, we considered a 10-agent system in a network with undirected communication topology described by the graph of Figure 8a, with the leader R_1 as the root. Figure 8b presents the desired formation. We set $a_{ij} = 1$ for connected agents and initial conditions $\mathbf{q}_x(0) = [5, 4, 6, 3.5, 5.5, 7, 3, 4.5, 6.5, 7.5]^T$, $\mathbf{q}_y(0) = (0)\mathbf{1}_N$ and $\mathbf{q}_\theta(0) = (\pi/2)\mathbf{1}_N$ for agents 1 to 10, respectively. The reference trajectory was a circle given by $\mathbf{m}(t) = [5 \cos(2\pi t/\tau), 5 \sin(2\pi t/\tau)]^T$, with $\tau = 180$ s. There were five fixed obstacles in the environment, represented as columns with different diameters in the positions $\alpha_{o_1} = [-3.5, -2.5]^T$, $\alpha_{o_2} = [-6, -2.5]^T$, both with associated security distance $\sigma_1 = 0.4$, $\alpha_{o_3} = [2, 2.5]^T$, $\alpha_{o_4} = [0, -5.8]^T$, both with security distance $\sigma_2 = 0.5$ and $\alpha_{o_5} = [-1, 5.3]^T$ with $\sigma_3 = 0.6$. The displacement vectors to define the desired formation were $\mathbf{z}_1 = [0, 1]^T$, $\mathbf{z}_2 = [-0.5, 0.5]^T$, $\mathbf{z}_3 = [0.5, 0.5]^T$, $\mathbf{z}_4 = [-1, 0]^T$, $\mathbf{z}_5 = [0, 0]^T$, $\mathbf{z}_6 = [1, 0]^T$, $\mathbf{z}_7 = [-1.5, -0.5]^T$, $\mathbf{z}_8 = [-0.5, -0.5]^T$, $\mathbf{z}_9 = [0.5, -0.5]^T$ and $\mathbf{z}_{10} = [1.5, -0.5]^T$, for R_1 to R_{10} respectively, with respect to the formation center. The control gains were set to $\gamma = 1$, $\lambda = 0.85$ and $k = 0.35$ for tracking, evasion and consensus, respectively.

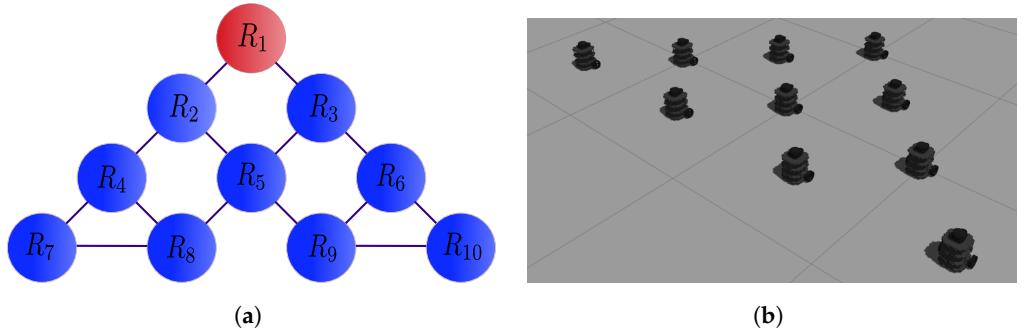


Figure 8. The 10-agent system. (a) Connected graph used in the simulations. (b) Specified formation in the dynamic simulator Gazebo.

The results to implement the proposed distributed control law for the 10-agent MAS are presented in Figures 9–13. One can see a video of this experiment in the link <https://drive.google.com/file/d/1KoZ4JIW9-lwD5kWcmxN5LNCiIsVQxOZ2/view> (accessed on 24 June 2021). Some snapshots of this experiment are shown in Figure 9; the group of robots initiated forming a line and they navigated in the specified triangular formation while obstacles were avoided when needed. The trajectories followed by the robots and the obstacle position with its security distance are depicted in Figure 10a, where one can observe several evasion actions of different agents to avoid collisions with the fixed obstacles and between robots. The trajectories of the virtual agents (variables α_{v_i}) for this experiment are presented in Figure 10b. It can be seen that the circular trajectory imposed by the leader was well tracked by the 10 agents except when evasion actions were performed, however, after a transient the group of robots returned to consensus. The final consensus value reached when the leader trajectory ended in 180 s was the point (5, −1) marked with a circle.

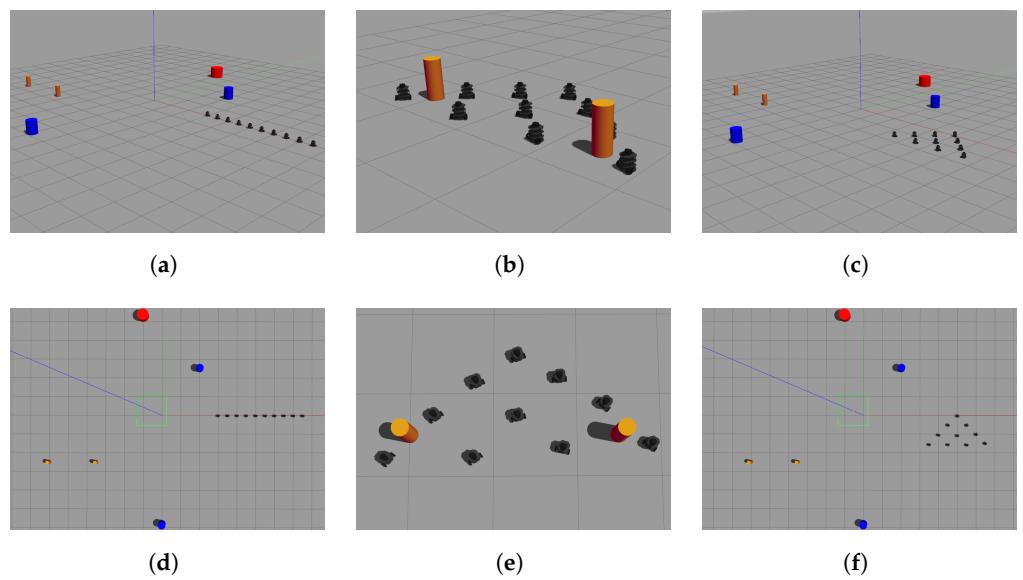


Figure 9. The 10-agent system. Snapshots of the DDRs motion in the simulation. (a) Initial position. (b) Intermediate position. (c) Final position. (d) Upper view of initial position. (e) Upper view of intermediate position. (f) Upper view of final position.

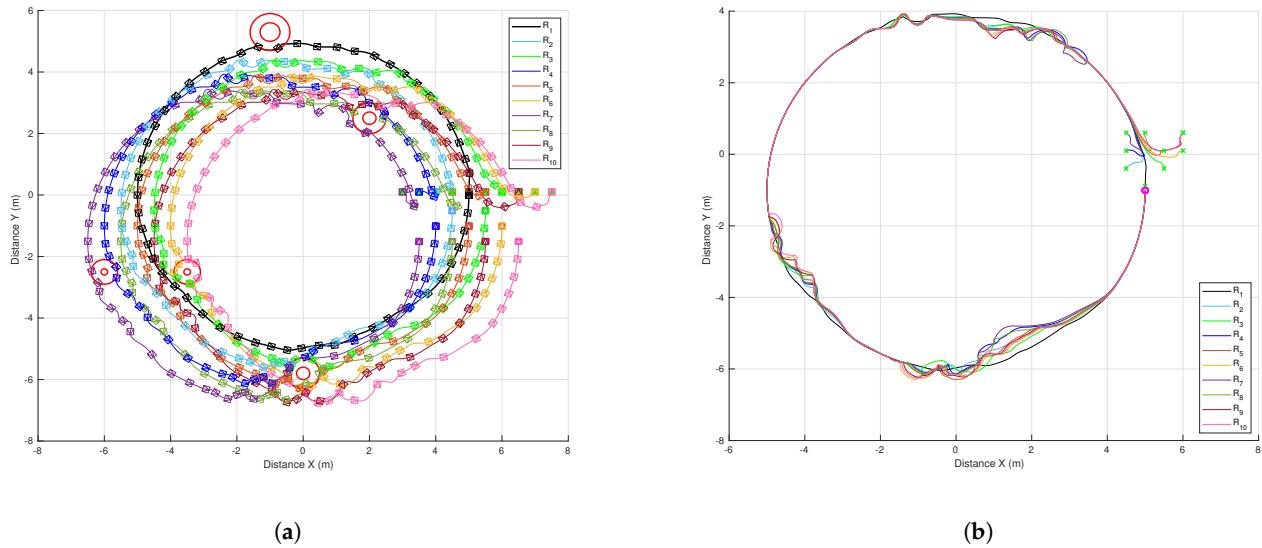


Figure 10. The 10-agent system. Trajectories of the DDRs. (a) Trajectories of real agents. (b) Trajectories of virtual agents.

In the case of the 10-agent system, due to the number of robots and the number of fixed obstacles, many more evasion actions were performed in comparison to the 4-agent experiment. Figure 11 shows the transition function along the simulation time for each robot. All the agents activated the multitask control (42) at least one time during the experiment to avoid collisions with the fixed obstacles whether only one agent had the transition function activated or to avoid other agents whether more than one agent had in high value the transition function. For instance, the leader avoided a fixed obstacle around time 50 s and agents R_4 and R_7 avoided each other around time 22 s.

Figure 12a presents the tracking error for the leader R_1 along the experiment. There was an initial transient since the leader did not initiate over the reference trajectory. In the time intervals from 22 s to 63 s and from 98 s to 165 s, it is clear the effect of the evasion actions performed by the followers that affected the tracking of the reference by the leader, however the tracking error converged to zero after those transients. In the same time intervals could be observed transients due to the evasion actions over the consensus

errors of the virtual MAS for coordinates x and y , presented in Figure 12b,c, respectively. Nevertheless, the consensus errors converged asymptotically to zero after the transients due to evasions and they were null at the final position where the leader stopped at time 180 s. Figure 13 shows the linear and angular velocities of all the robots. As expected, in the same time intervals where evasion actions were performed, the velocities had important changes but they returned to be all the same value during the intervals where the whole group moved in the desired formation, as around 80 s and from 170 s to the end. Furthermore, the velocities were continuous all the time since the function $h(t)$ defined for each robot generated a smooth transition between the control laws of Theorem 1 and the one of Theorem 2.

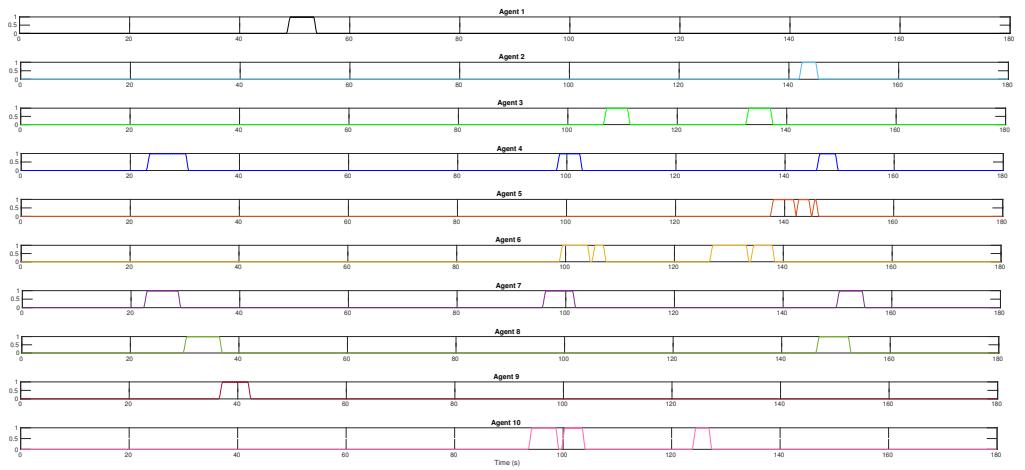


Figure 11. The 10-agent system. Task transition function activation.

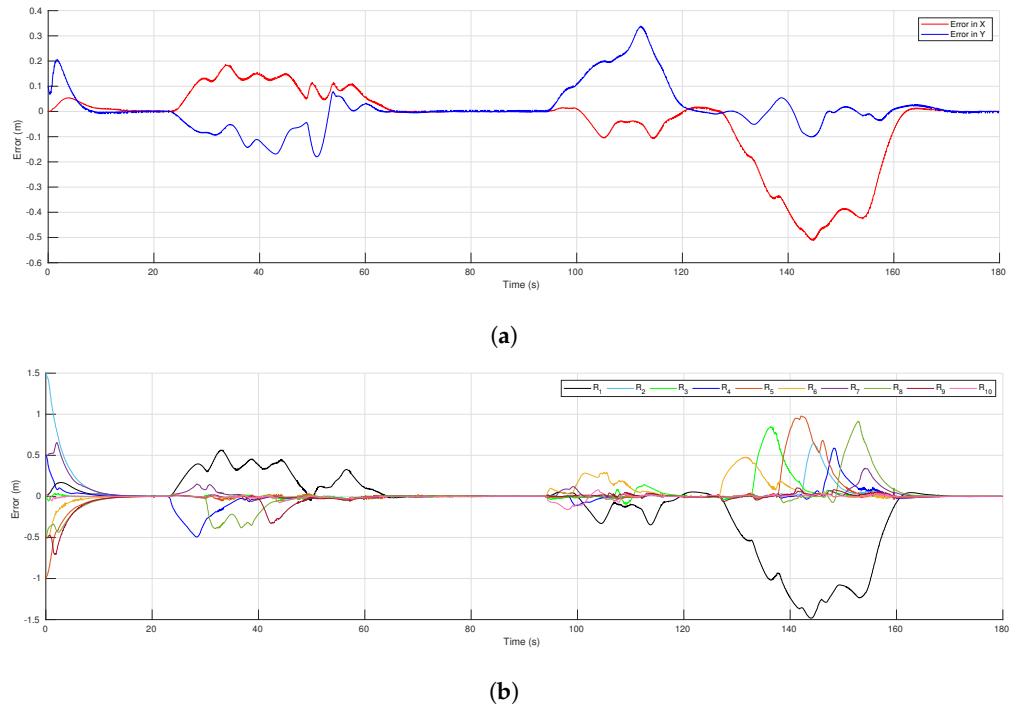


Figure 12. Cont.

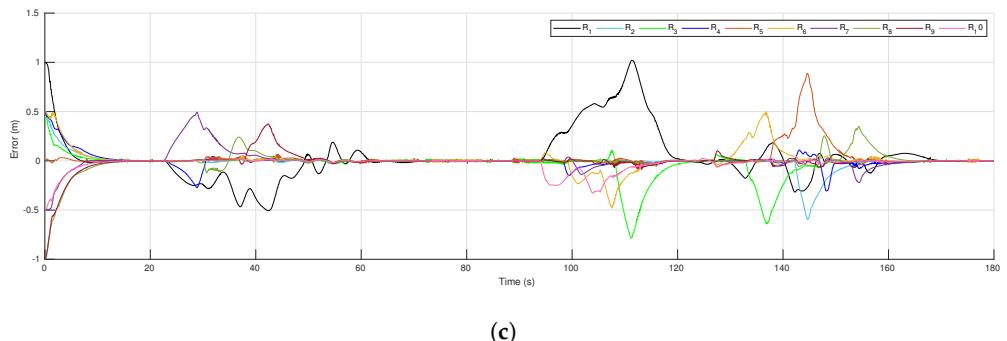


Figure 12. The 10-agent system. Error functions of the formation tracking. **(a)** Tracking errors of the leader. **(b)** Consensus errors in the x -axis. **(c)** Consensus errors in the y -axis.

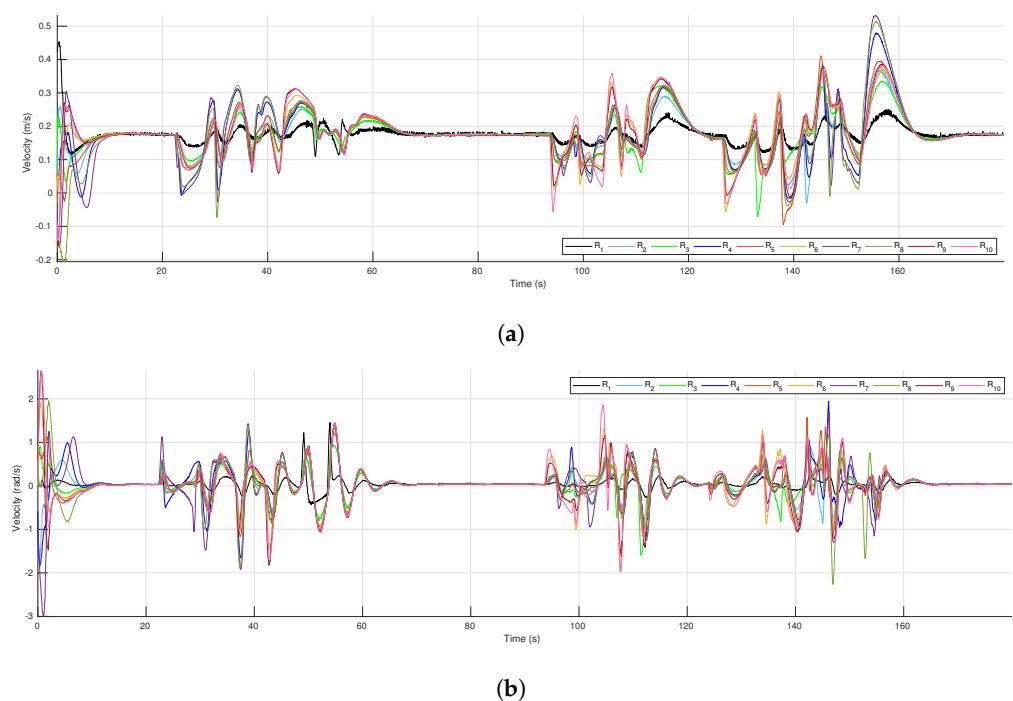


Figure 13. The 10-agent system. Control inputs (velocities) of the DMRs. **(a)** Linear velocities. **(b)** Angular velocities.

4.3. Environments with Unknown Polygonal Obstacles

This section presents results of simulations in a different kind of environment in comparison with the previous sections where circular obstacles were considered. In this case, we had a cluttered environment with large irregular obstacles with polygonal shapes. This result was implemented in MATLAB for a rapid prototyping of onboard sensors, since we simulated a scanning range finder for each robot, which allows the robots the detection of the nearest obstacle. When an obstacle was detected within the security distance $\sigma = 0.25$ the obstacle avoidance task was activated. The simulation setup corresponded to five robots and three obstacles of different shapes. The connectivity graph was circular and the desired formation specified by displacement vectors was a pentagon. The robot R_3 was the leader, which had to track a linear trajectory starting and finishing in the coordinates $(4, 7)$ and $(11, -2)$, respectively. This reference trajectory of the leader continued by following a sinusoidal cycle to finally reach the coordinate $(15, -2)$. Figure 14 shows the trajectories of the control points of each robot starting from arbitrary positions marked with asterisks. It can be seen that the robots avoided the obstacles in the environment and achieved the desired formation as a pentagon. The reference of the leader is depicted as a dash-dotted

line trespassing the pentagonal obstacle. The leader effectively avoided this obstacle and returned to track the reference. Figure 15 shows the continuous velocities of the control points (α_i) of each robot in the x and y coordinates. The reference trajectory of the leader was designed to start and finish motionless for each part of it. Thus, the linear part of the reference finished in 7 s while the sinusoidal part was performed from 7 to 11 s. After that, the velocities became null and the formation stopped since there was no consensus error among the virtual agents.

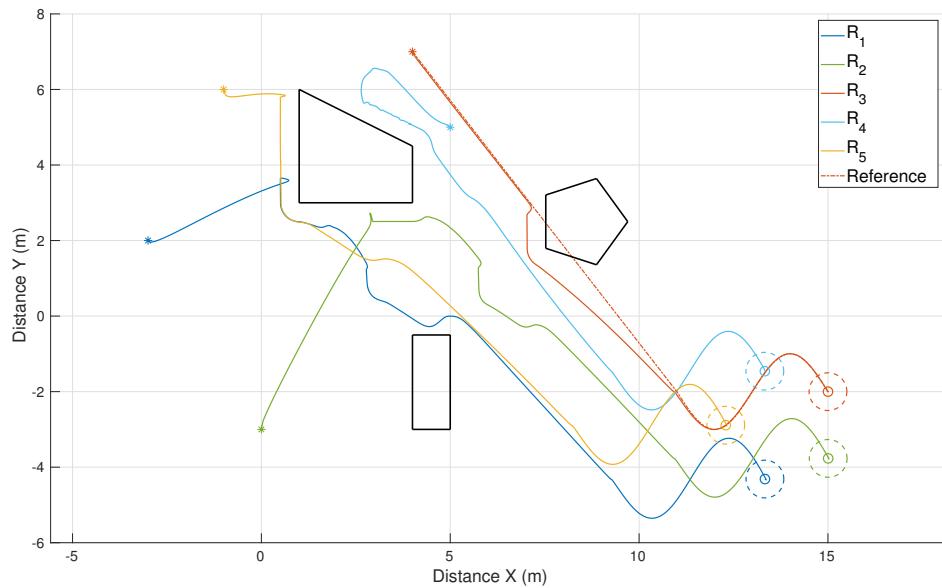


Figure 14. Environment with polygonal obstacles. Trajectories of five robots dealing with irregular unknown obstacles.

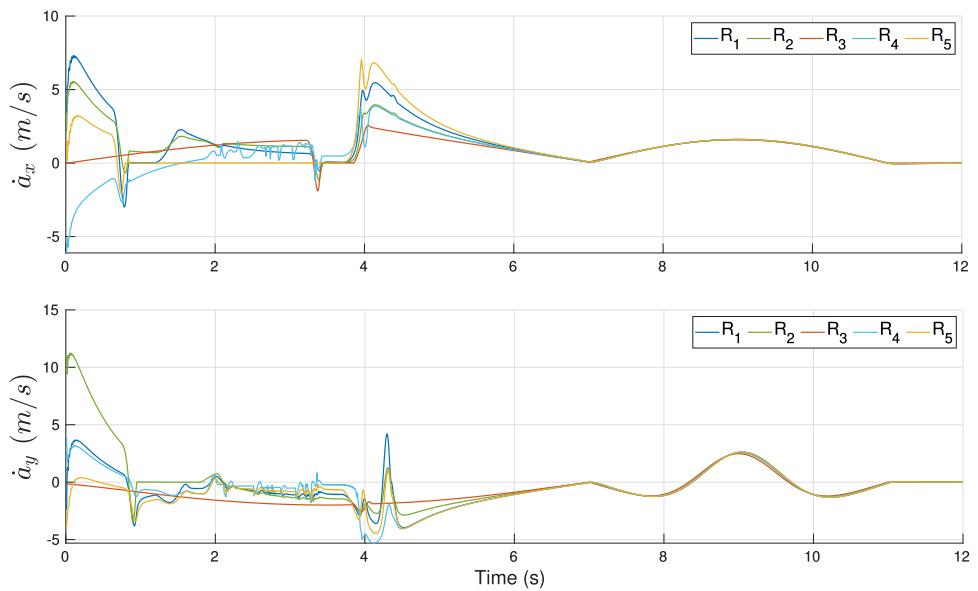


Figure 15. Environment with polygonal obstacles. Velocities of the control points of each robot.

4.4. Experiments with Real Robots

In this section, the proposed approach was evaluated in experiments with real robots using a group of three DDRs, integrated by a Pioneer 3DX (used as the leader agent) and two TurtleBots 2 (used as the followers). The gains were set to $\gamma = 0.8$, $\lambda = 0.8$ and $k = 0.12$ for tracking, evasion and consensus, respectively. The desired formation was triangular, specified by the displacement vectors $\mathbf{z}_1 = [0, 0]^T$, $\mathbf{z}_2 = 0.7[\cos(\pi/4), \cos(\pi/4)]^T$ and $\mathbf{z}_3 = 0.7[\cos(3\pi/4), \sin(3\pi/4)]^T$, with respect to the leader's R_1 position. The considered

graph was undirected and connected, such that each robot has communication with the other two. Each robot was connected via WiFi to a computer where the velocity commands were computed in a distributed way using ROS. Each agent published its computed velocities and also read the computed controls published by its neighbors at a sampling rate of 0.025 ms. The position of each robot and the obstacles position in the environment were obtained from an Optitrack Motion Capture System.

4.4.1. Linear Trajectory

As a first experiment, the leader performed the tracking of a linear trajectory, which was given by $\mathbf{m}(t) = [x, \lambda x + b]^T$, where $x = (p_{x_0} - p_{x_\tau})(1 + \cos(\pi t/\tau))/2 + p_{x_\tau}$ with $\tau = 30$ s, $\lambda = (p_{y_\tau} - p_{y_0})/(p_{x_\tau} - p_{x_0})$, $b = -\lambda p_{x_0} + p_{y_0}$ and $(p_{x_0}, p_{y_0}), (p_{x_\tau}, p_{y_\tau})$ were the initial and final coordinates of the trajectory, respectively. Besides, a fixed obstacle was at the position $\alpha_o = [0.91, 0.12]^T$ with security distance $\sigma = 0.4$.

The results of this experiment are presented in Figures 16–18 and the corresponding video can be seen in the link <https://drive.google.com/file/d/1RVNe52Gh6gmc2AtGyeuERfKY54dixHz-/view> (accessed on 24 June 2021). Figure 16 shows some snapshots of the video with the initial configuration at the top left and the final formation at the bottom right. Some intermediate positions are presented where one of the DDRs was avoiding the obstacle and also caused the others to move. This was clear in the trajectories of the robots as given by the motion capture system that can be observed in Figure 17a. The position of the obstacle is also shown with its security distance depicted as a red circle. Figure 17b shows the trajectories of the virtual agents computed from the position of the robots and the displacement vectors \mathbf{z}_i . One can observe that the robots moved in formation until R_3 had to avoid the obstacle, which generated a reaction over the motion of R_2 and a small effect on the leader R_1 . At time 30 s, the virtual agents reached consensus close to the coordinate (1.1, 2), which was the final point of the reference of the leader.

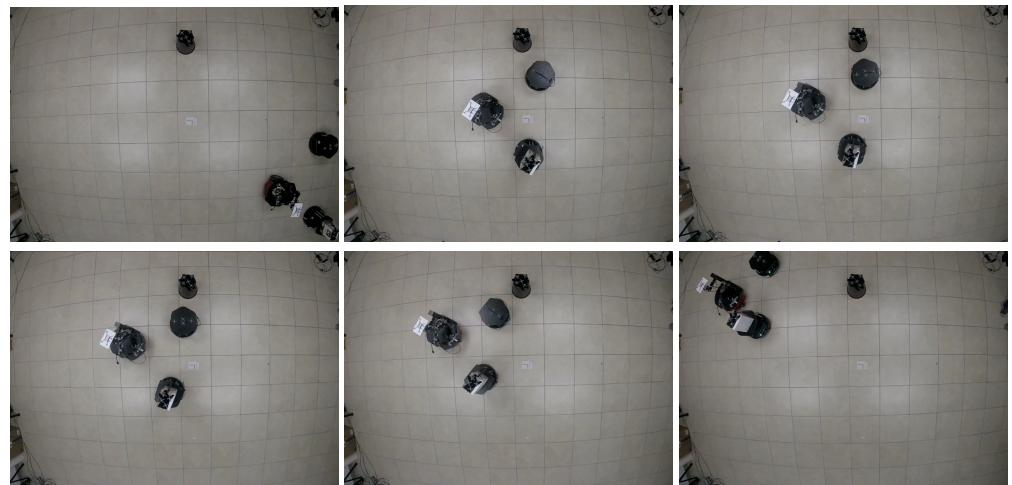


Figure 16. Three-agent system linear trajectory. Snapshots of the robots motion in the experiment. The initial configuration is at the top left and the final formation is at the bottom right. Intermediate positions when an obstacle is avoided are shown.

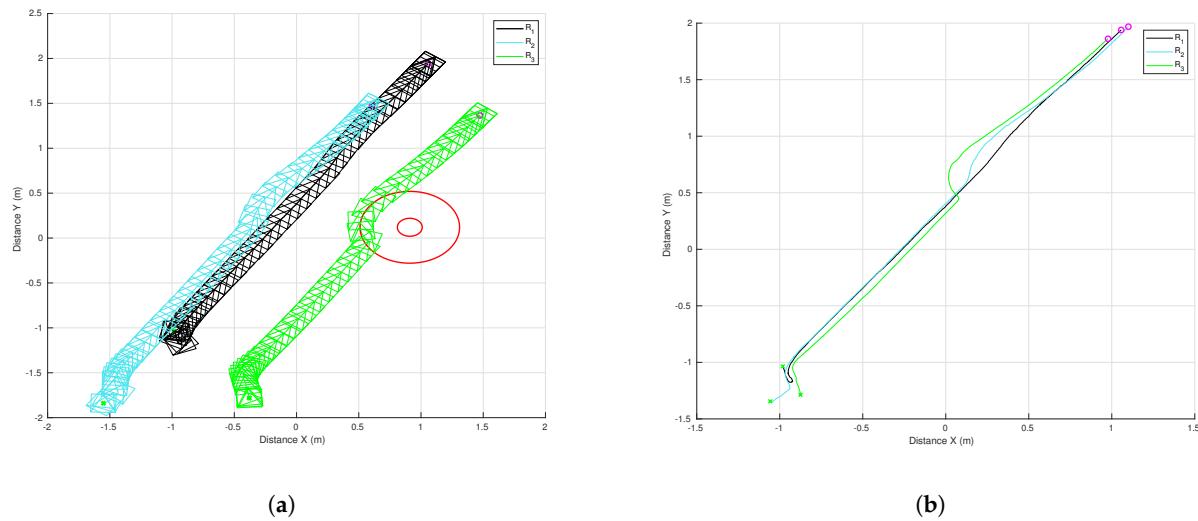


Figure 17. Three-agent system linear trajectory. **(a)** Trajectories of real agents. **(b)** Trajectories of virtual agents.

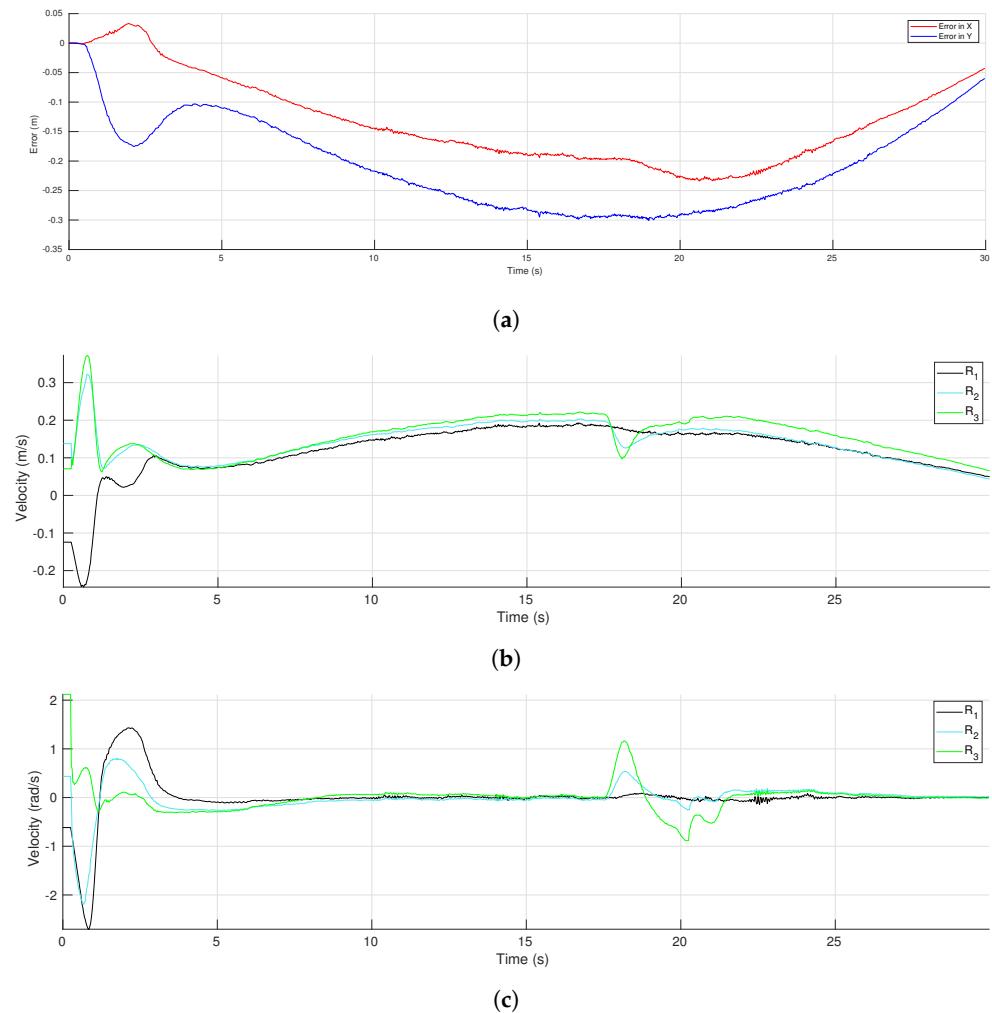


Figure 18. Three-agent system linear trajectory. **(a)** Tracking errors of the leader. **(b)** Linear velocities. **(c)** Angular velocities.

The evolution of the tracking errors for the leader \$R_1\$ with respect to time is shown in Figure 18a. One can observe an initial transient and a maximum error slightly after the middle of the time, which was expected given the form of the reference trajectory

with maximum velocity in the middle of the trajectory. After that middle point, the errors in both coordinates were reduced to reach low final values. The effect of the obstacle avoidance performed by R_3 could be appreciated around time 18 s with a smooth change in the x -coordinate of the tracking error. Figure 18b,c show the profiles of the computed robot velocities (linear and angular, respectively) for each agent. An initial transient was appreciated during the first 5 s and after that, the velocities evolved very close each other until the robot R_3 performed an evasion around time 18 s. This action yielded a larger change in the velocities of R_3 than for R_2 and had a low effect in the leader's velocities. The use of the transition function $h(t)$ allowed the proposed scheme to compute continuous velocities along the experiment.

4.4.2. Circular Trajectory

As a second experiment, we considered a circular trajectory, which was parameterized as $\mathbf{m}(t) = 1.1[\cos(2\pi t/\tau), \sin(2\pi t/\tau)]^T$, with $\tau = 70$ s, such that the radius of the circle was 1.1 m and the trajectory started at the coordinate (1.1,0) and returns to the same point in τ s. Besides, a fixed obstacle was at the position $\alpha_o = [0.6, -2.0]^T$, with security distance $\sigma = 0.55$.

Figures 19–21 present the results of this experiment and the corresponding video can be seen in the link https://drive.google.com/file/d/1bJZBO7dByTipDryC0XqHJurHE4f-d_i1/view?usp=sharing (accessed on 24 June 2021). Some snapshots of the video are presented in Figure 19. The initial configuration of the MAS is the image at the top left and the final formation is at the bottom right. Some intermediate positions are presented; in particular we show the evasion of the fixed obstacle, which causes a deviation of the three robots to later recover the trajectory. Figure 20a depicts the trajectories of the robots and shows how the robot R_2 performed the evasion when it overstepped the security region. The effect of the obstacle avoidance is also observed in Figure 20b, where the trajectories of the virtual agents are shown. They moved in consensus until the robot R_2 deviated to evade the obstacle. In this case, the deviation of the evading robot was more significant than in the previous experiment and then the three robots deviated in a similar way. At the end of the trajectory, at 70 s, the virtual agents returned to consensus close to the the final point (1.1,0) of the leader's reference.

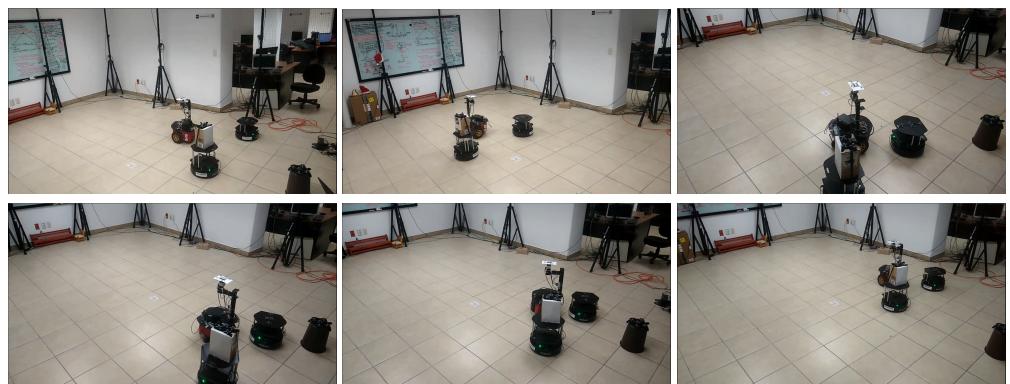


Figure 19. Three-agent system circular trajectory. Snapshots of the robots motion in the experiment. The initial configuration is at the top left and the final formation is at the bottom right. Intermediate positions when an obstacle is avoided are shown.

Figure 21a shows the evolution of the tracking errors for the leader R_1 with respect to time. After the initial transient, the errors remained at low values and they had a second transient due to the effect of the obstacle avoidance around time 53 s. Both coordinates were affected in this case, but at the end the errors converged asymptotically to zero as proven in Theorem 2. The consensus errors of the virtual MAS for coordinates x and y are shown in Figure 21b,c, respectively. The consensus errors were also maintained at low values and they increased around time 53 s due to the evasion task initiated by robot R_3 . The avoidance task finished around time 65 s and then the consensus errors converged to

reach a low value that allowed the MAS to return to the desired formation. Since the robot velocities depended on the errors shown in these figures, the proposed scheme provided continuous velocities along the experiment thanks to the smooth switching of control laws.

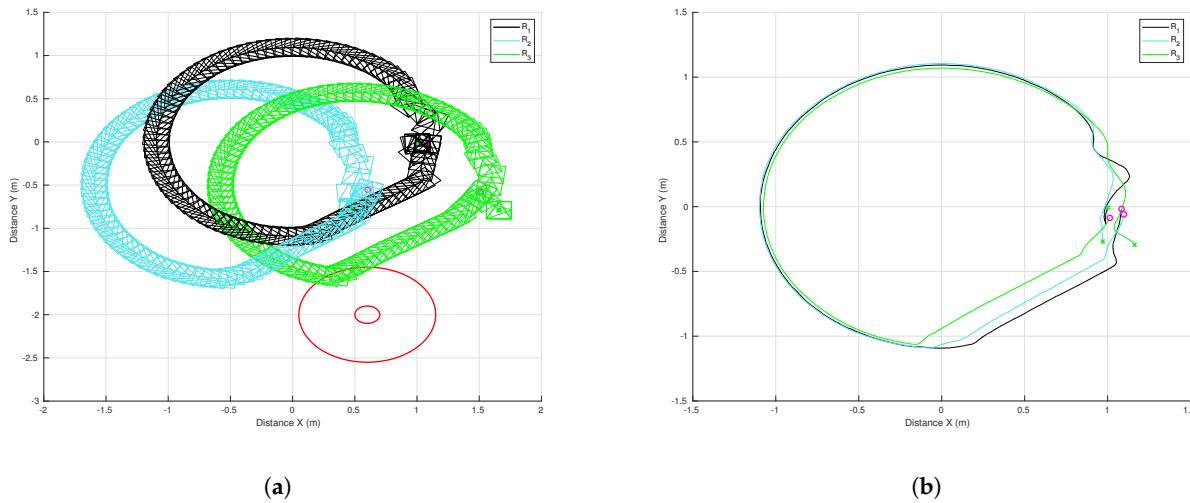


Figure 20. Three-agent system circular trajectory. **(a)** Trajectories of real agents. **(b)** Trajectories of virtual agents.

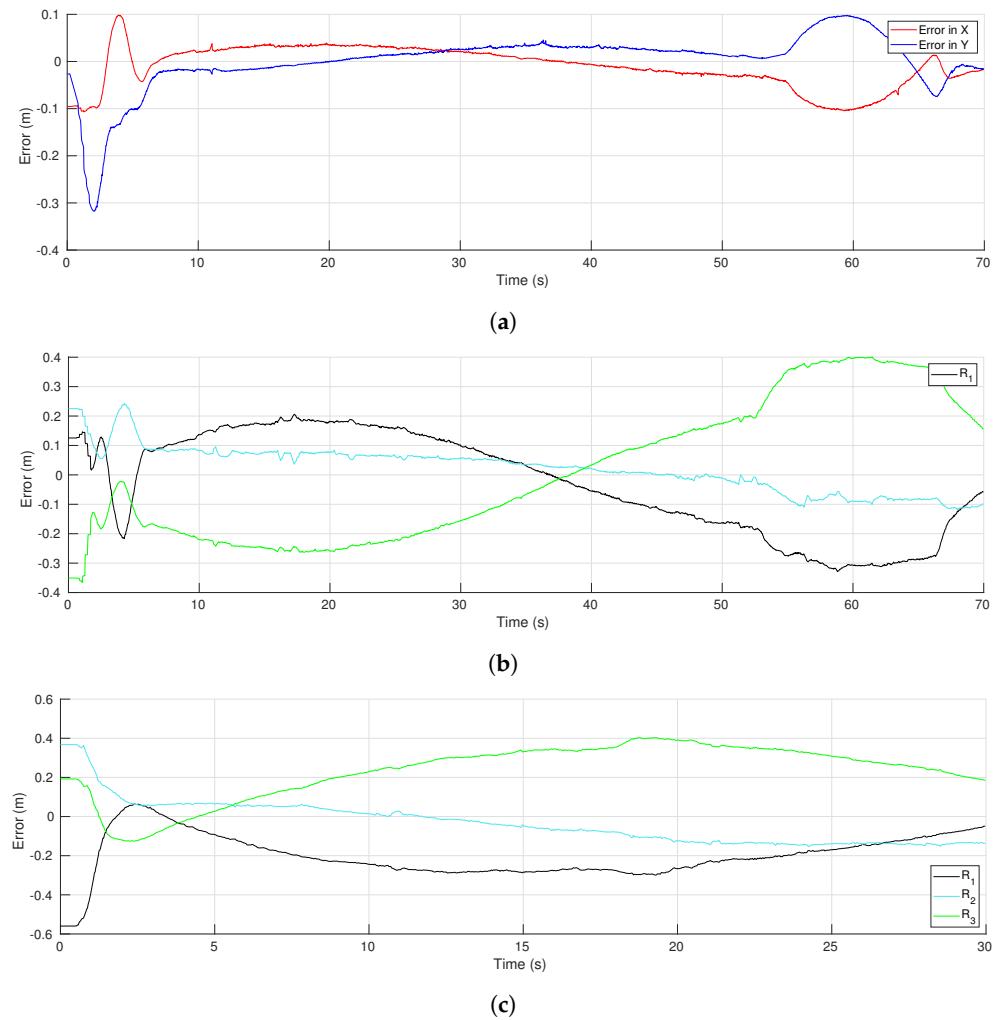


Figure 21. Three-agent system circular trajectory. Error functions of the formation tracking. **(a)** Tracking errors of the leader. **(b)** Consensus errors in the *x*-axis. **(c)** Consensus errors in the *y*-axis.

5. Conclusions and Discussion

In this paper, we have proposed a distributed control scheme to solve the problem of position formation of a group of differential-drive robots while tracking a predefined trajectory guided by a leader, all moving in an environment with obstacles. The obstacles can be the agents themselves or fixed objects in the environment. The proposed scheme is valid for connected communication topologies. A hierarchical task-based scheme has been introduced for the formation tracking problem with obstacle avoidance. Thus, whether a robot is close to an obstacle at a given distance, the task with higher priority is devoted to avoid the obstacle and the task with lower priority is the formation tracking, and whether no robots are near to the obstacles only the formation tracking control is executed.

We have formulated an adequate task function for the formation tracking and has been combined to solve both tasks simultaneously when needed, guaranteeing continuity of the velocities computed by the control law, independently of the activation/deactivation of the obstacle avoidance task. The proposed control scheme only needs relative information between neighboring agents and local measurements to avoid obstacles, in such a way that neither a global coordinate system nor a map of the environment are required. The convergence of the proposed control scheme to the formation tracking in spite of the execution of evasion actions is proven theoretically and its effectiveness is validated through realistic simulations and experiments using real robots.

As future work, we want to extend the proposed hierarchical scheme to include more tasks, for instance connectivity maintenance (focused on preserve the initial connected graph or guarantee the maintenance of a connected graph regardless of changes in the communication topology), consider more than one leader and different formations in sub-groups, as well as the implementation of the approach for long-distance navigation experiments in more general environments with complex obstacles. To do so, we will focus on developing experimental work relying only on local measurements using onboard sensing, such as laser range finders or depth cameras mounted on the robots, instead of a global positioning system.

Author Contributions: Conceptualization, H.M.B.; methodology, H.M.B. and J.B.M.; software, J.B.M.; validation, H.M.B. and J.B.M.; formal analysis, H.M.B., J.B.M. and D.G.-G.; investigation, H.M.B. and D.G.-G.; resources, H.M.B.; writing—original draft preparation, H.M.B., J.B.M. and D.G.-G.; visualization, H.M.B. and J.B.M.; supervision, H.M.B.; funding acquisition, H.M.B. and D.G.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by Intel Corporation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Eng. Antonio Sanchez Medel for his technical support in the realization of the experiments reported in this work.

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

References

1. Fan, F.; Wu, G.; Wang, M.; Cao, Q.; Yang, S. Multi-Robot Cyber Physical System for Sensing Environmental Variables of Transmission Line. *Sensors* **2018**, *18*, 3146. [[CrossRef](#)] [[PubMed](#)]
2. Queralta, J.P.; Taipalmaa, J.; Can Pullinen, B.; Sarker, V.K.; Nguyen Gia, T.; Tenhunen, H.; Gabbouj, M.; Raitoharju, J.; Westerlund, T. Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision. *IEEE Access* **2020**, *8*, 191617–191643. [[CrossRef](#)]
3. Davoodi, M.; Velni, J.M.; Li, C. Coverage Control with Multiple Ground Robots for Precision Agriculture. *Mech. Eng.* **2018**, *140*, 4–8. [[CrossRef](#)]
4. Recker, T.; Heinrich, M.; Raatz, A. A Comparison of Different Approaches for Formation Control of Nonholonomic Mobile Robots regarding Object Transport. *Procedia CIRP* **2021**, *96*, 248–253. [[CrossRef](#)]

5. Ailon, A.; Zohar, I. Control Strategies for Driving a Group of Nonholonomic Kinematic Mobile Robots in Formation Along a Time-Parameterized Path. *IEEE ASME Trans. Mechatron.* **2012**, *17*, 326–336. [[CrossRef](#)]
6. Ghommam, J.; Mehrjerdi, H.; Saad, M.; Mnif, F. Formation path following control of unicycle-type mobile robots. *Robot. Auton. Syst.* **2010**, *58*, 727–736. [[CrossRef](#)]
7. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control.* **2004**, *49*, 1520–1533. [[CrossRef](#)]
8. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [[CrossRef](#)]
9. Ren, W.; Beard, R.W. *Distributed Consensus in Multi-Vehicle Cooperative Control*; Springer: Berlin/Heidelberg, Germany, 2008.
10. Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [[CrossRef](#)]
11. Montijano, E.; Cristofalo, E.; Schwager, M.; Sagües, C. Distributed Formation Control of Non-Holonomic Robots without a Global Reference Frame. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5248–5254.
12. Roza, A.; Maggiore, M.; Scardovi, L. A Smooth Distributed Feedback for Formation Control of Unicycles. *IEEE Trans. Autom. Control* **2019**, *64*, 4998–5011. [[CrossRef](#)]
13. Sun, Z.; de Marina, H.G.; Seyboth, G.S.; Anderson, B.D.O.; Yu, C. Circular Formation Control of Multiple Unicycle-Type Agents With Nonidentical Constant Speeds. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 192–205. [[CrossRef](#)]
14. Ren, W.; Sorensen, N. Distributed coordination architecture for multi-robot formation control. *Robot. Auton. Syst.* **2008**, *56*, 324–333. [[CrossRef](#)]
15. Ren, W. Multi-vehicle consensus with a time-varying reference state. *Syst. Control Lett.* **2007**, *56*, 474–483. [[CrossRef](#)]
16. Kwon, J.; Chwa, D. Hierarchical Formation Control Based on a Vector Field Method for Wheeled Mobile Robots. *IEEE Trans. Robot.* **2012**, *28*, 1335–1345. [[CrossRef](#)]
17. Chu, X.; Peng, Z.; Wen, G.; Rahmani, A. Decentralised consensus-based formation tracking of multiple differential drive robots. *Int. J. Control.* **2017**, *90*, 2461–2470. [[CrossRef](#)]
18. Liu, T.; Jiang, Z.P. Distributed formation control of nonholonomic mobile robots without global position measurements. *Automatica* **2013**, *49*, 592–600. [[CrossRef](#)]
19. Peng, X.; Sun, Z.; Guo, K.; Geng, Z. Mobile Formation Coordination and Tracking Control for Multiple Nonholonomic Vehicles. *IEEE ASME Trans. Mechatron.* **2020**, *25*, 1231–1242. [[CrossRef](#)]
20. Saradagi, A.; Muralidharan, V.; Krishnan, V.; Menta, S.; Mahindrakar, A.D. Formation Control and Trajectory Tracking of Nonholonomic Mobile Robots. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 2250–2258. [[CrossRef](#)]
21. Yan, L.; Ma, B. Practical Formation Tracking Control of Multiple Unicycle Robots. *IEEE Access* **2019**, *7*, 113417–113426. [[CrossRef](#)]
22. Ning, B.; Han, Q.L.; Lu, Q. Fixed-Time Leader-Following Consensus for Multiple Wheeled Mobile Robots. *IEEE Trans. Cybern.* **2020**, *50*, 4381–4392. [[CrossRef](#)]
23. González-Sierra, J.; Aranda-Bricaire, E.; Rodríguez-Cortés, H.; Santiguillo-Salinas, J. Formation tracking for a group of differential-drive mobile robots using an attitude observer. *Int. J. Control.* **2021**, *94*, 89–102. [[CrossRef](#)]
24. Listmann, K.D.; Masalawala, M.V.; Adamy, J. Consensus for Formation Control of Nonholonomic Mobile Robots. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3886–3891.
25. Poonawala, H.A.; Satici, A.C.; Eckert, H.; Spong, M.W. Collision-Free Formation Control with Decentralized Connectivity Preservation for Nonholonomic-Wheeled Mobile Robots. *IEEE Trans. Control Netw. Syst.* **2015**, *2*, 122–130. [[CrossRef](#)]
26. Jin, J.; Gans, N. Collision-free formation and heading consensus of nonholonomic robots as a pose regulation problem. *Robot. Auton. Syst.* **2017**, *95*, 25–36. [[CrossRef](#)]
27. Dang, A.D.; La, H.M.; Nguyen, T.; Horn, J. Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force-based approach. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419847897. [[CrossRef](#)]
28. Yu, H.; Shi, P.; Lim, C.C.; Wang, D. Formation control for multi-robot systems with collision avoidance. *Int. J. Control.* **2019**, *92*, 2223–2234. [[CrossRef](#)]
29. Dai, S.; He, S.; Chen, X.; Jin, X. Adaptive Leader–Follower Formation Control of Nonholonomic Mobile Robots With Prescribed Transient and Steady-State Performance. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3662–3671. [[CrossRef](#)]
30. Kamel, M.A.; Zhang, Y. Decentralized Leader–Follower Formation Control with Obstacle Avoidance of Multiple Unicycle Mobile Robots. In Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 3–6 May 2015; pp. 406–411.
31. Wang, Z.; Wang, L.; Zhang, H.; Chen, Q.; Liu, J. Distributed regular polygon formation control and obstacle avoidance for non-holonomic wheeled mobile robots with directed communication topology. *IET Control Theory Appl.* **2020**, *14*, 1113–1122. [[CrossRef](#)]
32. Wang, Y.; Wang, D.; Zhu, S. Formation tracking of multi-vehicle systems in unknown environments using a multi-region control scheme. *Int. J. Control.* **2017**, *90*, 2760–2771. [[CrossRef](#)]
33. Samson, C.; Borgne, M.L.; Espiau, B. *Robot Control: The Task Function Approach*, 1st ed.; Oxford Engineering Science Series; Oxford University Press: New York, NY, USA, 1991; Volume 22.
34. Obregon-Flores, J.; Arechavaleta, G.; Becerra, H.M.; Morales-Diaz, A. Predefined-Time Robust Hierarchical Inverse Dynamics on Torque-Controlled Redundant Manipulators. *IEEE Trans. Robot.* **2021**, *37*, 962–978. [[CrossRef](#)]

35. Rosales, C.; Leica, P.; Sarcinelli-Filho, M.; Carelli, R. 3D Formation Control of Autonomous Vehicles Based on Null-Space. *J. Intell. Robot. Syst.* **2016**, *84*, 453–467. [[CrossRef](#)]
36. Antonelli, G.; Arrichiello, F.; Chiaverini, S. Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 1173–1182. [[CrossRef](#)]
37. Arechavaleta, G.; Morales-Díaz, A.; Pérez Villeda, H.; Castelán, M. Hierarchical Task-Based Control of Multirobot Systems with Terminal Attractors. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 334–341. [[CrossRef](#)]
38. Acosta, J.F.; González de Rivera, G.; Andaluz, V.H.; Garrido, J. Multirobot Heterogeneous Control Considering Secondary Objectives. *Sensors* **2019**, *19*, 4367. [[CrossRef](#)]
39. Trujillo, M.A.; Becerra, H.M.; Gómez-Gutiérrez, D.; Ruiz-León, J.; Ramírez-Treviño, A. Priority task-based formation control and obstacle avoidance of holonomic agents with continuous control inputs. *IFAC-PapersOnLine* **2018**, *51*, 216–222. [[CrossRef](#)]
40. Zhu, H.; Juhl, J.; Ferranti, L.; Alonso-Mora, J. Distributed Multi-Robot Formation Splitting and Merging in Dynamic Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9080–9086.
41. Kim, J. Distributed Rendezvous of Heterogeneous Robots with Motion-Based Power Level Estimation. *J. Intell. Robot. Syst.* **2020**, *100*, 1417–1427. [[CrossRef](#)]
42. Becerra, H.M.; Colunga, J.A.; Romero, J.G. Simultaneous convergence of position and orientation of wheeled mobile robots using trajectory planning and robust controllers. *Int. J. Adv. Robot. Syst.* **2018**, *15*. [[CrossRef](#)]
43. Siciliano, B. Kinematic control of redundant robot manipulators: A tutorial. *J. Intell. Robot. Syst.* **1990**, *3*, 201–212. [[CrossRef](#)]
44. Antonelli, G.; Arrichiello, F.; Chiaverini, S. The null-space-based behavioral control for autonomous robotic systems. *Intell. Serv. Robot.* **2008**, *1*, 27–39. [[CrossRef](#)]
45. Lee, J.; Mansard, N.; Park, J. Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control. *IEEE Trans. Robot.* **2012**, *28*, 1260–1277. [[CrossRef](#)]
46. Aldana-López, R.; Gómez-Gutiérrez, D.; Trujillo, M.A.; Navarro-Gutiérrez, M.; Becerra, H.M. A predefined-time first-order exact differentiator based on time-varying gains. *Int. J. Robust Nonlinear Control* **2021**, *31*, 5510–5522. [[CrossRef](#)]
47. Miao, Z.; Liu, Y.H.; Wang, Y.; Yi, G.; Fierro, R. Distributed Estimation and Control for Leader-Following Formations of Nonholonomic Mobile Robots. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1946–1954. [[CrossRef](#)]