

Article

Bio-Inspired Autonomous Navigation and Formation Controller for Differential Mobile Robots

Alejandro Juarez-Lora ¹ and Alejandro Rodriguez-Angeles ^{2,*}

¹ Centro de Investigacion en Computacion del Instituto Politecnico Nacional, CIC-IPN, Ciudad de Mexico 07738, Mexico

² Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional, Cinvestav-IPN, Ciudad de Mexico 07360, Mexico

* Correspondence: aangeles@cinvestav.mx

Abstract: This article proposes a decentralized controller for differential mobile robots, providing autonomous navigation and obstacle avoidance by enforcing a formation toward trajectory tracking. The control system relies on dynamic modeling, which integrates evasion forces from obstacles, formation forces, and path-following forces. The resulting control loop can be seen as a dynamic extension of the kinematic model for the differential mobile robot, producing linear and angular velocities fed to the mobile robot's kinematic model and thus passed to the low-level wheel controller. Using the Lyapunov method, the closed-loop stability is proven for the non-collision case. Experimental and simulated results that support the stability analysis and the performance of the proposed controller are shown.

Keywords: autonomous navigation; multi-agent systems; decentralized control; bio-inspired control; reaction control; formation



Citation: Juarez-Lora, A.; Rodriguez-Angeles, A. Bio-Inspired Autonomous Navigation and Formation Controller for Differential Mobile Robots. *Entropy* **2023**, *25*, 582. <https://doi.org/10.3390/e25040582>

Academic Editor: César Cruz-Hernández

Received: 15 February 2023

Revised: 16 March 2023

Accepted: 21 March 2023

Published: 28 March 2023



Copyright: © 2023 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

Autonomous driving on a multi-agent system is a highly researched goal pursued in robotics. *Deliberative* control techniques, which require precise recognition and knowledge of the surroundings and well-defined hierarchical structures, show an efficient performance in controlled environments; see [1], where a broad review of path planning strategies is presented, including environment and robot interaction modeling. In contrast, *reactive* control techniques, based on a robust stimulation–response behavior, yield a favorable execution on unknown environments [2]. The location of the agents (mobile robots) about a global frame of reference, navigation in unknown environments, and the introduction of multiple robots in the same workspace make this a challenging problem to solve. Multiple mobile robots systems present new challenges compared to single vehicle control, such as the heterogeneity of the agents, mixed traffic, cooperative behaviors, collisions among vehicles, static obstacles, and even pedestrians [3,4].

Mobile robots are electro-mechanical devices that can move on the bi-dimensional plane, and this restriction leads to 3 degrees of freedom, which are its location and orientation coordinates. Mobile robots can be classified according to their grade of mobility and directional capabilities. Type (2, 0) or differential robots have two wheels with individual speeds and no directional actuator. Thus, differential mobile robots are underactuated systems subject to non-holonomic constraints, which hinders the design of the navigation controller. Although the navigation control of mobile robots is not new, novel applications and developments are studied, due to the recent technological advances and cost reduction in electronics, as well as more powerful, highly integrated but smaller computing devices [5,6]. In [5], the application of multi-robot systems for harvester assisting purposes is presented such that a cooperative co-working set of robots must follow the farmer to help with weeding, harvesting, crop scouting, etc., rendering a human–machine collaboration

system. Other applications for multi-agent systems and social robots navigation, such as delivery robots, warehouses, indoor service robots, surveillance robots, etc., are listed in [7]. An interesting application is presented in [8], where a multi-robot system is proposed for defect detection and location on large surface metal plates; it is pointed out that there are several trade-offs between the performance in terms of defect location accuracy and the number of robots, their navigation capabilities, and ability to keep a formation.

The implementation of multi-agent swarm systems usually involves the usage of several data acquisition devices, such as inertial sensors [9], simultaneous localization and mapping (SLAM) [10], onboard cameras for computer vision [11], and relative radio positioning [12], in order to provide data redundancy and diversity to model environmental conditions properly [13]. There are even biomimetic sensors designed for navigation purposes in multi-robot systems; see [14], where a micro lenses array is integrated to aperture and field diaphragms to emulate an insect compound eye and later is combined to SLAM techniques for the navigation of a set of outdoor light robots. In general, the amount of data processing is directly proportional to the number of agents in the system, increasing power consumption in onboard processing platform solutions. Centralized swarm navigation schemes usually share all their gathered data through all the agents in the system, making them utterly dependent on the communication system's reliability [15]. To overcome the burden of data required to create a map, in [6], they propose using OpenStreetMaps, which are user-generated maps, publicly available, whose information is combined with lidar-based Naive-Valley-Path generation methodologies to render a local path that is free of obstacle collisions with other vehicles and pedestrians, thus combining deliberative and reactive techniques, providing a complete outdoor autonomous navigation system for unstructured environments. Another application of integrating deliberative and reactive techniques is found in [16], where a navigation controller that uses SLAM, path-planning techniques and exteroceptive sensors is applied to small-scale vessels or surface vehicles.

On the other hand, some species in nature have developed effective ways to achieve evasion, following other members and shaping formation, yielding to behavior models of biological systems, such as schools of fish, flocks of birds, or crowd dynamics, which are examples of reactive control techniques, and can be used for its implementation on robotic systems, more specifically, unicycle differential robots. In these reactive models, the individual actions of each agent are defined toward the fulfillment of a global objective [17,18] so as to have the most number of animals in a reduced space or some species of birds that can make more complex formations toward a leader.

Some advances in the research of crowd dynamics allow, with a set of simple rules, to predict the agents' behavior. For instance, in [19], potential functions and panel methods are embedded into an algorithm to create a collision-free path for differential agents. The simplicity, ease of implementation, and low computational requirements make this modeling approach effective in robot navigation. However, potential functions may suffer from several problems, such as local minima traps, dead locks between close obstacles, oscillations in the presence of obstacles, and inside narrow passages.

Another kind of solution is obtained from observing the animal's conduct when moving and interacting with each other, according to its goal. In [20,21], Helbing et al. (2000, 2005) analyzed crowd dynamics, describing a model of displacement behavior; if an individual wants to move to a desired point, it moves in the shortest path with the most comfortable speed; when an obstacle, like a wall or another individual appears, the individual starts evading it as soon as a comfort zone is invaded, i.e., a private space. This zone is preferred not to be invaded, so the individual tries to keep it clear.

There are other investigations, such as [17], which set navigation rules with the purpose of the agents reaching geometric shape formations regarding the position of the other agents in a sort of coordinated behavior. In humans, for example, making a line or a circle implies knowing the location of everybody else in the group such that everyone sets their position on the formation.

More recently, reinforcement learning (RL) and deep reinforcement learning (DRL) techniques have been applied for the navigation of autonomous vehicles; [4,7] present surveys of advances on applying RL and DRL to problems such as obstacle avoidance, indoor navigation, multi-robot navigation, and social navigation, considering heterogeneous fleets, unmanned vehicles, aerial vehicles and ships, and possible interaction between agents and humans. They point out that the results highly depend of the degree of cooperation and shared information between the agents. Particular applications of RL to the case of groups of autonomous vehicles give rise to the so-called multi-agent reinforcement learning (MARL), which is a more distributed framework in which several agents simultaneously learn cooperative or competitive behaviors [22]. This approach is being applied to mixed traffic problems and heterogeneous group of agents.

In this article, a decentralized control strategy is designed, allowing autonomous driving for a multi-agent system, using behavioral models to enable the trajectory tracking and collision avoidance of dynamic and static obstacles, based on the relative approaching velocity. A desired formation is simultaneously achieved by establishing some navigation restrictions, in this case, and without loss of generality, a circular formation. The objective of the research is then to use these models and to adapt them in a control strategy which allows a decentralized, autonomous navigation system.

In Section 2, the specifications of the control proposal are described, while in Section 3, the Lyapunov's stability test is applied on the closed-loop system, for the non-collision case, to obtain stability conditions to be satisfied by the control gains. In Section 4, the development of an experimental platform is shown. Section 5 compares the simulation and experimental results. Section 6 gives the conclusions of this work.

2. Design of the Autonomous Navigation Control Law

A differential mobile robot type (2, 0) has the kinematic model described in (1), where $[x_i, y_i]$ are the coordinates of the rotation center regarding a fixed frame of reference, and θ_i is its orientation, as shown in Figure 1. Subscript i identifies the i -th robot in a multi-agent system. For this model, the control inputs are the translational velocity V_i and rotational velocity W_i , [23].

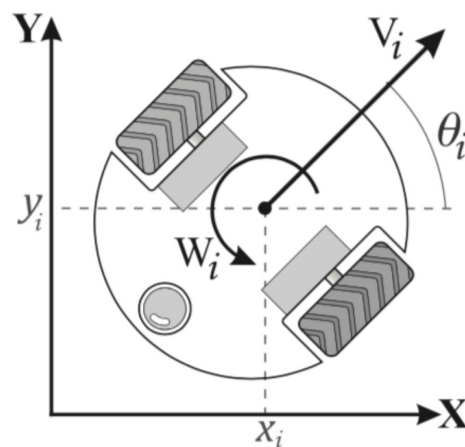


Figure 1. Type (2, 0) Mobile robot.

$$\begin{aligned}\dot{x}_i &= V_i \cos \theta_i \\ \dot{y}_i &= V_i \sin \theta_i \\ \dot{\theta}_i &= W_i\end{aligned}\quad (1)$$

As mentioned in the introduction, the proposed controller integrates a dynamic model based on reactive forces into the kinematic model, controlled by the translational velocity V_i and rotational velocity W_i . The integration can be depicted in a nested control scheme formed by an external and internal controller. The *external* control loop contains the

bio-inspired navigation model, which is a driven force dynamic model, whose output is processed by the *internal* control loop that corresponds to the kinematic model and is driven by the translational V_i and rotational velocity W_i . This interconnection is shown in Figure 2. In the following, the integrated controller is explained in detail.

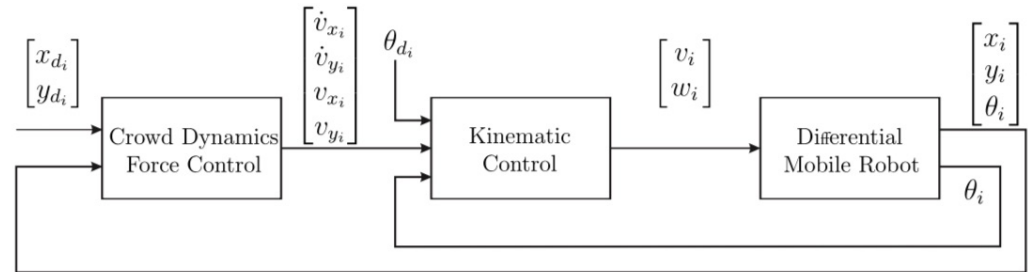


Figure 2. Scheme of the integrated control model.

2.1. External Control Loop

This control loop consists of the bio-inspired algorithm, which is built in a series of modifications of Helbing's crowd dynamics model, presented in [20,21], which portrays the behavior of an individual of mass m_i and velocity $v_i = [v_{x_i}, v_{y_i}]$ given by a set of socio-psychological and physical rules. The individual's position $p_i = [x_i, y_i]$ moves toward a reference point or goal position $p_{d_i} = [x_{d_i}, y_{d_i}]$. The desired individual's velocity, denoted as the vector v_i^0 , is reached in a characteristic transition time τ_i , and points towards a direction given by the position error vector $e_i = [x_{d_i} - x_i, y_{d_i} - y_i]$, therefore moving in a straight line to the goal position. At the same time, it intends to avoid obstacles and other individuals which invade its comfort zone r_i . Evasion forces for dynamic and static obstacles are denoted as f_{ij} and f_{iw} , respectively. This dynamic force model, given by (2), is then composed by attracting forces related to the goal position and repulsive forces based on the distance to obstacles, yielding a reactive control.

$$m_i \begin{bmatrix} \dot{v}_{x_i} \\ \dot{v}_{y_i} \end{bmatrix} = \frac{m_i}{\tau_i} (v_i^0 e_i - v_i) + \sum f_{ij} + \sum f_{iw} \quad (2)$$

The obstacle repulsive forces f_{ij} and f_{iw} are modeled by (3) and (4), and correspond to the evasion of dynamic or static obstacles, respectively, considering as dynamic obstacles other agents that get into the comfort zone of the i -th robot. These repulsive forces are based on relative distances that onboard sensors can quickly obtain:

$$f_{ij} = k_i g(r_{ij} - d_{ij}) n_{ij} - \kappa_i g(r_{ij} - d_{ij}) \Delta v_{ji}^t t_{ij} \quad (3)$$

$$f_{iw} = k_i g(r_i - d_{iw}) n_{ij} - \kappa_i g(r_i - d_{iw}) (v_i \cdot t_{iw}) t_{iw} \quad (4)$$

This model has terms as the distance between the mass center of agent i and j denoted by $d_{ij} = \|p_i - p_j\|$, the normalized vector of direction $n_{ij} = [n_{ijx}, n_{ijy}] = (p_i - p_j)/d_{ij}$, which points from the agent j toward agent i , the tangential vector $t_{ij} = [-n_{ijy}, n_{ijx}]$, the term g , which determines if the comfort zone of agent i interacts with the agent j comfort zone, i.e., $g = 1$ if $r_{ij} > d_{ij}$ (where $r_{ij} = r_i + r_j$, with r_i, r_j the radii of the comfort zones) and $g = 0$ otherwise. The difference in translational velocity vectors between the i and j agents represents a relative approaching velocity, and it is denoted by $\Delta v_{ji}^t = (v_j - v_i) \cdot t_{ij}$. The magnitude of the positive gains k and κ determines the influence of the normal and tangential components of acceleration, which permits modulating the response of acceleration and intensity of twist to avoid collisions. The repulsion forces against static obstacles f_{iw} comes as setting $v_j = 0$ in (3), with d_{iw} being the distance between the i agent and the obstacle once it gets into the agent's comfort zone.

The dynamic model (2) is used for designing a dynamic control law at acceleration level driven by forces, whose states are fed to the kinematic model of the mobile robots as depicted in Figure 2. For this purpose, consider that the mass m_i is unitary, and any obstacle, either dynamic or static, can be treated with the same evasion function, taking into consideration only its relative approaching velocity. Furthermore, friction forces are let out of the model as well as slippery forces in order to simplify the control design; nevertheless, taking into consideration such forces would improve the trajectory tracking performance. With all of the above considerations and based on the control objectives, the control law given by (5) is proposed:

$$\dot{v}_i = \begin{bmatrix} \dot{v}_{x_i} \\ \dot{v}_{y_i} \end{bmatrix} = k_p e_i + k_d \dot{e}_i + \sum f_{ij} + f_{ic} \quad (5)$$

where in order to achieve trajectory tracking, a PD controller is included with tuning gains k_p, k_d and fed by the position error (6); meanwhile, the total repulsive forces are the vector sum of all avoiding collision forces f_{ij} , related to dynamic and static obstacles that are inside the comfort zone, taking into account that the relative approaching velocity for dynamic obstacles (other vehicles) is given by $\Delta v_{ji}^t = (v_j - v_i) \cdot t_{ij}$, while for static obstacles, the approaching velocity corresponds only to the velocity of the agent itself as shown on the last terms of Equations (3) and (4). The term f_{ic} allows enforcing a desired geometric formation of the multi-agent system, then the position of each agent in the formation must be congruent with satisfying the desired trajectory of the agent. Otherwise, a conflict would arise, and none of the goals, nor trajectory tracking, nor formation would be satisfied. By definition of f_{ij} , several simple geometric shape formations may be generated [17], such as straight lines, arrow shapes, circles, etc.:

$$e_i = \begin{bmatrix} e_{x_i} \\ e_{y_i} \end{bmatrix} = \begin{bmatrix} x_{d_i} - x_i \\ y_{d_i} - y_i \end{bmatrix} \quad (6)$$

Based on the Cucker–Smale model modifications presented in [18], the function f_{ic} is defined as in (7) that corresponds to a circular formation:

$$f_{ic} = \gamma \left(1 - \frac{R}{\|d_{ic}\|} \right) d_{ic} \quad (7)$$

The function f_{ic} pretends to form a circle of radius R with the N agents of the system, regarding the geometric center (\bar{x}, \bar{y}) , which is given by the average position of the whole system, (8). The distribution of each agent on the circle formation takes into account the desired trajectory for each agent, as would be shown at the presented results:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (8)$$

In function (7), $d_{ic} = [\bar{x} - x_i, \bar{y} - y_i]^T$ is the vector distance from agent i toward the geometric center, and γ is a positive gain for tuning. By evaluating the control proposed at (5) and integrating it, the values $(\dot{v}_{x_i}, \dot{v}_{y_i}, v_{x_i}, v_{y_i})$ are obtained.

2.2. Internal Control Loop

This control loop transforms the velocities and accelerations obtained by the *external control loop* into the linear and angular velocities V_i and W_i needed to drive the vehicle. To achieve this, it is considered the work at [24], obtaining the relations given in (9) and (10):

$$V_i = k_v [v_{x_i} \cos(\theta_i) + v_{y_i} \sin(\theta_i)] \quad (9)$$

$$W_i = k_a \left[\frac{\dot{v}_{y_i} v_{x_i} - \dot{v}_{x_i} v_{y_i}}{v_{x_i}^2 + v_{y_i}^2 + \epsilon} \sin_c(e_{\theta_i}) - k_t e_{\theta_i} \right] \quad (10)$$

where k_v is a gain which modulates the intensity of the control signal V_i . The constant $\epsilon \approx 0$ is added in order to avoid singularities when v_{x_i}, v_{y_i} are zero. The function $\sin_c(\cdot)$ is considered the *cardinal sine*, and it is defined in (11):

$$\sin_c(\alpha) = \begin{cases} \frac{\sin(\alpha)}{\alpha} & \text{if } \alpha \neq 0 \\ 1 & \text{if } \alpha = 0 \end{cases} \quad (11)$$

Since it is intended to achieve a desired orientation, a proportional control is added with $e_{\theta_i} = (\theta_i - \theta_{d_i})$ as the orientation error regarding an angle of reference θ_{d_i} . The value of k_a, k_t modulates the control actions given by the orientation control.

3. Stability Analysis for the Non-Collision Case

The avoidance collision force term f_{ij} represents repulsive forces for dynamic and static obstacles, depending on the relative approaching velocity. Therefore, such forces depend on each possible scenario that the agent may encounter, and thus the stability analysis is carried out for the free collision case. From the simulation and experimental tests, it is concluded that the proper tuning of repulsive forces will not affect convergence to the desired position as far as there is not an obstacle inside the comfort zone, generating a conflict for the agent being in its desired position.

Using Lyapunov's stability test, setting $f_{ij} = 0$ for the collision-free case and taking into consideration the values of interest $(x_i, y_i, v_{x_i}, v_{y_i}, \theta_i)$ and their desired references $(x_{d_i}, y_{d_i}, v_{x_{d_i}}, v_{y_{d_i}}, \theta_{d_i})$, the next state variables are defined:

$$\begin{aligned} z_{1_i} &= e_{x_i} = x_{d_i} - x_i & z_{2_i} &= e_{y_i} = y_{d_i} - y_i \\ z_{3_i} &= e_{v_{x_i}} = v_{x_{d_i}} - v_{x_i} & z_{4_i} &= e_{v_{y_i}} = v_{y_{d_i}} - v_{y_i} \\ z_{5_i} &= e_{\theta_i} = \theta_i - \theta_{d_i} \end{aligned} \quad (12)$$

The closed loop for the i -th agent is given by

$$\begin{aligned} \dot{z}_{1_i} &= v_{x_{d_i}} \sin^2 \theta_i + (z_{4_i} - v_{y_{d_i}}) \sin(\theta_i) \cos(\theta_i) + z_{3_i} \cos^2(\theta_i) \\ \dot{z}_{2_i} &= v_{y_{d_i}} \cos^2(\theta_i) + (z_{3_i} - v_{x_{d_i}}) \sin(\theta_i) \cos(\theta_i) + z_{4_i} \sin^2(\theta_i) \\ \dot{z}_{3_i} &= \dot{v}_{x_{d_i}} - k_p z_{1_i} - k_d z_{3_i} - \gamma \beta_{i_x} \\ \dot{z}_{4_i} &= \dot{v}_{y_{d_i}} - k_p z_{2_i} - k_d z_{4_i} - \gamma \beta_{i_y} \\ \dot{z}_{5_i} &= -\dot{\theta}_{d_i} - k_a k_t z_{5_i} + k_a \left[\frac{(\dot{v}_{y_{d_i}} - \dot{z}_{4_i})(v_{x_{d_i}} - z_{3_i}) - (v_{y_{d_i}} - z_{4_i})(\dot{v}_{x_{d_i}} - \dot{z}_{3_i})}{(v_{x_{d_i}} - z_{3_i})^2 + (v_{y_{d_i}} - z_{4_i})^2 + \epsilon} \left(\frac{\sin(z_{5_i})}{z_{5_i}} \right) \right] \end{aligned} \quad (13)$$

With $\dot{\theta}_{d_i} = W_{d_i}$ as angular velocity reference and

$$\beta_i = \begin{bmatrix} \beta_{i_x} \\ \beta_{i_y} \end{bmatrix} = \left(1 - \frac{R}{\sqrt{(\bar{x} - x_{d_i} + z_{1_i})^2 + (\bar{y} - y_{d_i} + z_{2_i})^2}} \right) \begin{bmatrix} \bar{x} - x_{d_i} + z_{1_i} \\ \bar{y} - y_{d_i} + z_{2_i} \end{bmatrix} \quad (14)$$

Using the Lyapunov candidate function $V_i(z_i) = \frac{1}{2}z_i^T z_i \geq 0$, the time derivate of $V_i(z_i)$ is

$$\begin{aligned} \dot{V}_i(z_i) = & [\cos^2\theta_i - k_p]z_{1i}z_{3i} + [\sin^2\theta_i - k_p]z_{2i}z_{4i} \\ & + \left[\sin\theta_i\cos\theta_i + \frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right] z_{1i}z_{4i} \\ & + \left[\sin\theta_i\cos\theta_i - \frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right] z_{2i}z_{3i} \\ & - \left[\sin\theta_i\cos\theta_i + \frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right] z_{1i}v_{y_{d_i}} \\ & - \left[\sin\theta_i\cos\theta_i + \frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right] z_{1i}v_{x_{d_i}} \\ & + \left[z_{1i}\sin\theta_i + \gamma\beta_{iy} \left(\frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right) \right] v_{x_{d_i}} \\ & + \left[z_{2i}\cos\theta_i - \gamma\beta_{ix} \left(\frac{k_a k_p \sin(z_{5i})}{(v_{x_{d_i}} - z_{3i})^2 + (v_{y_{d_i}} - z_{4i})^2 + \epsilon} \right) \right] v_{y_{d_i}} \\ & - k_d [z_{3i}^2 + z_{4i}^2] - k_a k_t z_{5i}^2 - z_{5i}W_{d_i} \end{aligned} \quad (15)$$

In order to prove $\dot{V}_i(z_i) < 0$, we can upper bound it considering the algebraic properties $(a^2 + b^2) \geq 0$ and $\frac{1}{2}(a^2 + b^2) \geq -ab$. Furthermore, it is worth mentioning that the motors produce a maximal velocity and acceleration, which are positive and bounded, regardless of its spin direction. So, it results in $|v_{x_{d_i}}| < v_{x_{i_{max}}}$, $|v_{y_{d_i}}| < v_{y_{i_{max}}}$ and $|\dot{v}_{x_{d_i}}| \leq \dot{v}_{x_{i_{max}}}$, $|\dot{v}_{y_{d_i}}| \leq \dot{v}_{y_{i_{max}}}$. Replacing the bounds of the trigonometrical functions in (15) given by $-0.2 \leq \cos^2\theta_i + \sin\theta_i\cos\theta_i \leq 1.2$, $-0.2 \leq \sin^2\theta_i + \sin\theta_i\cos\theta_i \leq 1.2$, $|\sin(z_{5i})| \leq 1$ and considering the worst-case operation scenario, the function can be written as

$$\begin{aligned} \dot{V}(z_i) \leq & \frac{1}{2} \left[k_p \left(1 - \frac{k_a}{\epsilon} \right) + 0.2 \right] z_{1i}^2 + \frac{1}{2} \left[k_p \left(1 + \frac{k_a}{\epsilon} \right) + 0.2 \right] z_{2i}^2 \\ & + \frac{1}{2} \left[k_p \left(1 + \frac{k_a}{\epsilon} \right) + 0.2 \right] z_{3i}^2 + \frac{1}{2} \left[k_p \left(1 - \frac{k_a}{\epsilon} \right) + 0.2 \right] z_{4i}^2 \\ & + \left[v_{x_{i_{max}}} + \left(\frac{1}{2} - k_p \frac{k_a}{\epsilon} \right) v_{y_{i_{max}}} \right] z_{1i} + \left[v_{y_{i_{max}}} + \left(\frac{1}{2} - k_p \frac{k_a}{\epsilon} \right) v_{x_{i_{max}}} \right] z_{2i} \\ & + \left[\dot{v}_{x_{i_{max}}} - \gamma\beta_{ix} - \left(\gamma\beta_{iy} + k_d v_{y_{i_{max}}} \right) \frac{k_a}{\epsilon} \right] z_{3i} \\ & + \left[\dot{v}_{y_{i_{max}}} - \gamma\beta_{iy} + \left(\gamma\beta_{ix} + k_d v_{x_{i_{max}}} \right) \frac{k_a}{\epsilon} \right] z_{4i} \\ & + \frac{k_a}{\epsilon} \left[\gamma\beta_{iy} v_{x_{i_{max}}} - \gamma\beta_{ix} v_{y_{i_{max}}} \right] - k_d [z_{3i}^2 + z_{4i}^2] - k_a k_t z_{5i}^2 - z_{5i}W_{d_i} \end{aligned} \quad (16)$$

According to (7), the best-case scenario is when agent i is in a circular formation, i.e., $\|d_{ic}\| = R$, yielding to $\|\beta_i\| = 0$. The worst-case scenario is $\|d_{ic}\| > R$, which means the agent i is outside the circumference and $\|\beta_i\| \neq 0$. These considerations lead to the next list of conditions, which accomplish $\dot{V}_i(0) < 0$:

1. $k_p > 0, k_d > 0, k_a > 0, k_t > 0, \epsilon > 0, \gamma > 0$;
2. $k_p > \frac{0.2}{(1 - \frac{k_a}{\epsilon})}$;
3. $k_p > \frac{\epsilon}{k_a} \left(\frac{v_{x_{i_{max}}}}{v_{y_{i_{max}}}} + \frac{1}{2} \right)$;
4. $k_d > \frac{\dot{v}_{x_{i_{max}}}}{v_{y_{i_{max}}}} \frac{\epsilon}{k_a}$;
5. $\gamma > \frac{\frac{k_a}{\epsilon} k_d v_{y_{i_{max}}} - \dot{v}_{x_{i_{max}}}}{1 + \frac{k_a}{\epsilon}}, \gamma < \frac{\dot{v}_{y_{i_{max}}} + \frac{k_a}{\epsilon} k_d v_{x_{i_{max}}}}{1 + \frac{k_a}{\epsilon}}$.

Under the conditions given in (17), the closed-loop system is asymptotically stable only when $\|\beta_i\| = 0$ as $V_i(0) = 0$, $V_i(z_i) > 0$, $\dot{V}_i(0) = 0$ and $\dot{V}_i(z_i) < 0$. Nevertheless, when $\|\beta_i\| \neq 0$, there are unvanishing terms, which makes $\dot{V}_i(0) \neq 0$ and only retrieves practical stability, known as uniformly ultimately bounded (UUB) stability.

To perform stability analysis considering possible collisions with dynamic and static obstacles, each different reaction force for each possible situation during the navigation of the agent should be properly modeled, which is a cumbersome task. Nevertheless, simulation and experimental tests showed that tuning the reaction forces for dynamic obstacles would also work for static obstacles since for the last ones, the relative approaching velocity

is smaller than that of dynamic obstacles, thus requiring less aggressive evasion actions. Then, k, κ in expression (3) are tuned so the repulsive forces can produce a quick response to avoid collisions with dynamic obstacles; the same tuning is used for static obstacles, while maintaining the system's stability, assuming bounded and differentiable perturbations.

4. Experimental Platform Considerations

In order to prove the proposed control law, a MATLAB® [25] simulation and a physical implementation are prepared. During experimentation, four Turtlebot3 Burger® and one Turtlebot3 Waffle Pi® are used, giving a total of 5 agents, individually controlled using ROS in an onboard Raspberry Pi®, where the external control law is deployed, computing the control inputs V_i, W_i for each agent, delivered to the onboard OpenCR® power stage card through serial protocol. The description of the hardware and the programmed nodes in ROS is available in Figure 3.

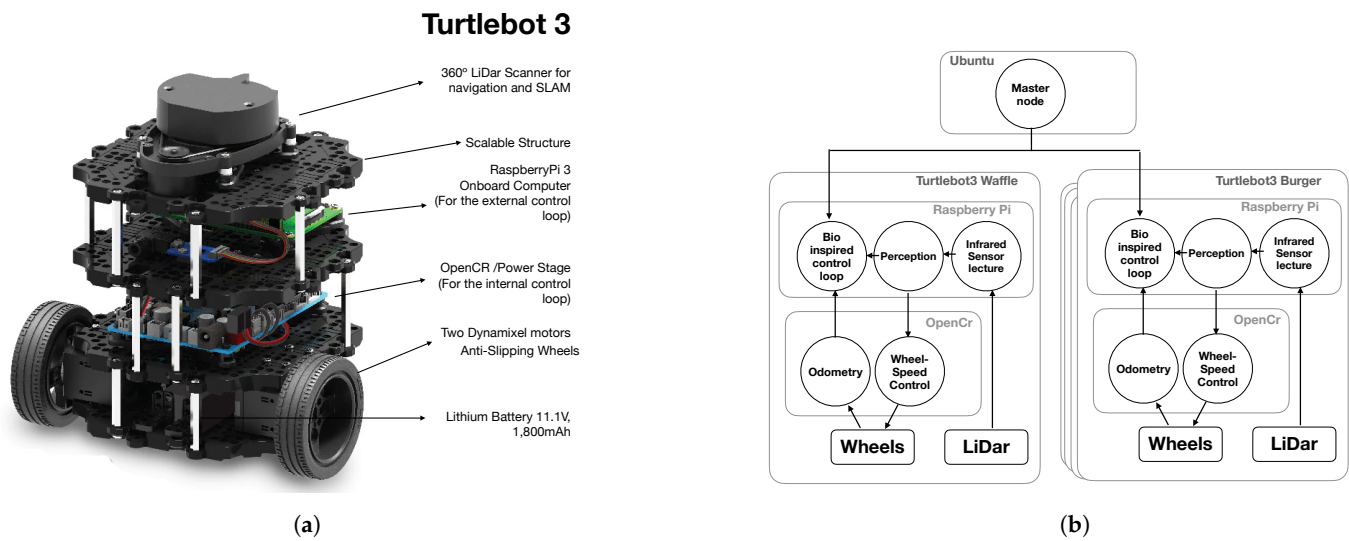


Figure 3. Description of the experimental platform. (a) Structure of the used differential re-configurable robot Turtlebot3, (b) coded nodes using ROS for the deployment of the internal and external control loops. Each rectangle represents a device, while each circle represents a coded node in ROS.

The initial position of each robot, on the same inertial global frame, is given, so each agent, using odometry, can compute its position on each given time. The used algorithms to obtain parameters needed in the control loop are described next.

4.1. Trajectory Reference Generation

The function in (18) retrieves a lemniscate figure, and it can be computed to obtain the reference position, velocity and acceleration of each agent $x_{d_i}(t), y_{d_i}(t)$:

$$\begin{aligned} x_{d_i}(t) &= a \cos(k_s t) + R \cos\left(\frac{360(i)}{N}\right) & y_{d_i}(t) &= b \sin(2k_s t) + R \sin\left(\frac{360(i)}{N}\right) \\ \dot{x}_{d_i}(t) &= -a \cdot k_s \cdot \sin(k_s t) & \dot{y}_{d_i}(t) &= 2b \cdot k_s \cdot \cos(2k_s t) \\ \ddot{x}_{d_i}(t) &= -a \cdot k_s^2 \cdot \cos(k_s t) & \ddot{y}_{d_i}(t) &= -4bk_s^2 \cdot \sin(k_s t) \end{aligned} \quad (18)$$

Here, a, b are the length and width of the lemniscate, R is the formation circle radius, N is the number of agents, the sub-index i refers to the i -th agent, and k_s is the temporary exchange rate.

The trajectory reference has a duration d , where if $t < d$, the function and its time derivatives are computed. If $t > d$, the last computed value on $t = d$ is used in order to obtain the final position coordinates of the trajectory, with reference velocity equal to zero.

Once the references are obtained, the expressions in (19)–(21) are used to obtain $\theta_{d_i}, V_{d_i}, W_{d_i}$:

$$\theta_{d_i} = \tan^{-1}\left(\frac{\dot{x}_{d_i}}{\dot{y}_{d_i}}\right) \quad (19)$$

$$V_{d_i} = \sqrt{\dot{x}_{d_i}^2 + \dot{y}_{d_i}^2} \quad (20)$$

$$W_{d_i} = \frac{\dot{x}_{d_i}\dot{y}_{d_i} - \dot{y}_{d_i}\dot{x}_{d_i}}{v_{d_i}^2 + \epsilon} \quad (21)$$

4.2. Perception

The agent's comfort zone consists of a continuous area around the agent. This area can be set as several shapes, which matches the surrounding obstacles. For simplicity, for the analysis performed in this work, it is chosen to be circular, assuming that potential obstacles in the surrounding space of the reactive behavior of the robot should conform to the circular-shaped area, as shown in a wide variety of cases in nature, e.g., in agriculture, as the robotic solution proposed in [5].

To measure the distance d_{ij} from the agent i to the agent j , and the angle θ_{ij} about the agent i 's translational axis, where agent j is located, it is necessary to implement the comfort zone properly. An infrared sensor emits a beam of light, which, when reflected on an object's surface, enables measurement of the distance from where it is located. Setting multiple infrared sensors around the agent is impractical due to the poor retrieved resolution due to the number of beams on the circumference, which is limited to the available space. This can be observed in Figure 4a, where an obstacle is not detected despite being in the comfort zone. A lidar system allows increasing the resolution, spinning an infrared beam to cover all the circumference. The LDS-01, used in this work, has a 1/360 resolution, scilicet, emits a beam each 0.0174 [rad] [26]. This is a 360-beam configuration around the agent and improves the environment perception, as many beams are now reflected in a single object.

In this event, the shortest registered distance is set as d_{ij} , while θ_{ij} is set as the average angle of all the triggered beam lights. When there are multiple beams activated, but these are not consecutive, it is considered to be multiple-obstacle detection, as shown in Figure 4b.

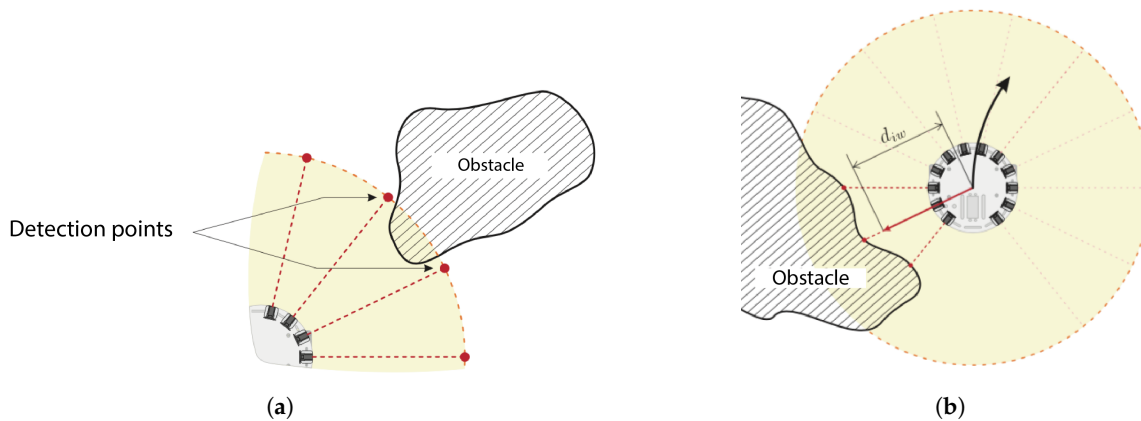


Figure 4. Detection scenarios regarding sensing. (a) Detection system issue, (b) multiple consecutive beam lights triggered.

4.3. Position and Velocity Estimation of the Other Agents

For computing collision avoidance function f_{ij} given by (3) and used in (5), the position and velocity of the j -th agent are needed. These are denoted as $p_j = [x_j, y_j]^T$ and $v_j = [\dot{x}_j, \dot{y}_j]^T = [v_{x_j}, v_{y_j}]^T$, and in order to compute them, the decentralized estimation algorithm starts from these suppositions:

- Each agent knows its own position regarding a global reference frame.
- All the obstacles detected in a radio detection R_d are considered agents.
- The comfort zone of the agent j is assumed to be the same radius as agent i .

The next algorithm is executed:

1. Agent i determines with its sensors the vector to the agent j , which is composed by the distance d_{ij} , and the angle from the global x axis frame, denoted as ρ_{ij} .
2. To obtain the ρ_{ij} angle, the orientation of the agent i in its inertial frame θ_i and the angle of the agent j detected by the sensor, denoted as θ_{ij} , are needed.
3. While θ_i increases, the value of θ_{ij} decreases, as shown in Figure 5a. This yields the value ρ_{ij} given by

$$\rho_{ij} = \theta_i + \theta_{ij} \quad (22)$$

4. Agent j 's position on the global reference frame is obtained by

$$x_j = x_i + d_{ij}\sin(\rho_{ij}), \quad y_j = y_i + d_{ij}\cos(\rho_{ij}) \quad (23)$$

5. The geometric center is computed by

$$\bar{x} = \frac{1}{N_i} \sum_{i=1}^{N_i} x_i, \quad \bar{y} = \frac{1}{N_i} \sum_{i=1}^{N_i} y_i \quad (24)$$

where N_i is the number of agents detected by the i -th agent. The fact that $N_i \neq N$ is possible must be remarked, due to these reasons:

- If the agent j is out of the sensor coverage area of the agent i ;
- If agent k is behind the agent j , agent i will not be detected.

Note: This only applies for the experimentation platform. In simulation, all variables are available, which disables the scenario depicted in Figure 5b.

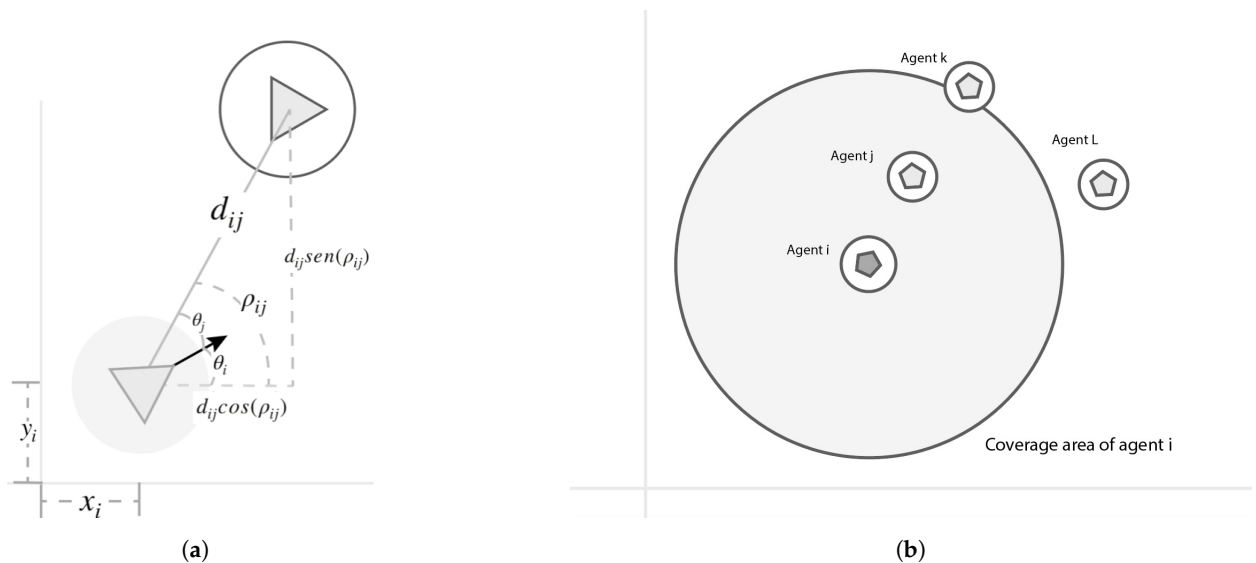


Figure 5. Detection scenarios regarding positioning. (a) Position estimation scheme. (b) Agent i cannot detect the agent k , as it is behind agent j . Agent L cannot be detected because it is outside the agent i coverage area.

Finally, to determine the velocity of agent j , the distance d_{ij} is compared with the distance on the previous execution step, denoted as $d_{ij[t-1]}$. The expression (25) obtains the velocity of agent j :

$$v_j = \frac{d_{ij} - d_{ij[t-1]}}{t - t_{-1}} \quad (25)$$

To obtain the velocity on its axis components and to evaluate the tangential velocity Δv_{ji}^t , the next functions are evaluated:

$$v_{x_j} = v_j \cos(\rho_{ij}), \quad v_{y_j} = v_j \sin(\rho_{ij}) \quad (26)$$

5. Simulation and Experimental Results

At simulation, comparison studies for the case of trajectory tracking and collision avoidance were carried out, considering other techniques proposed in the literature, such as potential repulsive fields, and the geometric obstacle avoidance control method (GOACM), obtaining similar performance; however, there is not a completely similar strategy that integrates formation, trajectory tracking, and collision avoidance of dynamic and static obstacles, for which full comparison of our proposed strategy could be done. Then, for the sake of space and considering that such comparison studies do not represent a fair layout, such results are not presented here.

The experimental configuration consists of a set of $N = 5$ agents, which has to follow a lemniscate trajectory given by (18), with the parameters on Table 1 and an execution period of $t = 100$ [s]. Considering the stability conditions given by (17), the upper bound for the control actions is chosen according to the specification of the TurtleBot 3. This is $V_{max} = 0.22$ [m/s] and $W_{max} = 2.84$ [rad/s], [26], and the tuning of the control gains is shown in Table 2. Lastly, the initial conditions of each agent are shown in Table 3.

Table 1. Specifications for the desired trajectory.

a	b	d	k_s	r_i	R	R_d
1 [m]	0.5 [m]	85 [s]	0.15 [m/s]	0.4 [m]	0.6 [m]	1.0 [m]

Table 2. Tuning gains used on the control loop.

k_p	k_d	k_a	k_v	k_t	γ	k	κ	ϵ
4	4	0.5	0.995	2.3	0.49	5	13.5	0.005

Table 3. Initial position of each agent.

	x_i	y_i	θ_i
Agent 1	0.5 m	−0.5 m	$\pi/2$
Agent 2	−0.5 m	−0.5 m	$\pi/4$
Agent 3	0.0 m	−0.5 m	0
Agent 4	1.0 m	−0.5 m	$-\pi/2$
Agent 5	1.5 m	−0.5 m	0

Simulation and experimental results are presented under the same conditions of desired trajectory and tuning gains for comparison purposes. It is essential to emphasize that simulations are an ideal case. At the same time, during experiments, odometry and lidar measurements are used to determine each agent's location and distance to possible obstacle collision. For the sake of space, only results that involve trajectory tracking, a desired circular formation and possible collisions between agents, i.e., dynamic obstacles, are presented. Since the proposed avoidance collision strategy is based on the relative approaching velocity, it is clear that more aggressive evasion actions are required for dynamic obstacles than for static ones because for dynamic obstacles the relative approaching velocity is higher than that of the proper agent, while for a static obstacle, the approaching velocity corresponds to that of the agent. Collision avoidance among agents is present at the transient, between 0 and 8 s, both in the simulation and experimental results because the initial position of the agents imply possible collisions while trying to get into their assigned position at the formation. Nevertheless, several cases considering static obstacles were tested both by simulation and experiments, showing good obstacle avoidance.

First, the simulation results are shown. Figure 6a shows the geometric center position of the whole system, compared with the ideal group formation reference. A better convergence is exhibited in sections where the curvature radius is prominent. Figure 6b shows the correspondent convergence error, and it can be deduced that for $t > 85$ [s], the

geometric center position converges to a standstill position. All the agents converge to the same external and internal control loop action, as shown in Figure 7a,b, thus moving in a synchronized way, while tracking the desired lemniscate trajectory.

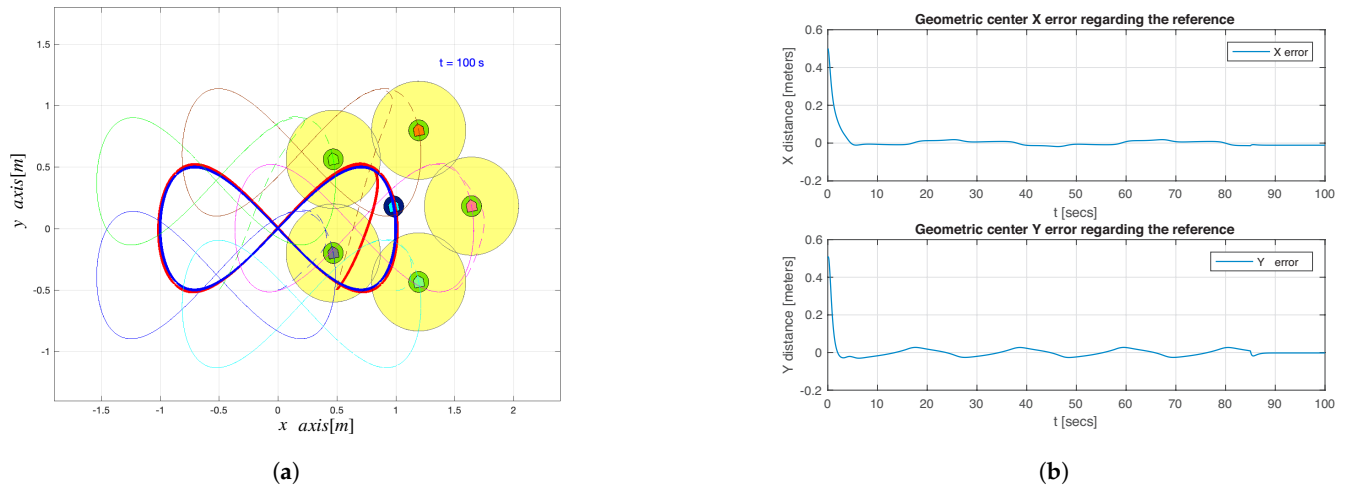


Figure 6. Detection scenarios regarding positioning. (a) Geometric center position (Simulation), (b) trajectory tracking error of the geometric center (Simulation).

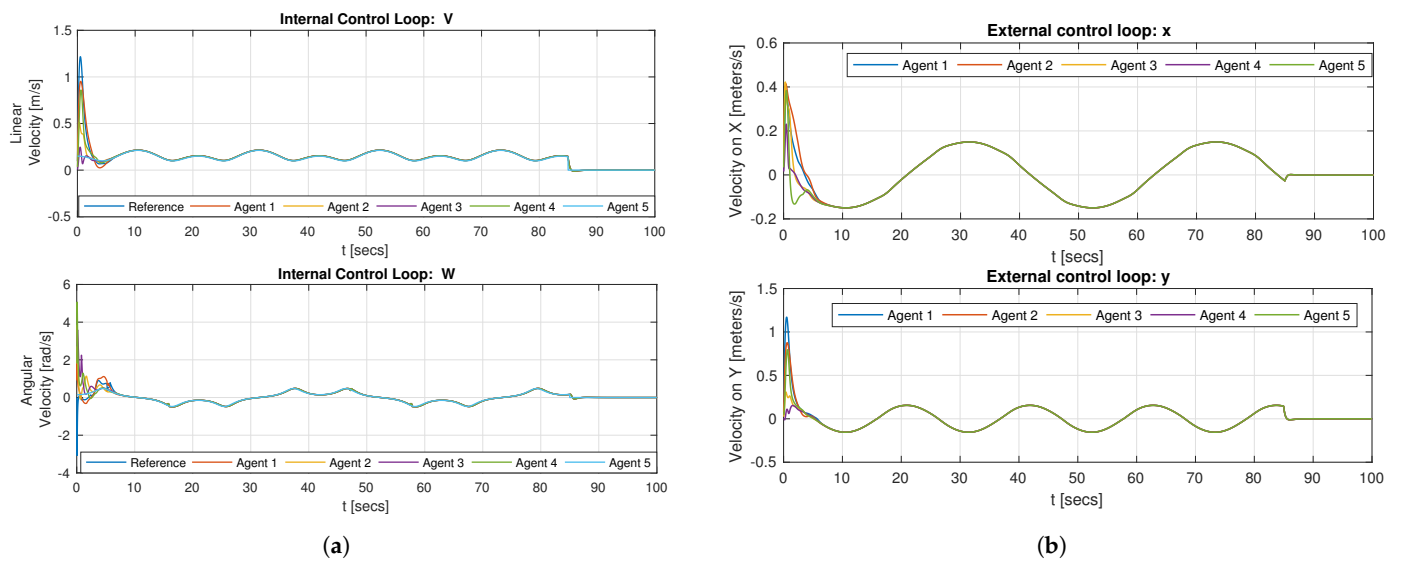


Figure 7. Control signals in simulation setup. (a) Internal control loop variables (Simulation), (b) external control loop variables (Simulation).

As for the experimental setup, technical difficulties and limitations are evident in Figures 8 and 9, where noise and abrupt changes are present. This is because of the lidar measurements and poor odometry location.

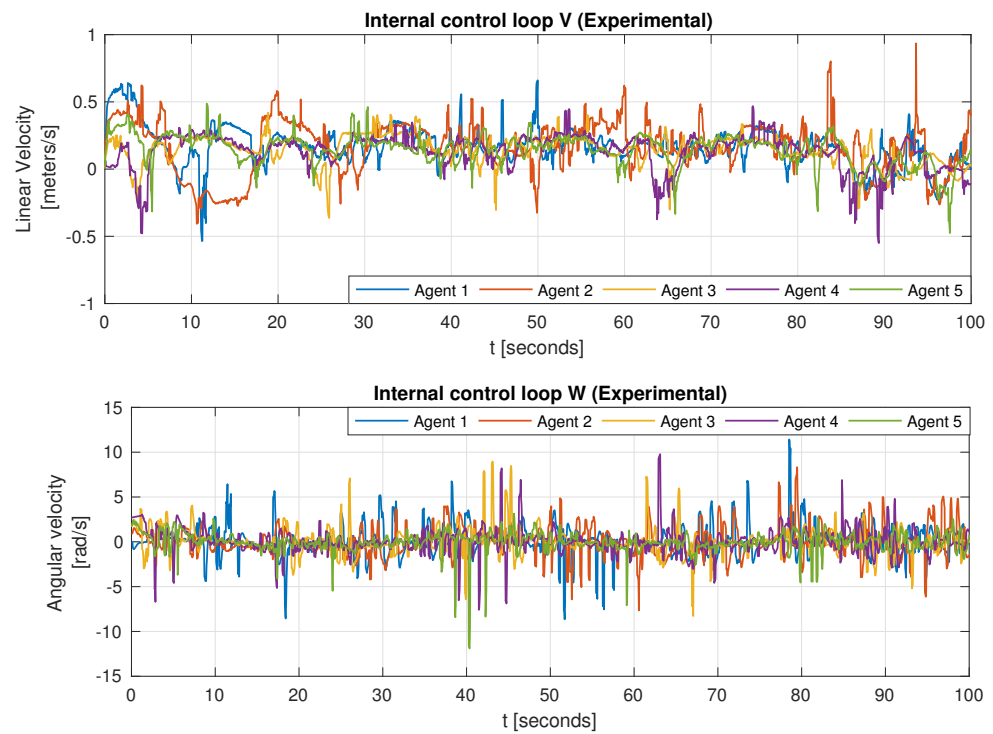


Figure 8. Internal control loop (experimental).

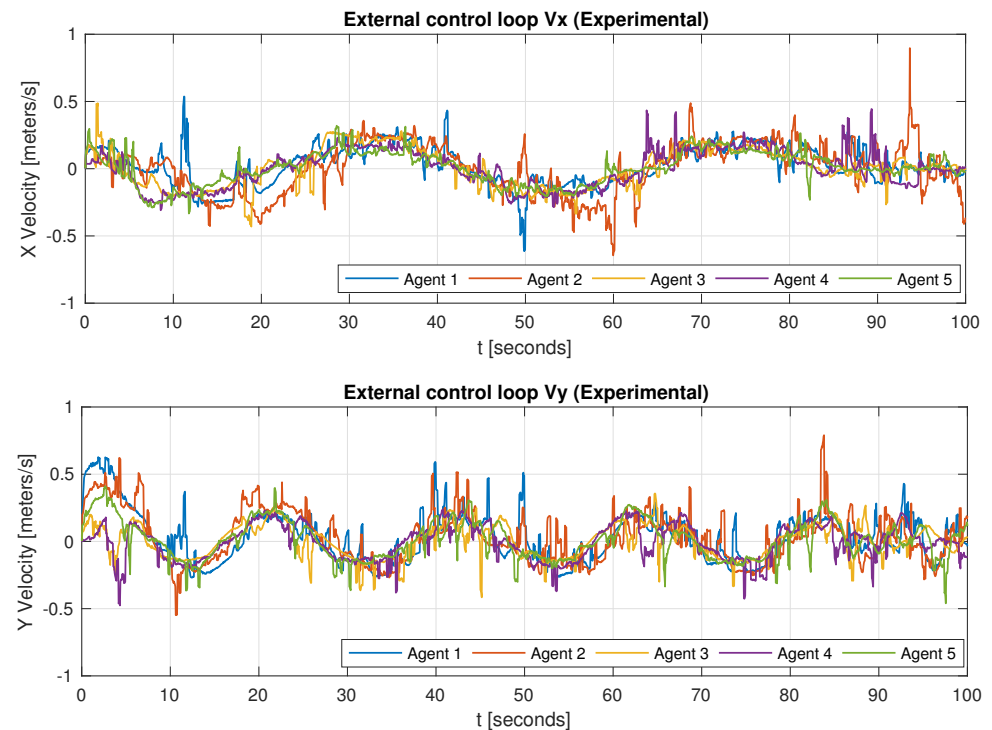


Figure 9. External control loop (experimental).

Nevertheless, the group tries to follow the lemniscate trajectory reference while generating the circular formation pattern as exhibited in Figure 10a. This can also be noted in the tracking errors in Figure 10b; for $t > 85$ [s], the desired trajectory stops, but since there are position errors of each agent concerning the circular formation, the agent keeps moving to

fit into the circumference, but at the end, the circular formation is achieved as shown in the experiment snapshot of Figure 11.

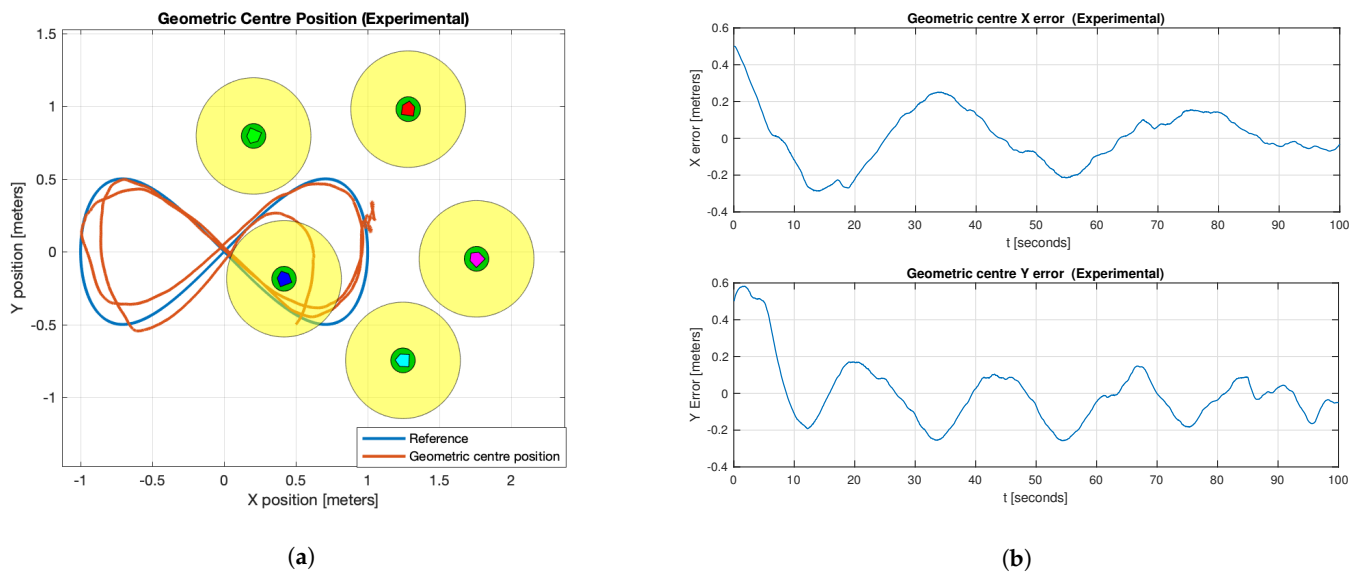


Figure 10. Experimental performance of the proposed controller. (a) Average geometric center (yellow). (b) Geometric center error position (experimental).

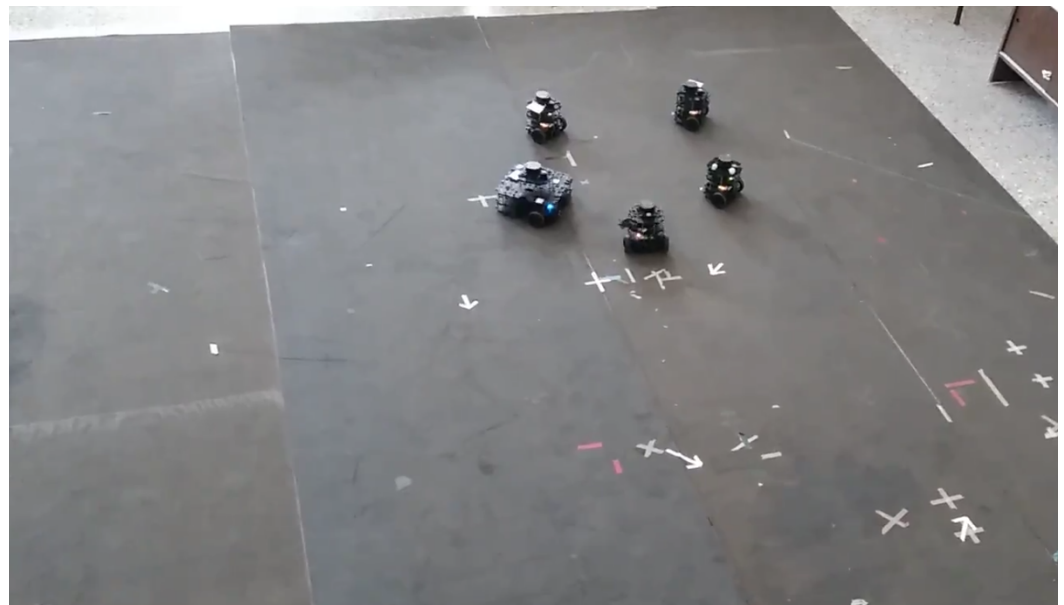


Figure 11. Actual experiment footage.

6. Conclusions

The proposed control law, which aims to generate a desired formation during path tracking, shows satisfactory performance in simulation and acceptable performance in the experimental setup, taking into account the technical limitations of the experimental platform. Better acquisition of the environment data, achieving recognition between the agents in the system, and other localization mechanisms, such as sensor data fusion and filtering, would be reflected in an improvement of the controller behavior.

The proposed controller can be seen as a dynamic extension of the standard kinematic control because of the incorporation of the force-driven model, which allows obstacle avoidance, trajectory tracking, and enforcing the formation. This dynamic force model

can be further modified to include some other goals related to the synchronization of the agents, enclosing, escorting, etc. The proposed controller is decentralized and highly relies on the perception capacities of each agent, but it could be easily implemented based on communication systems such that each agent shares its location and event information of detected obstacles. The overall multi-agent systems would enhance its performance and possible applications, such as harvesting, bodyguard formation, terrain coverage and a possible extension into flying vehicles.

Author Contributions: Conceptualization, A.J.-L. and A.R.-A.; methodology, A.J.-L. and A.R.-A.; software, A.J.-L.; validation, A.J.-L.; formal analysis, A.J.-L. and A.R.-A.; investigation, A.J.-L. and A.R.-A.; resources, A.J.-L. and A.R.-A.; data curation, A.J.-L. and A.R.-A.; writing—A.J.-L. and A.R.-A.; writing—review and editing, A.J.-L. and A.R.-A.; visualization, A.J.-L. and A.R.-A.; supervision, A.R.-A.; funding acquisition, A.R.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from the National Council for Science and Technology, CONACyT, Mexico, through the grant CB2017-2018-A1-S-26123.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Sánchez-Ibáñez, J.R.; Pérez-del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. *Sensors* **2021**, *21*, 7898. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Arkin, R.C. *Behavior-Based Robotics*; MIT Press: Cambridge, MA, USA, 1998.
3. Muzahid, A.J.M.; Kamarulzaman, S.F.; Rahman, M.A.; Murad, S.A.; Kamal, M.A.S.; Alenezi, A. Multiple vehicle cooperation and collision avoidance in automated vehicles: Survey and an AI-enabled conceptual framework. *Sci. Rep. Nat.* **2023**, *13*, 603. [\[CrossRef\]](#)
4. Wesselhöft, M.; Hinkeldeyn, J.; Kreutzfeldt, J. Controlling Fleets of Autonomous Mobile Robots with Reinforcement Learning: A Brief Survey. *Robotics* **2022**, *11*, 85. [\[CrossRef\]](#)
5. Loukatos, D.; Petrongonas, E.; Manes, K.; Kyrtopoulos, I.-V.; Dimou, V.; Arvanitis, K.G. A Synergy of Innovative Technologies towards Implementing an Autonomous DIY Electric Vehicle for Harvester-Assisting Purposes. *Machines* **2021**, *9*, 82. [\[CrossRef\]](#)
6. Li, J.; Qin, H.; Wang, J.; Li, J. OpenStreetMap-Based Autonomous Navigation for the Four Wheel-Legged Robot Via 3D-Lidar and CCD Camera. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2708–2717. [\[CrossRef\]](#)
7. Zhu, K.; Zhang, T. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Sci. Technol.* **2021**, *26*, 674–691. [\[CrossRef\]](#)
8. Zhang, J.; Niu, X.; Croxford, A.J.; Drinkwater, B.W. Strategies for guided acoustic wave inspection using mobile robots. *Proc. R. Soc. Publ.* **2022**, *478*, 20210762. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Shaeffer, D. MEMS inertial sensors: A tutorial overview. *IEEE Commun. Mag.* **2013**, *51*, 100–109. [\[CrossRef\]](#)
10. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 6. [\[CrossRef\]](#)
11. Granlund, G.; Knutsson, H. *Signal Processing for Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2013.
12. De Ponte Muller, F. Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles. *Sensors* **2017**, *17*, 271. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Sand, S.; Zhang, S.; Mühlegg, M.; Falconi, G.; Zhu, C.; Krüger, T.; Nowak, S. Swarm Exploration and Navigation on Mars. In Proceedings of the 2013 International Conference on Localization and GNSS (ICL-GNSS), Torino, Italy, 25–27 June 2013; pp. 1–6.
14. Li, J.; Chu, J.; Zhang, R.; Hu, H.; Tong, K.; Li, J. Biomimetic Navigation System using a Polarization Sensor and a Binocular Camera. *J. Opt. Soc. Am. A. Opt. Image. Sci. Vis.* **2022**, *39*, 847–854. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Ducatelle, F.; Di Caro, G.; Pinciroli, C.; Mondada, F.; Gambardella, L. Communication assisted navigation in robotic swarms: Self-organization and cooperation. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4981–4988.
16. Zakaria, W.N.W.; Mahmood, I. A.-T.; Shamsudin, A.U.; Rahman, M.A.A.; Tomari, M.R.M. ROS-based SLAM and Path Planning for Autonomous Unmanned Surface Vehicle Navigation System. In Proceedings of the 2022 IEEE 5th International Symposium in Robotics and Manufacturing Automation (ROMA), Malacca, Malaysia, 6–8 August 2022; pp. 1–6. [\[CrossRef\]](#)
17. Li, X.; Liu, Y.; Wu, J. Flocking Control and Pattern Motion in a Modified Cucker-Smale Model. *Korean Math. Soc.* **2016**, *53*, 1–6.
18. Park, J.; Kim, H.J.; Ha, S.-Y. Cucker-Smale Flocking With Inter-Particle Bonding Forces. *IEEE Trans. Autom. Control* **2018**, *55*, 2617–2623. [\[CrossRef\]](#)

19. Merheb, A.; Gazi, V.; Sezer-Uzol, N. Implementation Studies of Robot Swarm Navigation Using Potential Functions and Panel Methods. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2556–2567. [[CrossRef](#)]
20. Helbing, D.; Farkas, I.; Vicsek, T. Simulating dynamical features of escape panic. *Nature* **2000**, *407*, 487–490. [[CrossRef](#)] [[PubMed](#)]
21. Helbing, D.; Buzna, L.; Johansson, A.; Werner, T. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transp. Sci.* **2005**, *39*, 1–24. [[CrossRef](#)]
22. Dinneweth, J.; Boubezoul, A.; Mandiau, R.; Espié, S. Multi-agent reinforcement learning for autonomous vehicles: A survey. *Auton. Intell. Syst.* **2022**, *2*, 27. [[CrossRef](#)]
23. Canudas de Wit, C.; Siciliano, B. *Theory of Robot Control*; Tercera Edición; Springer: Berlin/Heidelberg, Germany, 1997.
24. Vázquez Chávez, L.F.; Rodríguez Ángeles, A. Bio-inspired decentralized autonomous robot mobile navigation control for multi agent systems. *Kibernetika* **2018**, *54*, 135–154.
25. MATLAB, version R2020a; The MathWorks Inc.: Natick, MA, USA, 2020.
26. TurtleBot3 Specifications; ROBOTIS Ltd.: Seoul, Republic of Korea, 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.