electronics

Article

Motion Coordination of Multiple Autonomous Mobile Robots under Hard and Soft Constraints

Spyridon Anogiatis [1], Panagiotis S. Trakas [1] and Charalampos P. Bechlioulis [1,2,*]

[1] Division of Signals and Control Systems, Department of Electrical and Computer Engineering, University of Patras, Rio, 26504 Patras, Greece; anogiatiss@ac.upatras.gr (S.A.); ptrakas@upatras.gr (P.S.T.)
[2] Athena Research Center, Robotics Institute, Artemidos 6 & Epidavrou, 15125 Marousi, Greece
* Correspondence: chmpechl@upatras.gr

Abstract: This paper presents a distributed approach to the motion control problem for a platoon of unicycle robots moving through an unknown environment filled with static obstacles under multiple hard and soft operational constraints. Each robot has an onboard camera to determine its relative position in relation to its predecessor and proximity sensors to detect and avoid nearby obstacles. Moreover, no robot apart from the leader can independently localize itself within the given workspace. To overcome this limitation, we propose a novel distributed control protocol for each robot of the fleet, utilizing the Adaptive Performance Control (APC) methodology. By utilizing the APC approach to address input constraints via the on-line modification of the error specifications, we ensure that each follower effectively tracks its predecessor without encountering collisions with obstacles, while simultaneously maintaining visual contact with its preceding robot, thus ensuring the inter-robot visual connectivity. Finally, extensive simulation results are presented to demonstrate the effectiveness of the presented control system along with a real-time experiment conducted on an actual robotic system to validate the feasibility of the proposed approach in real-world scenarios.

Keywords: adaptive performance control; input constraints; motion coordination; multi-agent systems; collision avoidance

check for updates

Citation: Anogiatis, S.; Trakas, P.S.; Bechlioulis, C.P. Motion Coordination of Multiple Autonomous Mobile Robots under Hard and Soft Constraints. *Electronics* **2024**, *13*, 2128. https://doi.org/10.3390/electronics13112128

Academic Editor: Giuseppe Prencipe

Received: 1 May 2024
Revised: 23 May 2024
Accepted: 27 May 2024
Published: 29 May 2024

1. Introduction

Robotics is a rapidly growing scientific field with many applications in multiple aspects of everyday life, like manufacturing, agriculture, medicine, exploration, transportation etc. As technology progresses, robots are becoming more and more popular due to their advantages such as efficiency, productivity, versatility and precision when performing repetitive tasks. As the demand is constantly growing, the industry searches for more efficient solutions in order to meet the share of the marketplace, resulting in studies exploring the possibility of cooperation of Multi-Agent Robotic Systems (MARS) [1,2]. Usually, one robot alone is able to solve a problem in one domain only, thus preventing it from doing anything else and also is incapable to solve the problem in case of a failure occurring as mentioned in [3]. Moreover, a single agent requires strong processing power due to the complex algorithms it has to run. On the other hand, MARS are known for their fault-tolerant capabilities, meaning that, if one agent fails, another one is sent to replace it. Another notable feature of MARS is its adaptability, particularly in scenarios where a fleet of mobile robots is assigned the task of exploring an unknown environment. It is reasonable to expect that such a fleet can complete the exploration faster than a single agent, as each robot can focus on a specific region of the workspace. Additionally, one might anticipate reduced energy consumption throughout the process.

MARS can be more effective when information is shared among all robots via a communication network hence resolving the same task much faster. Due to the MARS efficacy in completing the given task, one may wonder about the energy that has cost in the end. As said in [4], this is one of the variables playing a crucial role while solving a problem. Also, cooperation among the agents is a crucial aspect that many researchers focus their studies on. In [5] there are numerous parameters and challenges presented for MARS that help someone understand their structure (like having a leader, mobility, communication capabilities etc.) and many key problems that need to be solved for the system to work properly.

2. Related Literature

One of the most significant capabilities of MARS is cooperation among the agents in order to achieve efficiently a common goal. The first thing one must consider during the designing phase of a MARS, is to determine the kind of cooperation among the system's agents. In this work, we consider a multi-robot fleet consisted of unicycle mobile robots. Thus, the first step is to determine the formation architecture of the fleet. Many studies have been conducted referring around the subject of the formation control of multiple mobile robots as mentioned in [6–19]. Most of these works propose a formation based on the graph theory, meaning that they can perform from complex shapes to more simple, like a straight line (leader-follower formation). The authors in [6–14] propose a strategy focused on a communication network. The paper [20] introduces a unified coordinated control scheme developed for networked multi-robot systems, with a particular emphasis on object transportation. This scheme incorporates a discontinuous cooperative control law for individual sub-formations around targets, complemented by a continuous control protocol designed to tackle implementation challenges. Another approach many researchers have adopted is the use of artificial potential fields [21,22] as well as the exploitation of a reference point for their system like a camera placed in the ceiling or a 360° camera mounted on each robotic agent [15–19].

Moreover, the authors in [23–26] propose a vision based algorithm to solve the formation control problem thus introducing various constraints into the system's design. This means that each robot must keep its predecessor in its Field Of View (FOV). Particularly, in [23] the authors use a novel Lyapunov barrier function to deal with the system's constraints while using a recursive adaptive backstepping method and Neural Network approximation to solve the formation tracking problem. Similarly, the authors in [24] proposed a robust depth-based visual predictive controller, which optimizes the trajectory planned while taking into account the constraints presented by the visual feedback. On the other hand, in [25] a differential game is suggested where one agent stays within the FOV of the other in a set workspace. Moreover, in [26], the relative position and bearing angle between the predecessor and follower are obtained using a camera. This work integrates the kinematics of the robot with Lyapunov theory to effectively address formation control.

One common requirement for the aforementioned algorithms to work, is the fact that the follower robot should always have the preceding robot in its line of sight (LOS) which should never break when an obstruction occurs. That is why vision based algorithms suffer under presented occlusions in the workspace, so various researchers [27–30] have studied how to maintain visual connectivity between agents to resolve this problem. In [27], the researchers use an RGB-D camera to solve the formation control problem while detecting and avoiding obstacles as well as preserving the predecessor in the follower's FOV. Also, a similar approach is adopted in [28], where two methods are proposed for serial and parallel formation tracking based on Lyapunov theory and vision constraints, capable of obstacle avoidance and vision maintenance. Moreover, the authors in [29] approach the visual obstruction in a different way, they place virtual predecessor at the last known place it was shown and after some time they expect to have visual connectivity. In a similar fashion, in reference [30], a visual based algorithm with collision avoidance capabilities is proposed, considering the limitations of the hardware and applying them to a correction system where a negative gradient, achieves the formation tracking.

It is of vital importance to venture further into the obstacle avoidance methods like in [31–35]. In [31], a controller is designed based on a fuzzy cascaded PID method, which employs artificial potential algorithm to avoid obstacles and ensure the fleet stays in formation. Similarly, in [32] the authors propose an artificial potential function, which does not stuck in local minima, and by utilizing the Lyapunov theorem and Riccati equations they ensure stability of the suggested formation. Another interesting approach is proposed in [33], where the robot follows a human while avoiding obstacles. The task was formulated as a receding-horizon optimization problem and was solved under the Nonlinear Model Predictive Control (NMPC) framework, while an Extended Kalman Filter (EKF) is integrated into the robot's controller for the human's movements. Furthermore, the authors in [34] propose a scheme for cooperative reconnaissance of multiple unmanned ground vehicles. Initially, the robotic agents are exploring the partially known workspace and secondly the agents get into formation and transverse the set space. Lastly, in [35] a dynamic leader-follower formation robust controller is proposed. This approach utilizes the relative position of each robot pair, explores the situation where an agent fails and showcases how the fleet adapts in a case like this.

Contribution

In this work, we aim at extending the control algorithm in [36], that solves the motion coordination of multiple wheeled robots, by incorporating hard input constraints to the system along with the soft constraints regarding the output performance. When the velocity of a robot follower gets saturated, due to the actuation limitations standing as hard constraint, it is a possible that the robot starts losing visual contact with its predecessor due to the collision avoidance protocol. For this reason, we propose a new event-triggered decentralized control system that every follower sends a signal to its predecessor to slow down when some criteria are met, thus guaranteeing safe passage for all mobile robots of the fleet.

Each of these robotic agents is equipped with proximity sensors, which allow them to measure the distance to the nearest obstacles on the workspace and a front monocular camera with limited FOV that allows any robot follower to get its relative position and bearing angle to its preceding robot. Furthermore, the leader of the robot fleet is the only one capable of localizing itself within the given workspace and move around without compromising the safety of the platoon. The most crucial part of this work is the design of a decentralized control scheme based on the APC methodology [37] dealing with input-output constraints, which ensures that the platoon of robots navigates safely through the environment. Additionally, the proposed control protocol assures a-priori visual connectivity between a robot follower and its preceding robot and keeps the predecessor in its FOV at all times while avoiding any obstructions caused by the static obstacles. The main contributions of this work are outlined as follows:

1. In contrast with [36], we incorporate hard constraints regarding the actuation capacity of the system. This addition is crucial as it tackles a significant issue encountered when the follower robot reaches its maximum velocity and attempts to avoid an obstacle simultaneously. In such scenarios, maintaining the predefined distance from its predecessor becomes challenging, potentially resulting in an increase in inter-robot distance or even collisions with obstacles. To address this challenge, we dynamically relax the bounds on relative error, treating them as soft constraints that can be relaxed when they conflict with hard constraints. Meanwhile, we ensure adherence to hard constraints, including collision avoidance with obstacles and other agents as well as input limitations.
2. Contrary to the recent works [10,17,23], we consider multiple input and output, possibly conflicting, constraints, simultaneously. In the presence of multiple hard constraints, i.e., safety and input constraints, we propose a distributed control strategy that leverages unidirectional communication between each robot and its predecessor. By sending signals to force the predecessor to slow down when hard constraints are at risk of being breached, we ensure that the follower maintains visual connectivity and keeps pace.

3. The proposed control strategy is characterized by simple structure and easy gain selection, which boost its scalability. These characteristics are validated through multiple realistic and complex simulations, as well as a real-time experiment involving two Amigobots.

3. Problem Formulation and Preliminaries

First, let us consider $\mathcal{W} \subset \mathbb{R}^2$ to be a planar workspace occupied by n static obstacles $\mathcal{O}_i$, where $i \in \mathcal{J}_O$ with $\mathcal{J}_O \triangleq \{1,\ 2,\ \ldots,\ n\}$ and the free space is define as $\mathcal{W}_f := \mathcal{W} \setminus \bigcup_{i \in \mathcal{J}_O} \mathcal{O}_i$. Additionally, consider a fleet of $N+1$ robots $\mathcal{R}_i$, that are disk shaped, with radius r_i for $i \in \mathcal{J}_R$ with $\mathcal{J}_R \triangleq \{1,\ 2,\ \ldots,\ N\}$ and obey the unicycle model as follows:

$$\begin{aligned} \dot{p}_i &= n_i \cdot u_i \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{1}$$

where $p_i = [x_i, y_i]^T \in \mathbb{R}^2$ and $\theta_i \in \mathbb{R}$ represent the position and orientation of the $i-th$ robot with respect to its inertial coordinate frame, respectively and $n_i = [\cos\theta_i, \sin\theta_i]^T \in \mathbb{R}^2$. Additionally, $\tau_i = [u_i, \omega_i]^T \in \mathcal{T}_i \subset \mathbb{R}^2$ denotes the control input, containing the commanded linear and angular velocities, respectively. To account for actuation limitations, the velocity of each robot is constrained within the compact set:

$$\mathcal{T}_i := \{(u_i, \omega_i) : |u_i/\alpha_i| + |b_i\omega_i/\alpha_i| \leq 1\} \tag{2}$$

where α_i is the maximum wheel velocity and b_i represents half the distance between the two driving wheels. Each robot follower $\mathcal{R}_i$, $i \in \mathcal{J}_F \triangleq \{1,\ 2,\ \ldots,\ N\}$ is equipped with a monocular camera fixed at the robot's center that extracts the relative position $\tilde{p}_i = p_{i-1} - p_i$ of the robot $\mathcal{R}_{i-1}$ expressed in the camera's body frame as long as it is detectable. Furthermore, a robot follower's predecessor $\mathcal{R}_{i-1}$ is only visible if:

1. The robot $\mathcal{R}_{i-1}$ is located within the field of view $\mathcal{F}_i$ of robot $\mathcal{R}_i$'s camera, which is defined as a sector with angle $2\beta_{con \in (0,\pi)}$ and radius $d_{con} > 0$.
2. The line segment $\mathcal{L}_i$, or LOS, that connects $\mathcal{R}_{i-1}$ to $\mathcal{R}_i$ does not go through or be interrupted by an obstacle $\mathcal{O}_j, j \in \mathcal{J}_O$.

In addition, $d_{col} > r_{i-1} + r_i$ is the minimum distance allowed between the robots $\mathcal{R}_{i-1}$ and $\mathcal{R}_i$. Figure 1 shows in depth the parameters explained above.

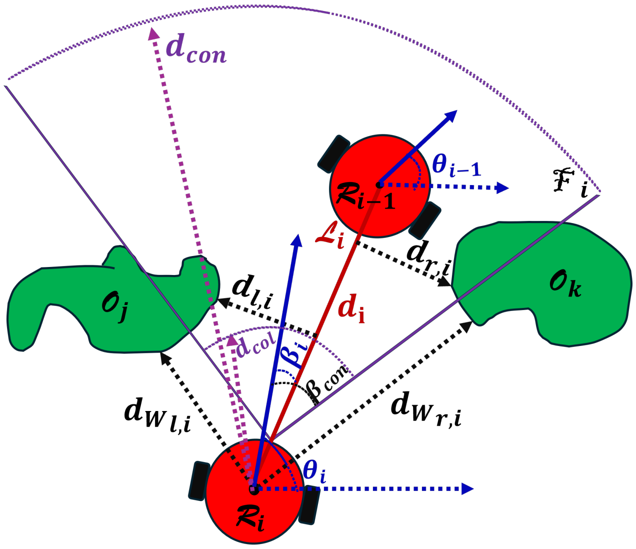

Figure 1. Robot $\mathcal{R}_i$ tracking its predecessor $\mathcal{R}_{i-1}$.

As mentioned before, all robots are equipped with proximity sensors that can detect obstacles in range d_{con} allowing to compute $d_{W_{l,i}}$, $d_{W_{r,i}}$ between the robot and the outer rim of the $\mathcal{W}_f$. The distances $d_{l,i}$, $d_{r,i}$, which are the minimum distances left or right to the LOS $\mathcal{L}_i$ can be computed using the sign line distance very effectively. Lastly, let us define the relative distance and angle of view corresponding to robot $\mathcal{R}_i$ and $\mathcal{R}_{i-1}$, respectively as:

$$\begin{aligned} d_i &= \|\tilde{p}_i\| \\ \beta_i &= \arctan\left(\frac{\tilde{y}_i}{\tilde{x}_i}\right) - \theta_i \end{aligned} \tag{3}$$

with $\tilde{p}_i = p_{i-1} - p_i = [\tilde{x}_i, \tilde{y}_i]^T$ for all followers $i \in J_F$. Now, the problem can be precisely formulated as follows.

Problem 1. *Assuming that the leader R_0 navigates within the workspace $\mathcal{W}_f$, the goal of this work is to design a decentralized control scheme for the constrained input $\tau_i \in \mathcal{T}_i, i \in \mathcal{J}_F$, such that the entire robotic platoon navigates safely within the workspace while there are no collisions to the static obstacles nor inter-robot ones. In this vein, there are specific constraints that must be held as follows:*

$$d_{col} < d_i(t), d_{W_{l,i}} > r_i, d_{W_{r,i}} > r_i \tag{4}$$

and also every predecessor robot $\mathcal{R}_{i-1}$ remain within the FOV $\mathcal{F}_i$ of follower robot $\mathcal{R}_i$ such that:

$$d_i(t) < d_{con}, |\beta_i(t)| < \beta_{con}, d_{l,i}(t) > r_i, d_{r,i}(t) > r_i \tag{5}$$

for all time $t \geq 0$ and $i \in \mathcal{J}_F$. Moreover, due to the aforementioned operational constraints the formation of the platoon should keep up a desired inter-robot distance $d_{des} \in [d_{col}, d_{con}]$ with zero angle of view, meaning that each follower keeps the preceding robot at the center of its camera and at a distance d_{des}.

To solve the aforementioned problem we operate under the assumption of unidirectional communication, where each robot is capable of transmitting a packet of information exclusively to its predecessor. Additionally, we assume that the initial configuration of robots meets the following condition:

$$d_{col} < d_i(0) < d_{con}, |\beta_i(t)| < \beta_{con} \tag{6}$$

$$d_{W_{l,i}}(0) > r_i, d_{W_{r,i}}(0) > r_i, d_{l,i}(0) > r_i, d_{r,i}(0) > r_i,\ i \in J_f. \tag{7}$$

Remark 1. *It should be emphasized that the above-mentioned assumptions are not restrictive because they make the problem feasible and ensure that initially all robots are safe and monitor their predecessors, allowing the proposed control method to be implemented. In the presence of input constraints, some form of communication between the robots becomes necessary to guarantee the fulfillment of output constraints at all times. In addition, if the robot fleet is folded and the initial conditions mentioned above are not eligible, then the collision and visibility constraints cannot be met simultaneously. In that scenario there must be a reordering of the robots of the fleet to alleviate the deadlock.*

Remark 2. *Note that in this work, we do not study the motion planning of the leader robot R_0 towards its goal position. Hence, the main goal of this paper is the coordination of the platoon of robots under the multiple constraints mentioned above.*

4. Controller Design

In this section, we design the control scheme adopting the APC methodology [37] to deal with input constraints. In this way, various safety requirements are ensured while collision avoidance and visibility maintenance are guaranteed, in the presence of hard input constraints. The design procedure can be divided in the following steps:

Step 1. Let us initially define the the distance and angle of view errors:

$$e_{d_i}(t) = d_i(t) - d_{des} \tag{8}$$

$$e_{\beta_i}(t) = \beta_i(t) \tag{9}$$

for each robot $\mathcal{R}_i, i \in J_F$. By differentiating e_{d_i} and e_{β_i} with respect to time and substituting $d_i(t)$, $\beta_i(t)$ to their equivalent in (3), the error dynamics are obtained as follows:

$$\dot{e}_{d_i} = -u_i \cos\beta_i + u_{i-1}\cos(\theta_i - \theta_{i-1} + \beta_i) \tag{10}$$

$$\dot{e}_{\beta_i} = -\omega_i + \frac{u_i}{d_i}\sin\beta_i + \frac{u_{i-1}}{d_i}\sin(\theta_i - \theta_{i-1} + \beta_i). \tag{11}$$

As observed in Figure 1, the distance between robots $\mathcal{R}_i$ and $\mathcal{R}_{i-1}$ is not influenced by their angular velocities; therefore, the two terms in (10) correspond to the robots' projected linear velocities in the direction of their LOS, which determine the rate at which their distance changes. However, the rate of change of the angle of view (11) is solely affected by the angular velocity of robot $\mathcal{R}_i$ and the cross-radial velocity of the robots.

The control objective is to design the velocity inputs of $\tau_i = [u_i, \omega_i]^T \in \mathcal{T}_i$, $i \in J_F$, such that the following output constraints are respected:

$$\underline{\rho}_{d_i}(t) < e_{d_i}(t) < \bar{\rho}_{d_i}(t) \tag{12}$$

$$\underline{\rho}_{\beta_i}(t) < e_{\beta_i}(t) < \bar{\rho}_{\beta_i}(t) \tag{13}$$

for all $t \geq 0$ and for properly designed performance functions $\underline{\rho}_{d_i}(t), \bar{\rho}_{d_i}(t), \underline{\rho}_{\beta_i}(t), \bar{\rho}_{\beta_i}(t)$ which incorporate the following safety constraints:

$$-(d_{des} - d_{col}) < \underline{\rho}_{d_i}(t) < \bar{\rho}_{d_i}(t) < d_{con} - d_{des} \tag{14}$$

$$-\beta_{con} < \underline{\rho}_{\beta_i}(t) < \bar{\rho}_{\beta_i}(t) < \beta_{con}. \tag{15}$$

These conditions, concerning the performance functions of distance and angle of view errors, guarantee that each follower keeps the preceding robot inside its camera FOV $\mathcal{F}_i$ and prevent collisions with it. The satisfaction of (12) and (13) leads, via (14) and (15), to:

$$-(d_{des} - d_{col}) < e_{d_i}(t) < d_{con} - d_{des} \tag{16}$$

$$-\beta_{con} < e_{\beta_i}(t) < \beta_{con} \tag{17}$$

and therefore owing to the definition of (e_{d_i}, e_{β_i}):

$$d_{col} < d_i(t) < d_{con} \tag{18}$$

$$-\beta_{con} < \beta_i < \beta_{con} \tag{19}$$

for all $t \geq 0$.

Step 2. Following the definition of tracking errors and their associated performance specifications, we proceed to design the ideal distributed control laws governing the linear and angular velocities. These control laws are intended to enforce the prescribed performance attributes on the robots, assuming the absence of input constraints. Next, we design the desired velocity signals for each robot $\mathcal{R}_i$, $i \in J_F$, that impose prescribed performance as dictated by (12) and (13). One more step that needs to be done is the calculation of the error transformation for the distance and angle of view. It is worth noting that for appropriately chosen initial value of the performance functions $\underline{\rho}_{d_i}(t)$, $\bar{\rho}_{d_i}(t)$, $\underline{\rho}_{\beta_i}(t)\underline{\rho}_{\beta_i}(t)$, $\bar{\rho}_{\beta_i}(t)$ the transformed errors are finite at $t = 0$. Thus, we maintain the transformed error signals $\epsilon_{d_i} := \ln\left(\frac{e_{d_i}(t)-\underline{\rho}_{d_i}(t)}{\bar{\rho}_{d_i}(t)-e_{d_i}(t)}\right)$ and $\epsilon_{\beta_i} := \ln\left(\frac{e_{\beta_i}(t)-\underline{\rho}_{\beta_i}(t)}{\bar{\rho}_{\beta_i}(t)-e_{\beta_i}(t)}\right)$ bounded for all time via the appropriate selection of the velocity control commands. Then, the satisfaction of (12) and (13) is guaranteed for all time, owing to the properties of the inverse error mappings ϵ_{d_i}, ϵ_{β_i}. Hence, the constrained problem at hand has been reformulated as a simple unconstrained stabilization problem of the transformed error signals $\epsilon_{d_i}(t)$ and $\epsilon_{\beta_i}(t)$, which is solved using the following velocity control protocol:

$$u_{d_i} = \frac{k_d}{\cos\beta_i}\epsilon_{d_i} \tag{20}$$

$$\omega_{d_i} = \frac{\min(\max(u_{d_i}, -\bar{u}_i), \bar{u}_i)}{d_i}\sin\beta_i + k_\beta\epsilon_{\beta_i} \tag{21}$$

with $\bar{u}_i$ denoting the maximum linear velocity of $i-$th robot and k_d, k_β are positive control gains.

Step 3. In step 2 we designed the reference control input $\tau_{d_i} := [u_{d_i}(t), \omega_{d_i}(t)]^T \in \mathbb{R}^2$ that ensures safe navigation with prescribed performance guarantees. Note that τ_{d_i} serves as the ideal control signal designed to enforce prescribed output performance specifications on robot i. Nevertheless, since τ_{d_i} is constrained within the compact set $\mathcal{T}_i$, we exploit a saturation function to produce a feasible control input that obeys the input constraints. Hence, by selecting $\bar{u}_i = a_i$ and $\bar{\omega}_i = a_i/b_i$, as the translational and rotational velocity saturation levels, respectively, we adopt a saturation function $\sigma(\cdot) : (-\infty,\infty)\times(-\infty,\infty) \to \mathcal{T}_i$ that maps the desired control signals $\tau_{d_i} \notin \mathcal{T}_i$ on the boundary of the set $\mathcal{T}_i := [-\bar{u}_i, \bar{u}_i] \times [-\bar{\omega}_i, \bar{\omega}_i]$, based on the radial distance of τ_{d_i} from the origin as depicted in Figure 2.

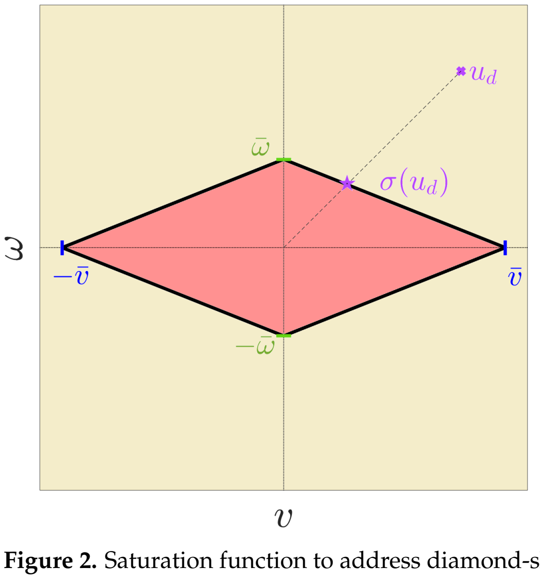

Figure 2. Saturation function to address diamond-shaped constraints; u_d denotes the desired control input and $\sigma(u_d)$ denotes the feasible, constrained, control input based on the radial distance of u_d from the origin.

Thus, the control input incorporating both input and output constraints is obtained by:

$$\tau_i^s := [u_i^s, \omega_i^s]^T = \sigma(\tau_{d_i}(t)) \in \mathcal{T}_i. \tag{22}$$

Note that in the presence of input limitations each robot might face challenges in keeping its predecessor within its FOV when executing saturated control commands, such as moving at its maximum linear velocity. Maintaining all robots within the FOV of their followers is critical, i.e., a hard constraint, for the coordinated navigation, as only the leader possesses knowledge about the desired path through the workspace. To address this practical problem, each robot sends a signal q_i to its predecessor. This signal serves to gradually decelerate the predecessor when the distance and FOV performance functions exceed a safety threshold. Specifically, each robot adopts the following distributed control input:

$$u_i = q_{i+1} u_i^s \tag{23}$$
$$\omega_i = \omega_i^s \tag{24}$$

where:

$$q_{i+1} := S\Big(\max(0, \bar{\rho}_{d_{i+1}}); \zeta_d\Big) S\Big(\max(0, \bar{\rho}_{\beta_{i+1}}); \zeta_\beta\Big)$$

with:

$$S(\chi; \bar{r}) \triangleq \begin{cases} 1 - \frac{\chi}{\bar{r}} & \text{if } \forall \chi \in [0, \bar{r}] \\ 0 & \text{if } \forall \chi \in [\bar{r}, \infty) \end{cases} \tag{25}$$

for some positive constants ζ_d, ζ_β, denoting the safety thresholds, selected to satisfy $\zeta_d < d_{con}$ and $\zeta_\beta < \beta_{con}$.

Step 4. Finally, we design the adaptive performance functions such that all operational and safety requirements are ensured along with input limitations, i.e., the soft error constraints regarding output performance, are adjusted dynamically, to meet the multiple hard constraints of the system. Note that designing such an adaptation mechanism is crucial for ensuring the boundedness of closed-loop signals, as singularities arise when the tracking error exceeds the specified performance envelope. Specifically, we address two common scenarios in which each robot in the fleet reacts to the presence of obstacles. Firstly, considering a static obstacle appearing either to the left or the right side of robot i and its follower, there is a possibility of the obstacle obstructing their path, potentially causing the follower to lose sight of its predecessor or collide with the obstacle. In such instances, the performance functions related to the FOV, i.e., $\underline{\rho}_{\beta_i}$ and $\bar{\rho}_{\beta_i}$, must be adjusted to ensure that the LOS $\mathcal{L}_i$ of the robot moves away from the obstacle, thereby preventing the risk of losing visibility or collision.

However, in a second scenario, another obstacle may emerge from the opposite side of the robot fleet, rendering the aforementioned maneuver ineffective as it attempts to avoid both obstacles, thereby conflicting with the control command. So, the angle of view is not going to deviate, meaning that the robot follower will probably not be able to successfully avoid the obstacle. A control strategy for the critical case mentioned above, is for the follower to approach its predecessor by reducing the distance performance functions $\underline{\rho}_{d_i}$, $\bar{\rho}_{d_i}$ while keeping the distance greater than d_{col}. Additionally, in presence of hard input constraints, that do not allow the satisfaction of the prescribed performance specifications, the performance functions have to be adjusted online to guarantee the best feasible output response w.r.t. the actuation limitations.

Driven by the aforementioned discussion, we introduce the adaptive performance functions that incorporate input-output and safety constraints as:

$$\dot{\underline{\rho}}_{d_i} = -\lambda(\underline{\rho}_{d_\infty} + \underline{\rho}_{d_i}) + f_{d,i}(f_{r,i} + f_{l,i}) \tag{26}$$
$$\dot{\bar{\rho}}_{d_i} = -\lambda(\bar{\rho}_{d_i} - \bar{\rho}_{d_\infty}) + f_{d,i}(f_{r,i} + f_{l,i}) + f_{u,i} \tag{27}$$
$$\dot{\underline{\rho}}_{\beta_i} = -\lambda(\underline{\rho}_{\beta_\infty} + \underline{\rho}_{\beta_i}) - f_{r,i} + f_{l,i} - f_{\omega,i} \tag{28}$$
$$\dot{\bar{\rho}}_{\beta_i} = -\lambda(\bar{\rho}_{\beta_i} - \bar{\rho}_{\beta_\infty}) - f_{r,i} + f_{l,i} + f_{\omega,i} \tag{29}$$

where:

$$
\begin{aligned}
f_{l,i} &:= \frac{S(min(d_{l,i}-r_i, d_{W_{l,i}}-r_i);\delta)}{min(d_{l,i}-r_i, d_{W_{l,i}}-r_i)}\\
f_{r,i} &:= \frac{S(min(d_{r,i}-r_i, d_{W_{r,i}}-r_i);\delta)}{min(d_{r,i}-r_i, d_{W_{r,i}}-r_i)}\\
f_{d,i} &:= S(|f_{r,i}-f_{l,i}|;\delta)\\
f_{u,i} &:= |u_{d_i}-u_i|\\
f_{\omega,i} &:= |\omega_{d_i}-\omega_i|
\end{aligned}
$$

with $\underline{\rho}_{d_i}(0) = -(d_{des}-d_{col})$, $\bar{\rho}_{d_{d_i}}(0) = d_{con}-d_{des}$, $\underline{\rho}_{\beta_i}(0) = -\beta_{con}$ and $\bar{\rho}_{\beta_i}(0) = \beta_{con}$, $\delta > 0$ and the bump function S given by (25). Note that the prescribed performance specifications are incorporated through the parameters of the first term in (26)–(29). Particularly, the parameter λ determines the exponential rate of convergence of the distance and angle of view errors e_{d_i}, e_{β_i} to compact sets close to the origin with sizes explicitly regulated by $\underline{\rho}_{d_\infty}$, $\bar{\rho}_{d\infty}$, $\underline{\rho}_{\beta_i}$, $\bar{\rho}_{\beta\infty}$. One point worth mentioning is, that if the constraints in (14) and (15) being held, the follower robot is maintaining its predecessor within its camera FOV, while it is avoiding any collisions between the two. Notice that when input saturation is active, i.e., $u_{d_i} \neq u_i$, $\omega_{d_i} \neq \omega_i$ the magnitude of the performance update laws (27)–(29) increases. This adjustment ensures the appropriate balance between input (hard) and output (soft) constraints, guaranteeing the boundedness of all closed-loop signals. On the other hand, when input saturation is inactive, the performance update laws return to their nominal form, exponentially fast. In particular, if a single obstacle intervenes from the left or right between the follower and the predecessor robot then the terms $f_{r,i}$ and $f_{l,i}$ will increase. This causes the distance performance functions to decrease, meaning that the follower is going to approach its preceding robot. Similarly, in the case of obstacles appearing on both sides of the leader-follower robots, the distance performance functions will increase and the robot follower will get closer to its predecessor. Furthermore, the angle of view performance functions will decrease or increase based on the obstacle's relative position to the robot, so it deviates its LOS away in order to begin executing the obstacle avoidance maneuver.

Finally, to ensure that the constraints bestowed in (14) and (15) regarding the performance functions $\underline{\rho}_{d_i}$, $\bar{\rho}_{d_i}$, $\underline{\rho}_{\beta_i}$ and $\bar{\rho}_{\beta_i}$ are always met, a projection operator [38] is necessary to be applied over the sets: $[-(d_{des}-d_{col}),\ d_{con}-d_{des}-2\rho_{d_\infty}]$, $[-(d_{des}-d_{col})+2\rho_{d_\infty},\ d_{con}-d_{des}]$, $[-\beta_{con},\ \beta_{con}-2\rho_{\beta_\infty}]$ and $[-\beta_{con}+2\rho_{\beta_\infty},\ \beta_{con}]$. The projection operator over a compact convex set $\Omega = [\rho_{\min}, \rho_{\max}]$ is defined as follows:

$$
\mathrm{Proj}(\dot{\rho},\rho) = \begin{cases} \dot{\rho}(1-w(\rho)) & \text{if } w(\rho) > 0 \text{ and } \dot{\rho}w'(\rho) \\ \dot{\rho} & \text{otherwise} \end{cases} \tag{30}
$$

where $w(\rho) = \frac{1}{1-\epsilon^2}\left(\left(\frac{2\rho-(\rho_{\max}-\rho_{\min})}{\rho_{\max}+\rho_{\min}}\right)^2 - \epsilon^2\right)$ for a positive number $\epsilon \in (0,\ 1)$. The proposed control algorithm (8)–(30) is summarized in Figure 3 for readers' convenience.

Remark 3. *In a multi-robot system, it is common to impose constraints that regulate the distances between all pairs of robots to prevent collisions. The constraints* (16) *and* (17) *establish the minimum and maximum allowable distance between any two robots, ensuring safe navigation and collision avoidance. As a result, while the algorithm primarily prioritize constraints with the predecessor robot and obstacles, it implicitly integrates constraints between all pairs of robots to uphold safe inter-agent distances.*

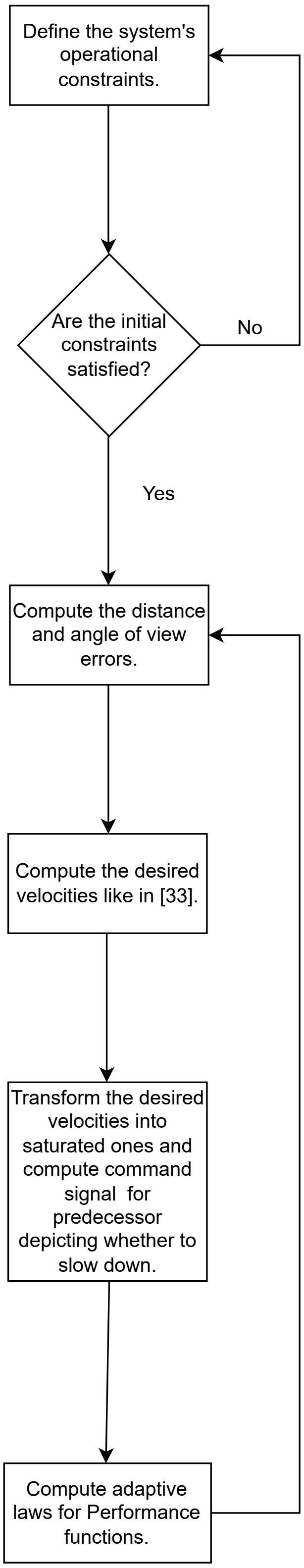
Define the system's operational constraints.
Are the initial constraints satisfied?
No
Yes
Compute the distance and angle of view errors.
Compute the desired velocities like in [33].
Transform the desired velocities into saturated ones and compute command signal for predecessor depicting whether to slow down.
Compute adaptive laws for Performance functions.

Figure 3. Algorithm Flowchart describing the proposed control approach (8)–(30).

Stability Analysis

Theorem 1. *Given a planar workspace cluttered with obstacles, consider a platoon of unicycle robots, described by* (1)*, operating under input constraints as well as safety and visual constraints* *as described in this section. Moreover, the leader robot R_0 tracks a feasible path within the workspace and the system initializes under the appropriate conditions so that all the constraints are initially met. The distributed control protocol (20)–(29) guarantees the safe navigation of the robot fleet within the workspace, ensuring collision avoidance with obstacles and maintaining visibility between robots for all $t \geq 0$.*

Proof. Based on the formulated problem discussed, the robot fleet takes the form of a line beginning from the leader robot and expanding backwards to the last follower. For this reason, the analysis will be conducted into pairs of follower and predecessor. Consider the Lyapunov function candidate:

$$V_i = \frac{1}{2}\epsilon_{d_i}^2 + \frac{1}{2}\epsilon_{\beta_i}^2$$

Differentiating with respect to time and applying the error dynamics (10) and (11):

$$\begin{aligned}
\dot{V}_i = {} & \frac{\epsilon_{d_i}(\bar{\rho}_{d_i} - \underline{\rho}_{d_i})}{(\bar{\rho}_{d_i} - e_{d_i})(e_{d_i} - \underline{\rho}_{d_i})}\Bigg(u_{i-1}\cos\left(\theta_i - \theta_{i-1} + \beta_i\right) - u_i\cos(\beta_i) \\
& \qquad - \frac{\dot{\underline{\rho}}_{d_i}(\bar{\rho}_{d_i} - e_{d_i}) + \dot{\bar{\rho}}_{d_i}(e_{d_i} - \underline{\rho}_{d_i})}{\bar{\rho}_{d_i} - \underline{\rho}_{d_i}}\Bigg) \\
& + \frac{\epsilon_{\beta_i}(\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i})}{(\bar{\rho}_{\beta_i} - e_{\beta_i})(e_{\beta_i} - \underline{\rho}_{\beta_i})}\Bigg(-\frac{u_{i-1}}{d_i}\sin\left(\theta_i - \theta_{i-1} + \beta_i\right) - \omega_i + \frac{u_i}{d_i}\sin(\beta_i) \\
& \qquad - \frac{\dot{\underline{\rho}}_{\beta_i}(\bar{\rho}_{\beta_i} - e_{\beta_i}) + \dot{\bar{\rho}}_{\beta_i}(e_{\beta_i} - \underline{\rho}_{\beta_i})}{\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i}}\Bigg).
\end{aligned}$$

Hence, substituting the control protocol (20)–(29) the above equation takes the form below:

$$\dot{V}_i = \frac{(\bar{\rho}_{d_i} - \underline{\rho}_{d_i})\epsilon_{d_i}(h_{d_i} - h_{u_i})}{(\bar{\rho}_{d_i} - e_{d_i})(e_{d_i} - \underline{\rho}_{d_i})} + \frac{(\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i})\epsilon_{\beta_i}(h_{\beta_i} - h_{\omega_i})}{(\bar{\rho}_{\beta_i} - e_{\beta_i})(e_{\beta_i} - \underline{\rho}_{\beta_i})} \tag{31}$$

with

$$\begin{aligned}
h_{d_i} :={} & u_{i-1}\cos\left(\theta_i - \theta_{i-1} + \beta_i\right) - u_i\cos\left(\beta_i\right) \\
& + \frac{1}{\bar{\rho}_{d_i} - \underline{\rho}_{d_i}}\Bigg[(\bar{\rho}_{d_i} - e_{d_i})\Big(\lambda(\underline{\rho}_{d_i} + \underline{\rho}_{d_\infty}) - f_{d,i}(f_{r,i} + f_{l,i})\Big) \\
& + (e_{d_i} - \underline{\rho}_{d_i})\Big(\lambda(\bar{\rho}_{d_i} - \bar{\rho}_{d_\infty}) - f_{d,i}(f_{r,i} + f_{l,i})\Big)\Bigg] \\
h_{u_i} :={} & \frac{|u_{d_i} - u_i|(\bar{\rho}_{d_i} - e_{d_i})}{(\bar{\rho}_{d_i} - \underline{\rho}_{d_i})}
\end{aligned} \tag{32}$$

$$\begin{aligned}
h_{\beta_i} :={} & -\frac{u_{i-1}}{d_i}\sin\left(\theta_i - \theta_{i-1} + \beta_i\right) - \omega_i + \frac{u_i}{d_i}\sin(\beta_i) \\
& + \frac{1}{\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i}}\Bigg[(\bar{\rho}_{\beta_i} - e_{\beta_i})\Big(\lambda(\underline{\rho}_{\beta_i} + \underline{\rho}_{\beta_\infty}) + f_{r,i} - f_{l,i}\Big) \\
& + (e_{\beta_i} - \underline{\rho}_{\beta_i})\Big(\lambda(\bar{\rho}_{\beta_i} - \bar{\rho}_{\beta_\infty}) + f_{r,i} - f_{l,i}\Big)\Bigg] \\
h_{\omega_i} :={} & |\omega_{d_i} - \omega_i|\left(\frac{(\bar{\rho}_{\beta_i} - e_{\beta_i}) + (e_{\beta_i} - \underline{\rho}_{\beta_i})}{\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i}}\right).
\end{aligned} \tag{33}$$

Note that the form of (31) is valid as long as the projection operator (30) on the performance functions is inactive. However, since (30) is activated to ensure that $d_i > d_{col} > 0$ and $|\beta_i| < \beta_{con} < \frac{\pi}{2}$, the corresponding tracking errors, as well as the transformed ones, decrease owing to the signal q_i sent from robot i to its predecessor and forces it to stop when the corresponding performance function exceeds a predefined safety threshold within a compact set where (30) is active. Thenceforward, the stability of the multi-agent system is concluded by solely studying the properties of (31).

Notice that the terms $(\bar{\rho}_{d_i} - \underline{\rho}_{d_i})$, $(\bar{\rho}_{d_i} - e_{d_i})$, $(e_{d_i} - \underline{\rho}_{d_i})$, $(\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i})$, $(\bar{\rho}_{\beta_i} - e_{\beta_i})$, $(e_{\beta_i} - \underline{\rho}_{\beta_i})$ are strictly positive due to (12) and (13). Owing to input saturation on both linear and angular velocities of all robots, there exist positive constants $\bar{h}_{d_i}, \bar{h}_{\beta_i}$ such that $\|h_{d_i}(\cdot)\|_\infty \leq \bar{h}_{d_i}$ and $\|h_{\beta_i}(\cdot)\|_\infty \leq \bar{h}_{\beta_i}$. Additionally, (32) and (33) are positive and radially unbounded functions, strictly increasing in ϵ_{d_i} and ϵ_{β_i}, respectively, which leads us to:

$$\dot{V}_i \leq \frac{(\bar{\rho}_{d_i} - \underline{\rho}_{d_i})\epsilon_{d_i}(\bar{h}_{d_i} - h_{u_i}(\epsilon_{d_i}))}{(\bar{\rho}_{d_i} - e_{d_i})(e_{d_i} - \underline{\rho}_{d_i})} + \frac{(\bar{\rho}_{\beta_i} - \underline{\rho}_{\beta_i})\epsilon_{\beta_i}(\bar{h}_{\beta_i} - h_{\omega_i}(\epsilon_{\beta_i}))}{(\bar{\rho}_{\beta_i} - e_{\beta_i})(e_{\beta_i} - \underline{\rho}_{\beta_i})}. \tag{34}$$

Hence, $\dot{V}_i < 0$ when $h_{u_i}(\epsilon_{d_i}) > \bar{h}_{d_i}$ and $h_{\omega_i}(\epsilon_{\beta_i}) > \bar{h}_{\beta_i}$. Moreover, provided that the safety and visibility criteria are initially met, then the $\epsilon_{d_i}(0)$ and $\epsilon_{\beta_i}(0)$ are properly clarified, from which someone derives that the transformed errors $\epsilon_{d_i}(t)$ and $\epsilon_{\beta_i}(t)$ are uniformly ultimately bounded. Consequently, all closed-loop signals remain bounded, the constraints (12) and (13) are fulfilled at all times and neither crashes nor inter-robot visibility breaks take place, which completes the proof. □

Remark 4. *It should be noted that the input limitations as well as the operational specifications are achieved by adaptively modifying the performance functions (26)–(29), thus simplifying the selection of the control gains k_d and k_β. Moreover, $\bar{\rho}_{d_i}$, $\underline{\rho}_{d_i}$, $\bar{\rho}_{\beta_i}$, $\underline{\rho}_{\beta_i}$ establish the maximum allowable steady-state distance and AOV error, respectively, setting prescribed performance specifications on the closed-loop system. Note that due to the adaptive performance laws (26)–(29), the aforementioned performance parameters can be selected arbitrarily small without jeopardizing the stability of the system. This adaptability allows for fine-tuning the algorithm's performance based on specific application requirements and enhances the robustness against input saturation. Furthermore, the threshold δ determines when the terms f_{d_i}, f_{r_i}, and f_{l_i} nullify. In particular, as the distance of the robot $\mathcal{R}_i$ and the LOS $\mathcal{L}_i$ with surrounding obstacles is greater than δ, these terms vanish, resulting in prescribed output response. Meanwhile, adjusting ϵ affects the sensitivity of the projection operator (30) to deviations in performance boundaries, balancing precision and flexibility in error correction.*

Remark 5. *The adaptive performance method proposed in this work, forces the distance and angle of view errors to remain rigorously in $\underline{\rho}_{d_i}(t)$, $\bar{\rho}_{d_i}(t)$, $\underline{\rho}_{\beta_i}(t)$ and $\bar{\rho}_{\beta_i}(t)$ at all times. Modulating the transformed errors $\epsilon_{d_i}(t)$ and $\epsilon_{\beta_i}(t)$ and keeping them bounded, results to the satisfaction of inequalities (12) and (13). Correspondingly, the current problem can be represented as stabilization of the transformed errors $\epsilon_{d_i}(t)$ and $\epsilon_{\beta_i}(t)$. By observing the introduced control protocol, it is evident that the performance functions act as barrier functions used in constraint optimization. Namely, the errors $\epsilon_{d_i}(t)$ and $\epsilon_{\beta_i}(t)$ can never reach their limits due to the proposed control protocol mentioned in this section.*

5. Results

5.1. Simulation Study A

In the first simulation study, we apply the proposed control protocol to a team of five circular robotic agents operating in a predefined workspace, cluttered by static obstacles as shown in Figure 4. The environment configuration was created using MATLAB, and the integration of the differential equations was conducted using the ode45 function. Furthermore, the parameters for this simulation study are specified as follows. The radius of each robot is $r_i = 0.2$ m, $i \in J_F$; the desired distance between agents is set to $d_{des} = 1$ m. Operational constraints enforced by sensors include $d_{col} = 0.5$ m, $d_{con} = 4$ m, and $\beta_{con} = 0.48$ rad. Moreover, the performance function parameters are selected as $\lambda = 1$, $\bar{\rho}_{d_\infty} = \underline{\rho}_{d_\infty} = 0.1$, and $\bar{\rho}_{\beta_\infty} = \underline{\rho}_{\beta_\infty} = 0.1$. The control parameters are set as $k_d = 1$, $k_\beta = 1$, $\delta = 0.5$, and $\epsilon = 0.75$. Additionally, the saturation limits for the followers are $\bar{u}_i = 0.5$ m/s and $\bar{\omega}_i = 0.56$ rad/s, $i \in J_F$, while the thresholds activating the bump function in Equation (23), signaling the predecessor to slow down, are $\zeta_d = 0.5$ and $\zeta_\beta = 0.85\beta_{con}$.

The leader robot (red circle) is commanded to navigate safely through the given space to the goal position without bumping into any obstacles. To achieve this, the leader employs a trajectory planning method known as Vector Field Orientation (VFO), as presented in [39]. This approach generates a feasible path from one waypoint to another, and once the leader robot approaches a certain distance (error tolerance) from its goal, it proceeds towards the next waypoint. Figure 4 illustrates ten consecutive snapshots of the robot team in the workspace every couple of seconds along with the camera FOV of each robot follower. As anticipated, at $t = 0$ s, all robots have their predecessors within their FOV, maintaining this visual connectivity throughout the simulation, even in scenarios featuring narrow passages and sharp corners.

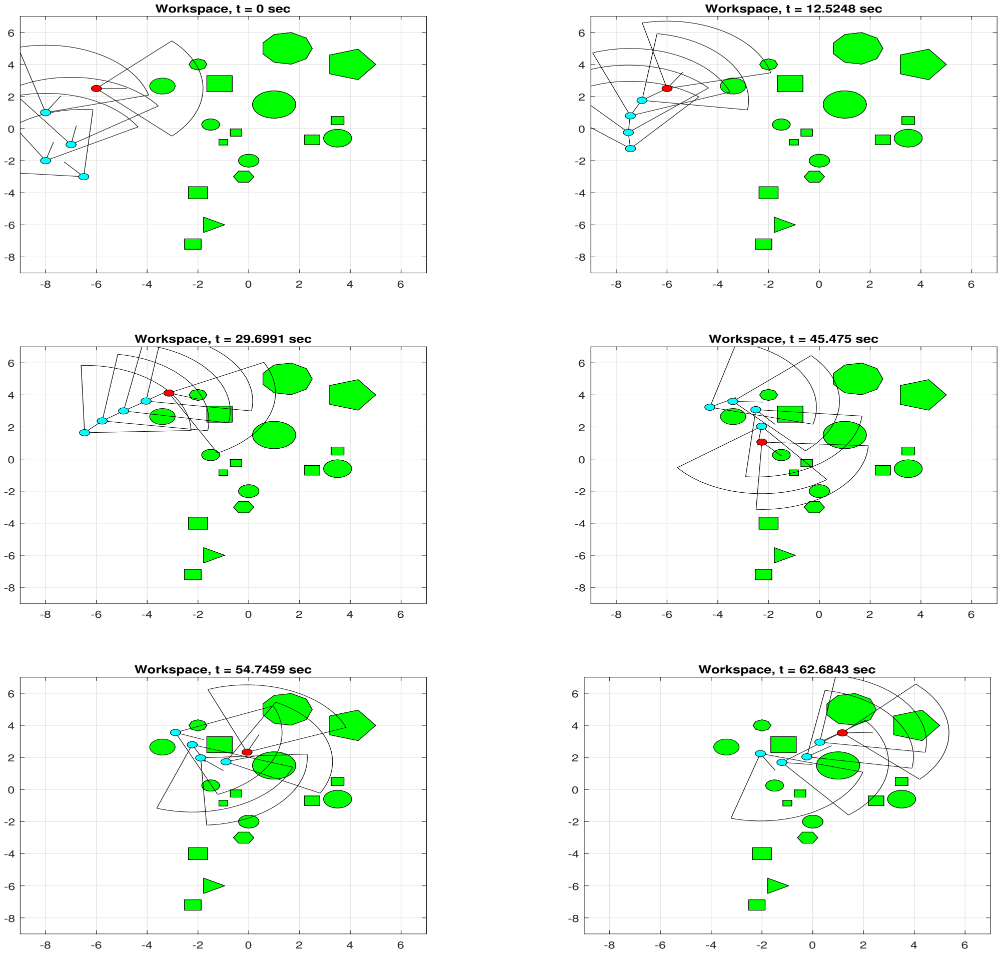

Figure 4. *Cont.*

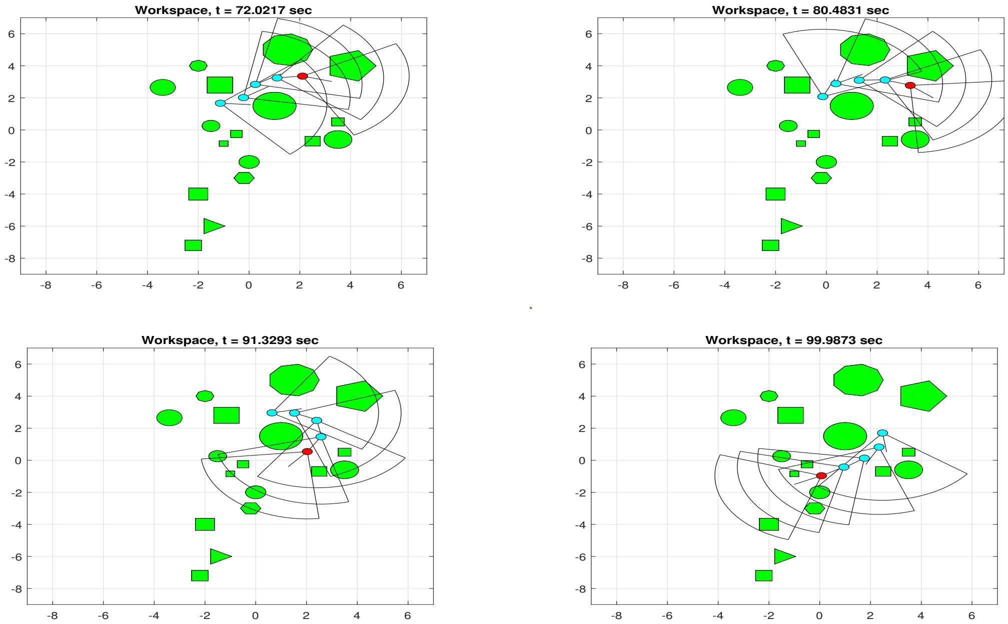

Figure 4. Consecutive snapshots of the robot fleet (the red circle is the leader while the cyan ones are the followers) navigating through the given workspace. Each camera's field of view is given by the black-colored quadrant.

Furthermore, in Figure 5 it is shown how the distance and angle of view errors evolve over time, as the system navigates through the workspace. Evidently, the system manages to keep the errors mentioned within the adaptive boundaries determined by the performance functions without compromising the robot fleet's safety, i.e., hard constraint. One point worth mentioning is the fact that the distance error of the first follower remains unaltered for approximately 12 s. This occurrence is expected, as each agent signals its predecessor to halt, via (23), under certain safety conditions. Activation of the bump function (25) triggers a gradual decrease in the speed of the robot's predecessor, extending even to the leader if necessary. This reduction continues until the robot comes to a complete stop, particularly if the follower maintains a relative distance greater than a specified threshold. This halting mechanism enables the robot to pause until the inter-agent distance aligns with safety constraints before resuming movement. By incorporating this approach, the algorithm guarantees the overall safety of the system despite arbitrary input constraints.

Nonetheless, by observing Figure 6 which depicts the linear and angular velocities of each robot follower one can notice their intensive behavior which can be explained by the fact that the system is trying to meet tight steady state performance specifications. The phenomenon where the first two followers stayed still until the others catch up becomes more clear by observing the velocities in Figure 6 and it should be noted that the leader robot waits as well, otherwise in case the leader started independently of others, the boundedness of closed-loop signals can not be a-priori guaranteed, owing to hard input constraints. Additionally, notice that the angular velocities change more drastically in order to avoid any occurring obstructions appearing, while the linear velocity is responsible to ensure the desired inter-robot distance. According to Remark 4 it is worth mentioning that the tracking response can be improved by fine tuning the control gains k_d, k_β separately for each agent.

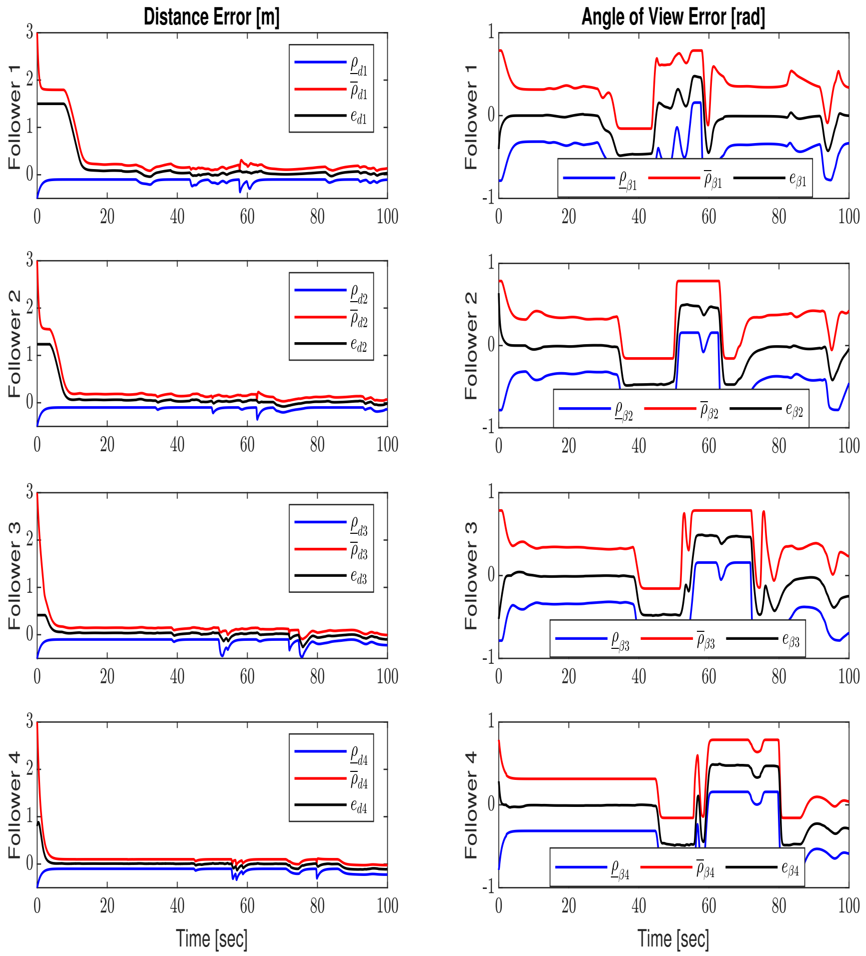

Figure 5. Distance and angle of view error response of each robot follower agent along with their dedicated performance function boundaries; red line is the upper bound, blue line is the lower bound and black line is the respected error.

5.2. Simulation Study B

In this paradigm we simulate the system in a more realistic environment and study the system response. Utilizing the Gazebo 11 Simulator, a suitable confined workspace was created to test the performance of the control algorithm. The system consists of 3 Pioneer3DX robotic agents equipped with a LiDAR sensor, a calibrated camera, and a corresponding-colored ball, in order to be detectable by each follower. By using the ROS framework we managed to implement our algorithm in the simulated environment.Knowing the intrinsic parameters of the camera we created a color detection algorithm corresponding to the color of the ball. Afterwards, we computed the ball's pose by transforming its coordinates from the image plane into real-world coordinates in a same manner as it was discussed in [40]. Also, the navigation part for the leader's movement across the workspace is handled with the help of existing ROS navigation libraries. The troubleshooting and parameters explanation for the navigation system was feasible due to the work presented in [41], thus by meddling them, it was ensured that leader robot moved in such way that the following robots are able to track it, as illustrated in Figure 7.

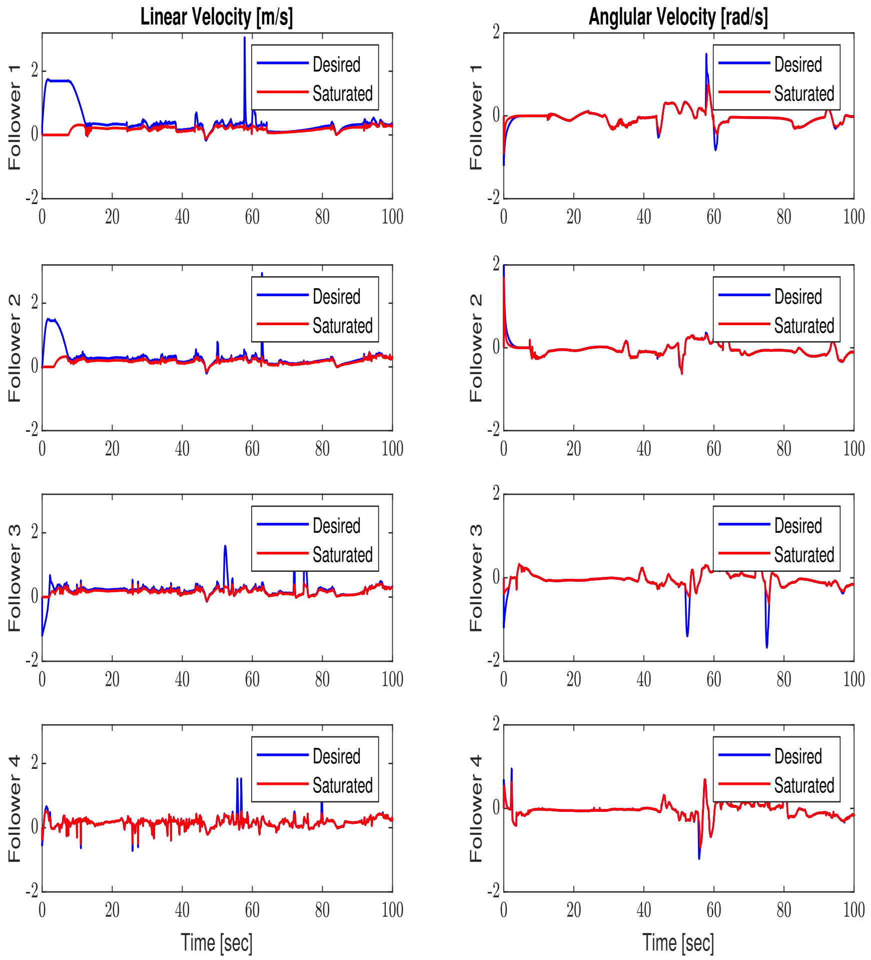

Figure 6. Linear and angular (desired (blue) and actual (red)) velocities of each robot.

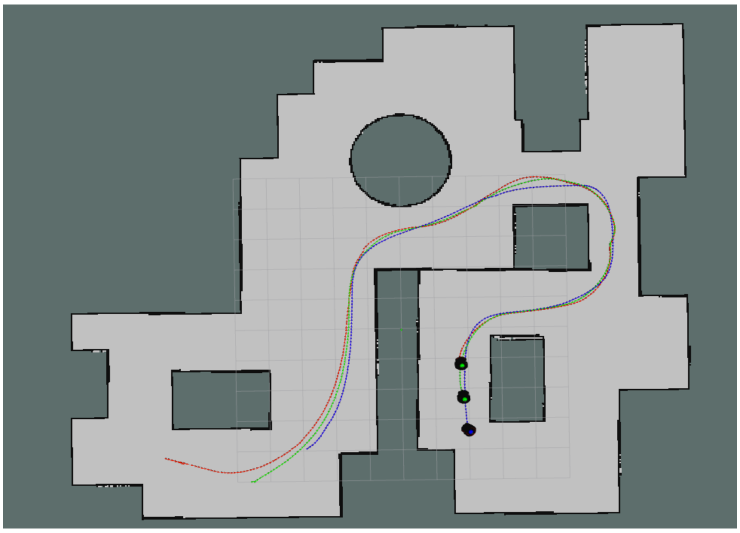

Figure 7. The trajectories of each robot on the simulated mapped workspace.

Subsequently, we provide information about the system's parameters in this simulation case. The robots' radius $r_i = 0.225$ m, $i \in J_F$, the desired distance between agents $d_{des} = 1$ m and the operational constraints (imposed by sensor limitations) are the following $d_{col} = 0.5$ m, $d_{con} = 12$ m and $\beta_{con} = 0.6929$ rad. Furthermore, the performance function parameters are set as follows $\lambda = 0.5$, $\bar{\rho}_{d_\infty} = \underline{\rho}_{d_\infty} = 0.1$, and $\bar{\rho}_{\beta_\infty} = \underline{\rho}_{\beta_\infty} = 0.1$. Finally, the rest of the parameters that need to be defined are those of the control protocol and are chosen as follows $k_d = 0.4, k_\beta = 0.5, \delta = 0.6$ and $\epsilon = 0.75$. The saturation limits for the followers are selected as $\bar{u}_i = 0.51$ m/s and $\bar{\omega}_i = 1.2$ rad/s, and the limits that activate the bump function indicating to the predecessor when to stop are $\zeta_d = 0.5$ and $\zeta_\beta = 0.85\beta_{con}$.

Figure 8 shows the distance and angle of view error responses of each robot follower along with the boundaries set by the control algorithm during the simulation and how they change depending on the topology of the workspace, while ensuring the fleet's safety.Furthermore, as it was shown in the first experiment we observe in Figure 8, that the distance error of the second follower stays unchanged for a period of time because its follower sent a signal to halt until the upper bound of the distance error is smaller than 0.5 m. As it is shown in Figure 9, the commanded velocities present an oscillatory behavior, because the controller tries to set the errors between the tight bounds provided by the PPC algorithm. One valid point is that each follower must match their predecessor's velocity in order to keep the desired distance set by user. So, the oscillatory behavior can easily be explained by the controller's action to reduce the error while keeping the error under tight bounds. However, in this work we dynamically relax those bounds in case that the follower was saturated thus the commanded velocities set by the controller were less oscillatory. In Figure 9, one can notice the linear velocities are swinging around 0.5 m/s, which is the commanded speed of the leader robot in the simulation, although due to the modification presented in this work the linear velocity of each predecessor is slightly less than 0.5 m/s because of the signal sent by their dedicated follower to slow down. Additionally, the angular velocities do not abide to the same concept as mentioned above, it is guided by the predecessor's relative position and the collision avoidance protocol, which was discussed previously. Lastly, there is a simulation video that clarifies the aforementioned results, available in GazeboVideo (https://youtu.be/AWY2_q-2Muw, accessed on 27 May 2024).

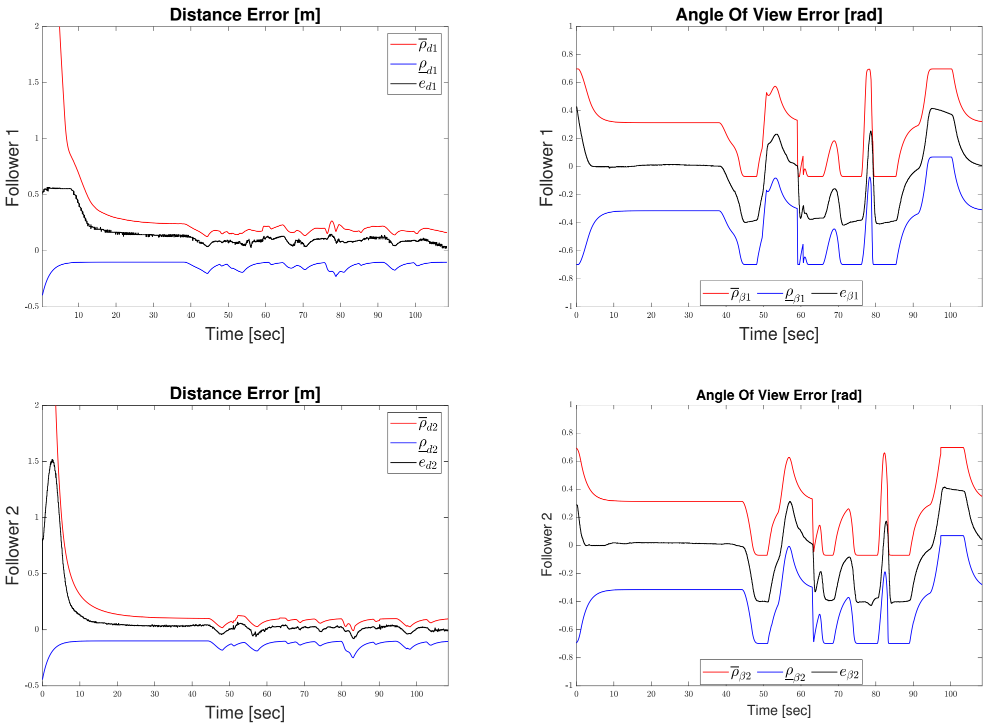

Figure 8. Distance and Angle of View error response during the simulation in Gazebo; red line depicts the upper bound of each respected error, blue line is the lower bound and black line signifies the error.

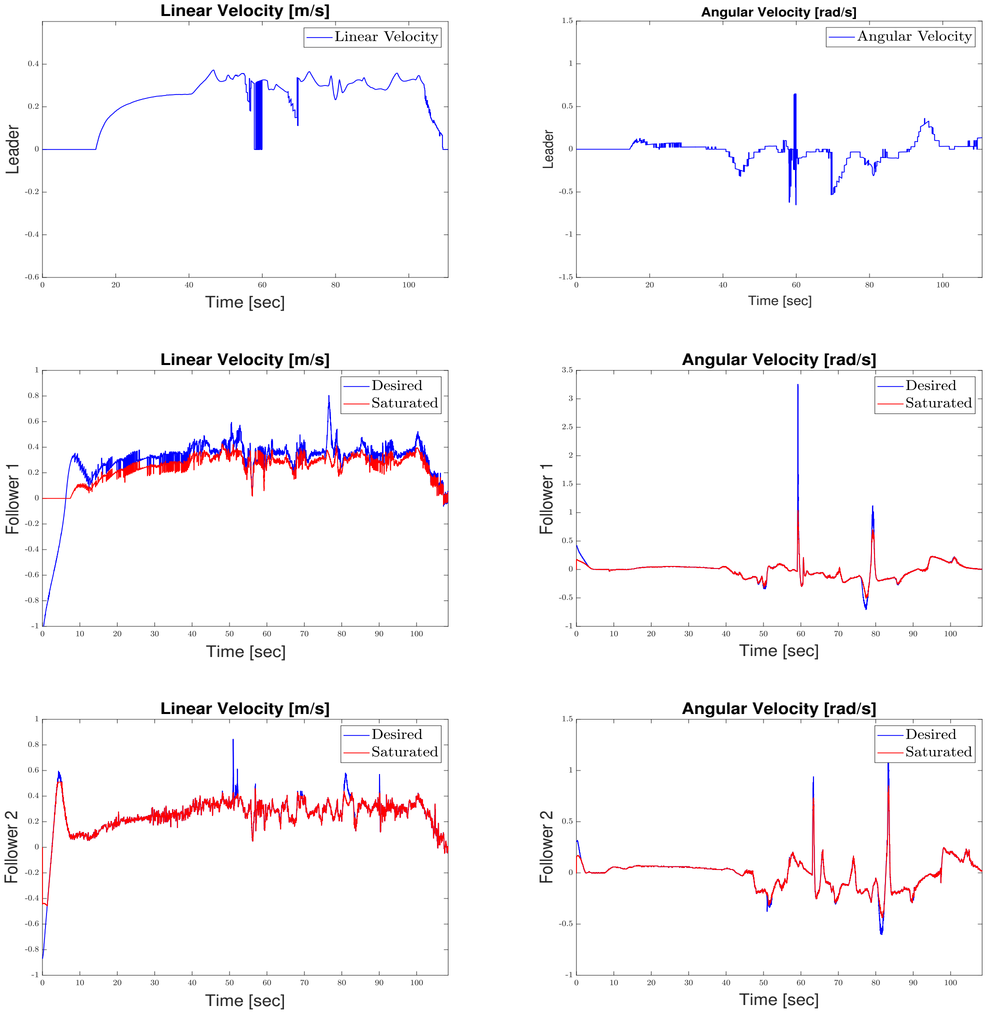

Figure 9. Linear and angular desired and actual velocities for each robot agent during the simulation in Gazebo; red line is the actual saturated velocity while blue is the corresponding desired velocity.

5.3. Real-World Experiment

In this case study, an experiment was conducted in the real world. The main goal is to showcase the performance and the robustness of the proposed control scheme on a real environment, where many uncertainties are presented like traction, measurement noise and various delays presented by the actuating hardware when giving a command to the robot's motors. For this procedure two AmigoBots are utilized, as shown in Figure 10, where one is the leader and the other one is the follower. The whole process is implemented leveraging the ROS framework and Python in the same way as it was mentioned in Simulation Study B. Both robots are equipped with a LiDAR sensor, an Odroid unit (mini computer) equipped with Ubuntu and a Logitech C270 HD webcam, with 30 fps frame rate and 720p analysis (no depth). Also, the leader has a green ball onboard, which is utilized by the follower for detecting it via an onboard camera. Another point worth mentioning is that the leader robot is moving through the dedicated space by tele-operation. It is worth noting that the proposed control scheme regulates the linear and angular velocities, taking into account the kinematic model of the robot. Given that many commercial robots, including those utilized in this experiment, are equipped with low-level micro-controllers, the compensation of robot dynamics is handled by these onboard controllers.

Subsequently, it is necessary to provide information about the real world's system parameters. The desired distance between agents is $d_{des} = 1$ m and the operational constraints (imposed by sensor limitations) are the following $d_{col} = 0.2$ m, $d_{con} = 12$ m and $\beta_{con} = 0.48$ rad. Furthermore, the performance function parameters are set as follows,

$\lambda = 0.3$, $\bar{\rho}_{d_\infty} = \underline{\rho}_{d_\infty} = 0.1$ and $\bar{\rho}_{\beta_\infty} = \underline{\rho}_{\beta_\infty} = 0.085$. Finally, the rest of the parameters that need to be defined are those of the control protocol and are chosen as follows $k_d = 0.095$, $k_\beta = 0.107, \delta = 0.5$ and $\epsilon = 0.75$. Finally, the saturation limits for the follower are $\bar{u}_i = 0.25$ m/s and $\bar{w}_{sat} = 0.33$ m/s, and the safety limits that activate the bump function indicating to the predecessor when to stop, are $\zeta_d = 0.5$ m and $\zeta_\beta = 0.85\beta_{con}$.

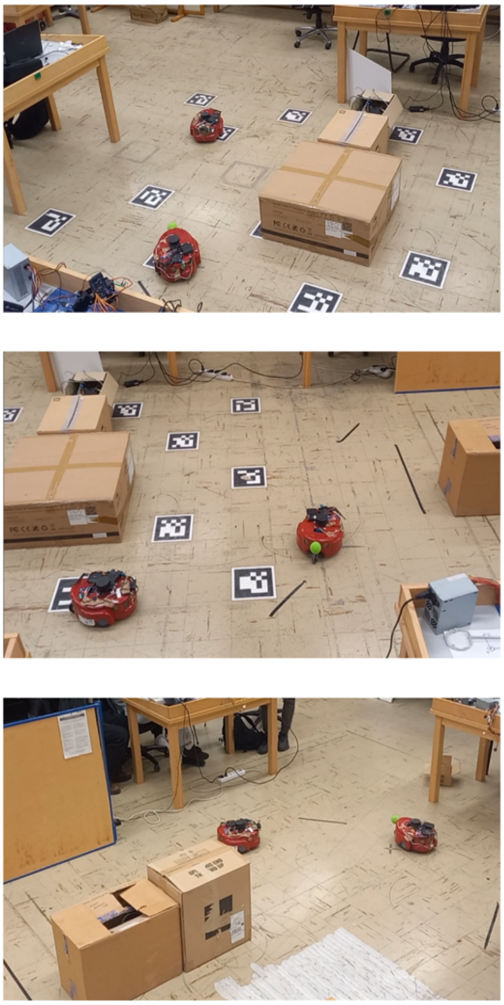

Figure 10. Real World Experiment workspace along with robots.

The distance and angle of view error responses of the follower robot along with the associated performance functions are shown in Figure 11. By observing the given subfigures one can notice that the control protocol manages to keep the errors bounded while ensuring safe navigation of the robot fleet through the workspace without losing visual connectivity at any time. One crucial point that must be pointed out is the oscillatory behavior of the distance and error responses. This behavior arises due to the follower's abrupt changes in commanded velocity; during the experiment, the follower made sudden stops when approaching its predecessor and then accelerated, repeating this cycle to satisfy the performance function's specifications at steady state while keeping the error within the performance boundaries. This underscores the necessity for selecting small control gains k_d and k_β to ensure satisfactory system behavior. It is important to note that the limited FOV of the camera posed challenges for the collision avoidance task of the control algorithm. Firstly, the sudden and extreme measurements related to the distance from the LOS, occasionally caused the follower to turn very rapidly, leading to a loss of visual connectivity. Secondly, when the robot initiated the collision avoidance protocol, the camera did not have sufficient space left in its FOV to adequately track the preceding robot as it maneuvered sharply around the obstacle. A video of the real-world experiment can be accessed via the following link (RealWorldVideo): https://youtu.be/MTC6EQjY-UA (accessed on 27 May 2024).

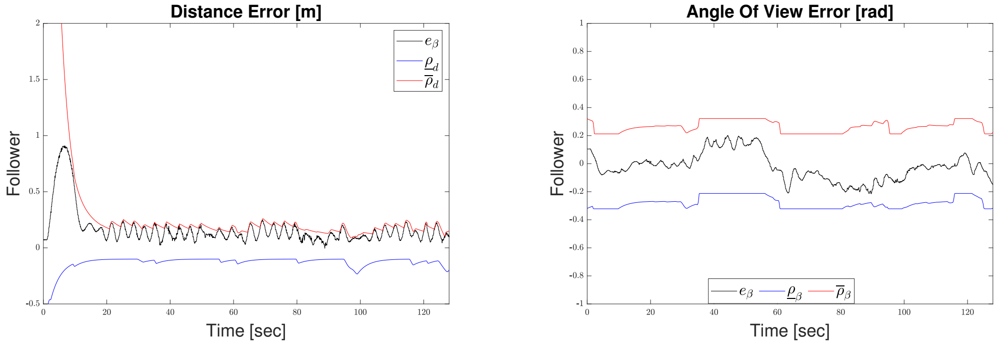

Figure 11. Error response of inter-robot distance and angle of view for the real world experiment; red line depicts the upper adaptive performance bound, while blue line is the lower adaptive performance and black line denote the respected errors.

It can be concluded that in the real-world scenario the commanded velocities given by the control scheme are more oscillatory Figure 12 than the two simulation studies conducted above. This phenomenon can be attributed to practical constraints such as slip, as well as delays introduced by measurements, both of which hinder the algorithm's performance, as expected. Figure 12 illustrates the velocity of each robotic agent, with the actual velocity of the robot follower depicted in red (which is the saturated velocity) and the desired velocity in blue. By saturating the follower's velocity, we were able to mitigate the high oscillations, thereby enhancing the efficiency of our system in maintaining a predetermined distance between the two robots.

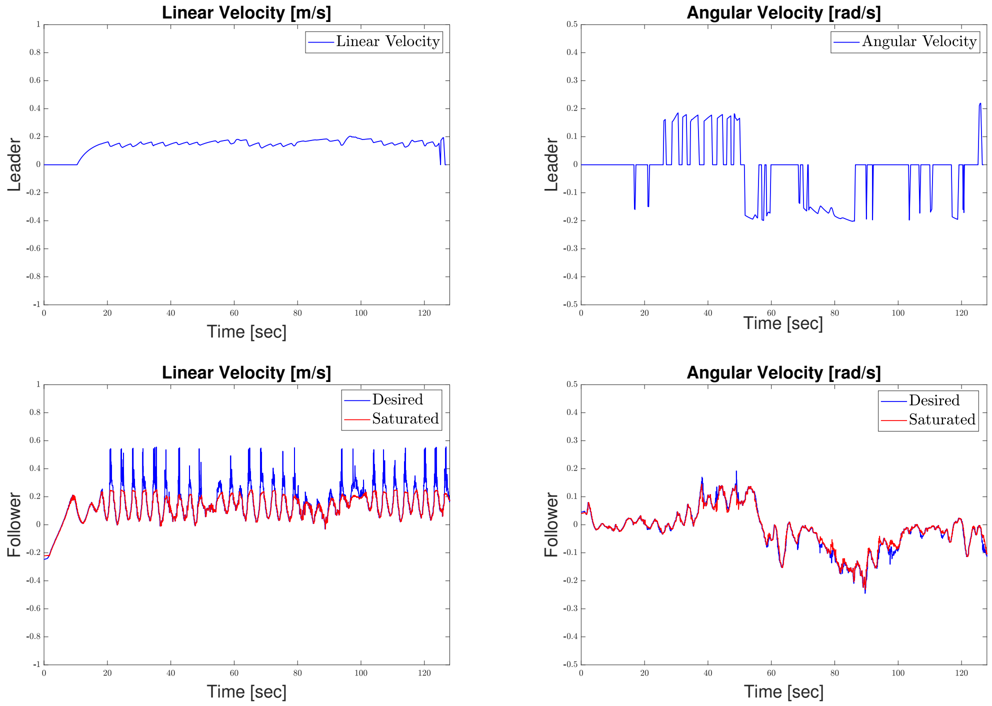

Figure 12. Linear and Angular Commanded Velocities given during the real-world experiment in the Laboratory. Red are the actual saturated velocities proposed in this paper while blue the desired velocities.

6. Conclusions

In the context of this work, the problem of coordinating the motion of a fleet of unicycle robot agents was addressed. In particular, the mobile robots, equipped with the introduced control protocol, are capable of safely navigating autonomously through an obstacle-cluttered unknown workspace. Given that each agent is subject to hard input constraints owing to actuation limitations, a robust distributed control scheme was designed to evade any collision while ensuring long term visual connection between each group of predecessor and follower. Finally, multiple case studies were conducted in various environments, i.e., MATLAB, Gazebo, real-world, in order to validate the efficiency of the proposed control strategy.

Future research efforts will be focused on addressing feedback delays and intermittent communication in order to enhance the efficiency and practicality of the proposed control protocol. More research is also needed to implement the discussed algorithm in more complex formations based on graph theory and ascertain its performance in dynamic environments. Moreover, we intend to further study the case of visual loss (partially or not) due to environmental obstructions, such as motion blur or light conditions. Finally, exploring the behavior of the platoon in the event of agent failure will contribute to its overall robustness and connectivity maintenance.

Author Contributions: Conceptualization, P.S.T. and C.P.B.; methodology, S.A. and P.S.T.; software, S.A.; validation, P.S.T.; formal analysis, S.A. and P.S.T.; investigation, S.A. and P.S.T.; writing—original draft preparation, S.A. and P.S.T.; writing—review and editing, C.P.B.; visualization, S.A.; supervision, P.S.T. and C.P.B.; project administration, C.P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the second call for research projects to support post-doctoral researchers (HFRI-PD19-370). The work of C.P.B. was also supported by the project "Applied Research for Autonomous Robotic Systems" (MIS 5200632) which is implemented within the framework of the National Recovery and Resilience Plan "Greece 2.0" (Measure: 16618-Basic and Applied Research) and is funded by the European Union—NextGenerationEU.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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.

Abbreviations

The following abbreviations are used in this manuscript:

APC	Adaptive Performance Control
EKF	Extended Kalman Filter
FOV	Field Of View
LiDAR	Light Detection And Ranging
LOS	Line Of Sight
MARS	Multi-Agent Robotic System
NMPC	Nonlinear Model Predictive Control
PF	Performance Function
PID	Proportional Integral Derivative
PPC	Prescribed Perfomance Control
RGB-D	Red Green Blue - Depth
ROS	Robot Operating System
URDF	Unified Robotics Description Format
UGV	Unmanned Ground Vehicle
VFO	Vector Field Orientation

References

1. Alqobali, R.; Alshmrani, M.; Alnasser, R.; Rashidi, A.; Alhmiedat, T.; Alia, O.M. A Survey on Robot Semantic Navigation Systems for Indoor Environments. *Appl. Sci.* **2024**, *14*, 89. [CrossRef]
2. Peng, Z.; Wang, J.; Wang, D.; Han, Q.L. An Overview of Recent Advances in Coordinated Control of Multiple Autonomous Surface Vehicles. *IEEE Trans. Ind. Inform.* **2021**, *17*, 732–745. [CrossRef]
3. Jiménez, A.C.; García-Díaz, V.; Bolaños, S. A Decentralized Framework for Multi-Agent Robotic Systems. *Sensors* **2018**, *18*, 417. [CrossRef] [PubMed]
4. Ismail, Z.; Sariff, N. A Survey and Analysis of Cooperative Multi-Agent Robot Systems: Challenges and Directions. *Appl. Mob. Robots* **2018**, *5*, 8–14. [CrossRef]
5. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-Agent Systems: A Survey. *IEEE Access* **2018**, *6*, 28573–28593. [CrossRef]
6. Shen, Z.; Liu, Y.; Li, Z.; Wu, Y. Distributed vehicular platoon control considering communication delays and packet dropouts. *J. Frankl. Inst.* **2024**, *361*, 106703. [CrossRef]
7. Guan, J.C.; Ren, H.W.; Tan, G.L. Distributed Dynamic Event-Triggered Control to Leader-Following Consensus of Nonlinear Multi-Agent Systems with Directed Graphs. *Entropy* **2024**, *26*, 113. [CrossRef]
8. Yu, X.; Hou, Z.; Chen, T. Data-Driven Distributed Adaptive Consensus Tracking of Nonlinear Multiagent Systems: A Controller-Based Dynamic Linearization Method. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 6953–6965. [CrossRef]
9. Li, W.; Du, X.; Xiao, J.; Zhang, L. Bipartite hybrid formation tracking control for heterogeneous multi-agent systems in multi-group cooperative-competitive networks. *Appl. Math. Comput.* **2023**, *456*, 128133. [CrossRef]
10. Reineh, M.S.; Li, P.; Jabbari, F. Leader–Follower Tracking for General Multiagent Systems With Unknown Leader Input and Limited Actuation. *IEEE Trans. Control Netw. Syst.* **2023**, *10*, 1149–1158. [CrossRef]
11. Zhang, J.; Zhang, H.; Sun, S.; Cai, Y. Adaptive Time-Varying Formation Tracking Control for Multiagent Systems With Nonzero Leader Input by Intermittent Communications. *IEEE Trans. Cybern.* **2023**, *53*, 5706–5715. [CrossRef] [PubMed]
12. Wang, J.; Lv, Y.; Shan, X.; Wang, H.; Wang, J. Research on Cooperative Control of Multiple Intelligent Networked Vehicles Based on the Improved Leader–Follower Method. *World Electr. Veh. J.* **2024**, *15*, 73. [CrossRef]
13. Gao, Y.; Jiao, J.; Hirche, S. H_2 suboptimal leader-follower consensus control of multi-agent systems. *IFAC-PapersOnLine* **2023**, *56*, 2614–2619. [CrossRef]
14. Oh-Hara, S.; Fujimori, A. Robustness in formation control of mobile robots using leader-follower method. *Adv. Mech. Eng.* **2023**, *15*. [CrossRef]
15. Li, Y.; Wu, C.; Tian, W. Formation Outlier Formation Transformation Based on Virtual Robots. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 17–19 November 2023; pp. 5121–5125. [CrossRef]
16. Sugiyama, D.; Azuma, S.i.; Ariizumi, R.; Asai, T. Relation between leader–follower consensus control and feedback vertex sets. *Adv. Robot.* **2023**, *37*, 37–45. [CrossRef]
17. Chu, X.; Ng, R.; Wang, H.; Au, K.W.S. Feedback Control for Collision-Free Nonholonomic Vehicle Navigation on SE(2) with Null Space Circumvention. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5594–5604. [CrossRef]
18. Sewlia, M.; Zelazo, D. Bearing-based formation stabilization using event-triggered control. *Int. J. Robust Nonlinear Control* **2024**, *34*, 4375–4387. [CrossRef]
19. Li, M.; Polyakov, A.; Zheng, G. On Generalized Homogeneous Leader-Following Consensus Control for Multiagent Systems. *IEEE Trans. Control Netw. Syst.* **2024**, *11*, 558–568. [CrossRef]
20. Hu, J.; Bhowmick, P.; Lanzon, A. Group Coordinated Control of Networked Mobile Robots With Applications to Object Transportation. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8269–8274. [CrossRef]
21. Cheah, C.C.; Hou, S.P.; Slotine, J.J.E. Region-based shape control for a swarm of robots. *Automatica* **2009**, *45*, 2406–2411. [CrossRef]
22. Trakas, P.S.; Bechlioulis, C.P.; Rovithakis, G.A. Coordinated Navigation of Holonomic Robot Swarms in Complex Workspaces via Dynamic Formation Tracking Control. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 28 June–1 July 2022; pp. 79–84. [CrossRef]
23. Dai, S.L.; Lu, K.; Fu, J. Adaptive Finite-Time Tracking Control of Nonholonomic Multirobot Formation Systems with Limited Field-of-View Sensors. *IEEE Trans. Cybern.* **2022**, *52*, 10695–10708. [CrossRef]
24. Fallah, M.M.H.; Janabi-Sharifi, F.; Sajjadi, S.; Mehrandezh, M. A Visual Predictive Control Framework for Robust and Constrained Multi-Agent Formation Control. *J. Intell. Robot. Syst. Theory Appl.* **2022**, *105*, 72. [CrossRef]
25. Lozano, E.; Becerra, I.; Ruiz, U.; Bravo, L.; Murrieta-Cid, R. A visibility-based pursuit-evasion game between two nonholonomic robots in environments with obstacles. *Auton. Robot.* **2022**, *46*, 349–371. [CrossRef]
26. Arteaga-Escamilla, C.M.; Castro-Linares, R.; Álvarez Gallegos, J. Leader–follower formation with reduction of the off-tracking and velocity estimation under visibility constraints. *Int. J. Adv. Robot. Syst.* **2021**, *18*. [CrossRef]
27. Guan, R.; Hu, G. Formation Tracking of Mobile Robots Under Obstacles Using Only an Active RGB-D Camera. *IEEE Trans. Ind. Electron.* **2024**, *71*, 4049–4058. [CrossRef]
28. Li, Y.; Liu, L.; Gan, Z.; Hu, X. Robust formation control for unicycle robots with directional sensor information. *Auton. Intell. Syst.* **2023**, *3*, 6. [CrossRef]
29. Wang, Y.; Wang, D. Vision-Based Leader-Follower Queue Maneuvering in Unknown Cluttered Environments. In *Collaborative Fleet Maneuvering for Multiple Autonomous Vehicle Systems*; Springer: Singapore, 2023; pp. 67–89. [CrossRef]

30. Wenjie, D.; Shaoping, W.; Chao, Z.; Qing, Z.; Sial, M.B. Formation control and obstacle avoidance of MAS with constraints of visibility and motion saturation. In Proceedings of the 2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 1–4 August 2021; pp. 2107–2113. [CrossRef]
31. Liu, Z.; Li, W.; Li, B.; Gao, S.; Ouyang, M.; Wang, T. Multi-robots Formation and Obstacle Avoidance Algorithm Based on Leader-Follower and Artificial Potential Field Method. In *Computer Supported Cooperative Work and Social Computing*; Sun, Y., Lu, T., Wang, T., Fan, H., Liu, D., Du, B., Eds.; Springer: Singapore, 2024; pp. 422–437.
32. Aljassani, A.M.; Ghani, S.N.; Al-Hajjar, A.M. Enhanced multi-agent systems formation and obstacle avoidance (EMAFOA) control algorithm. *Results Eng.* **2023**, *18*, 101151. [CrossRef]
33. Zhang, Y.; Huang, J.; Yu, J.; Zhu, Y.; Hasegawa, Y. Relative-Posture-Fixed Model Predictive Human-Following Control with Visibility Constraints in Obstacle Environments. In Proceedings of the 2023 International Conference on Advanced Robotics and Mechatronics (ICARM), Sanya, China, 8–10 July 2023; pp. 959–964. [CrossRef]
34. Haojie Zhang, T.Y.; Su, Z. A formation cooperative reconnaissance strategy for multi-UGVs in partially unknown environment. *J. Chin. Inst. Eng.* **2023**, *46*, 551–562. [CrossRef]
35. Cai, H.; Guo, S.; Gao, H. A Dynamic Leaderâ€"Follower Approach for Line Marching of Swarm Robots. *Unmanned Syst.* **2023**, *11*, 67–82. [CrossRef]
36. Bechlioulis, C.P.; Vlantis, P.; Kyriakopoulos, K.J. Coordination of Multiple Robotic Vehicles in Obstacle-Cluttered Environments. *Robotics* **2021**, *10*, 75. [CrossRef]
37. Trakas, P.S.; Bechlioulis, C.P. Approximation-free Adaptive Prescribed Performance Control for Unknown SISO Nonlinear Systems with Input Saturation. In Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC), Cancun, Mexico, 6–9 December 2022; pp. 4351–4356. [CrossRef]
38. Cai, Z.; de Queiroz, M.; Dawson, D. A sufficiently smooth projection operator. *IEEE Trans. Autom. Control* **2006**, *51*, 135–139. [CrossRef]
39. MichaÅ‚ek, M.; KozÅ‚owski, K. Motion planning and feedback control for a unicycle in a way point following task: The VFO approach. *Int. J. Appl. Math. Comput. Sci.* **2009**, *19*, 533–545. [CrossRef]
40. Van Zandycke, G.; De Vleeschouwer, C. 3D ball localization from a single calibrated image. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3472–3480.
41. Zheng, K. *ROS Navigation Tuning Guide*; Springer: Cham, Switzerland, 2017. [CrossRef]

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.