

Article

Adaptive Optimal Collision Avoidance of Dynamic Agents for Differential-Drive Robots

Diego Martinez-Baselga ^{1,2,*} , Diego Lanasa ¹ , Luis Riazuelo ¹  and Luis Montano ¹ 

¹ Institute of Engineering Research (I3A), University of Zaragoza, 50018 Zaragoza, Spain; 796180@unizar.es (D.L.); riazuelo@unizar.es (L.R.); montano@unizar.es (L.M.)

² Department of Cognitive Robotics, Delft University of Technology, 2628CD Delft, The Netherlands

* Correspondence: d.martinezbaselga@tudelft.nl or diegomartinez@unizar.es

Abstract

Efficient navigation in crowded and dynamic environments is crucial for robot integration into human spaces. AVOCADO (Adaptive Optimal Collision Avoidance Driven by Opinion) generates collision-free velocities using Velocity Obstacles and adaptation to the cooperation estimation among agents. However, it assumes holonomic motion and cannot handle non-holonomic constraints, such as those of differential-drive robots. We propose DD-AVOCADO, an extension of AVOCADO that incorporates differential-drive kinematics to compute feasible and safe velocities. The method combines AVOCADO-based planning with a non-holonomic controller and accounts for tracking errors to avoid collisions. Simulation results across diverse scenarios show a significant reduction in collisions and efficient navigation in scenarios with cooperative and non-cooperative agents, and hardware experiments demonstrate its applicability in robot platforms. The method has the potential to be applied to other dynamic models.

Keywords: collision avoidance; motion and path planning; multi-robot systems

1. Introduction

Autonomous navigation of mobile robots in dynamic, multi-agent environments remains a major challenge in robotics. It requires real-time decision-making to avoid moving obstacles, unpredictable trajectories, and potential motion conflicts among agents. To enable safe and cooperative interactions, planning algorithms must not only avoid collisions but also adapt to the behavior of surrounding agents.

Among existing approaches, Velocity Obstacle (VO) [1]-based methods have been widely used to compute collision-free velocities by modeling the relative motion of agents. A notable evolution of VO is Optimal Reciprocal Collision Avoidance (ORCA) [2], which introduces reciprocal cooperation to share collision avoidance responsibility. Although effective in dense and dynamic settings, ORCA assumes homogeneous agents and equal-responsibility sharing, which is unrealistic when cooperation levels vary.

AVOCADO [3] (Adaptive Optimal Collision Avoidance Driven by Opinion) improves on ORCA by dynamically estimating the cooperation level of each agent, enabling more flexible and adaptive collision-free planning in mixed-cooperation scenarios. It incorporates an adaptive control law that enables collision avoidance with both cooperative agents and dynamic obstacles. However, like VO and ORCA, AVOCADO assumes holonomic motion, limiting its applicability to real platforms with kinematic constraints. An example of this problem is shown in Figure 1. The previously mentioned planners do not consider the



Received: 11 February 2026

Revised: 17 March 2026

Accepted: 26 March 2026

Published: 30 March 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

robot constraints and assume that the robot may follow a trajectory like the one represented in blue. Nevertheless, when a non-holonomic controller tries to track the planned motion, it incurs a tracking error, represented in red. In densely populated scenarios, this error may lead to a collision.

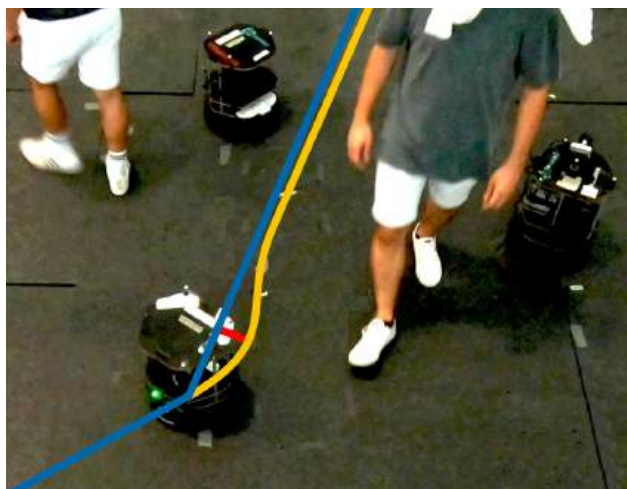


Figure 1. Scenario populated with three robots and two pedestrians. In blue are the past trajectory and a future non-colliding trajectory of a robot with no kinematic constraints considered. In yellow is a future non-colliding trajectory considering the kinematic restrictions. In red is the tracking error between both trajectories.

To address this limitation, extensions such as Non-Holonomic ORCA (NH-ORCA) [4] incorporate non-holonomic dynamics, adjusting admissible velocity regions based on tracking errors. However, no equivalent extension exists for AVOCADO, limiting the application of the planners to scenarios with reciprocal cooperation.

Unlike NH-ORCA, which assumes reciprocal cooperation among homogeneous robots executing the same algorithm, AVOCADO explicitly models heterogeneous agents whose cooperation level is unknown and dynamically estimated. Integrating non-holonomic feasibility into this framework requires ensuring that the adaptive collision avoidance constraints remain valid under bounded tracking errors of the differential-drive controller. This work introduces DD-AVOCADO (Differential Drive-Adaptive Optimal Collision Avoidance Driven by Opinion), an extension of AVOCADO for differential-drive robots. The proposed method integrates non-holonomic constraints into the planning process while preserving the logic of the original algorithm, ensuring that computed velocities are both safe and physically feasible for real robotic platforms. In this way, it combines the adaptability, efficiency and efficacy of AVOCADO with safety and applicability in differential-drive robots, with a potential extension to other platforms by considering other dynamic models.

The contributions of this work are threefold:

- **Incorporation of non-holonomic feasibility into AVOCADO.** We restrict the holonomic velocity space of AVOCADO to an admissible set that accounts for the tracking error of a differential-drive controller, ensuring that selected velocities are physically executable.
- **Tracking-error-aware collision avoidance within the AVOCADO framework.** We adapt the velocity obstacle constraints to account for tracking error while preserving AVOCADO's adaptive cooperation mechanism, enabling safe interaction with heterogeneous agents.

- **Validation in simulation and real robot experiments.** We evaluate the proposed approach in both simulated and real-world scenarios, showing improved safety compared to baseline methods while maintaining effective navigation.

Sections 2 and 3 state the background of the work, presenting the related work and preliminary concepts, respectively. Section 4 details the approach followed, and Section 5 evaluates the planner in simulation and hardware experiments. Finally, Section 6 concludes the work and discusses future extensions.

2. Related Work

Ensuring safe and efficient motion in dynamic environments has motivated a wide range of local planning strategies. Early efforts only included static obstacles to define the set of colliding velocities in the robot velocity space [5]. Force-based models such as potential fields [6] and social forces [7] relied on simple reactive dynamics but often lacked robustness. A very influential line of work, widely extended throughout the years [8], was introduced with Velocity Obstacles [1] (VO). It characterizes the set of velocities that would lead to future collisions considering the relative velocities and positions among the robot and the dynamic agents in the environment, enabling agents to plan directly in velocity space.

A major evolution was proposed with the Reciprocal Velocity Obstacles (RVO) [9], adapting the VO concepts to multi-robot navigation, where every robot in the environment collaborates equally in the collision avoidance maneuvers. Optimal Reciprocal Collision Avoidance (ORCA) [2] formulated the problem with linear constraints that avoided oscillations among agents, achieving efficient multi-robot collision avoidance without communication.

The assumption of equal cooperation has been removed in works like [10] to achieve more efficient navigation depending on the scenario. Nevertheless, this requires at least one-hop communication among agents, which is not realistic in real-world scenarios populated with heterogeneous and possibly non-cooperating agents. AVOCADO [3] proposes to use an adaptive control law to change the degree of cooperation it takes in the collision avoidance depending on the observed behavior of surrounding agents. In that sense, it takes full responsibility to avoid a dynamic agent whose velocity does not change in the presence of the robot, while it trusts other agents more when a reaction is perceived. To do so, it uses nonlinear opinion dynamics law [11,12], which has also been used in other works for fast and reactive navigation [13,14]. A major problem of the aforementioned methods is that they assume the robot follows single-integrator holonomic dynamics, which limits their applicability in real-world robotics.

There are multiple works that alleviate the problem of the applicability of ORCA to real-world robotic platforms. Van Den Berg et al. [15] proposes a method to translate the equations to second-order integrators, and Rufli et al. [16] to n-th order integrators. Alonso-Mora et al. [4] adapts it to non-holonomic platforms, extending other works that consider differential-drive [17,18] and car-like [19] robots. Nevertheless, these approaches incorporate non-holonomic dynamics assuming that all agents follow the same collision avoidance strategy and share reciprocal responsibility. In contrast, AVOCADO explicitly models heterogeneous agents whose level of cooperation is unknown and must be estimated online. The approach proposed in this work extends AVOCADO to differential-drive robots while preserving this adaptive interaction mechanism.

There are other approaches that solve the motion planning problems in different paradigms. End-to-end Deep Reinforcement Learning (DRL) planners [20–22] map sensory information into robot velocity commands. Cited works consider holonomic velocity commands, but there may also be steering commands for differential-drive robots [23].

Recent works have demonstrated that the combination of DRL planners with VO-based information results in high-performing policies [24–26]. However, despite their promising results, pure learning-based methods lack generalization capabilities and explicit collision avoidance constraints, and DRL planners usually do not consider kinematic restrictions.

Control solutions explicitly consider collision avoidance and kinematic models. A wide group of approaches involves Control Barrier Functions (CBFs) [27], which solve motion planning as an optimization problem with formal collision avoidance guarantees. They have been recently combined with VO for collision avoidance with dynamic obstacles with fixed velocities [28,29] and multi-robot navigation [30,31]. However, it is not clear how to extend them to unknown agent behaviors [32]. Other approaches use Model Predictive Control (MPC) methods [33–35], needing explicit predictions on other agents. Some MPC planners consider multiple possible scenario evolutions [36,37], but they are computationally costly. Reducing the time horizon can alleviate this, but also lead to local minima and oscillations. Multiple planning alternatives can be considered to escape from local minima [38,39]. However, MPC planners do not consider the effect of the robot on the dynamic agent motion, as they use the explicit predictions of surrounding agents' trajectories in the robot trajectory optimization. In this work, we propose a planner with explicit collision avoidance constraints that considers robot kinematic restrictions and adapts to dynamic agents' reactions to the robot motion.

3. Preliminaries and Problem Formulation

3.1. AVOCADO

AVOCADO [3] is a multi-agent motion planner designed to compute collision-free velocities in dynamic environments without assuming homogeneous cooperation among agents. Unlike ORCA [2], which enforces reciprocal responsibility sharing, AVOCADO dynamically estimates each agent's cooperation level and adapts its avoidance strategy accordingly.

The method considers single integrator holonomic dynamics:

$$\dot{\mathbf{p}}_r = \mathbf{v}_H, \tag{1}$$

where $\mathbf{p}_r \in \mathbb{R}^2$ is the position of the robot and $\mathbf{v}_H \in \mathbb{R}^2$ its holonomic velocity. The environment is populated by $N > 0$ dynamic agents. The robot and every agent $i \in \{1, \dots, N\}$ are represented by disks of radius r_r and r_i , respectively.

AVOCADO relies on the concept of VO [1], which defines, for each agent i , the set of relative velocities, \mathbf{v}_{rel} , leading to a collision within a time horizon τ :

$$VO_i = \{\mathbf{v}_{rel} \mid \exists t \in [0, \tau] \text{ s.t. } t\mathbf{v}_{rel} \in D_c(\mathbf{p}_i - \mathbf{p}_r, r_r + r_i)\}. \tag{2}$$

where $D_c(\mathbf{p}, r) = \{\mathbf{p}' \mid \|\mathbf{p}' - \mathbf{p}\| < r\}$ and $\|\bullet\|$ is the L2-norm of a vector.

To avoid collisions, the robot computes the minimum change \mathbf{u}_i required to leave VO_i :

$$\mathbf{u}_i = \left(\arg \min_{\mathbf{v} \in \partial VO_i} \|\mathbf{v}_{rel} - (\mathbf{v}_H^{pre} - \mathbf{v}_i)\| \right) - (\mathbf{v}_H^{pre} - \mathbf{v}_i), \tag{3}$$

where \mathbf{v}_H^{pre} is the preferred velocity, given by a higher-level motion planner, and ∂VO_i is the limit of VO_i . The avoidance responsibility is distributed according to the estimated cooperation $\alpha_i \in [0, 1]$, defining the admissible region:

$$OCA_i = \{\mathbf{v}_H \mid (\mathbf{v}_H - (\mathbf{v}_H^{pre} + (1 - \alpha_i)\mathbf{u}_i)) \cdot \mathbf{n}_i \geq 0\}, \tag{4}$$

where \mathbf{n}_i is the normal of ∂VO_i at the head of \mathbf{u}_i . This means that, if agent i cooperates in the collision avoidance α_i parts of \mathbf{u}_i , the robot must exert at least $1 - \alpha_i$ parts of \mathbf{u}_i to prevent collisions.

The degree of cooperation is unknown and needs to be estimated. AVOCADO estimates it using an adaptive control law based on a nonlinear opinion dynamics model [11,12]:

$$\dot{o}_i = -d_i x_i + d_i A_i \tanh(a_i o_i + c_i e_i) + b_i, \tag{5}$$

where $o_i = 2\alpha_i - 1$. Parameters d_i, a_i, b_i, c_i regulate opinion updating, tuning how fast the opinion vanishes in time (d_i), the opinion exchange among agents (a_i and c_i) and a prior bias on the degree of cooperation (b_i). A_i is the attention, measures interaction relevance and is modeled using the time to collision with agent i . e_i is a geometric cooperation cue derived from \mathbf{u}_i . This adaptive mechanism enables AVOCADO to handle heterogeneous and unpredictable behaviors without explicit communication.

The final velocity minimizes the deviation from the preferred velocity subject to all OCA_i constraints:

$$\mathbf{v}_H^* = \arg \min_{\mathbf{v}_H \in \bigcap_i OCA_i} \|\mathbf{v}_H - \mathbf{v}_H^{pre}\|. \tag{6}$$

3.2. Kinematic Model

We consider a differential-drive robot with a unicycle model with the same model and non-holonomic controller as in NH-ORCA [18]. The robot follows non-holonomic controls:

$$\mathbf{v}_{NH} = (v, \omega), \tag{7}$$

where v and ω are the linear and angular velocities, respectively. The robot moves considering the simple model of a unicycle:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v \cos \theta \\ v \sin \theta \\ \omega \end{pmatrix}, \tag{8}$$

where $\mathbf{p}_r = (x, y)$ and θ is the robot orientation.

The tracking error ε is the maximum deviation between the position following V_H under a holonomic controller and the position actually reached under a non-holonomic controller. A threshold ε_{max} bounds this error, considering the robot's actuation constraints v_{max}, ω_{max} (maximum linear and angular velocity, respectively), and $v_{max,\omega} = v_{max} - |\omega| \frac{l_W}{2}$, where l_W is the distance between the wheels. $v_{max,\omega}$ describes the robot maximum velocity limit given its angular velocity, based on the differential-drive constraints.

The robot follows the trajectories given by holonomic velocities $\mathbf{v}_H = (v_x, v_y)$ by rotating towards $\theta_H = \arctan(v_y, v_x)$ during T , where T is the minimum time set to achieve θ_H , and moving forward. The velocity that minimizes the tracking error [18] is

$$\omega = \frac{\theta_H}{T}, \quad v^* = |\mathbf{v}_H| \frac{\theta_H \sin(\theta_H)}{2(1 - \cos \theta_H)}. \tag{9}$$

Depending on kinematic limits (ω_{max}, v_{max}), motion is classified in three regions:

- Region R_{A_1} : $\omega \leq \omega_{max}, v^* \leq v_{max}$.
- Region R_{A_2} : $\omega \leq \omega_{max}, v^* > v_{max}$.
- Region R_B : $\omega > \omega_{max}$.

The controller, $f(\mathbf{v}_H) = (v, \omega)$, applies the following velocities depending on the velocity limits of the region:

$$f(\mathbf{v}_H) = \begin{cases} \left(v^*, \frac{\theta_H}{T} \right), & R_{A_1} \\ \left(v_{max}, \frac{\theta_H}{T} \right), & R_{A_2} \\ (0, \omega_{max}), & R_B \end{cases} \tag{10}$$

Using this controller, the tracking error, ε , is given by [18]

$$\varepsilon^2 = \begin{cases} \frac{2(1 - \cos \theta_H) - \sin^2 \theta_H T^2 |V_H|_{max}^2}{2(1 - \cos \theta_H)}, & R_{A_1} \\ |V_H|_{max}^2 T^2 - \frac{2|V_H|_{max} T^2 \sin \theta_H}{\theta_H} v_{max,\omega}, & R_{A_2} \\ + \frac{2T^2(1 - \cos \theta_H)}{\theta_H^2} v_{max,\omega}^2, & R_{A_2} \\ \left(|V_H|_{max} \frac{\theta_H}{\omega_{max}} \right)^2, & R_B \end{cases} \tag{11}$$

In this setup, the robots are assumed to have no acceleration limits, but could be easily included by considering a second-order integrator controller with saturation limits in acceleration as in [40] and extending the tracking error computation with this controller. Nevertheless, the focus of this work is on incorporating non-holonomic kinematic feasibility while maintaining computational efficiency. There are potential discontinuities in both linear and angular velocities, which could also be solved by changing the controller.

3.3. Problem Formulation

We formally describe the problem below.

Problem 1. *Given a robot with the kinematic model described in Section 3.2 navigating in an environment populated by $N > 0$ dynamic agents, we seek to find the non-holonomic control inputs that satisfy*

$$\mathbf{v}_{NH}^* = f \left(\arg \min_{\mathbf{v}_H} \|\mathbf{v}_H - \mathbf{v}_H^{pre}\| \right) \tag{12}$$

$$s.t. \quad \forall t \in [0, \tau] \text{ s.t. } t \mathbf{v}_{rel} \in \cup_{i=1}^N D_c(\mathbf{p}_i - \mathbf{p}_r, r_r + r_i), \tag{13}$$

where \mathbf{v}_{rel} is the relative velocity between the robot and agent i .

4. Method

The proposed method restricts AVOCADO’s planning space to holonomic velocities whose tracking error does not exceed a maximum threshold ε_{max} . To guarantee collision safety in the presence of this error, obstacles and the robot are inflated by ε_{max} , effectively increasing their radius. This ensures that the generated trajectories are safe and physically executable, and respect the robot’s kinematic constraints. In this way, we combine AVOCADO’s adaptation to surrounding agents’ degree of cooperation with NH-ORCA non-holonomic safety in the presence of kinematic constraints.

4.1. Admissible Velocity Set

A grid over (v_x, v_y) is defined. For each velocity, the tracking error ε is computed considering the controller f in Equation (10) and the robot’s kinematic limits over the

horizon T , using Equation (11). An example of the result of this computation is shown in Figure 2.

Velocities producing $\varepsilon \leq \varepsilon_{\max}$ form S_{AHV} . A rectangular approximation P_{AHV} simplifies integration in planning and control by producing a convex set of allowed velocities, while ensuring feasible velocities. The rectangle is represented in blue in Figure 2. P_{AHV} acts as a conservative convex envelope of the admissible velocity set S_{AHV} . This approximation simplifies integration into the quadratic program by producing linear constraints while ensuring that all selected velocities remain feasible for the differential-drive controller. The approximation trades representational accuracy for computational simplicity, enabling real-time execution.

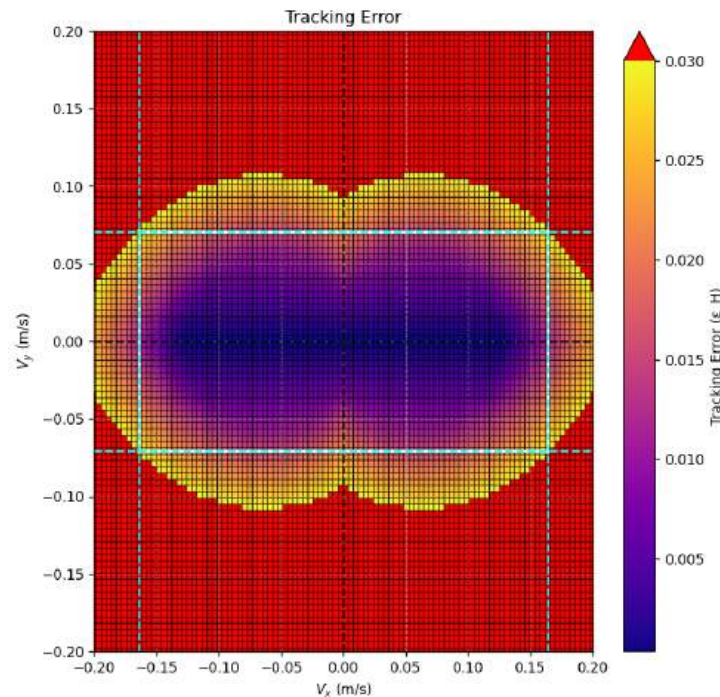


Figure 2. Admissible holonomic velocity set S_{AHV} for a differential-drive robot. The blue rectangle P_{AHV} approximates the set, with semi-infinite boundary lines extending each side (dashed). The parameters used in the computation are $v_{\max} = 0.15$ m/s, $\omega_{\max} = 4.96$ rad/s, $l_w = 0.0525$ m, $\varepsilon_{\max} = 0.03$ and $T = 0.35$ s.

The rectangle is rotated to align with the robot’s current heading θ using the standard planar rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \tag{14}$$

In this way, P_{AHV} can be computed offline once, and a matrix product is computed every control step. If the physical parameters of the robot (i.e., v_{\max} , ω_{\max} , l_w) change significantly due to environmental changes, such as actuation failures, P_{AHV} should be recomputed again. An example of a rotated P_{AHV} without extra computations is shown in Figure 3.

4.2. Integration into AVOCADO Logic

The core idea is to ensure that velocities selected by AVOCADO remain safe even when the robot’s non-holonomic controller deviates from the desired velocity given ε_{\max} .

First, we define the new VO_i restriction that considers ε_{\max} :

$$VO_{i,\varepsilon_{\max}} = \{ \mathbf{v} \mid \exists t \in [0, \tau] \quad \text{s.t.} \\ t\mathbf{v} \in D_c(\mathbf{p}_i - \mathbf{p}_r, r_r + r_i + 2\varepsilon_{\max}) \}. \tag{15}$$

Here, we assume agent i is using the same ϵ_{\max} . This may not hold but, in practice, the discrepancies between the modeled behavior and actual behavior of agent i are mitigated with Equation (5), and integrated as part of the opinion on agent i 's degree of cooperation. Thus, the adaptive cooperation estimation mechanism partially compensates for discrepancies between the modeled and actual behaviors of surrounding agents by adjusting the avoidance responsibility online. We acknowledge that this possible discrepancy is a limitation, which could be alleviated by expert knowledge or a deeper estimation via a perception algorithm.

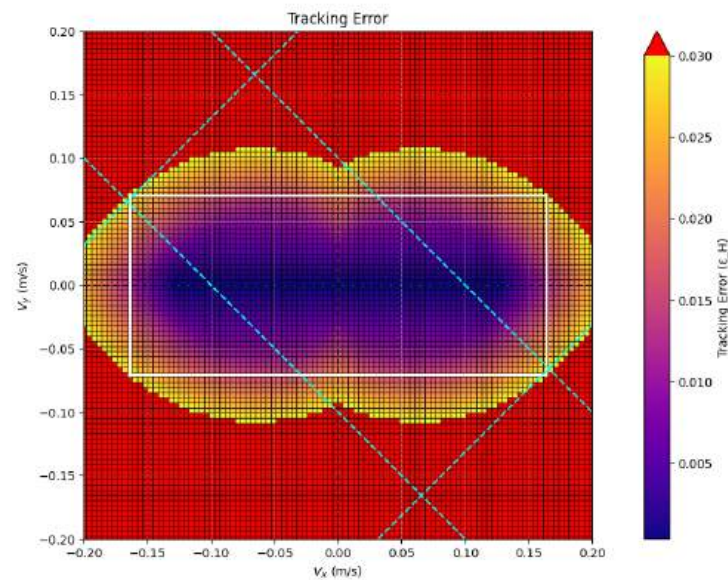


Figure 3. Admissible holonomic velocity set S_{AHV} and rectangle P_{AHV} , as in Figure 2. The rectangle is rotated given $\theta = \frac{\pi}{4}$ rad.

The change in the velocity of the system to avoid collisions is adapted to the newly defined $VO_{i,\epsilon_{\max}}$

$$\mathbf{u}_{i,\epsilon_{\max}} = \left(\arg \min_{\mathbf{v}_{rel} \in \partial VO_{i,\epsilon_{\max}}} \|\mathbf{v}_{rel} - (\mathbf{v}_H^{pre} - \mathbf{v}_i)\| \right) - (\mathbf{v}_H^{pre} - \mathbf{v}_i). \tag{16}$$

Note that the adaptive control law of AVOCADO keeps its properties unchanged. In this case, e_i must be defined with the projection over $\mathbf{u}_{i,\epsilon_{\max}}$. The collision avoidance region is defined considering the tracking error:

$$OCA_{i,\epsilon_{\max}} = \{ \mathbf{v}_H \mid (\mathbf{v}_H - (\mathbf{v}_H^{pre} + (1 - \alpha_i)\mathbf{u}_{i,\epsilon_{\max}})) \cdot \mathbf{n}_i \geq 0 \}. \tag{17}$$

Finally, the problem is formulated as a quadratic program considering the new collision avoidance restrictions, $OCA_{i,\epsilon_{\max}}$, and the kinematic restrictions, P_{AHV} , over the holonomic velocity set:

$$\mathbf{v}_{NH}^* = f \left(\arg \min_{\mathbf{v}_H} \|\mathbf{v}_H - \mathbf{v}_H^{pre}\| \right) \tag{18}$$

$$s.t. \quad \mathbf{v}_H \in \bigcap_i OCA_{i,\epsilon_{\max}} \tag{19}$$

$$\mathbf{v}_H \in P_{AHV}. \tag{20}$$

The linear constraints (19) and (20) are exemplified in Figure 4.

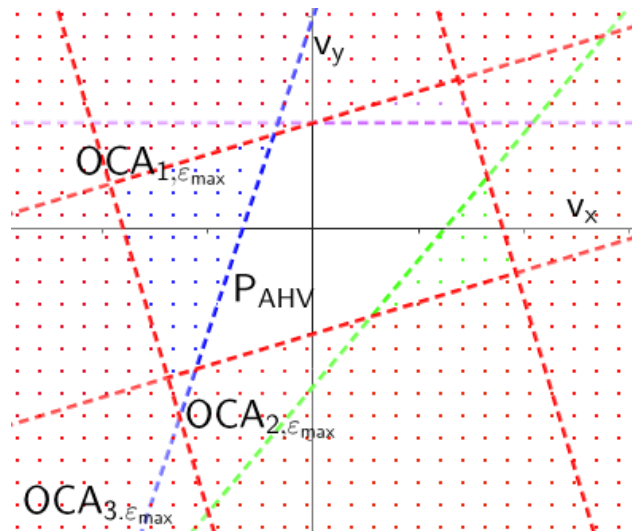


Figure 4. An example of the linear constraints present in the problem, including a rotated P_{AHV} (red) and three $OCA_{i,\epsilon_{max}}$, $i = \{1, 2, 3\}$ (purple, blue and green, respectively).

The summary of the control loop of DD-AVOCADO is represented in Algorithm 1. Lines 1 and 3 refer to Section 4.1, line 5 to Equation (5), line 6 to Equations (15)–(17), line 7 to Equation (18) and line 8 to Equation (10).

Algorithm 1: DD-AVOCADO control loop

Input: $v_{max}, \omega_{max}, l_W, f$
Data: ϵ_{max}

- 1 $P_{AHV} \leftarrow \text{ComputeConstraints}(v_{max}, \omega_{max}, l_W, \epsilon_{max})$
- 2 **for** Every control period **do**
- 3 $P_{AHV}^\theta \leftarrow \text{Rotate}(R(\theta), P_{AHV})$
- 4 **for** $i \in \{1, \dots, N\}$ **do**
- 5 $\alpha_i \leftarrow \text{UpdateOpinion}(\alpha_i, \mathbf{p}_r, \mathbf{p}_i, \mathbf{v}_r, \mathbf{v}_i)$
- 6 $OCA_{i,\epsilon_{max}} \leftarrow \text{GetOCA}(\alpha_i, \mathbf{p}_r, \mathbf{p}_i, \mathbf{v}_r, \mathbf{v}_i, \epsilon_{max})$
- 7 $\mathbf{v}_H \leftarrow \text{SolveProgram}(\mathbf{v}_H^{pre}, P_{AHV}, \bigcap_i OCA_{i,\epsilon_{max}})$
- 8 $\mathbf{v}_{NH} \leftarrow f(\mathbf{v}_H)$

5. Experiments

To evaluate DD-AVOCADO, we conducted simulations in dynamic multi-agent environments. The goal was to assess whether incorporating non-holonomic constraints improves system behavior and reduces collisions, especially in congested or non-cooperative scenarios. In all the experiments, we considered a naive high-level planner that sets \mathbf{v}_H^{pre} to the velocity that drives the robot straight to the goal. Even though we evaluated the planners in simple local planning environments with no static obstacles, the local planners could be combined with a more sophisticated global planner that considers the static map of the environment.

We measured the mean and the standard deviation of the computational time for one robot surrounded by 19 dynamic agents in 10,000 iterations. The mean results for one control step are 1.101 ± 0.032 ms, measured in a AMD Ryzen 7 5800X 8-core processor. This illustrates that the formulation is suitable for high-speed applications or low-capability devices. The computation of P_{AHV} takes less than 0.1 s for a 80×80 grid size.

5.1. Qualitative Results

Figures 5 and 6 illustrate the trajectories of multiple agents using DD-AVOCADO in representative scenarios.

Figure 5 shows a circular position-exchange scenario with 20 agents. Agents are initially distributed along a circumference, with each assigned a target corresponding to another agent's initial position, creating intentional crossings. Agent bodies are disks of radius 0.2 m; arrows indicate instantaneous orientation; and dashed lines depict trajectories. Cooperative agents are colored, and non-cooperative agents are shown in black. Cooperative agents use DD-AVOCADO with $v_{\max} = 0.5$ m/s and $|\omega|_{\max} = 0.96$ rad/s, respecting kinematic constraints and inter-agent interactions. Non-cooperative agents move directly toward their target at 0.25 m/s, ignoring other agents. Results demonstrate smooth, safe trajectories even under high congestion and non-cooperative behavior.

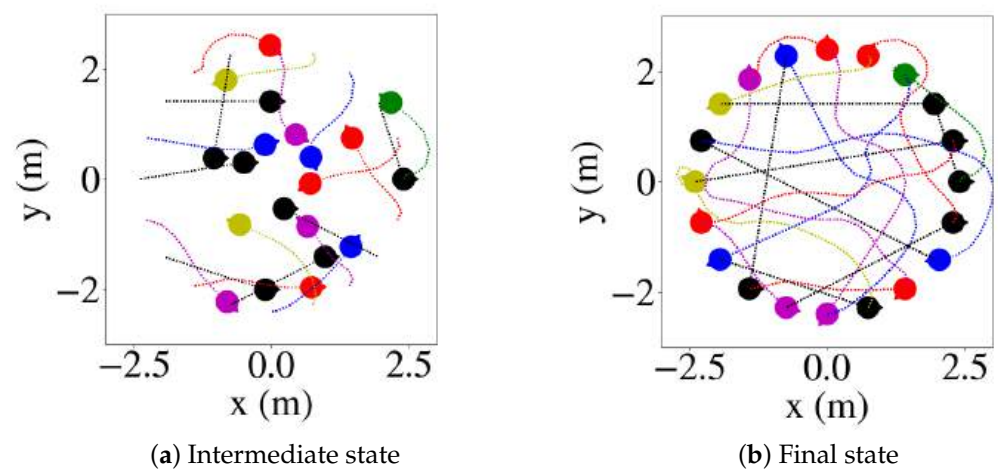


Figure 5. Circular exchange scenario with 20 agents. Cooperative agents are colored; non-cooperative are black. (a) Mid-simulation; (b) final positions and full trajectories.

Figure 6 shows the same scenario, but this time with all cooperative robots. Respecting kinematic constraints, agents navigate efficiently without collisions, even in congested environments.

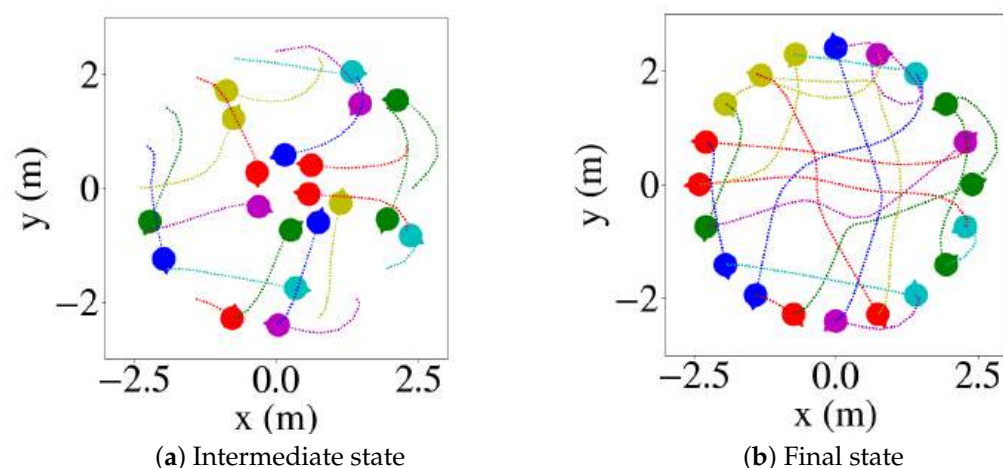


Figure 6. Successful trajectories in a dense scenario using DD-AVOCADO. Kinematic constraints and cooperation allow all agents to reach their goals collision-free.

Figure 7 shows a cooperative scenario with 15 agents comparing AVOCADO and DD-AVOCADO. We set the number of agents to 15 to simplify the visualization. On the

left (AVOCADO), multiple collisions occur early in the simulation, disrupting trajectories. On the right (DD-AVOCADO), all agents reach their targets safely.

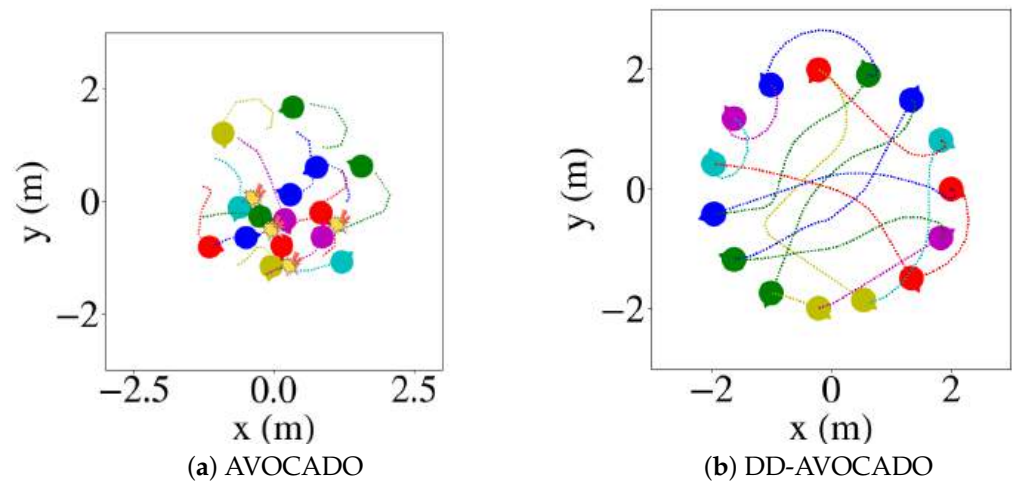


Figure 7. Circular scenario with 15 agents. (a) Five collisions occur with AVOCADO. (b) All agents reach goals using DD-AVOCADO. Cooperative agents are colored; non-cooperative are black.

By incorporating non-holonomic tracking errors into collision constraints and selecting only velocities within a maximum allowable error, all agents reach their goals collision-free. Without explicit kinematic constraints, as in AVOCADO, collisions occur inevitably, as the controller tries to reach velocities whose tracking error is unsafe. In contrast, DD-AVOCADO ensures safe, realistic, and efficient navigation.

Overall, simulations confirm DD-AVOCADO's effectiveness: agents respecting kinematic constraints complete trajectories safely, even in congested environments with non-cooperative agents, whereas ignoring these constraints leads to collisions and interrupted paths. This demonstrates that including non-holonomic constraints significantly improves safe and efficient navigation in dynamic multi-agent systems.

5.2. Simulation Setup

To evaluate robustness and quantify improvements over reference planners under different cooperation levels and physical limits, five representative scenarios were defined (Table 1). Each scenario was tested with four planners: ORCA [2], NH-ORCA [4], AVOCADO [3], and DD-AVOCADO. Simulations were conducted in a 3×3 m environment, with 25 repetitions per scenario using randomized initial and goal positions along a circle, avoiding overlaps.

Two maximum angular velocities ω_{\max} were considered, high (2.5 rad/s) and low (0.5 rad/s), to analyze how turning capabilities affect planning and collision avoidance. Table 1 summarizes all scenarios. In all scenarios, the v_{\max} of cooperative robots driven by the tested algorithm was set to 0.5 m/s and that of non-cooperative agents was set to 0.25 m/s.

Scenarios E1 and E2 test the planners in crowded cooperative and non-cooperative scenarios, respectively. E3 and E4 increase complexity by reducing the turning capabilities of the robots. Finally, E5 tests the planners in a scenario with fewer interactions, with the objective of also evaluating the planners in a less challenging environment. In contrast to ORCA and NH-ORCA, which assume reciprocal cooperation among robots executing the same algorithm, the experiments include heterogeneous cooperative and non-cooperative agents. This evaluates whether the proposed formulation preserves AVOCADO's adaptive interaction capabilities while enforcing kinematic feasibility.

Table 1. Simulation scenarios for each planning method.

Scenario	# Agents	% Coop.	ω_{\max} [rad/s]
E1	20	100%	2.5
E2	20	50%	2.5
E3	20	100%	0.5
E4	20	50%	0.5
E5	10	100%	2.5

5.3. Quantitative Results

Table 2 summarizes collision rates and average arrival times across the five scenarios and four planners. Collision rate is defined as the ratio of observed collisions to the maximum possible unique collisions across all repetitions, providing a measure of system safety.

Table 2. Collision rate (best in bold) and mean arrival time divided by the distance from the initial position to the goal, by scenario and method.

Scenario	Method	Coll. Rate [%]	Mean Time [s]
E1	ORCA	17.6	3.25
	NH-ORCA	0.0	5.03
	AVOCADO	13.2	3.25
	DD-AVOCADO	0.0	4.92
E2	ORCA	28.7	4.30
	NH-ORCA	6.1	5.37
	AVOCADO	27.6	4.24
	DD-AVOCADO	2.4	5.30
E3	ORCA	71.0	3.93
	NH-ORCA	0.0	23.57
	AVOCADO	66.8	4.13
	DD-AVOCADO	0.0	22.48
E4	ORCA	91.7	4.74
	NH-ORCA	31.2	12.38
	AVOCADO	89.1	4.71
	DD-AVOCADO	25.6	11.76
E5	ORCA	12.8	7.17
	NH-ORCA	0.0	9.85
	AVOCADO	10.1	8.26
	DD-AVOCADO	0.0	8.96

The results show that ORCA has high collision rates in all scenarios, as it does not consider robot kinematic constraints. NH-ORCA reduces collisions by enforcing kinematics, but still shows higher rates under low cooperation, as it considers that every agent in the environment is part of the multi-robot team. AVOCADO exhibits high collision rates (up to 88%), indicating poor adaptation to unmodeled kinematic constraints.

DD-AVOCADO consistently achieves the lowest collision rates, often 0% (E1, E3, E5), demonstrating that incorporating non-holonomic constraints is critical for safe multi-agent navigation. Even in challenging scenarios with low cooperation and low ω_{\max} (E2, E4), DD-AVOCADO maintains significantly lower collisions than NH-ORCA, highlighting its robustness.

Average arrival times are slightly higher for NH-ORCA and DD-AVOCADO due to prioritizing physically feasible, collision-free trajectories. Times to reach the goal are computed only over non-colliding trajectories. Since DD-AVOCADO achieves a higher success rate, it completes trajectories in scenarios that are more complex and where other planners often fail to reach the goal. In contrast, those planners typically succeed only in simpler scenarios where more direct paths are available. As a result, the mean navigation

time for DD-AVOCADO includes more difficult cases, which can increase the average value. While this affects the direct comparability of the metric, it is reported for transparency. Notably, despite this effect, DD-AVOCADO achieves lower navigation times than NH-ORCA while also exhibiting lower collision rates, demonstrating efficient navigation and safety, benefiting from the adaptation of the opinion dynamic law.

In summary, quantitative results confirm that DD-AVOCADO effectively reduces collisions for an efficient multi-agent navigation, by combining both AVOCADO's adaptation to other agents' uncertain behavior and NH-ORCA's consideration of kinematic constraints.

5.4. Hardware Experiments

We tested DD-AVOCADO through hardware experiments in scenarios with three robots and two pedestrians. The robots used are Turtlebot 2 platforms. The physical parameters of the robots are $v_{max} = 0.65$ m/s, $\omega_{max} = \pi$ rad/s and $l_w = 0.23$ m. We set fixed $\epsilon_{max} = 0.1$ m and $T = 0.35$ s, and a control timestep of 0.05 s. The pedestrians were not told to have a specific behavior, and they both avoided and did not avoid the robots. A motion capture system was used to measure the positions of the robots and pedestrians over time, and an Extended Kalman Filter with a constant velocity assumption was employed to track their positions and velocities.

During the experiments, the robots were set in a circle and had to reach their respective antipodal position in the circle. When all of them reached their goals, the experiment was restarted uninterruptedly. The pedestrians wandered in between the robots.

During the experiments, no collisions were observed, even in situations where the pedestrians were not cooperating with the robots and even blocking their way. The robots were able to collaborate with each other with efficient maneuvers, while actively avoiding collisions with pedestrians. An example of the trajectories of the robots and humans in an interval of the experiments is represented in Figure 8, where we can see how the robots effectively reached their goals.

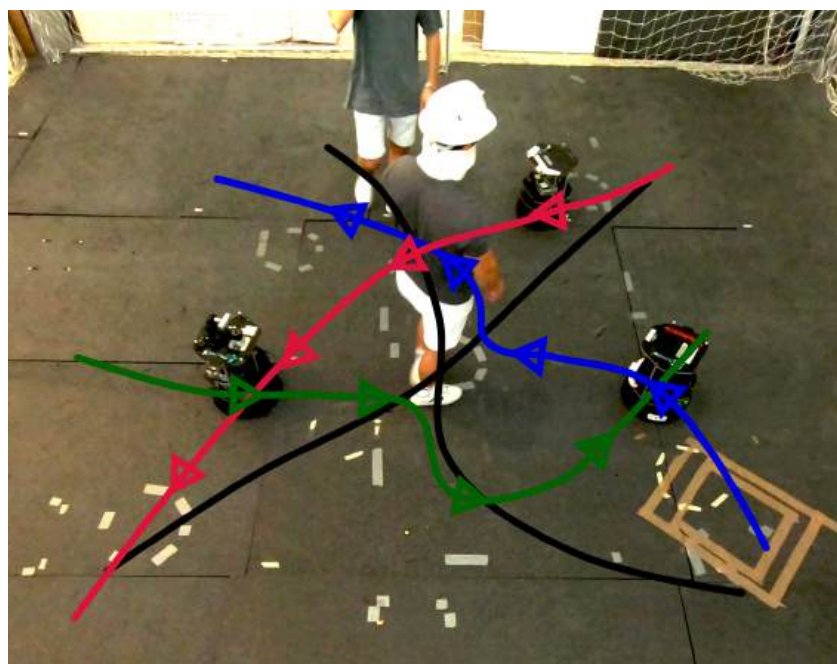


Figure 8. Trajectories of the robots (blue, red and green) and the pedestrians (black) in an interval of the experiments.

In addition, we conducted a set of experiments that simulated a robot with double the distance between wheels by setting $l_w = 0.46$ m. This shows the performance of the planner

facing different physical capabilities. The robot's rotation was limited with respect to the other settings, resulting in longer trajectories with larger curves. An example of trajectories under these modified physical constraints is shown in Figure 9. Even in these settings, the robots showed effective navigation without collisions.

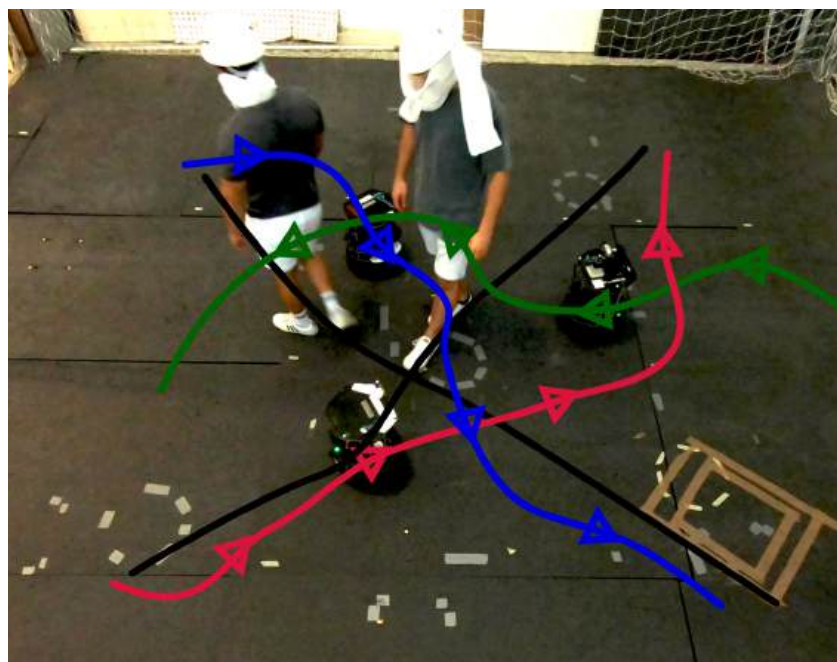


Figure 9. Trajectories of the robots (blue, red and green) and the pedestrians (black) in an interval of the experiments. The robot is constrained with $l_w = 0.46$ m.

The experimental results are also provided as a Video Abstract accompanying this paper.

6. Conclusions

This work has presented DD-AVOCADO, an extension of the AVOCADO multi-agent planner that incorporates the kinematic constraints of differential-drive robots. By considering the tracking error between desired holonomic velocities and the actual non-holonomic execution, a safe admissible velocity set is defined and enforced as geometric constraints in the velocity space. This ensures that planned velocities are both collision-free and physically executable. Simulation results in dense and partially cooperative scenarios show that DD-AVOCADO significantly reduces collisions compared to AVOCADO, ORCA, and NH-ORCA, while maintaining feasible and smooth trajectories, demonstrating its effectiveness for safe and realistic multi-agent navigation. The proposed formulation enables AVOCADO to be deployed on differential-drive robots while preserving its ability to interact safely with heterogeneous agents.

This work is now limited to differential-drive robots. Future work may extend it to other dynamic models by considering other controller and dynamic equations. In addition, we plan to extend the dimensionality of the planner to allow collision avoidance of aerial platforms as drones. The work is also limited to a constant velocity assumption for the model of the VO and pure reactive behavior. Further research could involve studying other global high-level planners that let the robot navigate through complex scenarios, rather than simply proposing velocities that point to the 2-D goal position. Finally, the rectangular approximation of the admissible velocity set makes the optimization suboptimal. Further research could study computing the nonlinear optimization in a trackable way.

Author Contributions: Conceptualization, D.M.-B., D.L., L.R. and L.M.; methodology, D.M.-B., D.L., L.R. and L.M.; software, D.M.-B., D.L.; validation, D.M.-B., D.L., L.R. and L.M.; formal analysis, D.M.-B., D.L., L.R. and L.M.; investigation, D.M.-B., D.L., L.R. and L.M.; resources, D.M.-B., D.L.; data curation, D.M.-B., D.L.; writing—original draft preparation, D.M.-B., D.L.; writing—review and editing, D.M.-B., D.L., L.R. and L.M.; visualization, D.M.-B., D.L.; supervision, L.R. and L.M.; project administration, L.M.; funding acquisition, L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by MICIU/AEI/10.13039/501100011033 and ERDF/EU under grants PID2022-139615OB-I00 and PRE2020-094415 from the Government of Aragón under grant DGA-T45-23R.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: All data is contained within the article.

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

References

1. Fiorini, P.; Shiller, Z. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [[CrossRef](#)]
2. Van Den Berg, J.; Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 3–19.
3. Martínez-Baselga, D.; Sebastián, E.; Montijano, E.; Riazuelo, L.; Sagüés, C.; Montano, L. AVOCADO: Adaptive Optimal Collision Avoidance Driven by Opinion. *IEEE Trans. Robot.* **2025**, *41*, 2495–2511. [[CrossRef](#)]
4. Alonso-Mora, J.; Beardsley, P.; Siegwart, R. Cooperative collision avoidance for nonholonomic robots. *IEEE Trans. Robot.* **2018**, *34*, 404–420. [[CrossRef](#)]
5. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **2002**, *4*, 23–33. [[CrossRef](#)]
6. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [[CrossRef](#)]
7. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Phys. Rev. E* **1995**, *51*, 4282. [[CrossRef](#)]
8. Vesentini, F.; Muradore, R.; Fiorini, P. A survey on velocity obstacle paradigm. *Robot. Auton. Syst.* **2024**, *174*, 104645. [[CrossRef](#)]
9. Van den Berg, J.; Lin, M.; Manocha, D. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2008; pp. 1928–1935.
10. Guo, K.; Wang, D.; Fan, T.; Pan, J. VR-ORCA: Variable responsibility optimal reciprocal collision avoidance. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4520–4527. [[CrossRef](#)]
11. Bizyaeva, A.; Franci, A.; Leonard, N.E. Nonlinear opinion dynamics with tunable sensitivity. *IEEE Trans. Autom. Control* **2022**, *68*, 1415–1430. [[CrossRef](#)]
12. Leonard, N.E.; Bizyaeva, A.; Franci, A. Fast and flexible multiagent decision-making. *Annu. Rev. Control Robot. Auton. Syst.* **2024**, *7*, 19–45. [[CrossRef](#)]
13. Cathcart, C.; Santos, M.; Park, S.; Leonard, N.E. Proactive opinion-driven robot navigation around human movers. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 4052–4058.
14. Amorim, G.; Bizyaeva, A.; Franci, A.; Leonard, N.E. Spatially-Invariant Opinion Dynamics on the Circle. *IEEE Control Syst. Lett.* **2024**, *8*, 3231–3236. [[CrossRef](#)]
15. Van Den Berg, J.; Snape, J.; Guy, S.J.; Manocha, D. Reciprocal collision avoidance with acceleration-velocity obstacles. In *2011 IEEE International Conference on Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2011; pp. 3475–3482.
16. Rufli, M.; Alonso-Mora, J.; Siegwart, R. Reciprocal collision avoidance with motion continuity constraints. *IEEE Trans. Robot.* **2013**, *29*, 899–912. [[CrossRef](#)]
17. Snape, J.; Van Den Berg, J.; Guy, S.J.; Manocha, D. Smooth and collision-free navigation for multiple robots under differential-drive constraints. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: Piscataway, NJ, USA, 2010; pp. 4584–4589.
18. Alonso-Mora, J.; Breitenmoser, A.; Rufli, M.; Beardsley, P.; Siegwart, R. Optimal reciprocal collision avoidance for multiple non-holonomic robots. In *Distributed Autonomous Robotic Systems: The 10th International Symposium*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 203–216.
19. Alonso-Mora, J.; Breitenmoser, A.; Beardsley, P.; Siegwart, R. Reciprocal collision avoidance for multiple car-like robots. In *2012 IEEE International Conference on Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2012; pp. 360–366.

20. Everett, M.; Chen, Y.F.; How, J.P. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 3052–3059.
21. Chen, C.; Liu, Y.; Kreiss, S.; Alahi, A. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2019; pp. 6015–6022.
22. Chen, C.; Hu, S.; Nikdel, P.; Mori, G.; Savva, M. Relational graph learning for crowd navigation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 10007–10013.
23. Fan, T.; Long, P.; Liu, W.; Pan, J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot. Res.* **2020**, *39*, 856–892. [[CrossRef](#)]
24. Qin, J.; Qin, J.; Qiu, J.; Liu, Q.; Li, M.; Ma, Q. SRL-ORCA: A socially aware multi-agent mapless navigation algorithm in complex dynamic scenes. *IEEE Robot. Autom. Lett.* **2023**, *9*, 143–150. [[CrossRef](#)]
25. Xie, Z.; Dames, P. Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles. *IEEE Trans. Robot.* **2023**, *39*, 2700–2719. [[CrossRef](#)]
26. Martinez-Baselga, D.; Riazuelo, L.; Montano, L. RUMOR: Reinforcement learning for understanding a model of the real world for navigation in dynamic environments. *Robot. Auton. Syst.* **2025**, *191*, 105020. [[CrossRef](#)]
27. Ames, A.D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; Tabuada, P. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*; IEEE: Piscataway, NJ, USA, 2019; pp. 3420–3431.
28. Tayal, M.; Singh, R.; Keshavan, J.; Kolathaya, S. Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach. In *2024 American Control Conference (ACC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 3722–3727.
29. Tayal, M.; Kolathaya, S. Polygonal Cone Control Barrier Functions (PolyC2BF) for safe navigation in cluttered environments. In *2024 European Control Conference (ECC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 2212–2217.
30. Roncero, A.S.; Muchacho, R.I.C.; Ögren, P. Multi-agent obstacle avoidance using velocity obstacles and control barrier functions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 6638–6644.
31. Huang, J.; Zeng, J.; Chi, X.; Sreenath, K.; Liu, Z.; Su, H. Dynamic collision avoidance using velocity obstacle-based control barrier functions. *IEEE Trans. Control Syst. Technol.* **2025**, *33*, 1601–1615. [[CrossRef](#)]
32. Long, K.; Dhiman, V.; Leok, M.; Cortés, J.; Atanasov, N. Safe control synthesis with uncertain dynamics and constraints. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7295–7302. [[CrossRef](#)]
33. Brito, B.; Floor, B.; Ferranti, L.; Alonso-Mora, J. Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4459–4466. [[CrossRef](#)]
34. Adajania, V.K.; Sharma, A.; Gupta, A.; Masnavi, H.; Krishna, K.M.; Singh, A.K. Multi-modal model predictive control through batch non-holonomic trajectory optimization: Application to highway driving. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4220–4227. [[CrossRef](#)]
35. Mavrogiannis, C.; Balasubramanian, K.; Poddar, S.; Gandra, A.; Srinivasa, S.S. Winding through: Crowd navigation via topological invariance. *IEEE Robot. Autom. Lett.* **2022**, *8*, 121–128. [[CrossRef](#)]
36. Ren, K.; Ahn, H.; Kamgarpour, M. Chance-constrained trajectory planning with multimodal environmental uncertainty. *IEEE Control Syst. Lett.* **2022**, *7*, 13–18. [[CrossRef](#)]
37. de Groot, O.; Ferranti, L.; Gavrila, D.M.; Alonso-Mora, J. Scenario-based motion planning with bounded probability of collision. *Int. J. Robot. Res.* **2025**, *44*, 1507–1525. [[CrossRef](#)]
38. de Groot, O.; Ferranti, L.; Gavrila, D.; Alonso-Mora, J. Globally Guided Trajectory Planning in Dynamic Environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2023; pp. 10118–10124.
39. de Groot, O.; Ferranti, L.; Gavrila, D.M.; Alonso-Mora, J. Topology-Driven Parallel Trajectory Optimization in Dynamic Environments. *IEEE Trans. Robot.* **2025**, *41*, 110–126. [[CrossRef](#)]
40. Alonso-Mora, J.; Gohl, P.; Watson, S.; Siegwart, R.; Beardsley, P. Shared control of autonomous vehicles based on velocity space optimization. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2014; pp. 1639–1645.

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.