2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Fast Decentralized Multi-Agent Collision Avoidance Based on Safe-Reachable Sets

Zikai Ouyang*, Junwei Liu*, Haibo Lu† and Wei Zhang†

Abstract— **This paper presents a decentralized multi-agent collision avoidance method for systems with single integrator dynamics and identical maximum speeds. The key to our approach lies in the concept of safe-reachable sets, which define the set of positions that each agent can reach while avoiding collisions with its neighbors for any admissible controllers. With this concept, we develop a distributed controller by solving an online convex program, which is shown to guarantee collision-free trajectories. Furthermore, under a no temporary deadlock condition, we establish that each agent converges to its target position. Our approach is also efficient in terms of makespan, representing the total time needed for convergence. Simulation results demonstrate the effectiveness of our approach in terms of safety, convergence, and efficiency.**

I. INTRODUCTION

Multi-agent collision avoidance is a fundamental problem in robotics. Key challenges of this problem include safety, scalability, and stability. To ensure safety, all agents must remain collision-free at all times, with each agent assigned a safety radius to prevent collisions. Scalability, another important property, implies that the approaches scale linearly with respect to the number of agents as those that scale exponentially may not be appropriate for a large group of agents. Stability is also critical and it refers to the successful convergence of each agent to its intended goal state. While a common issue that disrupts stability is a deadlock, which occurs when several agents get stuck together before reaching their goal states. In such cases, convergence cannot be achieved. Multi-agent collision avoidance is widely used in various fields, such as multi-agent navigation [1], warehouse inventory [2], traffic management [3], etc.

Multi-agent collision avoidance can be solved by centralized and decentralized methods. Centralized methods, such as those proposed in [4] and [5], view all agents as a composite system and each agent computes its input based on the information of all agents. For instance, the authors in [5] formulate the multi-agent collision avoidance problem as a differential game and establish safety and convergence guarantees for single integrators under certain assumptions. However, centralized methods have some limitations in practice as they scale poorly to large-scale systems. On the other hand, decentralized methods allow each agent to compute its input based on local information, making them more scalable and practical for a system with a large number of agents. This paper focuses on solving multi-agent collision avoidance in a decentralized manner.

Considerable effort has been devoted to developing decentralized collision avoidance methods for multi-agent systems. One of the most popular methods is based on the reciprocal velocity obstacle (RVO) [6], [7], [8], [9]. RVO extends the concept of velocity obstacle (VO) [10] by assuming that agents share equal responsibility for avoiding collisions. Using RVO, the optimal reciprocal collision avoidance (ORCA) method linearizes the input constraint induced by RVO and reduces the multi-agent collision avoidance problem to solving a low-dimensional linear program [7]. In general, ORCA cannot guarantee convergence because conflicting RVOs may cause deadlocks. Unlike RVO-based approaches, the control barrier function (CBF) approach transforms state constraints induced by safety into linear input constraints through the construction of barrier functions for each agent [11]. CBF formulates the multi-agent collision avoidance problem as a quadratic programming (QP) problem. However, the intersection of multiple linear constraints may be empty in dense conditions, violating the safety requirement. Furthermore, the paper [11] considers the concept of temporary deadlock, where at least one agent's solution to the QP problem is zero, and proposes a heuristic approach to resolve two out of three types of temporary deadlock.

Another kind of method is the geometric-based approach, focusing on using geometrical features to analyze the behavior of agents in the environment. Buffered Voronoi cell (BVC) [12] is a typical geometric-based approach that extends the concept of Voronoi cell, which has been widely used in multi-agent systems [13], [14], through retracting the edge of a Voronoi cell by a safety radius for each agent. Collision avoidance is performed by planning the agents' paths within their respective BVCs using a framework of model predictive control. In particular, the state constraints used during the predictive horizon are based on the same BVC with the current position information. BVC ensures

* These authors contributed equally to this work.

† Corresponding authors.

This work was supported in part by the National Natural Science Foundation of China under Grant 62073159 and Grant 62003180, in part by the Science, Technology, and Innovation Commission of Shenzhen Municipality under Grant JCYJ20220530115010023, in part by Shenzhen Science and Technology Program under Grant JCYJ20200109141601708, and in part by the Shenzhen Key Laboratory of Control Theory and Intelligent Systems under Grant ZDSYS20220330161800001.

Zikai Ouyang and Wei Zhang are with the Shenzhen Key Laboratory of Control Theory and Intelligent Systems, School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China; and also with the Peng Cheng Laboratory, Shenzhen 518000, China ouyang2022@mail.sustech.edu.cn, zhangw3@sustech.edu.cn

Junwei Liu is with the Shenzhen Key Laboratory of Control Theory and Intelligent Systems, School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China liujw@sustech.edu.cn

Haibo Lu is with the Peng Cheng Laboratory, Shenzhen 518000, China luhb@pcl.ac.cn

collision-free motion for discrete-time single-integrator systems, but fails to guarantee convergence due to the possibility of deadlocks. Recently, the authors in [15] propose heuristic methods to resolve certain types of deadlocks, and the paper [16] extends BVC to weighted BVC.

In this paper, we study the problem of steering each agent to its target position while avoiding collisions with other agents. We propose a decentralized collision avoidance method based on safe-reachable sets (SRSs) for a class of multi-agent systems with single-integrator dynamics. We derive a distributed version of SRSs to represent the set of all positions that an agent can reach while avoiding collisions with neighbors for any admissible controllers. The concept of SRS was first introduced in [17], and has been recently reformulated in [18], [19] for solving multiplayer adversarial games. Motivated by the definition of SRS, we construct an objective function and formulate the multi-agent collision problem as an online robust optimization problem, where each agent considers the worst-case scenarios created by neighboring agents, thereby ensuring safety regardless of their decisions. We then reduce this problem to solving an online convex program, which enables the design of a distributed controller. Our method is distributed as only the neighbors' information is used in the convex program to compute the input of each agent.

The contribution of this paper can be summarized as follows. Firstly, we prove that our method guarantees collision avoidance for single integrators with identical maximum speeds. While some other methods like ORCA and CBF require the optimization problem that generates input to be feasible all the time for collision-free motion. Secondly, our method also guarantees convergence under the no temporary deadlock condition. To the best of our knowledge, such a sufficient convergence guarantee is not given in existing decentralized methods. Thirdly, our method excels in terms of makespan, which represents the total time required for convergence. Simulation results demonstrate that our approach outperforms BVC under this metric.

The remainder of this paper is organized as follows. In Section II, we formulate the multi-agent collision avoidance problem. In Section III, we design a distributed controller based on the concept of SRSs. Section IV provides simulation results to demonstrate the effectiveness of our method. We state our conclusions and future work in Section V.

II. Problem Formulation

We consider a team of N agents indexed by $\mathcal{N} = \{1, 2, \ldots, N\}$ moving on the Euclidean space $\mathbb{R}^n$. The dynamics of the agents are described by the single integrator model

$$\dot{x}_i = u_i,\ i \in \mathcal{N} \tag{1}$$

where $x_i \in \mathbb{R}^n$ and $u_i \in \mathcal{U}$ are the position and velocity of agent i, respectively. We denote x_i and u_i as the state and control input of agent i, respectively and $\mathcal{U}$ as the input constraint set given by

$$\mathcal{U} = \{v \in \mathbb{R}^n \mid ||v|| \leq v_{\max}\} \tag{2}$$

with v_{max} being the maximum speed. We define the joint state as $x = [x_0^T, x_1^T, \ldots, x_N^T]^T$. Each agent i, $i \in \mathcal{N}$, has a safety radius r_i to avoid collisions. Agent i and agent j are said to be collision-free if their distance is greater than the sum of their safety radii, i.e., $\|x_i - x_j\| \geq r_{ij}$, where $r_{ij} = r_i + r_j$. Let $h_{ij}(x_i, x_j) = \|x_i - x_j\|^2 - r_{ij}^2$. Then the collision-free set for all agents can be written as

$$\mathcal{C} = \{x \in \mathbb{R}^{nN} \mid h_{ij}(x_i, x_j) \geq 0, \forall 1 \leq i < j \leq N\}. \tag{3}$$

Each agent i is assumed to have a sensing radius $R_i > 0$, and its neighborhood set is defined as

$$\mathcal{N}_i = \{j \in \mathcal{N} | \|x_i - x_j\| \leq R_i, j \neq i\}$$

The sensing region of agent i is expressed as

$$\mathcal{X}_i = \{y \in \mathbb{R}^n \mid g_i(y) \leq 0\}$$

where $g_i(y) = ||y - x_i||^2 - R_i^2$. To avoid collisions, each agent only accesses the agents within its sensing region. Thus, a distributed controller of agent i is such that

$$u_i = \pi_i(z_i) \tag{4}$$

where $z_i = [x_i^T; x_j^T, j \in \mathcal{N}_i]^T$ and $\pi_i : \mathbb{R}^{(|\mathcal{N}_i|+1)n} \to \mathcal{U}$ is the mapping from the state of agent i and its neighbors to the input constraint set. Define $\tilde{u}_i = [u_j^T, j \in \mathcal{N}_i]^T$. Any controller π_i satisfying the constraint $u_i \in \mathcal{U}$ is called an admissible controller. The collision-free condition of agent i, namely the ith collision-free condition, can be defined as

$$h_{ij}(x_i, x_j) \geq 0, \forall j \in \mathcal{N}_i. \tag{5}$$

We aim to design a distributed controller of the form (4) to steer each agent to its target position while avoiding collisions with other agents, with constraints on input and sensing scale. Formally, our problem can be stated as follows.

Problem 1: Consider a team of N agents with dynamics (1), input constraint $u_i \in \mathcal{U}$, and target position $x_{f,i}$, for $i \in \mathcal{N}$. The problem is to find a distributed controller of the form (4) satisfying the following two properties:

- Safety: For any initial joint state satisfying $x(0) \in \mathcal{C}$, all joint states along the system trajectory are in a collision-free configuration, i.e., $x(t) \in \mathcal{C}$, for all $t > 0$.
- Convergence: The position of agent i converges to the target position, i.e., $\lim_{t\to\infty} x_i(t) = x_{f,i}$, for all $i \in \mathcal{N}$.

III. Safe-Reachable Set-Based Approach

A. Safe-reachable set

We start with a formal definition of the safe-reachable set for each agent. In the centralized version of the safe-reachable set, as employed in [18], [19] for solving multiplayer adversarial games, each agent's safe-reachable set contains all positions that the agent can reach while satisfying certain safety constraints. In this study, we introduce a distributed version of the safe-reachable set and apply it to the multi-agent collision avoidance scenario.

Definition 1: Given a team of N agents with dynamics (1) and input constraint $u_i \in \mathcal{U}$ for all $i \in \mathcal{N}$, the ith safe-reachable set (SRS), denoted by Ω_i, consists of all points y

within the ith sensing region $\mathcal{X}_i$ satisfying the following two properties:

1) (Reachability) There exists a time instant τ and an admissible controller π_i over the interval $[0, \tau]$ such that agent i can reach position y at time τ.
2) (Safety) For any admissible controller π_j of agent j, $j \in \mathcal{N}_i$, the ith collision-free condition (5) holds for all $t \in [0, \tau]$.

According to Proposition 1 in [19] or the result on page 144 of [17], we can similarly represent the ith SRS as a sublevel set as follows:

$$\Omega_i(z_i) = \bigcap_{j \in \mathcal{N}_i} \{y \in \mathcal{X}_i \mid c_{ij}(y, x_{ij}) \leq 0\} \tag{6}$$

where $c_{ij}(y, x_{ij}) = (||y - x_i|| + r_{ij})^2 - ||y - x_j||^2$ with $x_{ij} = (x_i, x_j)$. This leads to the following properties of SRS.

Proposition 1: (i) The ith SRS Ω_i is nonempty at z_i if and only if the ith collision-free condition (5) holds at z_i. (ii) The ith nonempty SRS Ω_i is closed and convex.

Proof: (i) If the ith collision-free condition (5) holds, i.e., $\|x_i - x_j\| \geq r_{ij}, \forall j \in \mathcal{N}_i$, then $c_{ij}(x_i, x_{ij}) = r_{ij}^2 - \|x_i - x_j\|^2 \leq 0, \forall j \in \mathcal{N}_i$. Thus, the ith SRS Ω_i is nonempty. Conversely, if the ith SRS Ω_i is nonempty, then $\exists\ y \in \mathcal{X}_i, c_{ij}(y, x_{ij}) \leq 0$, for all $j \in \mathcal{N}_i$, which means $||y - x_i|| + r_{ij} - ||y - x_j|| \leq 0, \forall j \in \mathcal{N}_i$. Then, $\|x_i - x_j\| \geq |||y - x_i\| - \|y - x_j||| \geq r_{ij}$, for all $j \in \mathcal{N}_i$, implying that the ith collision-free condition holds. (ii) Note that $c_{ij}(y, x_{ij})$ can be expanded as $c_{ij}(y) = 2(x_j - x_i)^T y + 2r_{ij}\|y - x_i\| + r_{ij}^2 + x_i^T x_i - x_j^T x_j$, which is a sum of convex functions of the argument y. $c_{ij}(y)$, $g_i(y)$ are both convex function. Therefore, the nonempty ith SRS Ω_i is a closed and convex set from the definition (6). □

The construction of the ith SRS is carried out in a distributed manner, as the safety property is checked only for agents in the neighborhood of agent i. When no agents are located within the sensing region of agent i, the ith SRS reduces to its sensing region, as shown in Fig. 1.

Remark 1: SRS Ω_i is different from the concept of BVC[12]. In our distributed framework, the BVC of agent i can be rewritten as $\bar{\mathcal{V}}_i = \bigcap_{j \in \mathcal{N}_i} \{y \in \mathcal{X}_i \mid (y - \frac{x_i + x_j}{2})^T (x_j - x_i) + \frac{r_{ij}}{2}\|x_j - x_i\| \leq 0\}$ and (6) can be recast in a similar form as $\Omega_i = \bigcap_{j \in \mathcal{N}_i} \{y \in \mathcal{X}_i \mid (y - \frac{x_i + x_j}{2})^T (x_j - x_i) + \frac{r_{ij}^2}{2} + r_{ij}\|y - x_i\| \leq 0\}$. When $r_{ij} = 0$ for all $j \in \mathcal{N}_i$, both $\bar{\mathcal{V}}_i$ and Ω_i reduce to the Voronoi cell. Otherwise, $\bar{\mathcal{V}}_i$ and Ω_i have different expressions. Importantly, the main difference between SRS and BVC is that the property (i) in Proposition 1 holds only for SRS. In other words, a nonempty BVC does not imply that the ith collision-free condition holds, whereas a nonempty SRS does.

B. Distributed controller design

Building upon the concept of SRS, we next derive our distributed controller. Recall that the goal of each agent is to minimize the distance between the agent and its target position while avoiding collisions. According to the definition of SRS, agent i can safely reach any point within Ω_i.

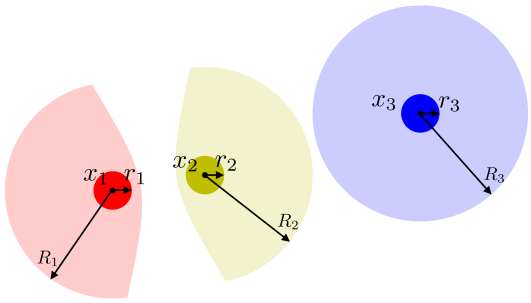

Fig. 1: Example of three agents with their corresponding SRSs. Each agent i is assigned a safety radius r_i and sensing radius R_i, and both the agent and its SRS are plotted in the same color. Agent 3 does not have any neighbors within its sensing region, and thus its SRS at this moment is equivalent to its sensing region. In contrast, for agents 1 and 2, their SRSs are obtained by considering the safety radius of each other, as their neighborhood sets are not empty.

Bearing this in mind, we define the ith objective function as the squared distance between the ith target position and the ith SRS:

$$\Psi_i(z_i) = \min_{y \in \Omega_i(z_i)} ||y - x_{f,i}||^2.$$

Since the ith nonempty SRS Ω_i is convex, according to property (ii) in Proposition 1, the value of Ψ_i can be determined using the following convex program:

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & ||y - x_{f,i}||^2 \\ \text{s.t.} \quad & c_{ij}(y, x_{ij}) \leq 0,\ j \in \mathcal{N}_i \\ & g_i(y) \leq 0. \end{aligned} \tag{7}$$

It is worth noting that the convex program (7) is equivalent to finding the Euclidean projection [20] of a point $x_{f,i}$ onto the ith SRS Ω_i. Moreover, we can demonstrate the uniqueness of the solution for the convex program (7) as below.

Proposition 2: If the ith SRS Ω_i is nonempty at z_i, then the convex program (7) admits a unique solution.

Proof: According to property (ii) of Proposition 1, the ith nonempty SRS Ω_i is both closed and convex. As stated on page 397 of [20], the Euclidean projection of a point on a convex and closed set is unique. Thus, the Euclidean projection of a point $x_{f,i}$ on a convex and closed set Ω_i is unique. This implies that the convex program (7) admits a unique solution. □

After defining the ith objective function, we formulate the decentralized multi-agent collision avoidance problem for agent i as an online robust minimization problem:

$$\begin{aligned} & \min_{u_i} \max_{\tilde{u}_i} J_i = \Psi_i(z_i) \\ & \text{s.t.} \quad \dot{x}_i = u_i,\ \dot{x}_j = u_j, j \in N_i \\ & \qquad u_i \in \mathcal{U},\ \tilde{u}_i = [u_j^T, j \in \mathcal{N}_i]^T \in \mathcal{U}^{|\mathcal{N}_i|}. \end{aligned} \tag{8}$$

Here, the robustness stems from the fact that for agent i, the inputs of its neighbors are unknown and worst-case scenarios should be considered to avoid collisions. The Hamiltonian of (8) is defined as $H(q, u_i, \tilde{u}_i) = p_i^T u_i + \sum_{j \in \mathcal{N}_i} p_j^T u_j$,

where $p_i, p_j, j \in \mathcal{N}_i$ are the costates of the system and $q = [p_i^T; p_j^T, j \in \mathcal{N}_i]^T$ is the Lagrange multiplier vector. According to Theorem 8.2 in [21], the necessary condition for problem (8) is given by

$$p_i^{*T} = \frac{\partial \Psi_i}{\partial x_i}, p_j^{*T} = \frac{\partial \Psi_i}{\partial x_j}, j \in \mathcal{N}_i$$
$$H(u_i^*, \tilde{u}_i) \leq H(u_i^*, \tilde{u}_i^*) \leq H(u_i, \tilde{u}_i^*), \forall u_i \in \mathcal{U}, \forall \tilde{u}_i \in \mathcal{U}^{|\mathcal{N}_i|}$$

where $(u_i^*, \tilde{u}_i^*)$ is the strategy pair in the saddle-point equilibrium. Observe that, the function H comprises two independent parts: $p_i^T u_i$ and $\sum_{j\in\mathcal{N}_i} p_j^T u_j$. Thus, minimizing H with respect to u_i is equivalent to minimizing $p_i^T u_i$. As such, we can express u_i^* as follows:

$$u_i^* = \begin{cases} -\frac{v_{max}}{||p_i||} p_i, & p_i \neq \mathbf{0} \\ \text{Undetermined}, & p_i = \mathbf{0} \end{cases} \tag{9}$$

where "undetermined" means u_i^* can be any value within $\mathcal{U}$. Note that, the value of u_i^* is related to the partial derivative of Ψ_i with respect to x_i. To proceed, we need the following proposition.

Proposition 3: For almost every z_i, there is a set of nonnegative constants λ_{ij}^*, $j \in \mathcal{N}_i$ such that

$$\frac{\partial \Psi_i}{\partial x_i} = \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial x_i}(\eta_i, x_{ij})$$

and $\frac{\partial \Psi_i}{\partial x_j} = \lambda_{ij}^* \frac{\partial c_{ij}}{\partial x_j}(\eta_i, x_{ij})$ for $j \in \mathcal{N}_i$, where η_i is the solution to the convex program (7). Moreover, $c_{ij}(\eta_i, x_{ij}) < 0$ implies that $\frac{\partial \Psi_i}{\partial x_j} = 0$.

Proof: Consider the Lagrangian function of (7):

$$L_i(y, \hat{\lambda}) = ||y - x_{f,i}||^2 + \sum_{j\in\mathcal{N}_i} \lambda_{ij} c_{ij}(y, x_{ij}) + \lambda g_i(y)$$

where $\hat{\lambda} = (\lambda_{ij}, j \in \mathcal{N}_i; \lambda)$. The KKT conditions imply that there is $\hat{\lambda}^*$ with nonnegative components such that

$$2(\eta_i - x_{f,i})^T + 2\lambda^*(\eta_i - x_i)^T + \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial y}(\eta_i, x_{ij}) = 0 \tag{10}$$
$$\lambda_{ij}^* c_{ij}(\eta_i, x_{ij}) = 0, \forall j \in \mathcal{N}_i \tag{11}$$

On the other hand, the complementary slackness (11) implies that the value of λ_{ij}^* and c_{ij} can be divided into two cases:

$$\begin{cases} c_{ij}(\eta_i, x_{ij}) = 0, & \lambda_{ij}^* > 0 \\ c_{ij}(\eta_i, x_{ij}) \leq 0, & \lambda_{ij}^* = 0. \end{cases}$$

For the case of $\lambda_{ij}^* = 0$, $\lambda_{ij}^* \dot{c}_{ij}(\eta_i, x_{ij}) = 0$ is satisfied. For the case of $\lambda_{ij}^* > 0$, $c_{ij}(\eta_i, x_{ij}) = 0$, and $\dot{c}_{ij}(\eta_i, x_{ij}) = 0$ or $\dot{c}_{ij}(\eta_i, x_{ij}) \neq 0$. Note that, the case of $c_{ij}(\eta_i, x_{ij}) = 0$ and $\dot{c}_{ij}(\eta_i, x_{ij}) \neq 0$ has measure zero, as stated in real analysis (see, for example, [22]). Thus, for almost every z_i, $\lambda_{ij}^* \dot{c}_{ij}(\eta_i, x_{ij}) = 0$, i.e.,

$$\lambda_{ij}^* \left(\frac{\partial c_{ij}}{\partial y}(\eta_i, x_{ij})\dot{\eta}_i + \frac{\partial c_{ij}}{\partial x_j}(\eta_i, x_{ij})u_j + \frac{\partial c_{ij}}{\partial x_i}(\eta_i, x_{ij})u_i\right) = 0. \tag{12}$$

Similarly, for almost every z_i, we obtain $\lambda^* \dot{g}_i(\eta_i) = 0$, i.e.,

$$\lambda^* \dot{g}_i(\eta_i) = 2\lambda^*(\eta_i - x_i)^T \dot{\eta}_i = 0. \tag{13}$$

Combining equations (10), (12), and (13), we obtain the derivative of Ψ_i as:

$$\begin{aligned} \dot{\Psi}_i &= 2(\eta_i - x_{f,i})^T \dot{\eta}_i \\ &= -2\lambda^*(\eta_i - x_i)^T \dot{\eta}_i - \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial y}(\eta_i, x_{ij})\dot{\eta}_i \\ &= -\sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial y}(\eta_i, x_{ij})\dot{\eta}_i \\ &= \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial x_i}(\eta_i, x_{ij})u_i + \sum_{j\in\mathcal{N}_i} \lambda_{ij}^* \frac{\partial c_{ij}}{\partial x_j}(\eta_i, x_{ij})u_j. \end{aligned}$$

for almost every z_i, which completes the proof by examining $\dot{\Psi}_i = \frac{\partial \Psi_i}{\partial x_i} u_i + \sum_{j\in\mathcal{N}_i} \frac{\partial \Psi_i}{\partial x_j} u_j$. □

We are now ready to derive our distributed controller as follows. Note that when $\eta_i = x_i$, $u_i^* = 0$. Conversely, when $\eta_i \neq x_i$, $\frac{\partial c_{ij}}{\partial x_i}$ is given by

$$\frac{\partial c_{ij}(\eta_i)}{\partial x_i} = 2(x_i - \eta_i)^T (1 + \frac{r_{ij}}{\|\eta_i - x_i\|}).$$

According to (9), u_i is undetermined when $p_i = \mathbf{0}$, thus we only need to consider the case of $p_i \neq \mathbf{0}$. The control input that minimizes $\dot{\Psi}_i$ is given by

$$\begin{aligned} u_i^* &= -\frac{v_{max}}{||p_i||} p_i = \frac{-\frac{\partial \Psi_i}{\partial x_i}^T}{\|\frac{\partial \Psi_i}{\partial x_i}^T\|} v_{max} \\ &= -\frac{(x_i - \eta_i) * 2\sum_{j\in\mathcal{N}_i} \lambda_{ij}^* (1 + \frac{r_{ij}}{\|\eta_i - x_i\|})}{\|(x_i - \eta_i) * 2\sum_{j\in\mathcal{N}_i} \lambda_{ij}^* (1 + \frac{r_{ij}}{\|\eta_i - x_i\|})\|} v_{max} \\ &= \frac{\eta_i - x_i}{\|\eta_i - x_i\|} v_{max}. \end{aligned} \tag{14}$$

Since u_i is undetermined when $p_i = \mathbf{0}$ and $\eta_i \neq x_i$, we choose the form of (14) for the control input u_i. Consequently, taking into account all the possibilities of u_i^*, we arrive at a distributed controller given by

$$\pi_i(z_i) = v_{max} \mathscr{N}(\eta_i - x_i) \tag{15}$$

where η_i is the unique solution to the convex program (7) and the normalizer of a vector v is defined as

$$\mathscr{N}(v) = \begin{cases} \frac{v}{\|v\|}, & v \neq \mathbf{0} \\ \mathbf{0}, & v = \mathbf{0}. \end{cases}$$

The controller (15) is distributed because the computation of η_i only requires the information of agent i and its neighbors.

C. Safety

We proceed to analyze the collision avoidance capability of our controller (15). To this end, let us recall the definition of set invariance. Consider a dynamical system of the form

$$\dot{e}(t) = f(e(t)) \tag{16}$$

accosiated with a state constraint set $\mathcal{I} = \{e \in \mathbb{R}^n \mid h(e) \geq 0\}$ and its boundary $\partial\mathcal{I} = \{e \in \mathbb{R}^n \mid h(e) = 0\}$. The set

$\mathcal{I}$ is said to be invariant [23] for system (16) if $e(0) \in \mathcal{C}$ implies $e(t) \in \mathcal{C}$ for all $t > 0$. The necessary and sufficient condition for set invariance is given by Nagumo's Theorem as follows.

Lemma 1 (Nagumo, [24]): The set $\mathcal{I}$ is invariant for the system (16) if and only if $\dot{h}(e) \geq 0,\ \forall e \in \partial\mathcal{I}$.

We are now prepared to present the main result for collision avoidance, which demonstrates that our controller can ensure collision-free motion for all agents, provided that each agent initilizes in a collision-free configuration.

Theorem 1: If the team of N agents is initially in a collision-free configuration, i.e., $x(0) \in \mathcal{C}$, and the sensing radius R_i satisfies $R_i \geq r_{ij}$, for all $i \neq j, i, j \in \mathcal{N}$, the distributed controller (15) ensures that all the future positions of the agents are collision-free, i.e., $x(t) \in \mathcal{C}$, $\forall t > 0$.

Proof: For agent i and the agents beyond its sensing region, we have

$$\|x_i - x_j\| > R_i \geq r_{ij}, \forall j \notin \mathcal{N}_i, j \in \mathcal{N}. \tag{17}$$

Thus, for each agent, we only need to consider the agents within its sensing area to avoid collisions. This is equivalent to showing that the $\mathcal{C}$ is invariant, with the definition of the collision-free set $\mathcal{C}$ given in (3). According to Lemma 1, to prove that the set $\mathcal{C}$ is invariant under the system (1) with the distributed controller (15), we need to establish that $\dot{h}_{ij} \geq 0$ for all $i \in \mathcal{N}$ and $j \in \mathcal{N}_i$ when $h_{ij} = 0$.

Taking the derivative of h_{ij}, we get

$$\begin{aligned}\dot{h}_{ij} &= 2(x_i - x_j)^T(u_i - u_j) \\ &= 2(x_i - x_j)^T(\mathcal{N}(\eta_i - x_i) - \mathcal{N}(\eta_j - x_j))v_{max}\end{aligned} \tag{18}$$

where $\eta_i \in \Omega_i$ and $\eta_j \in \Omega_j$, implying that the following inequalities are satisfied

$$\begin{aligned}(||\eta_i - x_i|| + r_{ij})^2 - ||\eta_i - x_j||^2 \leq 0 \\ (||\eta_j - x_j|| + r_{ij})^2 - ||\eta_j - x_i||^2 \leq 0.\end{aligned} \tag{19}$$

Note that $h_{ij} = 0$ implies that

$$r_{ij} = \|x_i - x_j\|, \forall i \neq j, i, j \in \mathcal{N}. \tag{20}$$

Substituting (20) into (19), we obtain

$$\begin{aligned}(x_i - x_j)^T(\eta_i - x_i) \geq r_{ij}\|\eta_i - x_i\| \\ -(x_i - x_j)^T(\eta_j - x_j) \geq r_{ij}\|\eta_j - x_j\|\end{aligned}$$

which can be rewritten as

$$\begin{aligned}(x_i - x_j)^T\mathcal{N}(\eta_i - x_i) \geq 0 \\ -(x_i - x_j)^T\mathcal{N}(\eta_j - x_j) \geq 0.\end{aligned} \tag{21}$$

Therefore, substituting (21) into (18) yields $\dot{h}_{ij} \geq 0$. □

Remark 2: Our method has advantages over existing methods such as CBF and ORCA in terms of recursively ensuring collision avoidance. For a multi-agent system with single integrators, our method ensures safety once the initial positions meet the collision-free condition, as stated in Theorem 1. In contrast, the CBF and ORCA methods require the corresponding optimization problem to remain feasible all the time to guarantee safety, even if the initial collision-free condition is satisfied.

D. Convergence

Finally, we show that our controller can guarantee convergence under certain conditions. One of the primary challenges in ensuring convergence is the potential occurrence of deadlocks [25], where agents become trapped in a state that prevents them from moving toward their target positions. Among various definitions of deadlocks, temporary deadlock refers to a situation in which agents become momentarily stuck during the convergence process [11]. To better present our result, we provide a formal definition of temporary deadlock below.

Definition 2: Given an admissible control law, agent i is said to be in a temporary deadlock configuration at time $t_0 \geq 0$ if $x_i(t_0) \neq x_{f,i}$ and $u_i(t_0) = \mathbf{0}$.

We are now ready to present the convergence result for our distributed controller under the assumption of no temporary deadlock.

Theorem 2: If no temporary deadlock occurs during the whole process, i.e., $u_i(t) \neq 0$ when $x_i(t) \neq x_{f,i}$, and both the initial positions and target positions are in a collision-free configuration, then the distributed controller (15) ensures that the position of each agent converges to its target position.

Proof: Consider the Lyapunov function candidate $V_i(x_i) = \|x_i - x_{f,i}\|^2$ for the ith subsystem of system (1) with the controller (15). The derivative of $V(x_i)$ along the trajectories of the system is given by

$$\begin{aligned}\dot{V}(x_i) &= 2(x_i - x_{f,i})^T u_i \\ &= 2v_{max}(x_i - x_{f,i})^T\mathcal{N}(\eta_i - x_i) \\ &= \begin{cases} -\frac{2(x_{f,i} - x_i)^T(\eta_i - x_i)}{\|\eta_i - x_i\|}v_{max}, & \eta_i \neq x_i \\ 0, & \eta_i = x_i \end{cases}\end{aligned}$$

where η_i is the solution of (7), i.e., $\|\eta_i - x_{f,i}\| = \min_{y \in \Omega_i(z_i)} \|y - x_{f,i}\|$. As shown in Theorem 1, the ith collision-free condition holds at z_i, which implies that $c_{ij}(x_i, x_{ij}) \leq 0, \forall j \in \mathcal{N}_i$, and thus $x_i \in \Omega_i(z_i)$. As a result, the following inequality holds

$$\|\eta_i - x_{f,i}\| \leq \|x_i - x_{f,i}\| \tag{22}$$

where the equality holds if and only if $\eta_i = x_i$. Under the no temporary deadlock condition, $\eta_i \neq x_i$ when $x_i \neq x_{f,i}$. Hence, the inequality (22) strictly holds, which implies that the time derivative of V satisfies

$$\begin{aligned}\dot{V}(x) &= -\frac{2(x_{f,i} - x_i)^T(\eta_i - x_i)}{\|\eta_i - x_i\|}v_{max} \\ &\leq -\frac{2(x_{f,i} - x_i)^T(\eta_i - x_i) - \|\eta_i - x_i\|^2}{\|\eta_i - x_i\|}v_{max} \\ &= \frac{2x_{f,i}^T x_i - 2x_{f,i}^T\eta_i - x_i^T x_i - \eta_i^T\eta_i}{\|\eta_i - x_i\|}v_{max} \\ &= \frac{\|\eta_i - x_{f,i}\|^2 - \|x_i - x_{f,i}\|^2}{\|\eta_i - x_i\|}v_{max} < 0\end{aligned}$$

where the last strict inequality holds as $\eta_i \neq x_i$, which is guaranteed by the no temporary deadlock condition. Therefore, in light of Lyapunov's stability theorem [26], we can conclude that x_i converges to $x_{f,i}$. □

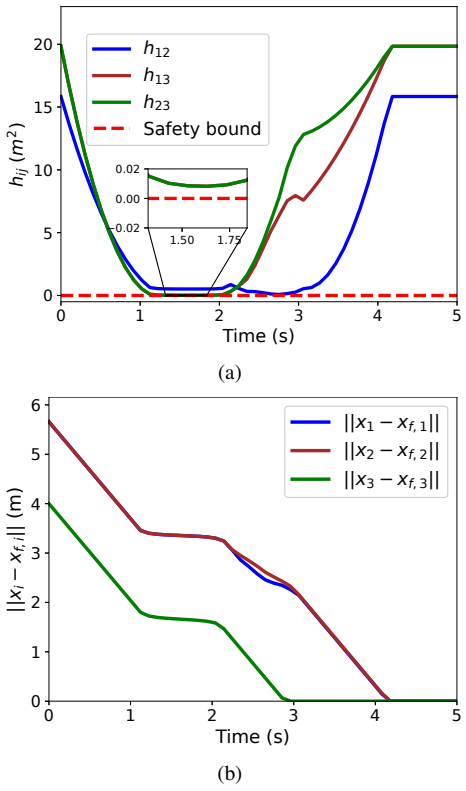

Fig. 2: Performance of our SRS-based controller in the simulation. (a) Safety. (b) Convergence.

Remark 3: Theorem 2 guarantees convergence under the no temporary deadlock condition, which is not provided in existing decentralized methods. Most multi-agent collision avoidance approaches rely on heuristic methods to handle deadlock situations, often analyzing specific case studies without ensuring convergence in general. It is worth noting that, although the no temporary deadlock condition is conservative for convergence guarantees, there exist examples where our controllers can achieve convergence despite the occurrence of temporary deadlocks.

IV. Simulation Results

In this section, we simulate N single integrator robots to reach their target positions and compare the performance of our SRS-based distributed controller (15) to the CBF-based controller [11] and the BVC-based controller [12]. Our simulations are carried out on a laptop equipped with a 2.3 GHz 14-Core Intel Core $i7$ processor with 24 GB of memory. The convex program in (7) is solved using the CVXPY solver [27]. In our simulations, we employ circular agents with identical safety radii of $r_i = 0.2\text{m}$ and maximum speeds of $v_{max} = 2\text{m/s}$. We adopt a time step of 0.1 seconds to ensure accurate and reliable simulation results.

A. SRS-based controller versus CBF-based controller

Consider the case where there are $N = 3$ agents and they move toward their target positions, as shown in Fig. 3(a). The initial positions of the agents are $x_1(0) = [-2, -2]^T$, $x_2(0) = [-2, 2]^T$ and $x_3(0) = [2, 0]^T$, and their target

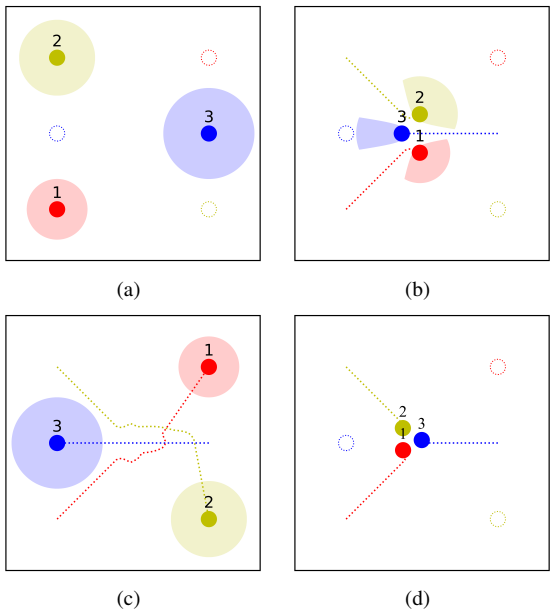

Fig. 3: Simulation of three single integrator robots using different controllers with distinct colors. The light-colored regions denote the respective SRSs associated with agents at various time instants. (a) At initial time. (b) At time 2.2s (SRS). (c) At time 4.2s (SRS). (d) At time 2.4s (CBF).

positions are $x_{f,1} = [2, 2]^T$, $x_{f,2} = [2, -2]^T$ and $x_{f,3} = [-2, 0]^T$. The sensing radii are $R_1 = 0.8\text{m}$, $R_2 = 1.0\text{m}$ and $R_3 = 1.2\text{m}$. The safety performance under our distributed controller is shown in Fig. 2(a), where the value of h_{ij} remains larger than zero throughout the simulation, meaning that the safety condition $x(t) \in \mathcal{C}$ holds for all $t \geq 0$. In addition, the value of $\|x_i - x_{f,i}\|$ converges to zeros for $i \in \{1, 2, 3\}$, as shown in Fig. 2(b), indicating the arrivals of the agents at their target positions. The collision-free trajectories of the three agents with the proposed SRS-based distributed controller (15) are depicted in Fig. 3(b) and Fig. 3(c). Fig. 3(c) shows that each agent successfully arrives at its target position with our SRS-based approach, while the agents with the CBF-based approach get stuck in a deadlock configuration as shown in Fig. 3(d).

B. SRS-based controller versus BVC-based controller

We also conduct a comparison between the SRS-based controller and the BVC-based controller in a scenario where N agents are positioned around two edges of a rectangle and randomly move towards positions near the opposite edge. The makespan, defined as the time elapsed from the start of the process to the end, is used to compare the two methods, as shown in Table I. Specifically, we perform 10 simulations for different agent sizes and compute the makespan for each approach. Both methods guarantee safety, but our SRS approach achieves an average improvement of 16.64% (size 10), 17.79% (size 20), and 15.58% (size 30) in the makespan. Fig. 4 illustrates a ten-agent scenario with the SRS-based controller and shows the velocity profile along the trajectory.

TABLE I: Average makespan comparison

Size	SRS	BVC	Improvement(%)
10	5.53s	6.45s	**16.64**
20	9.50s	11.20s	**17.79**
30	13.93s	16.10s	**15.58**

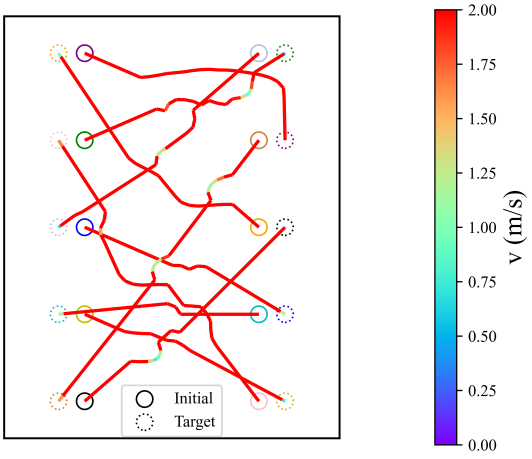

Fig. 4: Velocity profile with the SRS-based controller. Each agent moves from the initial position (the solid hollow circle) to the target position (the dotted hollow circle).

V. CONCLUSIONS

In this paper, we have proposed a geometric-based method based on safe-reachable sets to address the decentralized multi-agent collision avoidance problem. Specifically, we introduce a distributed version of the safe-reachable set concept for multi-agent collision avoidance scenarios. Utilizing the safe-reachable sets, we design our distributed controller by online solving a convex program at each time instant. Moreover, our method guarantees collision avoidance for single integrators with identical maximum speeds and guarantees convergence under the no temporary deadlock condition. Through simulations conducted in different scenarios, we demonstrate the effectiveness of our approach in terms of safety, convergence, and efficiency. In future work, we will extend our approach to handle general heterogeneous single integrator robots.

REFERENCES

[1] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.

[2] S. Dergachev and K. Yakovlev, "Distributed multi-agent navigation based on reciprocal collision avoidance and locally confined multi-agent path finding," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1489–1494, 2021.

[3] T. Chu, S. Qu, and J. Wang, "Large-scale multi-agent reinforcement learning using image-based state representation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 7592–7597, 2016.

[4] M. Chen, J. C. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1695–1700, 2016.

[5] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4229–4235, 2017.

[6] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935, 2008.

[7] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*, pp. 3–19, Springer, 2011.

[8] S. H. Arul and D. Manocha, "V-rvo: Decentralized multi-agent collision avoidance using voronoi diagrams and reciprocal velocity obstacles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8097–8104, IEEE, 2021.

[9] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 404–420, 2018.

[10] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

[11] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[12] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

[13] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[14] H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 4835–4840, 2011.

[15] M. Abdullhak and A. Vardy, "Deadlock prediction and recovery for distributed collision avoidance with buffered voronoi cells," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 429–436, 2021.

[16] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Weighted buffered voronoi cells for distributed semi-cooperative behavior," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5611–5617, 2020.

[17] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[18] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin, "Cooperative pursuit with voronoi partitions," *Automatica*, vol. 72, pp. 64–72, 2016.

[19] J. Liu, Z. Ouyang, J. Yang, H. Chen, H. Lu, and W. Zhang, "Coordinated defense allocation in reach-avoid scenarios with efficient online optimization," *submitted to arxiv*.

[20] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[21] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[22] H. L. Royden and P. Fitzpatrick, *Real analysis*, vol. 32. Macmillan New York, 1988.

[23] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[24] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, IEEE, 2019.

[25] D. Cappello, S. Garcin, Z. Mao, M. Sassano, A. Paranjape, and T. Mylvaganam, "A hybrid controller for multi-agent collision avoidance via a differential game formulation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1750–1757, 2020.

[26] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel.

[27] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.