

Distributed Encirclement and Capture of Multiple Pursuers With Collision Avoidance

Xinyi Wang , Graduate Student Member, IEEE, Lele Xi , Yulong Ding ,
and Ben M. Chen , Fellow, IEEE

I. INTRODUCTION

Abstract—In this article, we propose a distributed algorithm for cooperatively pursuing an adversarial evader in an unbounded environment with cluttered obstacles. The algorithm relies on constructing the buffered evader-centered bounded Voronoi cell (B-ECBVC) in real time for each pursuer to safely chase the evader among obstacles. Based on the B-ECBVC, an encirclement control law and a capture strategy are proposed. Specifically, the control law drives each pursuer toward the centroid of its B-ECBVC to trap the evader, while the capture strategy guides it to reduce the distance between a team of pursuers and the evader by adaptively compressing the B-ECBVC. By integrating the control law and the capture strategy, the pursuers can rapidly approach the evader, while simultaneously maintaining the encirclement. To guarantee collision avoidance, a rapid and reliable approach for creating secure regions of pursuers by integrating separating hyperplanes and buffered terms into B-ECBVCs. In addition, the proposed pursuit method is further extended to a higher order dynamics system with avoiding moving obstacles. Our B-ECBVC approach is validated with various escape policies of the evader in dense obstacle environments. Moreover, real-time experiments with an autonomous evader and a human evader are implemented in a multiple mobile robot platform to validate the effectiveness of our approach.

Index Terms—Distributed planning, multiagent systems, pursuit–evasion, Voronoi diagram.

Manuscript received 10 November 2022; revised 30 March 2023 and 21 June 2023; accepted 25 July 2023. Date of publication 31 August 2023; date of current version 29 February 2024. The work was supported in part by the Research Grants Council of Hong Kong SAR under Grant 14206821 and Grant 14217922, in part by the Hong Kong Centre for Logistics Robotics, in part by the National Natural Science Foundation of China under Grant 62088101, in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities. (Corresponding author: Yulong Ding.)

Xinyi Wang and Ben M. Chen are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China (e-mail: xywangmae@link.cuhk.edu.hk; bmchen@cuhk.edu.hk).

Lele Xi is with the Hebei University of Science and Technology, Shijiazhuang 051432, China (e-mail: maa@hebust.edu.cn).

Yulong Ding is with the Department of Control Science and Engineering, Tongji University, Shanghai 200092, China, and also with the Frontiers Science Center for Intelligent Autonomous Systems, Ministry of Education, Shanghai 200092, China (e-mail: dingyulong@tongji.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIE.2023.3301511>.

Digital Object Identifier 10.1109/TIE.2023.3301511

MULTIROBOT pursuit–evasion (MPE) game draws considerable research attention recently with many real-world applications, including area surveillance [1], target tracking [2], [3], [4], and wildlife monitoring [5]. In this kind game, a team of pursuers applies a cooperative control strategy to capture an evader, while the evader attempts to escape simultaneously. As a typical application scenario of MPE game, multiple robots collaboratively capture an evader in an unbounded environment with obstacles. One of the main challenges is the cooperation among robots to limit the adversarial evader movements and capture them safely.

One classic formulation of the MPE problem is based on a noncooperative differential game that utilizes Hamilton-Jacobi-Issacs (HJI) partial differential equation [6]. However, it is difficult to generate strategies when only the initial configurations are known, and has huge computational overhead in the case of large-scale players [7]. Another formulation of the MPE problem is based on intelligent methods. Selvakuma and Bakolas [8] employed min–max Q-learning to subsequently obtain the optimal pursuit action for the evader at each stage. Fu et al. [9] proposed a UAV pursuit–evasion strategy based on reinforcement learning and imitation learning to flexibly adjust the speed and attitude to pursue the evader. This kind of method may occur the curse of dimensionality problem [10] and need high-frequency interaction among pursuers for coordination [11], which may not be suitable in real-world pursuit scenarios with limited communication resources.

To obtain fast and robust pursuit strategies, recent studies use the Voronoi diagram to solve MPE problems in a distributed fashion. Compared with the abovementioned methods, it is scalable for a vast number of robots, which can be executed with less information sharing only according to relative positions of neighboring pursuers [12]. Pierson et al. [7] proposed an area-minimization (AM) pursuit strategy to chase an evader in a bounded environment. The pursuers moved to the midpoint of the Voronoi boundary shared with the evader to shrink its safe-reachable area. Tian et al. [13] introduced a capturing strategy based on AM and an obstacle-aware buffered Voronoi cell (OAVC) to ensure collision and obstacle avoidance during MPE. Another type of Voronoi-based method is based on weighted Voronoi tessellation that assigns a probability density for the location of the evader and then pursuers are guided to approach the evader by a move-to-centroid strategy. Pierson

et al. [14] modified the weighted Voronoi cell into OAVC to avoid collisions when pursuing the evader.

The existing voronoi-based studies mainly focuses on how to pursue the evader in a bounded environment. The evader often needs to be forced into a corner before it can be captured. However, in reality, many pursuit–evasion problems occur in unbounded environments such as the forest and urban city, making it easier for the evader to escape. Encirclement is crucial especially when confronted with a more intelligent evader.

In recent years, several works [15], [16], [17], [18] attempt to encircle the adversarial evader within the capture domain by strengthening teamwork and collaboration among the pursuers. Wang et al. [17] proposed an encirclement guaranteed partitioning method to separate the pursuers in four different quadrants of the frame centered at the evader. They primarily decouple the encirclement and capture problems, making pursuers unable to flexibly adjust encirclement and capture strategies. Fang et al. [18] introduced a distributed pursuit algorithm that can balance between surrounding and hunting (SH) the evader. However, its success rate of capturing depends on the strict initial spatial conditions of all players. In reality, it is rarely seen that desired spatial distribution can be precisely satisfied at the initial state.

The previous approaches have limitations in achieving effective and safe capture of the evader in unbounded environments with cluttered obstacles. To address these issues, this article proposes a distributed pursuit algorithm based on a buffered evader-centered bounded Voronoi cell (B-ECBVC), where collision-free trajectories are generated for cooperatively forming an encirclement and capturing the evader. The main contributions are summarized as follows.

- 1) A move-to-centroid control law is designed based on the B-ECBVCs, which guarantees that the evader can be encircled in a convex hull formed by a team of pursuers for random initial configurations. A cooperative capture strategy is developed, in which each pursuer adaptively compresses the boundaries of B-ECBVCs to rapidly approach evader. The capture strategy, combined with the designed control law, can effectively restrict the motions of the evader and simultaneously reduce the capture distance.
- 2) A fast and reliable approach is proposed to generate safe regions of pursuers by integrating separating hyperplanes and buffered terms into B-ECBVCs. Collision avoidance between robots and dense obstacles or among robots can be guaranteed in real time.
- 3) Simulations are carried out to show that compared with [7], [14], [18], our method can achieve efficient encirclement capture performance, and compared with [11], have a high capture success rate in obstacle environments. Moreover, real-time experiments with autonomous evader and human evader are implemented to verify that for random initial configurations, our method encourages the effective pursuit of various escape policies.

The rest of this article is organized as follows. Section II defines the formulation of MPE problem. Section III proposes ECBVC-based encirclement and capture strategy. Section IV

presents distributed pursuit strategy with collision avoidance. Section V gives the extension pursuit method with higher order dynamics. Section VI provides the simulation and comparison results. Section VII shows experimental results from hardware implementation with ground mobile robots. Finally, Section VIII concludes this article.

II. PROBLEM FORMULATION

Consider an MPE problem in an unbounded environment $\mathcal{W} \subseteq \mathbb{R}^2$, involving a team of n pursuers localized at $\mathcal{P} = [p_1, \dots, p_n]^T$ at time t , one evader, and m static obstacles. Denote O_o as obstacle o and $\mathcal{O} := \{O_1, \dots, O_m\}$ as a convex set of obstacles, where $o \in \mathcal{I}_o = \{1, \dots, m\}$. The goal of the pursuers is to improve the advantages of teamwork to trap the evader and capture it by having at least one of the pursuers come within capture distance r_c without collisions. The geometry of pursuer i and evader can usually be modeled as compact convex sets \mathcal{R}_i with centered position $p_e = (x_e, y_e)$ and $p_i = (x_i, y_i)$, respectively, where $i \in \mathcal{I} = \{1, \dots, n\}$. Let $\|\cdot\|$ denote the standard Euclidean norm. In the MPE problem, generally, the robot can be described as an integrator dynamic model [7], [13], [19]

$$\begin{aligned} \dot{p}_e &= v_e, \quad \|v_e\| \leq v_{e,\max} \\ \dot{p}_i &= v_i, \quad \|v_i\| \leq v_{p,\max} \quad \forall i \in \mathcal{I} \end{aligned} \quad (1)$$

where v_e and v_i are velocities of evader and pursuers subjected to their maximum speed $v_{e,\max}$ and $v_{p,\max}$, respectively.

Assuming that the escape policy of the evader is unavailable, a team of pursuers needs to cooperatively form a convex hull to restrict its motion. A convex hull formed by n_c pursuers is denoted as

$$\Omega = \left\{ p \in \mathcal{W} \mid p = \sum_{i=1}^{n_c} \lambda_i p_i, \lambda_i \geq 0, \sum_{i=1}^{n_c} \lambda_i = 1 \right\}.$$

Since an encirclement guaranteed partition requires at least four pursuers according to [17], we set $4 \leq n_c \leq n$. The encirclement distance d_e at time t is then defined as

$$d_e = \begin{cases} -\min_{p \in \Omega} \|p - p_e\|, & \text{if } p_e \in \Omega \\ \min_{p \in \Omega} \|p - p_e\|, & \text{otherwise.} \end{cases} \quad (2)$$

Definition 1 (Encirclement condition [17]): The evader is said to be encircled by the pursuers, if $p_e \in \Omega$ or $d_e \leq 0$ with $t \geq 0$.

Once satisfying the encirclement condition, the region where the evader is allowed us to freely move may still remain large. Hence, at least one of the pursuers needs to move close enough to the evader to capture it. The minimum distance between a team of pursuers and an evader is defined as the capture distance, that is

$$d_c = \min_{i \in \mathcal{I}} d(\mathcal{R}_i, \mathcal{R}_e) \quad (3)$$

where $d(\mathcal{R}_i, \mathcal{R}_e) := \inf\{\|q_i - q_e\| \mid q_i \in \mathcal{R}_i, q_e \in \mathcal{R}_e\}$.

Definition 2 (Capture condition [7]): The evader is said to be captured by the pursuers, if the capture distance is smaller than the capture radius r_c , i.e., $d_c \leq r_c$ with $t > 0$.

The time at which the encirclement condition and the capture condition hold can be defined as the encirclement time t_e and the capture time t_c , respectively. Moreover, in the process of pursuing the evader, pursuers are required to avoid collisions with others and obstacles.

Definition 3 (Collision-free Configuration [13]): A collision avoidance condition can be denoted as $d_{io} = d(\mathcal{R}_i, \mathcal{O}_o) > r_i$ and $d_{ij} = d(\mathcal{R}_i, \mathcal{R}_j) > r_i + r_j$, where r_i and r_j are safety radius for pursuer i and j , respectively.

Based on the above, the MPE problem in this article can be stated in a precise way as follows.

Problem 1 (MPE): Given random initial configuration $\mathcal{P}(0) \in \mathcal{W}$ with $d_c(0) > r_c$, find a cooperative control law v_i for each pursuer i such that $d_e \leq 0$, $d_c \leq r_c$ for some $t_c < \infty$ and $d_{io} \geq r_i$, $d_{ij} \geq r_i + r_j$ for all time in an unbounded environment.

III. ENCIRCLEMENT AND CAPTURE VIA ECBVC

A. Evader-Centered Bounded Voronoi Cell

The generalized Voronoi partition can be regarded as the intersection of a set of maximum margin separating hyperplanes [20], i.e.,

$$\begin{aligned} \mathcal{V}_i &= \{p \in \mathcal{W} \mid \|p - p_i^*\| \leq \min\{\|p - p_i^*\|, \|p - p_j^*\|\}\} \\ (p_i^*, p_j^*, p_e^*) &= \arg \min_{p_i \in \mathcal{R}_i, p_j \in \mathcal{R}_j, p_e \in \mathcal{R}_e} \{d(p_i, p_j), d(p_i, p_e)\} \\ \mathcal{V}_e &= \{p \in \mathcal{W} \mid \|p - p_e^*\| \leq \|p - p_i^*\| \\ (p_e^*, p_i^*) &= \arg \min_{p_e \in \mathcal{R}_e, p_i \in \mathcal{R}_i} \{d(p_e, p_i)\}. \end{aligned} \quad (4)$$

Let \mathbf{p}_i and \mathbf{p}_e denote the vectors from the origin to p_i and p_e , respectively. Therefore, the generalized Voronoi partition in (4) can be simplified into the standard Voronoi partition, $\mathcal{V}(\mathcal{P}) = \{\mathcal{V}_e, \mathcal{V}_1, \dots, \mathcal{V}_n\}$ generated by positions of all robots \mathcal{P} in a form of the linear separator

$$\begin{aligned} \mathcal{V}_e &= \left\{ \mathbf{p} \in \mathcal{W} \mid \mathbf{a}_{ei}^T \mathbf{p} \leq b_{ei} \right\} \\ \mathcal{V}_i &= \left\{ \mathbf{p} \in \mathcal{W} \mid \mathbf{a}_{ij}^T \mathbf{p} \leq b_{ij} \quad \forall j \neq i, i, j \in \mathcal{I} \right. \\ &\quad \left. \text{and } \mathbf{a}_{ie}^T \mathbf{p} \leq b_{ie} \right\} \end{aligned} \quad (5)$$

where $\mathbf{a}_{\kappa_1 \kappa_2}$, $b_{\kappa_1 \kappa_2}$ with $\kappa_1, \kappa_2 \in \{i, j, e\}$ and $\kappa_1 \neq \kappa_2$ are the Voronoi linear separator parameters

$$\begin{aligned} \mathbf{a}_{\kappa_1 \kappa_2} &= \mathbf{p}_{\kappa_1 \kappa_2} = \mathbf{p}_{\kappa_1} - \mathbf{p}_{\kappa_2} \\ b_{\kappa_1 \kappa_2} &= \mathbf{p}_{\kappa_1 \kappa_2}^T \frac{\mathbf{p}_{\kappa_1} + \mathbf{p}_{\kappa_2}}{2}. \end{aligned} \quad (6)$$

To guide pursuers to encircle and capture the evader, we introduce the ECBVC for each pursuer, which is the intersection of hyperplanes formed by neighboring robots within the limit of a bounded rectangular region. The region denoted as \mathcal{E} , is centered at the position of the evader with the lower bound s_l and upper bound s_u , respectively, as shown in Fig. 1.

The definition of \mathcal{E} can be given as follows:

$$\mathcal{E} = \{ \mathbf{p} \in \mathcal{W} \mid s_l \leq \mathbf{p} \leq s_u \} \quad (7)$$

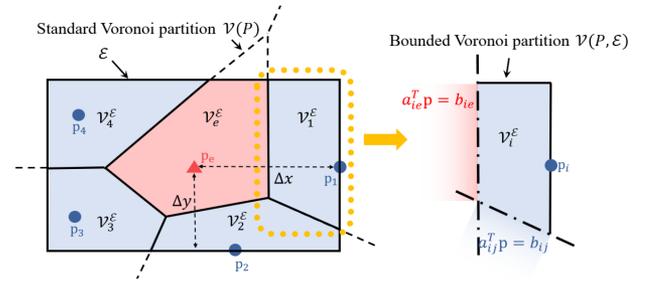


Fig. 1. Evader-centered bounded Voronoi cell.

where

$$\begin{aligned} s_l &= \mathbf{p}_e - [\Delta x \quad \Delta y]^T \\ s_u &= \mathbf{p}_e + [\Delta x \quad \Delta y]^T. \end{aligned} \quad (8)$$

The Δx and Δy represent the half length of the boundaries along the x -axis and y -axis, respectively. Combined with the standard Voronoi partition, we now give the formal definition of ECBVC as follows.

Definition 4 (Evader-centered bounded Voronoi cell): Given a team of pursuers and an evader with a set of positions \mathcal{P} and the bounded rectangular region \mathcal{E} , the evader-centered bounded Voronoi partition $\mathcal{V}(\mathcal{P}, \mathcal{E}) = \{\mathcal{V}_e^E, \mathcal{V}_1^E, \dots, \mathcal{V}_n^E\}$ can be defined as convex polygon regions

$$\begin{aligned} \mathcal{V}_e^E &= \left\{ \mathbf{p} \in \mathcal{W} \mid \mathbf{a}_{ei}^T \mathbf{p} \leq b_{ei} \quad \forall i \in \mathcal{I} \right. \\ &\quad \left. \text{and } s_l \leq \mathbf{p} \leq s_u \right\} \\ \mathcal{V}_i^E &= \left\{ \mathbf{p} \in \mathcal{W} \mid \mathbf{a}_{ij}^T \mathbf{p} \leq b_{ij} \quad \forall j \neq i, i, j \in \mathcal{I} \right. \\ &\quad \left. \mathbf{a}_{ie}^T \mathbf{p} \leq b_{ie} \right. \\ &\quad \left. \text{and } s_l \leq \mathbf{p} \leq s_u \right\}. \end{aligned} \quad (9)$$

B. ECBVC-Based Encirclement

The aim of encirclement is to block the evader's movements in an unbounded environment. In this section, a cooperative encirclement strategy v_i based on the ECBVC \mathcal{V}_i^E is designed for each pursuer i to scatter around the evader and keep it inside the convex hull.

To drive the pursuers around the evader, we need to give boundary limits of their Voronoi cells with considering the localization of the evader. Let the maximum distance between a team of pursuers and evader as Δx and Δy along the x -axis and y -axis, respectively

$$\begin{aligned} \Delta x &= \max_{i \in \mathcal{I}} \|x_i - x_e\| \\ \Delta y &= \max_{j \in \mathcal{I}} \|y_j - y_e\|. \end{aligned} \quad (10)$$

Therefore, the parameter of upper bound s_u and lower bound s_l of \mathcal{E} can be given using (10) during $0 \leq t \leq t_e$.

The performance in scattering around the evader by a team of pursuers p_1, \dots, p_n is assessed using an energy function

$\mathcal{H}(\mathcal{E}, \mathcal{P})$ [21]:

$$\mathcal{H}(\mathcal{E}, \mathcal{P}) = \sum_{i=1}^n \mathcal{H}_i(\mathcal{E}, \mathcal{P}) = \sum_{i=1}^n \int_{\mathcal{V}_i^\varepsilon} \|p - p_i\|^2 \phi(p) dp \quad (11)$$

where $\phi(p)$ is a distribution density function measuring probability in the environment, and we set it as a uniform density, that is $\phi(p) = C$.

The partial derivative of $\mathcal{H}(\mathcal{E}, \mathcal{P})$ is that

$$\frac{\partial \mathcal{H}(\mathcal{E}, \mathcal{P})}{\partial p_i} = M_{\mathcal{V}_i^\varepsilon} (p_i - C_{\mathcal{V}_i^\varepsilon}) \quad (12)$$

where $M_{\mathcal{V}_i^\varepsilon}$ and $C_{\mathcal{V}_i^\varepsilon}$ are the mass and mass centroid of each ECBVC. It can be observed that $\frac{\partial \mathcal{H}(\mathcal{E}, \mathcal{P})}{\partial p_i}$ only depends on its own position and the position of its Voronoi neighbors. The optimal positions for each pursuer, where the value of \mathcal{H} is minimized, occur when they are located at the centroid of their respective Voronoi cells. Thus, we choose a dissipative control law v_i in which each pursuer i follows its negative gradient component and moves over its dominance region $\mathcal{V}_i^\varepsilon$:

$$v_i = -\|v_{p, \max}\| \frac{p_i - C_{\mathcal{V}_i^\varepsilon}}{\|p_i - C_{\mathcal{V}_i^\varepsilon}\|}. \quad (13)$$

The abovementioned reactive feedback control law can ensure that each pursuer's location finally converges to its $C_{\mathcal{V}_i^\varepsilon}$. As a result, a centroidal Voronoi tessellation (CVT) [22] is constructed where each generator p_i coincides with its $C_{\mathcal{V}_i^\varepsilon}$.

Lemma 1: If p_1, \dots, p_n converge to $C_{\mathcal{V}_1^\varepsilon}, \dots, C_{\mathcal{V}_n^\varepsilon}$ by adopting control law in (13), then p_e will passively converge to $C_{\mathcal{V}_e^\varepsilon}$.

Proof: The ECBVCs $\mathcal{V}_i^\varepsilon$ and $\mathcal{V}_e^\varepsilon$ can be uniformly discretized as a set of points $\mathcal{S}_i = \{s_i | s_i \in \mathcal{V}_i^\varepsilon\}$ for each pursuer i and $\mathcal{S}_e = \{s_e | s_e \in \mathcal{V}_e^\varepsilon\}$ for the evader, respectively. The point number in \mathcal{S}_i and \mathcal{S}_e can be represented by $|\mathcal{S}_i|$ and $|\mathcal{S}_e|$, respectively.

Since $\|s_i - p_i\| \leq \min\{\|s_i - p_e\|, \|s_i - p_j\|\}$, each point in \mathcal{S}_i is closer to its own generator p_i than to others. Therefore, \mathcal{S}_i can be regarded as a cluster set generated by a cluster center p_i [23], that is, $\mathcal{S}_i = \{s_i \in \mathcal{V}_i^\varepsilon | \|s_i - p_i\| \leq \min\{\|s_i - p_e\|, \|s_i - p_j\|\}\}$. Correspondingly, cluster set $\mathcal{S}_e = \{s_e \in \mathcal{V}_e^\varepsilon | \|s_e - p_e\| \leq \|s_e - p_i\|\}$ with generator p_e as its cluster center.

When all pursuers use the control law in (13) to update their positions recursively, their Voronoi partitions will asymptotically converge to a CVT. The CVT corresponds to the optimal clustering for cluster sets $\mathcal{S}_1, \dots, \mathcal{S}_n$ whose cluster centers are $C_{\mathcal{V}_1^\varepsilon}, \dots, C_{\mathcal{V}_n^\varepsilon}$ [21]. Therefore, the cluster sets adjacent to \mathcal{S}_e are also optimal clustering. Correspondingly, \mathcal{S}_e can be automatically separated since $\|s_e - p_e\| \leq \|s_e - p_i\|$ according to (5). It means that the clustering process for \mathcal{S}_e is completed in a passive way, and \mathcal{S}_e is also an optimal clustering with the cluster center p_e . When $|\mathcal{S}_e| \rightarrow \infty$

$$p_e = \lim_{|\mathcal{S}_e| \rightarrow \infty} \frac{\sum_{s_e \in \mathcal{V}_e^\varepsilon} s_e}{|\mathcal{S}_e|} = \frac{\int_{\mathcal{V}_e^\varepsilon} p dp}{\int_{\mathcal{V}_e^\varepsilon} dp} = C_{\mathcal{V}_e^\varepsilon}. \quad \square$$

Proposition 1 (Properties of CVT [24]): If $\phi(p) = C$, p_e, p_1, \dots, p_n can be evenly spaced when they form a CVT asymptotically.

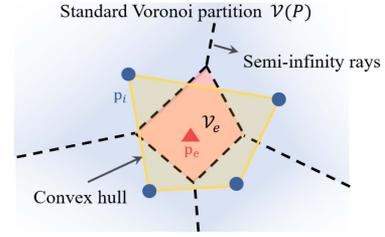


Fig. 2. Encirclement formed by pursuers.

To prove that the evader will be encircled by the team of pursuers in a convex hull, the following lemma is introduced.

Lemma 2 (Theorem 5.10 in [25]): p_i is a vertex of the convex hull of the set \mathcal{P} if and only if its corresponding \mathcal{V}_i shares half-infinite rays (edges) with its neighboring Voronoi cells in standard Voronoi partition $\mathcal{V}(\mathcal{P})$.

Fig. 2 provides an example of the convex hull formed by four pursuers. \mathcal{V}_i is the corresponding Voronoi cell of p_i and shares half-infinite rays (edges) with its neighboring Voronoi cells. Therefore, the enclosed area formed by the p_1, \dots, p_4 is a convex hull according to Lemma 2.

Theorem 1: Assuming $v_{e, \max} < v_{p, \max}$, given a noncollinear random initial configuration $\mathcal{P}(0)$ with $d_c(0) > r_c$, each pursuer adopts control law in (13), the evader will be encircled eventually by a team of pursuers in a convex hull.

Proof: Since the movement of the evader will cause $C_{\mathcal{V}_i^\varepsilon}$ changes accordingly for each pursuer i , it can be proven that the distance change of $C_{\mathcal{V}_i^\varepsilon}$ is smaller than $v_{e, \max} \Delta t$ by computing the analytical expression for the change of $C_{\mathcal{V}_i^\varepsilon}$ in a local coordinate system when $v_{e, \max} \Delta t$ changes. Due to $v_{e, \max} < v_{p, \max}$, p_i will converge to $C_{\mathcal{V}_i^\varepsilon}$ eventually. According to Lemma 1, when p_1, \dots, p_n converge to $C_{\mathcal{V}_1^\varepsilon}, \dots, C_{\mathcal{V}_n^\varepsilon}$ by adopting the control law in (13), p_e will passively converge to $C_{\mathcal{V}_e^\varepsilon}$. Thus, all robots are located at the centroids of their corresponding Voronoi cells eventually, which forms a CVT. As a result, the positions of all pursuers are evenly distributed around the center according to Proposition 1.

Since the \mathcal{V}_e is located at the center of \mathcal{E} , there are n_c Voronoi cells with $n_c \leq n$ sharing semi-infinite rays with their neighbors at the boundary of \mathcal{E} . The pursuers that correspond to these n_c Voronoi cells, can form a convex hull according to Lemma 2. \square

C. Cooperative Capture Via Resizing ECBVC

Once trapping the evader in a convex hull, pursuers are required to rapidly decrease the capture distance while also persistently maintaining the encirclement to prevent evader from escaping. Therefore, the capture strategy is designed to decrease d_c by continually shrinking the boundary of \mathcal{E} .

The shrinkage amount is determined by the movement distance of pursuers. To ensure that the evader can be encircled, the control law v_i for each pursuer i still adopts (13). Therefore, the movement distance of pursuer i during a replanning time step Δt is $D_i(\Delta t) = \|v_i(t) \Delta t\|$. Let $X_i(\Delta t)$ and $Y_i(\Delta t)$ as the x -component and the y -component of $D_i(\Delta t)$, i.e., $D_i(\Delta t) = \sqrt{X_i^2(\Delta t) + Y_i^2(\Delta t)}$. As shown in Fig. 3, the movement distance of the pursuers that are farthest from the evader along the

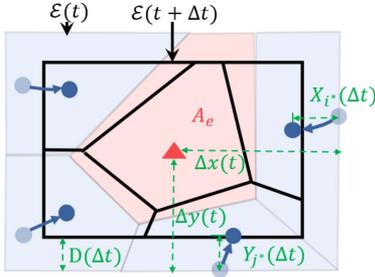


Fig. 3. Area change of the evader Voronoi cell when compressing region \mathcal{E} .

x -axis and y -axis, respectively, are

$$D_X(\Delta t) = \|X_{i^*}(\Delta t)\| \text{ with } i^* = \arg \max_{i \in \mathcal{I}} \|x_i(t) - x_e(t)\|$$

$$D_Y(\Delta t) = \|Y_{j^*}(\Delta t)\| \text{ with } j^* = \arg \max_{j \in \mathcal{I}} \|y_j(t) - y_e(t)\|.$$

The shrinkage amount for $\mathcal{E}(t)$ with Δt can be defined as

$$D(\Delta t) = \min(D_X(\Delta t), D_Y(\Delta t)). \quad (14)$$

If the length difference between the different sides of $\mathcal{E}(t)$ is large at time t_e , the difference will grow even larger and have an impact on the encirclement. Therefore, to shrink $\mathcal{E}(t)$ uniformly, an adaptive shrinkage factor γ is introduced to adjust $D(\Delta t)$ according to $\Delta x(t)$ and $\Delta y(t)$, that is

$$\gamma = \frac{\Delta x(t)}{\Delta x(t) + \Delta y(t)}.$$

As a result, $\Delta x(t + \Delta t)$ and the $\Delta y(t + \Delta t)$ during $t_e \leq t \leq t_c$ can be calculated using the following recursive equations:

$$\begin{aligned} \Delta x(t + \Delta t) &= \Delta x(t) - D(\Delta t)\gamma \\ \Delta y(t + \Delta t) &= \Delta y(t) - D(\Delta t)(1 - \gamma). \end{aligned} \quad (15)$$

Proposition 2: Given $p_e \in \Omega$, if \mathcal{E} is shrinking with time t according to (15), then the area of $\mathcal{V}_e^\mathcal{E}$, denoted by A_e , will decrease gradually.

Proof: Shrinking \mathcal{E} will cause $C_{\mathcal{V}_1^\mathcal{E}}, \dots, C_{\mathcal{V}_n^\mathcal{E}}$ to shift inwards the region. When pursuers move to the centroids of their ECBVCs, the area of each cell in $\mathcal{V}(\mathcal{P}, \mathcal{E})$ will tend to become even according to Proposition 1. Therefore, as the entire area of \mathcal{E} shrinks with time, the area of $\mathcal{V}_1^\mathcal{E}, \dots, \mathcal{V}_n^\mathcal{E}$ will decrease gradually. Due to $p_e \in \Omega$, A_e will also decrease gradually under the influence of the pursuer that surrounds it. \square

According to Proposition 2, A_e can decrease gradually using the capture strategy in (15). Therefore, d_c will also be decreased until satisfying the capture condition $d_c \leq r_c$.

IV. PURSUIT WITH COLLISION AVOIDANCE

This section describes our distributed encirclement and capture algorithm with guaranteed collision avoidance.

Inner-robot collisions can be avoided by calculating separating hyperplanes among a team $\mathbf{a}_{ij}^\top \mathbf{p} \leq b_{ij}$ using (5). In a real-world environment, robot-obstacle collisions also need to

be considered to guarantee safety when pursuing the evader. To avoid collisions strictly in real time, a set of separating hyperplanes between robots and obstacles can be integrated into ECBVC. The separating hyperplane is determined by the distance between $\mathbf{p}_i(t)$ and a bounded convex polytope of obstacle $\Psi_o = \text{conv}(\psi_1, \dots, \psi_{m_o})$ formed by the obstacle vertices $\psi_1, \dots, \psi_{m_o}$. The parameters \mathbf{a}_{io}, b_{io} of the separating hyperplane can be computed as the following quadratic program problem:

$$\begin{aligned} \min \quad & \mathbf{a}_{io}^\top \mathbf{a}_{io} \\ \text{s.t.} \quad & \mathbf{a}_{io}^\top \psi_l - b_{io} \geq 1 \quad \forall l \in 1, \dots, m_o \\ & \mathbf{a}_{io}^\top \mathbf{p}_i - b_{io} \leq 1. \end{aligned} \quad (16)$$

After getting \mathbf{a}_{io}, b_{io} , the robot-obstacles avoidance constraints $b_{io} = \min \mathbf{a}_{io}^\top \Psi_o$ can be constructed. Combined with (5), an ECBVC without collisions at each time t will be obtained.

We further consider the geometric size of the pursuers by using the buffered Voronoi cell. One property of the buffered Voronoi cell is that if the position of the pursuer i is inside its corresponding Voronoi cell $p_i \in \mathcal{V}_i^{\mathcal{E}, b}$, then the whole body of the pursuer is inside of it [12]. Therefore, we denote safety buffer terms $\beta_{ij} = r_i \|\mathbf{a}_{ij}\|$ and $\beta_{io} = r_i \|\mathbf{a}_{io}\|$ to divide each pursuer into collision-free regions and a boundary buffer term $\beta_i = r_i$ to ensure the body of each pursuer within its corresponding B-ECBVC. Note that this buffer term can be further extended to account for the robot with different sizes in different axes. Then, the definition of the B-ECBVC for each pursuer i is given as follows.

Definition 5 (B-ECBVC): Given p_i with $r_i, \forall i \in \mathcal{I}$, satisfying inner-robot and robot-obstacle collision-free conditions, the B-ECBVC $\mathcal{V}_i^{\mathcal{E}, b}$ can be defined as

$$\begin{aligned} \mathcal{V}_i^{\mathcal{E}, b} = \{ \mathbf{p} \in \mathcal{W} \mid & \mathbf{a}_{ij}^\top \mathbf{p} \leq b_{ij} - \beta_{ij} \quad \forall j \neq i, i, j \in \mathcal{I} \\ & \mathbf{a}_{io}^\top \mathbf{p} \leq b_{io} - \beta_{io} \\ & \mathbf{a}_{ie}^\top \mathbf{p} \leq b_{ie} \\ & \text{and } \mathbf{s}_l - [\beta_i \ \beta_i]^\top \leq \mathbf{p} \leq \mathbf{s}_u + [\beta_i \ \beta_i]^\top \}. \end{aligned} \quad (17)$$

It can be observed that the positions of pursuers will be updated in their corresponding B-ECBVCs by adopting the control law of (13). Therefore, if the positions of a pursuer are constrained in the corresponding B-ECBVCs, i.e., for initialization $d_{io}(0) \geq r_i$, $d_{ij}(0) \geq r_i + r_j$, then using the reactive feedback control law in (13), its moving path will always be in a collision-free configuration in the future [12], as shown in Fig. 4.

Another concern about pursuing in obstacles environments is that nonconvex obstacles may cause pursuers to get trapped in local minima. To overcome this issue, we incorporate the navigation function in our previous work [2] into a density map to provide feasible pursuit direction in nonconvex obstacle environments. Then, we update Voronoi region for each pursuer and make the pursuer move to the centroid of its region. The density map is defined as $\phi(p) = e^{-g(p)}$ with point $p \in \mathcal{W} \setminus \mathcal{O}$, where $g(p)$ is a navigation function, that is $g : \mathcal{W} \setminus \mathcal{O} \rightarrow \mathbb{R}^+$

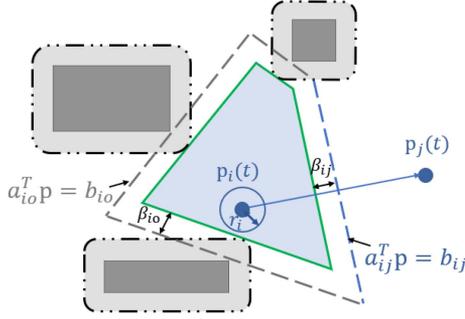


Fig. 4. Illustration of guaranteed collision avoidance. The blue shaded areas represent a B-ECBVC $\mathcal{V}_i^{e,b}$.

that approximates the minimum length of a collision-free path from any p to p_e using Dijkstra's algorithm.

V. EXTENDED PURSUIT METHOD WITH HIGHER ORDER DYNAMICS SYSTEMS

The previous sections mainly discussed the cooperative pursuit method in which the model of pursuers is a first-order system. This section will introduce how to extend our method to higher order systems.

Consider a triple-integrator dynamic system for a quadrotor $\dot{\mathbf{p}} = \mathbf{v}$, $\dot{\mathbf{v}} = \mathbf{a}$, $\dot{\mathbf{a}} = \mathbf{j}$, with differential flatness [26]. $\mathbf{x}_i = [\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i]^T$ denotes the state of the robot. To ensure that the entire pursuit process is smooth and dynamically feasible, a k th order clamped uniform B-spline $\mathbf{C}(u)$ has been selected to represent the trajectory

$$\mathbf{C}(u) = \sum_{j=0}^{M-1} Q_j N_{j,k}(u) \quad (18)$$

where Q_j represents the control point coordinates, j is the control point index, M is the number of control points, and u is the parameter that can be linear transferred to time t . The basis functions $N_{j,k}(u)$ can be calculated recursively using the function in [3]. Then, we formulate a trajectory generation problem to persistently pursue the evader as an optimization problem in a receding horizon fashion that is called model predictive control (MPC) based local motion planning method:

Problem 2 (MPC-based Local Motion Planning):

$$\begin{aligned} \min \quad & \int_t^{t+\Delta t} \lambda_u J_u + \lambda_d dt J_d \\ \text{s.t.} \quad & \mathbf{C}(t_{\text{ini}}) = \mathbf{p}_i(0), \mathbf{C}(t_{\text{end}}) = \mathbf{C}_{\mathcal{V}_i^{e,b}} \\ & \mathbf{C} \in \mathcal{V}_i^{e,b} \\ & \left\| \frac{d^{n_b} \mathbf{C}(u)}{dt^{n_b}} \right\| \leq u_{n_b}^{\max} \\ & \forall i \in \mathcal{I} \end{aligned} \quad (19)$$

where λ_u and λ_d are the tradeoff parameters. For each cost term, $J_u = \left\| \frac{d^{n_b} \mathbf{C}(u)}{dt^{n_b}} \right\|^2$ penalizes the aggressiveness of flight by minimizing the integration of the square of $n_b = 1, 2, 3, 4$.

Algorithm 1: Distributed Pursuit Using B-ECBVCs.

```

1: Input: Position information  $\mathcal{P}$ , Obstacles  $\mathcal{O}$ 
2: Output: Pursuit trajectory  $\mathbf{x}$ 
3: for Each pursuer  $i \in \mathcal{I}$  do
4:   while  $d_c(t) > r_c$  do
5:      $\mathbf{s}_l(t), \mathbf{s}_u(t) \leftarrow$  Get boundary constraints via (8)
6:     if  $d_e(t) \leq 0$  then
7:        $\mathbf{s}_l(t), \mathbf{s}_u(t) \leftarrow$  Reconstruct boundaries via (15)
8:     end if
9:      $\mathcal{V}_i^{e,b}(t), \mathbf{C}_{\mathcal{V}_i^{e,b}}(t) \leftarrow$  Generate B-ECBVC and
       centroid via (17)
10:     $\mathbf{x}_i \leftarrow$  Generate smooth trajectory via (19);
11:   end while
12: end for

```

$J_d = e^{-\|d_{\text{dyn}}(t)\|}$ is defined as the collision cost with moving obstacles, which is used to penalize the nearest distances $d_{\text{dyn}}(t)$ between the quadrotor and moving obstacles. The first term of constraints represents the performance of following the next-to-go point, which is determined by the pursuit path. The second indicates the safe constraints, which limit the trajectory within pursuer's B-ECBVC. The third term $u_{n_b}^{\max}$ is the maximum value of kinodynamic constraints.

By solving Problem 2, at each time step, a dynamically feasible trajectory can be generated to guide the quadrotor following along the pursuit path. The optimization problem is solved again at next time step taking into account changes in the environment. Hence, it has the ability to enhance system's resilience in the face of external perturbations. Algorithm 1 presents the scheme of our distributed encirclement and capture strategy for the MPE problem without collisions.

VI. SIMULATIONS

This section performs some simulations to evaluate the proposed algorithm. We provide some trials to validate the effectiveness of our algorithm with initial randomized configurations and different evader's policies and compare our method with other state-of-the-art methods. In both simulations and real-world experiments, each robot autonomously calculates its own pursuit or escape strategy in a distributed way, only based on the position of other players. All simulations are conducted in MATLAB with an Intel i7 CPU@2.60GHz computer under the same settings. The results are obtained with the following parameters: The innercollision safety radius for pursuer i with $i \in \mathcal{I}$ and evader are set as $r_i = 0.3$ m, $r_e = 0.3$ m, respectively. The capture radius is set as $r_c = 1.0$ m. The control input limits are set as $v_{p,\max} = 0.3$ m/s and $v_{e,\max} = 0.2$ m/s, respectively. The replanning time step $\Delta t = 0.24$ s.

A. Performance Under Various Escaping Policies

In the simulation, the evader adopts a first-order system and its integrator dynamic model is $\dot{p}_e = v_e$ with $\|v_e\| \leq v_{e,\max}$. To evaluate the generalizability of our method, different policies of the evader are selected as follows.

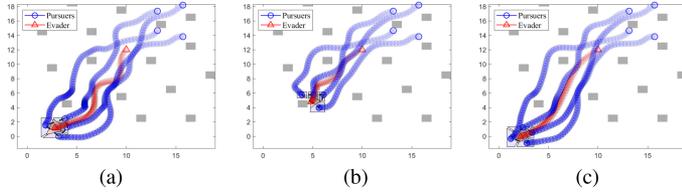


Fig. 5. Simulation of four pursuers and one evader with snapshots for three escape policies at final capture time in obstacle-dense environments. (a) MCP, $t_{c1} = 80.16$ s. (b) GP, $t_{c1} = 55.20$ s. (c) MCP, $t_{c1} = 78.00$ s.

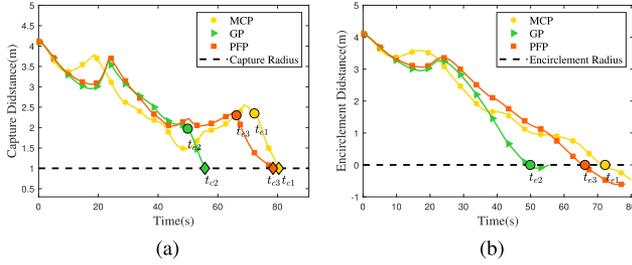


Fig. 6. Distance over time for three escape policies, where circles and diamonds represent the encirclement and capture time, respectively. (a) Capture distance $d_c(t)$. (b) Encircled distance $d_e(t)$.

- 1) The move-to-centroid policy (MCP) in [7] makes the evader far away from neighbor pursuers by moving to its Voronoi centroid. The control law v_{e1} using MCP is

$$v_{e1} = \|v_{e,\max}\| \frac{C_{V_e^w} - p_e}{\|C_{V_e^w} - p_e\|}. \quad (20)$$

- 2) The greedy policy (GP) makes the greedy choice for the evader at each time and keeps it away from the nearest pursuer g . The control law v_{e2} using GP is

$$v_{e2} = \|v_{e,\max}\| \frac{p_g - p_e}{\|p_g - p_e\|}. \quad (21)$$

- 3) The potential field policy (PFP) in [18] makes the evader react more responsively to the threat posed by a team of pursuers. The control law v_{e3} using PFP is

$$v_{e3} = \|v_{e,\max}\| \frac{\sum_i \frac{k_e(p_e - p_i)}{r_i} + p_e}{\left\| \sum_i \frac{k_e(p_e - p_i)}{r_i} + p_e \right\|} \quad (22)$$

where the repulsion gain factor k_e is a tradeoff between surrounding and capture ability.

Fig. 5 shows the trajectories of four pursuers encircling and capturing one evader over time. Starting at the nonencircled initial configuration of staying away from the evader, pursuers can gradually approach it by adopting the control law in (13). The B-ECBVCs of each pursuer and the evader are shaded in blue and red, respectively, shrinking as time passes. Then, pursuers successfully trap the evader in a convex hull, as shown in Fig. 6, where d_e equals to zeros. Once satisfying the encirclement condition, the pursuers adaptively resize the boundaries of their B-ECBVCs to quickly reduce d_c . Over time, the evader is finally captured ($d_c = r_c$) regardless of its various escape policies.

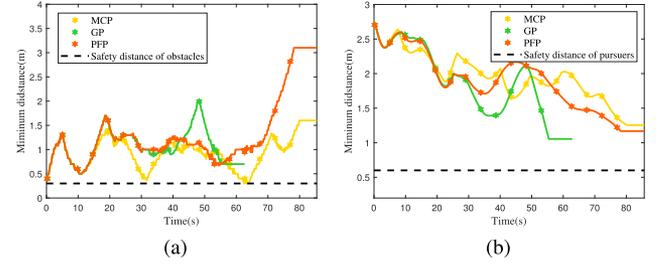


Fig. 7. Distance to evaluate collisions when pursuing the evader over time. (a) Minimum distance between a team of pursuers and obstacles $D_o(t)$. (b) Minimum distance among a team of pursuers $D_p(t)$.

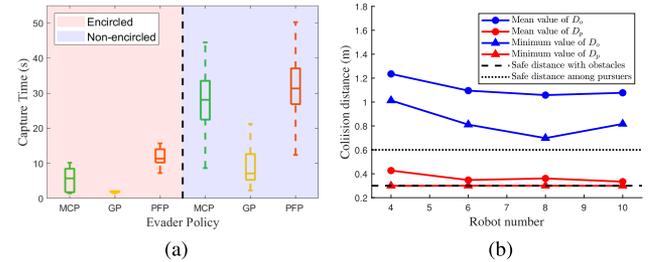


Fig. 8. Capture time and collision distance of 40 trials. (a) Capture time under different escape policies of the evader. (b) Collision distance in obstacle-dense environments with different numbers of robots.

Compared with GP, the evader with MCP and PFP seems more cunning. In MCP, the evader moves far away from its Voronoi edges formed by neighboring pursuers. Similar to MCP, each pursuer using PFP has a repulsive force on the evader and the magnitude of the force depends on the distance. As a result, the capture time for MCP ($t_{c1} = 80.16$ s) and PFP ($t_{c3} = 78.00$ s) are longer than that of GP ($t_{c2} = 55.20$ s). On the other hand, the evader using the GP is only reactive to the nearest pursuer, making it easier to catch.

Besides, thanks to the safe convex regions generated by a set of separating hyperplanes, collisions among obstacles and other teammates can be strictly avoided when pursuing. As shown in Fig. 7, the minimum distance between a team of pursuers and obstacles, denoted by $D_o = \min_{i \in \mathcal{I}} d_{io}$, is larger than r_i while the minimum distance among a team of pursuers, denoted by $D_p = \min_{i,j \in \mathcal{I}} d_{ij}$, is larger than $r_i + r_j$ for all time.

B. Performance on Random Initial Configurations

A series of simulations on random initial configurations of pursuers are conducted to analyze the performance of the proposed algorithm. A total of 40 initial configurations for four pursuers and the evader are randomly generated within the box region $[0, 20] \times [0, 20]$ in an unbounded environment, in which 20 of them, the evader is encircled. Fig. 8(a) reports the mean and variance of capture time for different evader's policies. t_c required for the evader-encircled initialization is less than that for the nonencircled initialization. It is because that resizing boundaries of the B-ECBVCs can reduce the d_c effectively. It can also be observed that the evader with PFP and MCP is more

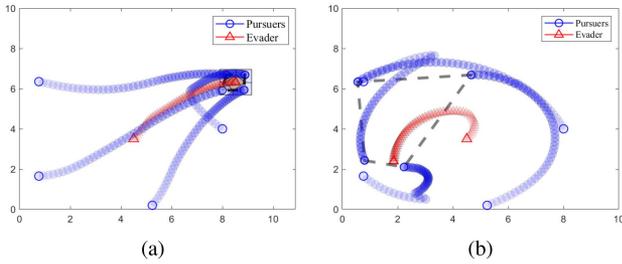


Fig. 9. Trajectories obtained by our method and SH at final capture time. The black dotted line represents encirclement. (a) Ours with PFP, $t_c = 8.50$ s. (b) SH with PFP, $t_c = 9.26$ s.

sensitive to the threat of pursuers and the pursuers need to take some time to trap the evader. Nevertheless, the evader can always be caught within a finite time.

Another 40 trials with the random initial configurations of pursuers in obstacle-dense environments are conducted, and the collision distance is shown in Fig. 8(b) when the number of pursuers is $n_i, i \in \{4, 6, 8, 10\}$. Due to the more crowded initial positions, both D_o and D_p decrease with the increasing number of robots. Nevertheless, the minimum values of D_o and D_p for 40 trials [the line with triangle markers in Fig. 8(b)] show that collision avoidance can always be guaranteed by integrating the separating hyperplanes and the buffered terms into B-ECBVCs.

C. Comparative Results

This section introduces comparisons with several state-of-the-art methods:

- 1) SH method [18], which designs a distributed control law to balance SH performance;
- 2) AM method [7], in which pursuit strategy is designed to move to the boundary center of the evader's Voronoi cell;
- 3) obstacle-aware Voronoi cell (OAVC) method [14], in which pursuers chase an evader guided by an information density map of the evader;
- 4) multiagent deep deterministic policy gradient (MADDPG) method [11], which is a deep reinforcement learning algorithm based on the actor-critic framework.

In this part, the maximum velocity ratio is $v_{e,max}/v_{p,max} = 0.9$.

1) Pursuit in an Obstacle-Free Environment: A comparative experiment between our method, SH and AM are implemented in an obstacle-free environment since SH and AM do not consider obstacles. To be fair, we will compare the proposed method with SH and AM under PFP and MCP, respectively, in the same initial configuration with $r_c = 0.5$ m and $r_i = 0$. As shown in Fig. 9, compared with our method, SH method needs to spend much time blocking the evader under policies PFP when confronted with a random initial position ($t_c = 9.26$ s). This is because the performance of this method relies on strict spatial distributions. In contrast, our method can flexibly adjust the encirclement and capture strategy regardless of escape policies with shorter capture time ($t_c = 8.50$ s). In Fig. 10, using the AM method, pursuers fail to trap the evader before capturing when

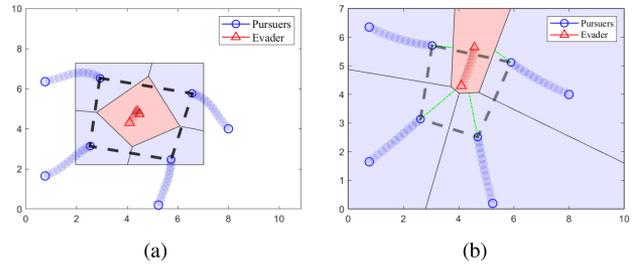


Fig. 10. Trajectories obtained by our method and AM at middle time. The black dotted line represents the encirclement. (a) Our with MCP, $t = 2.38$ s (b) AM with MCP, $t = 2.38$ s.

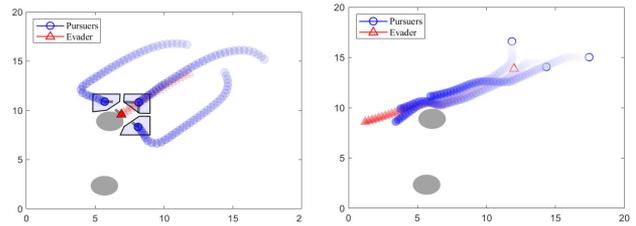


Fig. 11. Performance comparison with MADDPG of one trail.

TABLE I
COMPARISON OF SUCCESS RATE WITH MADDPG

Algorithm	Escape policy			
	Random	GP	MCP	PFP
Ours	100%	100%	100%	90%
MADDPG [11]	100 %	80%	35%	25%

$t = 2.38$ s. This is because the AM method is too greedy to move directly toward the evader in an unbounded environment, making the evader easier to escape. However, our method can keep the evader in the encirclement condition until satisfying the capture condition.

2) Pursuit in an Obstacle Environment: We compare the proposed method with MADDPG.¹ In simulations, 20 initial configurations for three pursuers and two obstacles are randomly generated in an unbounded environment. An example of compared trajectories is given in Fig. 11 and the success rate (catching the evader in a defined space limit with collision avoidance) is shown in Table I. It can be observed that MADDPG struggles to adapt to more intelligent evaders who utilize an advanced escape strategy, whereas it was able to capture a randomly moving evader. This is due to MADDPG's emphasis on optimizing local rewards without any explicit consideration of coordination, which gives the evader an advantage in escaping. In addition, pursuers experience difficulty in learning generalizable policies that are effective across a range of escaping strategies of the evader. In contrast, our algorithm can handle various escape policies of the evader and guarantee collision-free, indicating its greater robustness in cluttered environments.

¹The source code of MADDPG can be found in <https://github.com/starry-sky6688/MADDPG/>

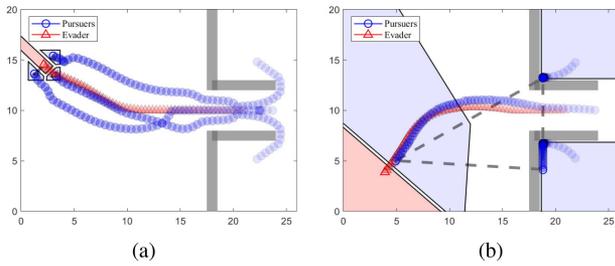


Fig. 12. Comparison between our method and OAVC method with MCP in a nonconvex environment. (a) Our method, capture time $t_c = 28.8$ s (b) OAVC, $t = 28.8$ s

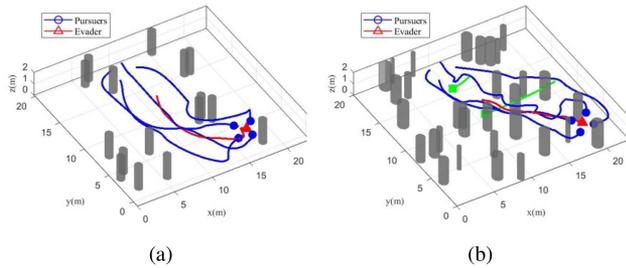


Fig. 13. Pursuit trajectories with four quadrotors in a cluttered environment. The greed curves are the trajectories of moving obstacles.

A comparative simulation is also conducted between the proposed method and the OAVC method for a highly nonconvex environment. Three pursuers and one evader scatter around a long corridor environment where the evader can get away from the corridor. The paths obtained by our method are shown in Fig. 12(a). When the evader hides in the corridor, two initially blocked pursuers can autonomously find collision-free and deadlock-free paths to approach the evader with the guidance of the navigation function. Once exiting from the corridor, the pursuers quickly surround the evader and simultaneously reduce the distance between them. However, as shown in Fig. 12(b), the OAVC method drives the pursuers to move toward the evader's high-density value without considering the effect of obstacles, resulting in the pursuers getting stuck in corners. The OAVC method is greedy and inadvertently creates opportunities for the evader to escape.

D. Results for Higher Order Dynamics Systems

This part validates the performance of the extended method for MPE using quadrotors that are triple-integrator dynamic systems. The dynamic constraints of the quadrotor are set to be $v = [1.0, -0.5, 1.0]$, $a = [2.0, -0.5, 1.5]$, and $j = [5.0, -2.0, 3.0]$ with maximum limit horizontally, minimum and maximum limits vertically. Due to the requirement of higher control frequency of the quadrotor than mobile ground vehicle, the replanning time step is set to be 0.1 s. As shown in Fig. 13(a), pursuers can encircle and capture the evader using smooth trajectories to fast follow the pursuit path.

We further validate the robustness in a dynamic and cluttered environment with 32 static obstacles and two dynamic obstacles shuttling horizontally. As shown in Fig. 13(b), the MPC-based

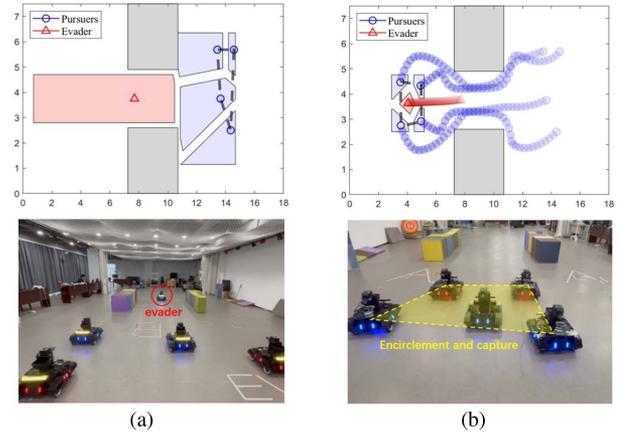


Fig. 14. Pursuing an autonomous evader. (a) $t = 0.0$ s. (b) $t_c = 24.02$ s

local motion planner is capable of maneuvering automatically and adjusting the behavior by replanning a safe trajectory to resist collision with moving obstacles.

VII. EXPERIMENTS

In this section, we test the effectiveness of our algorithm in a real-world environment with $8\text{ m} \times 16\text{ m}$. All experiments are conducted with the same parameters as simulations. We use Robomaster as the platform, which can be formulate as a single integral dynamic model. Each robot is equipped with an onboard computer NX running ROS, allowing the robot to calculate its own policy autonomously in real time. Communication was achieved through the ROS network, and each pursuer only need obtain the position of other robots without knowing their policies. The full video can be found on the <https://youtu.be/H3owgGUoTko>.

We first conduct experiments using autonomous robots in a corridor environment. The PFP strategy is adopted by an autonomous evader since it is a more cunning escape policy. As shown in Fig. 14(a), a group of pursuers is initially at one side of the corridor. The corridor easily causes multiple robots to get stuck or shake back and forth if they cannot cooperatively find directions. As expected, they smoothly navigate through the corridor and approach the evader in a cooperative way by the proposed method, as shown in Fig. 14(b). Once coming out of the corridor, pursuers adaptively adjust the strategy to encircle the evader. Eventually, the evader is captured eventually at $t_c = 24.02$ s.

We also treat a human as an evader to further verify the effectiveness of our method, as shown in Fig. 15. The human moves at an uneven speed, causing randomness and unpredictable moving directions that can react to pursuit more intelligently. In this experiment, the human holds a localization stick, and pursuers can obtain the real-time position of the human through the motion capture system. As shown in Fig. 15, given the nonencircled initial configuration, the team of pursuers can encircle the human from both sides and successfully trap her at $t_e = 5.21$ s. After encirclement, a team of pursuers approach the human quickly and capture it at $t_c = 23.31$ s eventually. Despite having unequal

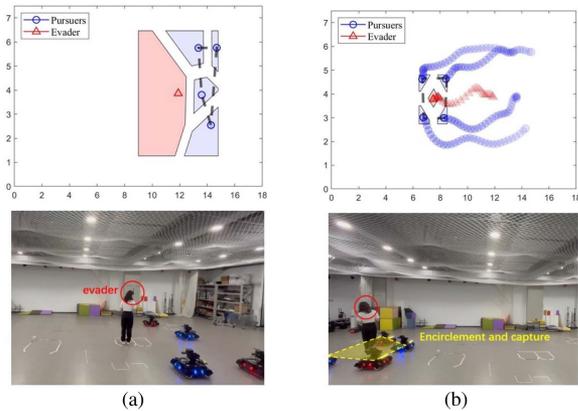


Fig. 15. Pursuing a human as an evader. (a) $t = 0.0$ s (b) $t_c = 23.31$ s

information between the pursuers and the evader, pursuers can still encircle and capture the evader in a cooperative way using our algorithm.

VIII. CONCLUSION

This article develops a distributed encirclement and capture algorithm in an obstacle-dense environment with guaranteed collision avoidance. We give a theoretical analysis to ensure that a team of pursuers can eventually encircle the evader in a convex hull. By continually compressing the boundary of B-ECBVCs, the capture condition will be satisfied while maintaining the encirclement. We also validated our method in both simulations and experiments with autonomous evaders and a human evader to show the effectiveness of our method under various unknown escape policies. Future work includes extensions to multiple evaders and MPE problems with a time constraint.

REFERENCES

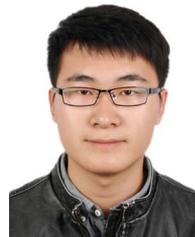
- [1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auton. Robots*, vol. 31, no. 4, pp. 299–316, 2011.
- [2] L. Xi, X. Wang, L. Jiao, S. Lai, Z. Peng, and B. M. Chen, "GTO-MPC-based target chasing using a quadrotor in cluttered environments," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6026–6035, Jun. 2022.
- [3] L. Xi, Z. Peng, L. Jiao, and B. M. Chen, "Smooth quadrotor trajectory generation for tracking a moving target in cluttered environments," *Sci. China Inf. Sci.*, vol. 64, no. 7, pp. 1–16, 2021.
- [4] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5014–5024, Jun. 2016.
- [5] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, Jan. 2016.
- [6] L. Cheng and Y. Yuan, "Multi-player obstacle avoidance pursuit-evasion games with adaptive parameter estimation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 5, pp. 5171–5181, May 2023, doi: [10.1109/TIE.2022.3187577](https://doi.org/10.1109/TIE.2022.3187577).
- [7] A. Pierson, Z. Wang, and M. Schwager, "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 530–537, Feb. 2016.
- [8] J. Selvakumar and E. Bakolas, "Min-max Q-learning for multi-player pursuit-evasion games," *Neurocomputing*, vol. 475, pp. 1–14, 2022.
- [9] X. Fu, J. Zhu, Z. Wei, H. Wang, and S. Li, "A UAV pursuit-evasion strategy based on DDPG and imitation learning," *Int. J. Aerosp. Eng.*, vol. 2022, pp. 1–14, 2022.
- [10] Z. Zhou and H. Xu, "Decentralized optimal large scale multi-player pursuit-evasion strategies: A mean field game approach with reinforcement learning," *Neurocomputing*, vol. 484, pp. 46–58, 2022.

- [11] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6379–6390, 2017.
- [12] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered Voronoi cells," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1047–1054, Feb. 2017.
- [13] B. Tian, P. Li, H. Lu, Q. Zong, and L. He, "Distributed pursuit of an evader with collision and obstacle avoidance," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13512–13520, Dec. 2022, doi: [10.1109/TCYB.2021.3112572](https://doi.org/10.1109/TCYB.2021.3112572).
- [14] A. Pierson and D. Rus, "Distributed target tracking in cluttered environments with guaranteed collision avoidance," in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst.*, 2017, pp. 83–89.
- [15] J. Liao, C. Liu, and H. H. Liu, "Model predictive control for cooperative hunting in obstacle rich and dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5089–5095.
- [16] J. Chen, W. Zha, Z. Peng, and D. Gu, "Multi-player pursuit-evasion games with one superior evader," *Automatica*, vol. 71, pp. 24–32, 2016.
- [17] C. Wang, H. Chen, J. Pan, and W. Zhang, "Encirclement guaranteed cooperative pursuit with robust model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1473–1479.
- [18] X. Fang, C. Wang, L. Xie, and J. Chen, "Cooperative pursuit with multi-pursuer and one faster free-moving evader," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1405–1414, Mar. 2022.
- [19] 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.
- [20] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds," *Int. J. Robot. Res.*, vol. 38, no. 2–3, pp. 196–223, 2019.
- [21] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.
- [22] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Feb. 2004.
- [23] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009.
- [24] Y. Song, B. Wang, Z. Shi, K. R. Pattipati, and S. Gupta, "Distributed algorithms for energy-efficient even self-deployment in mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1035–1047, May 2013.
- [25] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York, NY, USA: Springer, 2012.
- [26] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2520–2525.



Xinyi Wang (Graduate Student Member) received the bachelor's degree in flight vehicle design and engineering from Xiamen University, Xiamen, China, in 2019. She is currently working toward the Ph.D. degree in mechanical and automation engineering with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong.

Her current research interests include motion planning, multiagent pursuit-evasion problem, and optimization.



Lele Xi received the bachelor's degree in automation and master's degree in computer science and technology from the Hebei University of Science and Technology, Shijiazhuang, China, in 2014 and 2017, respectively, and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2022.

He is currently a Lecturer with the Hebei University of Science and Technology. He was a Research Assistant with the Department of Mechanical and Automation, Chinese University of Hong Kong, Hong Kong, in 2019–2021. His current research interests include motion planning, target tracking, and reinforcement learning, specifically in the area of aerial robotics.



Yulong Ding received the bachelor's degree in automation engineering from the Qilu University of Technology, Jinan, China, in 2012, the master's degree in computer science and technology from Xiamen University, Xiamen, China, in 2015, and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2020.

He is currently an Assistant Professor with Tongji University, Shanghai, China. His current research interests include heterogeneous multi-agent systems and task planning of multirobot systems.



Ben M. Chen (Fellow, IEEE) received the bachelor's degree in computer science (control theory) from Xiamen University, Xiamen, Fujian, China, in 1983, the master of science degree in electrical engineering (digital communications) from Gonzaga University, Spokane, Washington, USA, in 1988, and the Ph.D. degree in electrical engineering and computer science (automatic control) from Washington State University, Pullman, Washington, USA, in 1991.

He is currently a Professor of Mechanical and Automation Engineering with the Chinese University of Hong Kong (CUHK), Hong Kong. He was a Provost's Chair Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, before joining CUHK in 2018. He was an Assistant Professor with the Department of Electrical Engineering, State University of New York, Stony Brook, NY, USA, in 1992–1993. His current research interests include unmanned systems and their applications.

Dr. Chen is a Fellow of Academy of Engineering, Singapore. He had served on the editorial boards of a dozen international journals including *Automatica* and IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He currently serves as an Editor-in-Chief of Unmanned Systems and an Editor of *International Journal of Robust and Nonlinear Control*.