Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers

Alyssa Pierson, Zijian Wang, and Mac Schwager

Abstract—**We propose a distributed algorithm for the cooperative pursuit of multiple evaders using multiple pursuers in a bounded convex environment. The algorithm is suitable for intercepting rogue drones in protected airspace, among other applications. The pursuers do not know the evaders' policy, but by using a global "area-minimization" strategy based on a Voronoi tessellation of the environment, we guarantee the capture of all evaders in finite time. We present a decentralized version of this policy applicable in two-dimensional (2-D) and 3-D environments, and show in multiple simulations that it outperforms other decentralized multipursuer heuristics. Experiments with both autonomous and human-controlled robots were conducted to demonstrate the practicality of the approach. Specifically, human-controlled evaders are not able to avoid capture with the algorithm.**

Index Terms—**Distributed robot systems, path planning for multiple mobile robots or agents, autonomous agents.**

I. Introduction

THIS paper considers the problem of coordinating a group of pursuer robots to capture a group of evader robots within a convex, bounded environment. The pursuers do not know the evaders' policy, but instead move to minimize the safe-reachable area of an evader to guarantee capture. Our pursuit strategy is inspired by the area-minimization policy in [1]–[3] for pursuers chasing one evader in a 2D environment.

We present three main results in this paper. First, we extend the results of [1]–[3] to environments of arbitrary dimension, making it practical for aerial robots in 3D environments. Next, we propose a pursuer algorithm for the case of multiple evaders, and prove that it guarantees the capture of all evaders in finite time, however, this algorithm requires global information. Finally, we present a local, decentralized version of the multi-evader pursuit algorithm that performs as well as the global policy in simulation, and can be implemented on-board robots with local sensing and communication. Our approach is decentralized among the pursuers, wherein each pursuer only needs information about itself and its Voronoi neighbors to compute its control algorithm. Simulation results demonstrate the performance of our algorithm in 2D and 3D environments. Hardware experiments were conducted in a motion capture environment using Ouijabots and GoPiGo robots, shown in Fig. 1. Each robot runs its control strategy on board its Raspberry Pi 2. We also conducted experiments with a human-controlled evader, which could not avoid capture.

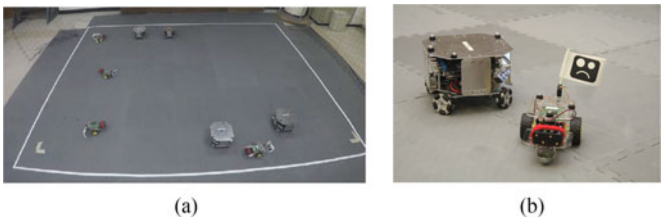


Fig. 1. Experiments were conducted using Ouijabot pursuers and GoPiGo evaders. The evaders are always captured, even when they are controlled by a human operator.

Our algorithm is useful in a number of applications of emerging importance, such as security and surveillance, search and rescue, and wildlife monitoring. The problem is inspired by the classic game of "cops and robbers," [4], [5] where the "cop" attempts to capture the "robber" while the robber simultaneously attempts to avoid capture. With the rise of recreational and industrial use of drones comes a significant threat of drones wandering into restricted airspace over airports, public buildings, protected parklands, or other sensitive areas. Our algorithm provides a practical method by which a fleet of autonomous pursuit drones can neutralize such threats. It is equally applicable to intercepting rogue watercraft in seaports, as well as vehicles or suspicious people on land. The algorithm is also useful for search and rescue applications, where survivors may not know they are being sought, therefore a search strategy must assume no knowledge of the survivors' policy. The algorithm may be useful for wildlife monitoring, where the pursuers are autonomous vehicles tasked with tracking or tagging wildlife, and the wildlife may react to the pursuers as threats.

A. Related Work

For multi-agent pursuit-evasion problems, one common approach is to formulate them as a differential game and solve the associated Hamilton-Jacobi-Isaacs (HJI) partial differential equation [6]. Here, the evader(s) maximize their capture time, while the pursuers minimize the capture time of the evaders. Optimal trajectories with respect to capture time are found by either solving the equations directly or with numerical approximations [7]–[10]. In practice, the differential games approach

Manuscript received September 10, 2016; accepted December 8, 2016. Date of publication December 28, 2016; date of current version January 16, 2017. This work was supported by National Science Foundation under Grants CNS-1330036 and IIS-1646921, and the SAIL-Toyota Center for AI Research. This paper was recommended for publication by Associate Editor. J. Ota and Editor N. Y. Chong upon evaluation of the reviewers' comments.

A. Pierson is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215 USA (e-mail: pierson@bu.edu).

Z. Wang and M. Schwager are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: zjwang@stanford.edu; schwager@stanford.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2016.2645516

poses several challenges. Many techniques require backwards-solving from a terminal condition, which is difficult to determine from initial conditions. Furthermore, the computational complexity of these methods creates a limitation on the number of agents in the game. Another approach is to formulate the pursuit-evasion problem on a graph and solve for the movement between nodes [11]. This approach decomposes the environment to reduce the possible actions of pursuers [12], [13], greatly reducing the computational complexity.

In games with multiple pursuers, coordination among the pursuers can lead to a more efficient capture of the evader(s). Several techniques have been proposed to coordinate the pursuers. One approach is the "sweep-pursuit-capture" strategy, where pursuers form a chain to simultaneous sweep the environment and encircle evaders [14]. This chain formation can also be used to prevent evaders from escaping narrow environments [15], and is relevant to the problem of clearing environments in search and rescue applications [16]. With a probabilistic approach, pursuers employ a greedy pursuit strategy that becomes computationally feasible [17]. Another approach uses Model Predictive Control (MPC) to prevent pursuers from colliding with agents or obstacles in the environment while tracking an evader [18], [19].

Our work builds upon pursuer-evader problems that leverage Voronoi tessellations in designing the pursuer control strategy. When the evader's dynamics are known, the pursuers can tessellate the environment and determine an optimal target assignment [20]. For an evader with unknown dynamics, Huang *et al.* propose an "area-minimization" policy that reduces the safe-reachable area of the evader to guarantee capture in the plane [1]–[3]. This area-minimization strategy has been extended to contain evaders in unbounded environments [21] or in non-holonomic systems [22].

In this paper, we derive a generalized formulation of the area-minimization policy, which guarantees capture of a group of evaders in $\mathbb{R}^N$. The remainder of the paper is organized as follows: Section II defines the multi-pursuer, multi-evader problem. Section III, presents our generalized capture guarantees of a single evader in convex, bounded environments in $\mathbb{R}^N$. Section IV extends these results to multiple evaders. Simulations of multi-evader pursuit in 2D and 3D environments are presented in Section V. Finally, results from our decentralized hardware implementation with ground vehicles are presented in Section VI.

II. Problem Formulation

Consider a group of n_p pursuer and n_e evader agents in a bounded, convex environment $Q \subset \mathbb{R}^N$, with points in Q denoted q. We denote the positions of the pursuers as $x_p^j \in Q$, for $j \in \{1, ..., n_p\}$ and the evaders as $x_e^i \in Q$ for $i \in \{1, ..., n_e\}$. The entire group of $n = n_p + n_e$ agents is denoted $x^T = [\ldots, x_p^{j\,T}, \ldots \mid \ldots, x_e^{i\,T}, \ldots] = [\ldots, x^{k\,T}, \ldots]$ for $k \in \{1, ..., n\}$. We assume all agents have integrator dynamics,

$$\dot{x}_e^i = u_e^i, \quad \|u_e(t)\| \le v_{\max},$$
$$\dot{x}_p^j = u_p^j, \quad \|u_p(t)\| \le v_{\max},$$

where u_e and u_p are the control inputs moving the pursuers and evaders through the environment, subject to the same maximum speed $v_{\max}$. Without loss of generality, we let $v_{\max} = 1$ for the remainder of this paper.

The pursuers' goal is to capture all evaders in the environment. If the pursuers capture an evader at time t_c, it remains captured for all $t > t_c$. An evader i is captured at t_c when

$$\min_{j \in n_p} \|x_e^i(t_c) - x_p^j(t_c)\| < r_c,$$

where $r_c > 0$ is the capture radius.

While we do not know the evader's control policy, we know that it can only avoid capture by moving within its safe-reachable set. The safe-reachable set is defined as all points in Q that an evader i can reach before any other agent. For agents with equal maximum velocities with integrator dynamics, this set is equal to the Voronoi partition, defined

$$V_i = \{q \in Q \| \|q - x^i\| \le \|q - x^j\|, \forall j \ne i, \ \ i, j \le n\}.$$

We define Voronoi neighbors as agents that share a Voronoi boundary, and denote the set of neighbors $\mathcal{N}_i$. In the following sections, we present the pursuer's area-minimization strategy that reduces the evader's reachable set until it is captured.

III. Pursuit of a Single Evader

In [2], the authors propose their "area-minimization strategy" for the pursuit of a single evader by multiple pursuers in $\mathbb{R}^2$. They prove that by driving to the midpoint of the shared Voronoi boundary between a pursuer and evader, the pursuer is guaranteed to capture the evader in finite time. Here, we present our proof of guaranteed capture of a single evader for environments in $\mathbb{R}^N$. Consistent with previous work, we find that the pursuers will drive to the centroid of the shared Voronoi boundary with the evader.

To prove guaranteed capture in finite time, we present our proof in a similar structure to [2]. Despite these parallels in structure, our derivation and proof techniques are different in that we represent edges and center points, facets and centroids, etc, in a general integral form that is independent of dimension. For consistency with existing literature, we denote A_e as the safe-reachable area of the evader and the pursuers' strategy as the "area-minimization" policy for $\mathbb{R}^N$.

A. Area-Minimization Policy

This section described the area-minimization strategy for the pursuers from [2], but re-cast in our integral notation. The safe-reachable area of an evader, A_e, is defined as the points in Q that the evader can reach before any other agent. For agents with equal speeds, this reduces to the Voronoi cell, V_e, of the evader. The area A_e is calculated

$$A_e = \int_{V_e} dq, \tag{1}$$

where V_e is the Voronoi cell of the evader in $\mathbb{R}^N$. The dynamics of A_e are

$$\dot{A}_e = \frac{\partial A_e}{\partial x_e}\dot{x}_e + \sum_{j=1}^{n_p} \frac{\partial A_e}{\partial x_p^j}\dot{x}_p^j.$$

The pursuers choose a strategy that will decrease the area over time. From this formulation, we can decouple the pursuers into their individual contributions $\frac{\partial A_e}{\partial x_p^j}\dot{x}_p^j$. For each pursuer, let

$$u_p^j = -\frac{\frac{\partial A_e}{\partial x_p^j}}{\|\frac{\partial A_e}{\partial x_p^j}\|}. \tag{2}$$

This policy follows the gradient of A_e, moving in the direction of the fastest decrease of the area. We refer to this as the "area-minimization" strategy for consistency with existing literature.

Lemma 1: For the evader x_e and its safe-reachable area A_e in (1) and pursuer x_p^j, the gradient $\frac{\partial A_e}{\partial x_p^j}$ is equivalent to

$$\frac{\partial A_e}{\partial x_p^j} = \frac{L_j}{\|x_p^j - x_e\|}\left(x_p^j - C_{b_j}\right). \tag{3}$$

where b_j is the shared Voronoi boundary between x_p^j and x_e, L_j is the area of the boundary, and C_{b_j} is the centroid of the boundary.

Proof: For the evader area A_e defined in (1), using Leibniz Integral Rule, the derivative of A_e reduces to

$$\dot{A}_e = \sum_{j \in \mathcal{N}_e} \int_{b_j} \left[\frac{\left(x_p^j - q\right)^T \dot{x}_p^j}{\|x_p^j - x_e\|} - \frac{\left(x_e - q\right)^T \dot{x}_e}{\|x_p^j - x_e\|} \right] dq$$

where $\mathcal{N}_e$ is the set of the evader's Voronoi neighbors and b_j is the shared Voronoi boundary between the evader and pursuer j. Define $L_j = \int_{b_j} dq$ and $C_{b_j} = \frac{1}{L_j}\int_{b_j} q dq$, noting that $L_j > 0$ and C_{b_j} is the centroid of the boundary b_j. Thus, $\dot{A}_e$ reduces to

$$\begin{aligned}\dot{A}_e &= \sum_{j \in \mathcal{N}_e} \frac{L_j \left(x_p^j - C_{b_j}\right)^T}{\|x_p^j - x_e\|}\dot{x}_p^j - \sum_{j \in \mathcal{N}_e} \frac{L_j \left(x_e^j - C_{b_j}\right)^T}{\|x_p^j - x_e\|}\dot{x}_e \\ &= \sum_{j \in \mathcal{N}_e} \frac{\partial A_e}{\partial x_p^j}\dot{x}_p^j + \frac{\partial A_e}{\partial x_e}\dot{x}_e.\end{aligned}$$

■

Using Lemma 1 and plugging (3) into (2) the puruser control policy reduces to

$$u_p^j = \frac{\left(C_{b_j} - x_p^j\right)}{\|C_{b_j} - x_p^j\|}. \tag{4}$$

Note this policy directs the pursuers towards the centroid of the shared Voronoi boundary between the purser and the evader. In 2D, the centroid is the midpoint of the shared Voronoi boundary edge (as found by different means in [2]), and in 3D, it is the centroid of the shared Voronoi boundary face.

B. Proof of Guaranteed Capture

The pursuers' strategy in (4) is designed to decrease A_e. To prove this guarantees capture, we parallel the proof technique from [2]. We first show that A_e is always non-increasing. When $\dot{A}_e = 0$, we show the distance between the pursuer and evader is strictly decreasing. We then show that bounds on these dynamics lead to guaranteed capture in finite time. For the remainder of this section, we use a single pursuer in our proof of guaranteed capture, and it is easily seen that the results hold with multiple pursuers.

Lemma 2: Consider a single pursuer, single evader in Q. Under the proposed pursuer strategy (4), the area A_e satisfies $\dot{A}_e \leq 0$ for any admissible evader control strategy. Furthermore, $\dot{A}_e = 0$ if and only if the evader uses the following controller:

$$u_e^* = \frac{(C_b - x_e)}{\|C_b - x_e\|}, \tag{5}$$

where C_b is the centroid of the shared Voronoi boundary between pursuer x_p and evader x_e.

Proof: For a single pursuer, single evader scenario, the dynamics of A_e reduce to

$$\dot{A}_e = \frac{L}{\|x_p - x_e\|}\left[(x_p - C_b)^T \dot{x}_p - (x_e - C_b)^T \dot{x}_e\right]. \tag{6}$$

Plugging in (4) into (6),

$$\dot{A}_e = \frac{L}{\|x_p - x_e\|}\left[-\|x_p - C_b\| - (x_e - C_b)^T \dot{x}_e\right].$$

We see that for this single pursuer case, $\dot{A}_e \leq 0$ and furthermore, for $\dot{A}_e = 0$, we find

$$\dot{x}_e = \frac{(C_b - x_e)}{\|C_b - x_e\|}.$$

■

By Lemma 2, the only evader policy to keep the area constant is to move towards the shared centroid C_b of the Voronoi boundary. Define z as the distance between the pursuer and evader,

$$z = \|x_p - x_e\|^2 = (x_p - x_e)^T (x_p - x_e).$$

The following proves z is strictly decreasing when $\dot{A}_e = 0$.

Lemma 3: For the pursuit strategy, (4), if $\dot{A}_e = 0$, then

$$\dot{z} = \frac{-2\|x_p - x_e\|^2}{\|C_b - x_p\|} < 0.$$

Proof: The dynamics of z are

$$\dot{z} = 2(x_p - x_e)^T (\dot{x}_p - \dot{x}_e).$$

For $\dot{A}_e = 0$, by Lemma 2, we know that the evader dynamics are given by (5). Since C_b exists on the shared Voronoi boundary, $\|C_b - x_p\| = \|C_b - x_e\|$. Plugging in our pursuer strategy (4) and evader strategy (5),

$$\begin{aligned}\dot{z} &= 2(x_p - x_e)^T \left(\tfrac{C_b - x_p - C_b - x_e}{\|C_b - x_p\|}\right), \\ &= \tfrac{-2\|x_p - x_e\|^2}{\|C_b - x_p\|}.\end{aligned}$$

■

By Lemmas 2 and 3, we have that the evader's area A_e is non-increasing, and that when $\dot{A}_e = 0$, z is strictly decreasing. However, there is a possibility that although A_e may be decreasing, z increases and remains within the range $\left[r_c^2, \ell_{\max}^2\right]$, where $\ell_{\max}$ is the maximum distance between any two points in Q. If the evader were to remain in this range, it would never be captured. The following lemma proves this cannot happen.

Lemma 4: Under the pursuer strategy u_p in (4), if $\dot{A}_e \geq -\beta$ for some constant $\beta > 0$, then $\dot{z} \leq -f(\beta)$, where $f(\beta)$ is given by

$$f(\beta) = \ell_{\max} - \frac{\ell_{\max}}{\ell_{\min}}\beta.$$

Proof: First, examine the case when $\dot{A}_e \geq -\beta$, thus

$$-\left(x_e - C_b\right)^T \dot{x}_e \geq \frac{-\beta\|x_p - x_e\|}{L} + \|C_b - x_p\|.$$

Rearranging this expression, we find

$$\left(x_e - x_p\right)^T \dot{x}_e \leq \frac{-\beta\|x_p - x_e\|}{L} \leq \frac{\ell_{\max}}{\ell_{\min}}\beta,$$

where $\ell_{\min}$ is a lower bound on L, defined by the geometry of Q. Substituting this expression into $\dot{z}$,

$$\dot{z} \leq \frac{\ell_{\max}}{\ell_{\min}}\beta - \left(x_p - x_e\right)^T \dot{x}_p \leq \frac{\ell_{\max}}{\ell_{\min}}\beta - \ell_{\max}.$$

■

Lemma 4 also implies that when $\dot{A}_e < -\beta$, $\dot{z} > f(\beta)$. We assume that β is chosen such that $f(\beta) > 0$. We now present our theorem proving that under the pursuer strategy (4), the evader is captured in finite time. Define an "cost-to-capture" function of the system,

$$E = kA_e + z, \tag{7}$$

where $k = \frac{4\ell_{\max} + f(\beta)}{\beta} > 0$. For capture to occur, $E = 0$, either as A_e or z goes to zero.

Theorem III.1: For the cost-to-capture function in (7) and pursuer strategy (4), if capture has not occurred before time t_0, then for $t > t_0$,

$$E(t) < E(t_0).$$

Proof: Lemma 4 gives us the following conditions, which much be true at any given time:

1) *Condition 1:* $\dot{A}_e \geq -\beta$ and $\dot{z} \leq -f(\beta)$, or
2) *Condition 2:* $\dot{z} > f(\beta)$ and $\dot{A}_e < -\beta$.

The derivative of E is

$$\dot{E} = k\dot{A}_e + \dot{z},$$

and we know $\dot{A}_e \leq 0$. Under Condition 1, $\dot{E} \leq -f(\beta)$. Under Condition 2, we see that $\dot{A} < -\beta$ and $\dot{z} > f(\beta)$, however, since the agents are restricted by maximum speeds, $\dot{z} \leq 4\ell_{\max}$. Thus, for $k = \frac{4\ell_{\max} - f(\beta)}{\beta}$,

$$\dot{E} < -k\beta + 4\ell_{\max} < -f(\beta),$$

which implies the cost to capture decreases to zero in finite time, ensuring capture of the evader. ■

IV. Extension to Multiple Evaders

Next, we present our algorithms for extending the cooperative pursuit to multiple evaders in the environment. In Section III, we presented the area-minimization strategy to reduce the safe-reachable area of a single evader, which guarantees capture in finite time. Here, we present a global area-minimization strategy that guarantees the capture of all evaders in finite time. We also present a decentralized version of the algorithm that we later implement in hardware.

Let A_{e_i} be the safe-reachable area of a single evader, and

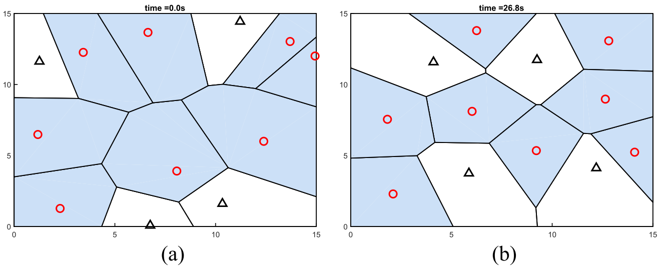
$$A_e = \sum_{i=1}^{n_e} A_{e_i},$$

for all evaders in the environment. Its derivative is

$$\dot{A}_e = \sum_{i=1}^{n_e} \frac{\partial A_{e_i}}{\partial x_e^i}\dot{x}_e^i + \sum_{i=1}^{n_e}\sum_{k\in\mathcal{N}_{e_i}} \frac{\partial A_{e_i}}{\partial x_e^k}\dot{x}_e^k + \sum_{j=1}^{n_p}\sum_{i\in\mathcal{N}_{p_j}} \frac{\partial A_{e_i}}{\partial x_p^j}\dot{x}_p^j.$$

Algorithm 1: Global Multi-Evader Pursuit.

```
Calculate nearest evader $x_e^\kappa$
Coordinate with other pursuers to determine $P_\kappa$
while $(\min_{j\in P_\kappa}\|x_p^j - x_e^\kappa\|) > r_c$ do
    Compute new Voronoi tessellation $\bar{V}_\kappa$ (8)
    Pursue $x_e^\kappa$ using (9)
end while
Once $x_e^\kappa$ is captured, update target
```



(a) (b)

Fig. 2. Symmetry trap when pursuers (triangles) attempt to minimize the area of all evaders (circles). The pursuers get trapped in the final configuration. (a) Initial Configuration. (b) Final Configuration.

A. Naive Approach

When designing the pursuer control policy, it is tempting to extend the results from Section III and allow the pursuers to minimize the area of all neighboring evaders, ie,

$$u_p^j = -\sum_{i\in\mathcal{N}_{p_j}} \frac{\partial A_{e_i}}{\partial x_p^j}.$$

Under this policy, the pursuer attempts to simultaneously reduce all neighboring evaders' safe areas. For the single-evader case, this control policy guaranteed capture of the evader in finite time. However, in the multi-evader case, this policy cannot guarantee capture, and generally performs poorly. Consider when the evaders are arranged symmetrically around the pursuer, then the pursuer can be caught in a "symmetry trap" as the components of u_p^j sum to zero. Fig. 2 illustrates a symmetry trap scenario wherein the pursuers fail to capture any evaders.

B. Global Policy With Guaranteed Capture

To avoid the symmetry trap, we propose a policy, requiring global information, in which pursuers target their nearest evader, and coordinate with other pursuers that share the same target. This effectively turns the multi-evader problem into parallel single-evader pursuit problems. Using this strategy, we can guarantee the capture of all evaders in finite time. Algorithm 1 presents a high-level overview of the strategy.

Instead of pursuing all neighboring evaders, each pursuer targets only its nearest evader. For a targeted evader κ, denote the set of pursuers assigned to that target as P_κ. Let $\bar{V}_\kappa$ denote the Voronoi partition between evader κ and its pursuers, defined

$$\bar{V}_\kappa = \{q|\|x_e^\kappa - q\| \leq \|x_p^j - q\|, \ \ j \in P_\kappa\}. \tag{8}$$

The pursuers then use $\bar{V}_\kappa$ in their control policy,

$$u_p^j = \frac{\left(C_{\bar{b}_{\kappa j}} - x_p^j\right)}{\|C_{\bar{b}_{\kappa j}} - x_p^j\|}, \tag{9}$$

where $\bar{b}_{\kappa j}$ is the Voronoi boundary between x_e^κ and x_p^j in $\bar{V}_\kappa$ and $C_{\bar{b}_{\kappa j}}$ is the centroid of that shared boundary. Note that multiple pursuers may target the same evader, and there may be evaders that are initially not targeted, but this does not prevent the capture of all evaders.

Proposition 1: By Algorithm 1, using control law (9), the pursuers can capture all evaders in finite time.

Proof: First, consider an evader κ initially targeted by pursuers $j \in P_\kappa$. To calculate (10), pursuer j ignores all other evaders $i \neq \kappa$ and pursuers $j \notin P_k$ to calculate V^κ. This becomes a single-evader sub-problem for $j \in P_\kappa$ pursuing evader κ. By Theorem III.1, pursuers $j \in P_\kappa$ capture evader κ in finite time.

Now consider an evader that is not initially targeted by any pursuers. Since all targeted evaders will be captured in finite time, there exists some time $\tau < \infty$ that a non-targeted evader will be targeted. Thus, all evaders in the environment will be captured in finite time. ■

In designing our global algorithm, we do not claim it is optimal, only that it guarantees capture. The only known method to find an optimal strategy in this setting is to solve the HJI equations, which is intractable for large numbers of agents.

C. Decentralized Local Policy

In the previous section, we proposed a global policy for the pursuers that guarantees capture of all evaders in finite time. However, this policy requires coordination and communication among all pursuers, making it difficult to implement. Here, we present a decentralized, local heuristic that performs comparably to the global algorithm in practice.

Instead of requiring pursuers to calculate $\bar{V}_\kappa$ for u_p^j, consider instead that each pursuer calculates its Voronoi cell based on all agents, which can be computed in a decentralized fashion [23]. From V_j, the pursuer then chooses the nearest evader from its Voronoi neighbors as its target x_e^κ. Overall, the control policy is

$$u_p^j = \frac{(C_{b_{\kappa j}} - x_p^j)}{\|C_{b_{\kappa j}} - x_p^j\|}, \tag{10}$$

where $C_{b_{\kappa j}}$ is the centroid of the shared Voronoi boundary between x_e^κ and x_p^j. If there are no evader Voronoi neighbors, then the pursuer moves directly towards the nearest evader in the environment. There are two key differences between the decentralized local policy and the global policy:

1) Pursuer x_p^j calculates V_j and $C_{b_{\kappa j}}$ using all agents in the environment.
2) Pursuers are allowed to switch targets over time.

These two relaxations of the policy ensure that at every time step, a pursuer j only needs local information about its Voronoi neighbors to compute its control law. Furthermore, since it includes all nearby pursuers in calculating V_j, the pursuer policies are cooperative with one another. Under this heuristic, we cannot guarantee capture in finite time, however, in all simulations and experiments we have observed that capture is achieved. We know of one theoretical counter-example for circular environments when all agents start equally spaced on a constant radius, creating a symmetry trap. This appears to be highly unlikely in practice, and we have not found counter-examples for non-circular environments. Randomized simulations demonstrate its performance. Algorithm 2 summarizes the main steps.

Algorithm 2: Local Area-Minimization Policy.

```
Calculate Voronoi tessellation with all agents
Determine nearest evader from Voronoi neighbors, $e^\kappa$
Compute $\dot{x}_p^j$ (10)
If no neighbor evader exists, move directly towards
  nearest evader
```

V. Simulations

Simulations were performed in Matlab to verify the behavior of our algorithm. We also compare our performance to a baseline strategy wherein the pursuer directly chases the evader [24]. Although our algorithm works for any evader policy, we need to implement an evader control law for simulations. If the evaders simply run away from pursuers, they will get trapped against a wall. If the evader attempts to maximize its area A_e, then by Lemma 2 the evader drives directly towards the pursuer (clearly a bad idea!). Similarly, an area-minimization policy only helps the pursuers. With these considerations, based on a comparison of several policies, we choose a "move-to-centroid" control, common in Voronoi-based multi-agent literature [23]. Here,

$$u_e^i = \frac{(C_{V_i} - x_e^i)}{\|C_{V_i} - x_e^i\|}, \tag{11}$$

where V_i is the Voronoi cell of the evader calculated using all agents, and C_{V_i} is the centroid. Intuitively, this policy drives the evader away from edges of its cell, balancing the threat of neighboring pursuers and avoiding environmental boundaries. We choose this policy for its simple, decentralized nature as well as its natural threat-avoidance properties.

A. Multi-Evader Pursuit in 2-D

Fig. 3 shows $n_p = 4$ pursuers and $n_e = 8$ evaders over time. The pursuers use the decentralized area-minimization policy from Algorithm 2, which targets their nearest Voronoi evader neighbor. In Fig. 3, the evader's safe-reachable area is shaded in blue. Over time, we see the evader's area shrinking, with successful capture of all agents by the end of the simulation. Fig. 3(f) plots the minimum distance an evader is to any pursuer over time, with the black horizontal line showing the capture radius. Although this value may temporarily increase, over time every evader is captured.

B. Multi-Evader Pursuit in 3-D

Fig. 4 shows a 3D simulation of $n_p = 4$ and $n_e = 2$ evaders. The Voronoi cells of the evaders are shaded in blue, showing the total volume of the evader's safe-reachable area. The pursuers use the decentralized heuristic from Algorithm 2, and over time, successfully capture both evaders.

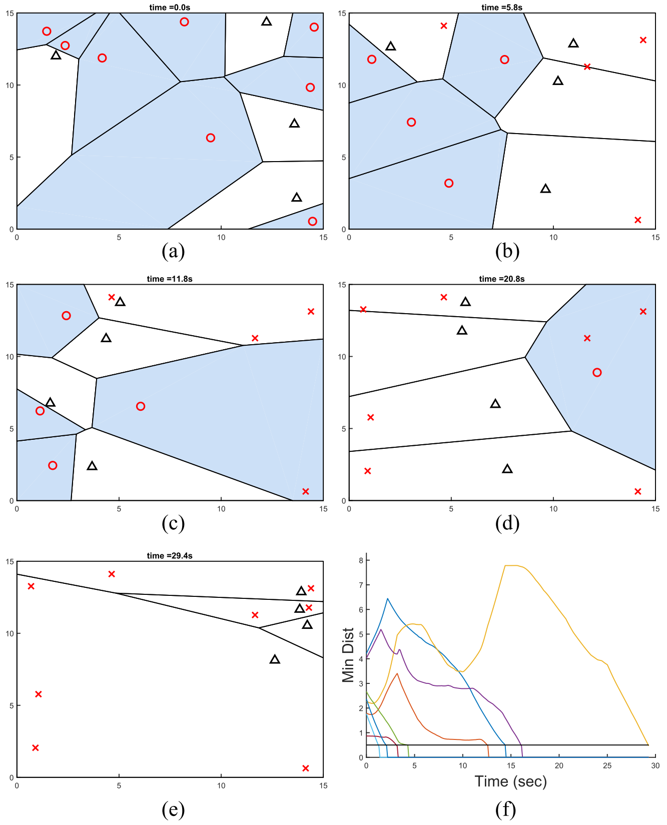


Fig. 3. (a)–(e) Simulation of four pursuers (triangles) and eight evaders (circles), with captured evaders as x's and safe-reachable area shaded. (f) Minimum distance over time of evaders to any pursuer, with capture radius marked in black.

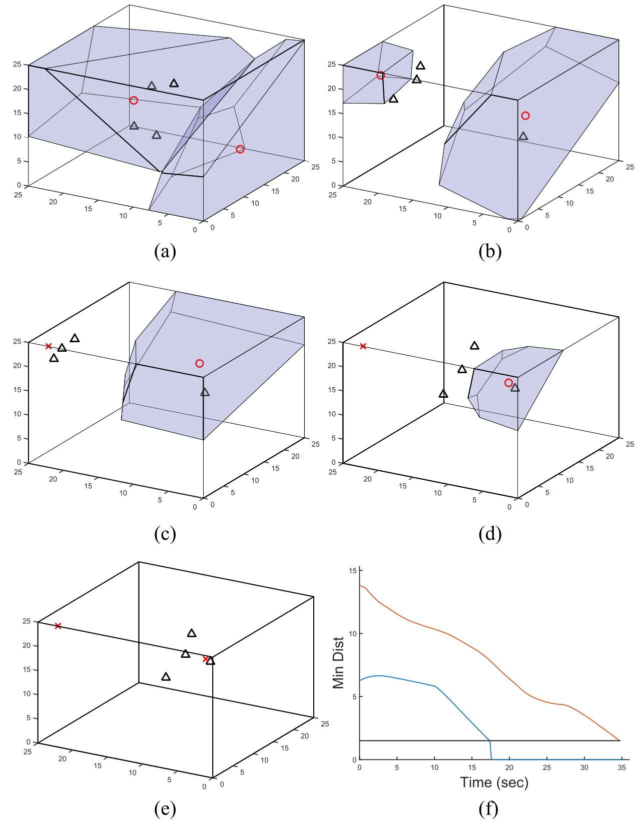


Fig. 4. (a)–(e) Simulation of four pursuers (triangles) and two evaders (circles), with captured evaders as x's and the safe-reachable area shaded. (f) Minimum distance over time of evaders to any pursuer, with capture radius marked in black.

C. Comparison to Other Methods

To analyze the performance of our global and local area-minimization policies, we conducted trials across a variety of scenarios in 2D and 3D. As a baseline, we compare these policies to two other comparable deterministic pursuit strategies. A summary of all pursuer policies is given in Table I. We do not compare to HJI-based strategies, which are intractable for large groups. The first baseline strategy we call the "Direct Charge" strategy. Here, a pursuer drives directly towards its nearest evader, e^{κ}, guaranteeing capture in closed, simply connected domains [24]. For a target $\kappa = \arg\min_{i \in n_e} \|x_p^j - x_e^i\|$,

$$u_p^j = \frac{\left(x_e^{\kappa} - x_p^j\right)}{\|x_e^{\kappa} - x_p^j\|}. \tag{12}$$

A limitation of this policy is a lack of cooperation between pursuers. The next strategy we call the "Hungarian Direct Charge" policy, where the pursuers are assigned a target κ with a Hungarian algorithm, then pursue targets with (12).

Fig. 5(a) summarizes the mean final capture time for seven different pursuer-evader combinations in 2D. For each combination, 100 trials with randomized initial configurations were run for each pursuer strategy. From Fig. 5(a), both the global and local area-minimization policies dominate the baseline strategies. Furthermore, note that our decentralized local policy performs just as well as the global policy in almost all scenarios. The one exception is for the combination of $n_p = 1$ pursuer chasing $n_e = 4$ evaders. Here, the local policy performs worse, but in practice, the global policy is already "decentralized" in implementation.

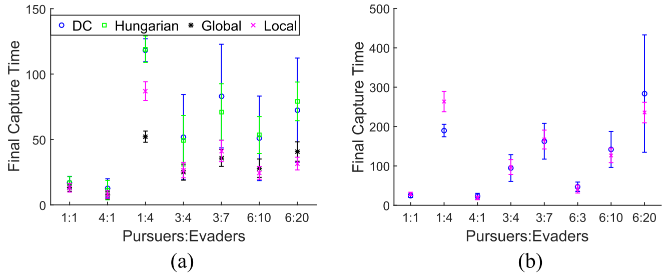
TABLE I
SUMMARY OF PURSUER POLICIES

Policy	Assignment	Controller
Direct Charge (DC)	Nearest Evader	(12)
Hungarian DC	Hungarian Alg.	(12)
Global Area-Min	Nearest Evader	(9)
Local Area-Min	Nearest Neighbor	(10)

Fig. 5(b) summarizes the mean final capture time for seven different pursuer:evader scenarios in 3D. For each scenario, 50 randomized trials were run for each pursuer strategy, comparing Algorithm 2 with the Direct Charge baseline policy. From these comparisons with other policies, we find that the decentralized area-minimization heuristic in Algorithm 2 is an effective pursuer policy to capture multiple evaders in a convex, bounded environment.

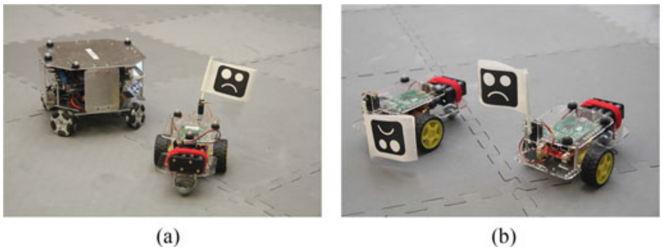
VI. Experiments

Here, we present our experimental results, which demonstrate our decentralized area-minimization policy in 2D. The experiments were conducted in the Autonomous Systems Lab (ASL) at Stanford University, with both autonomous evaders and a human-controlled evader. The environment is 4 m × 3 m,



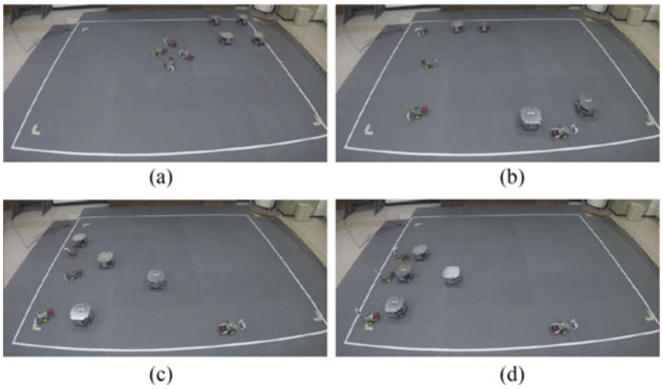
(a) (b)

Fig. 5. Mean final capture time for the pursuer policies in Table I over randomized trials. The local area-minimization policy from Algorithm 2 is shown in pink x's. (a) 2-D. (b) 3-D.



(a) (b)

Fig. 6. (a) Picture of Ouijabot and GoPiGo robots used in the experiment. (b) The capture status of the GoPiGos are indicated via flags.



(a) (b) (c) (d)

Fig. 7. Still frames of the experiment over time for all autonomous evaders.

seen in Figs. 7 and 9. For the pursuers, we used four custom Ouijabots designed in our lab [25]. The evaders are the Dexter Industries GoPiGo robots.[1] Both robotic platforms are equipped with a Raspberry Pi 2 running Linux and ROS, allowing us to implement Algorithm 2 and the evaders' "move-to-centroid" controller completely on-board. Localization is performed with Vicon,[2] with position data broadcasted over the ROS network. Except for this position broadcast, no other communication occurs between any robots, and the robots have no knowledge about other robots' policies other than knowing if a neighbor is a pursuer or evader. The maximum velocity for all agents is capped at 0.2 m/s.

To visualize when an evader is captured during experiments, the GoPiGos are equipped with status flags, pictured in Fig. 6. While free, an evader's flag is down. Once captured, the flag is raised.

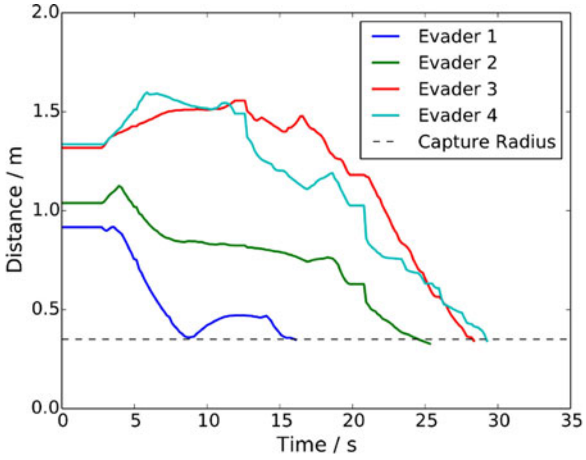
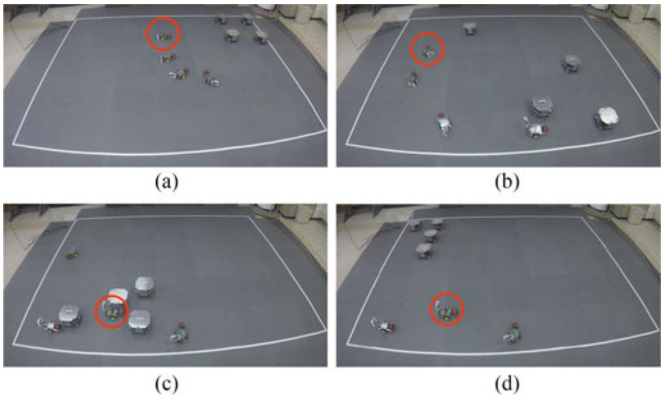


Fig. 8. Minimum distance of each evader to any pursuer over time. The capture radius is denoted with the dotted line.



(a) (b) (c) (d)

Fig. 9. Still frames of the experiment over time. The human-controlled evader is circled in red. (a) t = 0 s. (b) t = 15.3 s. (c) t = 29.7 s. (d) t = 44.8 s.

A. *Four Autonomous Evaders*

In the first experiment, all agents are autonomous and utilize on-board controllers. Frames from the video are shown in Fig. 7, and the full video can be found on the MSL website.[3] Initially, all pursuers start in one corner of the environment to give the evaders the greatest advantage. Despite noisy actuation and network delays, the experiment performs as expected, with all pursuers capturing all evaders. Fig. 8 plots the minimum distance to any pursuer from the evaders over time.

B. *One Human Evader + Three Autonomous Evaders*

The evaders in the first experiment use a decentralized move-to-centroid control policy. The policy performs well, but one may wonder if a human-controlled evader, with greater planning and predication capabilities, can do better. For this experiment, we convert one evader into a human-controlled evader. Here, we tele-operate the robot with a joystick, while all other robots remain autonomous. Unlike the other evaders, a human-controlled evader has full knowledge of the system, including how the pursuers react. Despite these advantages, the pursuer agents still

[1] http://www.dexterindustries.com/gopigo/

[2] https://www.vicon.com/

[3] http://msl.stanford.edu

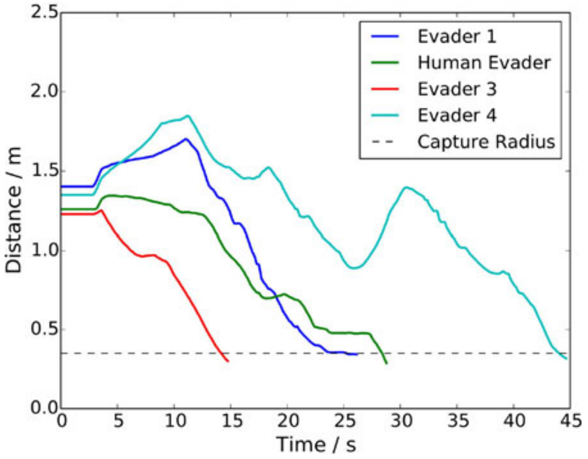


Fig. 10. Minimum distance of each evader to any pursuer over time. The capture radius is denoted by the dotten line.

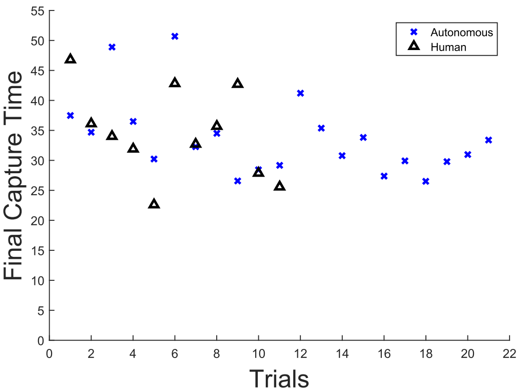


Fig. 11. Final capture time of evaders over different trials. All-autonomous trials are marked with x's and human-evader trials with triangles.

successfully capture all evaders. Fig. 9 shows still frames from our experiment video, and Fig. 10 plots the minimum distance between any pursuer and the evaders over time. Note that in Fig. 10, the human-controlled evader is not the last evader to be captured. Finally, Fig. 11 plots the final capture time over the different autonomous and human-controlled trials.

VII. Conclusion

In this paper, we presented our algorithm to control multiple pursuers to capture multiple evaders in a bounded, convex environment in $\mathbb{R}^N$. We also presented a distributed version of this algorithm, which is shown to perform similarly to the global policy in simulations. Experiments were conducted with the distributed algorithm driving Ouijabots to pursue GoPiGo evaders. In the experiments, we included a human-controlled evader that was unable to escape capture. Future extensions of this work will explore the pursuit of evaders in environments with obstacles, in unbounded environments, and implementation on aerial robots in 3D environments.

Acknowledgment

The authors would like to thank Zhengyuan Zhou for his insightful discussions during the preparation of this paper.

References

[1] H. Huang, W. Zhang, J. Ding, D. Stipanovic, and C. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *Proc. 2011 50th IEEE Conf., Decis. Control Eur. Control Conf.*, Dec. 2011, pp. 4835–4840.

[2] H. Huang, Z. Zhou, W. Zhang, J. Ding, D. M. Stipanovic, and C. J. Tomlin, "Safe-reachable area cooperative pursuit," *IEEE Trans. Robot.*, 2012, submitted for publication.

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

[4] R. Nowakowski and P. Winkler, "Vertex-to-vertex pursuit in a graph," *Discrete Math.*, vol. 43, no. 2, pp. 235–239, 1983.

[5] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Appl. Math.*, vol. 8, no. 1, pp. 1–12, 1984.

[6] R. Isaacs, *Differential Games: A Mathematical Theory With Applications to Warfare and Pursuit, Control and Optimization*. North Chelmsford, MA, USA: Courier Corporation, 1999.

[7] M. Falcone and R. Ferretti, "Semi-Lagrangian schemes for Hamilton–Jacobi equations, discrete representation formulae and godunov methods," *J. Comput. Phys.*, vol. 175, no. 2, pp. 559–575, 2002.

[8] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[9] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proc. 2011 IEEE Int. Conf., Robot. Autom.*, May 2011, pp. 1451–1456.

[10] M. V. Ramana and M. Kothari, "A cooperative pursuit-evasion game of a high speed evader," in *Proc. 2015 54th IEEE Conf. Decis. Control*, Dec. 2015, pp. 2969–2974.

[11] T. D. Parsons, *Pursuit-evasion in a Graph*. Berlin, Germany: Springer, 1978, pp. 426–441.

[12] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 875–884, Oct. 2005.

[13] N. M. Stiffler and J. M. O'Kane, "Pursuit-evasion with fixed beams," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 4251–4258.

[14] S. D. Bopardikar, F. Bullo, and J. P. Hespanha, "Cooperative pursuit with sensing limitations," in *Proc. 2007 Amer. Control Conf.*, Jul. 2007, pp. 5394–5399.

[15] A. Kolling and S. Carpin, "Pursuit-evasion on trees by robot teams," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 32–47, Feb. 2010.

[16] G. Hollinger, S. Singh, and A. Kehagias, "Improving the efficiency of clearing with multi-agent teams," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1088–1105, 2010.

[17] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

[18] A. Ataei and I. C. Paschalidis, "Quadrotor deployment for emergency response in smart cities: A robust MPC approach," in *Proc. 2015 54th IEEE Conf. Decis. Control*, Dec. 2015, pp. 5130–5135.

[19] A. Pierson, A. Ataei, I. C. Paschalidis, and M. Schwager, "Cooperative multi-quadrotor pursuit of an evader in an environment with no-fly zones," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 320–326.

[20] E. Bakolas and P. Tsiotras, "Optimal pursuit of moving targets using dynamic Voronoi diagrams," in *Proc. 49th IEEE Conf. Decis. Control*, Dec. 2010, pp. 7431–7436.

[21] S. Pan, H. Huang, J. Ding, W. Zhang, D. Stipanovic, and C. Tomlin, "Pursuit, evasion and defense in the plane," in *Proc. 2012 Amer. Control Conf.*, 2012, pp. 4167–4173.

[22] M. Kothari, J. G. Manathara, and I. Postlethwaite, "A cooperative pursuit-evasion game for non-holonomic systems," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 1977–1984, 2014.

[23] 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, Apr. 2004.

[24] S. Alexander, R. Bishop, and R. Ghrist, "Pursuit and evasion in non-convex domains of arbitrary dimensions," in *Proc. Robot., Sci. Syst.*, Philadelphia, USA, Aug. 2006. doi:10.15607/RSS.2006.II.015.

[25] Z. Wang, G. Yang, X. Su, and M. Schwager, "OuijaBots: Omnidirectional robots for cooperative object transport with rotation control using no communication," in *Proc. Int. Conf. Distrib. Auton. Robot. Syst.*, London, U.K., Nov. 2016.