Safe-Reachable Area Cooperative Pursuit

Haomiao Huang, Zhengyuan Zhou, Wei Zhang, Jerry Ding, Dušan M. Stipanović, and Claire J. Tomlin

***Abstract*—This paper considers a pursuit-evasion game where a number of pursuers are attempting to capture a single evader. Cooperation among multiple agents can be difficult to achieve, as this coordination may require considering all possible actions of the agents in question. This work presents a decentralized, real-time algorithm for cooperative pursuit of a single evader by multiple pursuers in bounded, simply-connected planar domains. The pursuers share state information but compute their inputs independently. No assumptions are made about the evader's control strategies other than requiring evader control inputs to conform to a limit on speed, and proof of guaranteed capture is shown when the domain is convex and the players' models are identical. Simulation results are presented showing the effectiveness of this strategy, and experimental results using the pursuit strategy to guide human players in a pursuit-evasion game are also presented.**

I. Introduction

Cooperation between agents is often a source of considerable difficulty for adversarial games with many agents. Computing solutions over the joint input space of multiple agents can greatly increase the computational complexity of the problem. This paper considers a multi-agent pursuit-evasion game, with a number of pursuers attempting to capture a single evader in a simply connected planar region. The pursuers may have speed equal to or faster than the evader, and the objective is to find a successful cooperative pursuit strategy for the pursuing agents.

The challenge is to find a solution strategy that induces effective cooperation among the pursuers without incurring a significant computational burden. To accomplish this, an approach is taken that allows computations to occur only in the state space of single agents instead of the high dimensional joint state space of all agents. This allows solutions to be computed quickly and in real-time.

This work presents a decentralized, guaranteed pursuit strategy where the pursuers cooperatively minimize the area of the evader's safe-reachable set. The evader's safe-reachable set is the set of all points that the evader can move to without being captured along the way by a pursuer. The set can be computed using a modified fast marching method (FMM) algorithm [1], or analytically for convex game domains when the pursuers and evader have equal speeds [2]. Each pursuer influences the evader's safe-reachable set only on a portion of the safe-reachable set's boundary, analogous to the shared Voronoi boundary in a Voronoi decomposition. Thus each pursuer's input decouples from that of the other pursuers and can be computed independently, but their inputs are coupled through the safe-reachable set and the evader's position, giving rise to cooperation between the pursuers. The pursuit algorithm is thus decentralized in the sense that the pursuers compute their control actions independently given the agent positions, which is the only shared information, resulting in real-time computation of the control inputs. Some elements of this pursuit algorithm were first presented in [3] for convex domains where the evader and pursuers have equal speeds. This work includes the previous results and also extends the algorithm to more general domain geometries and agent speeds, and includes further simulation and experimental results.

For convex domains and equal speeds, this pursuit strategy results in guaranteed capture of the evader in finite time regardless of the strategy or inputs of the evader. Empirically, the algorithm is shown to result in effective cooperation between pursuing agents, resulting in superior performance to techniques such as the pure pursuit strategy, where the pursuers attempt to minimize the instantaneous distance to the evader. This ability to engender cooperation is one of the key advantages of the strategy proposed in this chapter.

In addition, reachability and reachable sets are natural, intuitive tools to use with human agents and hierarchical control frameworks [4], [5]. In this particular case, the safe-reachable set of the evader can be used to create intuitive visualizations for a pursuing human agent. In experiments based on the **BE**rkeley **A**utonomy and **R**obotics in **CA**pture-the-flag **T**estbed (BEARCAT) with human agents, human pursuers utilizing the pursuit strategy with appropriate visualization of the set were able to capture an evader even in the presence of disturbances such as GPS noise and communications delay.

The following sections will present the pursuit strategy in detail, as well as simulation and experimental results demonstrating its use. First, related work is presented in Section II. The pursuit-evasion problem in question is defined in Section III, and its formulation as a differential game is discussed. Section IV lays out the pursuit strategy, first for the case of equal-speed pursuers and evader in a convex domain, and then for the more general case of non-convex domains and unequal speeds. A number of simulations are presented comparing the performance of the proposed strategy against the pure pursuit strategy. Section V describes these simulations and some comparative tests that were conducted for evaluating the algorithm. Experimental results are also presented for games involving human agents, and these are described in Section VI. These experiments demonstrate the feasibility of the safe-reachable area minimization strategy not only for autonomous agents, but also as a tool for guiding and coordinating human agents. Finally, Section VII concludes the

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567).

H. Huang, Z. Zhou, J. Ding, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA {haomiao, zhengyuan, jding, tomlin}@eecs.berkeley.edu

W. Zhang is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH, USA zhang@ece.osu.edu

D. M. Stipanović is with the Control and Decision Group, Coordinated Science Laboratory, University of Illinois, Urbana, IL, USA dusan@illinois.edu

paper with a discussion of the algorithm and results.

II. Related Work

The most complete approach to solving pursuit-evasion games is to formulate the problem as a differential game, and then solve the appropriate related Hamilton-Jacobi-Isaacs(HJI) partial differential equation(PDE) [6], [7], [8], [9]. The game is defined by an appropriate value function representing the capture time, with the evader attempting to maximize the time and pursuers attempting to minimize. This value can be computed via a related HJI equation, with appropriate boundary conditions, and this can be used to synthesize trajectories and controls that are optimal with respect to capture time. Solutions are typically found either using the method of characteristics [6], [8], where single trajectories are found by integrating backward from a known terminal condition, or via numerical approximation of the HJI equation on grids [9], [10], [11], [5].

The characteristic solutions are useful in understanding optimal solutions qualitatively, but can be of limited utility in extracting control inputs. The method requires backward integration from a terminal point, which makes it difficult to generate strategies when only the initial state of the agents is known. HJI computation on grids suffers from the curse of dimensionality: computing solutions to HJI equations is computationally infeasible for large problems, as the grid required for approximating the value function grows exponentially as the size of the state space increases.

For certain games and game configurations, it is possible to construct strategies for the agents geometrically. For example, pure pursuit, where a pursuer minimizes the instantaneous distance to the evader, is the optimal single-pursuer strategy in an open environment [6] and is guaranteed to capture the evader in simply-connected regions [12]. Strategies have also been found for coordinating groups of pursuers in open, unbounded spaces [13]. In general, such geometric methods are computationally efficient in generating control strategies, but are limited to relatively simple game environment with no obstacles. Inhomogeneous speed constraints, such as that occasioned by varying terrain, also present a challenge to geometric approaches.

In addition to games played in continuous time on continuous spaces, there has also been research into discrete games played on graphs, with the agents taking turns to move [14], [15]. This work on pursuit-evasion games on graphs has also led to results for games in continuous spaces, particularly for a class of games known as visibility pursuit-evasion [16], [17], [18]. The main difficulty with regards to discrete games is computational complexity. For either pursuit-evasion on graphs or visibility pursuit-evasion, solution strategies are usually found by searching over a large set of discrete actions, limiting the size of problems that can be practically solved, especially when multiple agents are involved on a side.

When the computation required to find optimal or guaranteed strategies is intractable due to either problem complexity or the need for real-time solutions, a form of model-predictive control (MPC) is often employed. In an MPC formulation, an optimization problem is solved over the control actions of one side while using a model to predict opponent actions. This solution is implemented for a short time period, and the optimization is then re-solved using the new agent states. Feedback via this recomputation is used to correct for errors in the prediction model. This strategy has been used for a number of games, for example in complex pursuit-evasion games such as air combat, where the roles of the agents may change over time [19], [20].

In general, MPC approaches work best when the predictive model used is a good approximation of the actual strategy of the opponent. When this is true, control inputs can be quickly and efficiently generated for the controlled agents using standard optimization tools. However, proofs of optimality and guarantees for the solutions are usually not available.

The overview above highlights the trade-offs between optimality (in terms of capture time), guarantees for capture, problem complexity, and computational speed that must be made in control design in adversarial games. Optimal, guaranteed solutions come at the price of either simplified problems, slow computation, or both, while real-time computation usually requires simple problems or loss of optimality and completeness. The research presented here is no exception to this rule. As the ultimate goal of this work is to provide useful, practical solutions, optimality may be sacrificed for computationally efficient solutions that nonetheless possess guarantees with respect to capture. The following sections will discuss the problem formulation and the strategy formulated to meet these needs.

III. The Cooperative Pursuit Problem

Consider a multi-agent pursuit-evasion game involving N pursuers and a single evader, taking place in an open, simply connected region Ω in $\mathbb{R}^2$. Let $x_e \in \mathbb{R}^2$ be the position of the evader and $x_p^i \in \mathbb{R}^2$ be the position of pursuer i. The equations of motion are

$$\begin{aligned} \dot{x_e} &= d,\ x_e(0) = x_e^0, \\ \dot{x_p^i} &= u_i,\ x_p^i(0) = x_p^{i,0},\ i = 1, ..., N, \end{aligned} \tag{1}$$

where d and u_i are the velocity control inputs of the evader and pursuers, respectively, and $x_e^0, x_p^{i,0} \in \Omega$ are the initial evader and pursuer positions. The respective agent inputs are constrained to lie within sets U_i for the pursuers and D for the evader. In this paper, U_i and D are assumed to be the following:

$$D = \{d \mid ||d|| \leq v_{e,max}\},\ U_i = \{u_i \mid ||u_i|| \leq v_{i,max}\}, \tag{2}$$

for some maximum speeds $v_{e,max}$ for the evader and $v_{i,max}$ for each pursuer. The motions of the evader and pursuers, as described by equation (1), are also constrained to lie within the region Ω, with

$$x_e(t),\ x_p^i(t) \in \Omega,\ \forall t \geq 0. \tag{3}$$

Any velocity input $d(t)$ or $u_i(t)$ which satisfies the constraints in equations (2) and (3) is called an admissible input for the evader or pursuer i, respectively.

The goal of the pursuers is to capture the evader by having at least one of the pursuers bring the evader within a distance r_c of the pursuer. Let $C(t)$ be the set of all positions in which the evader is captured at time t, that is

$$C(t) = \{y \mid \exists i, ||y - x_p^i(t)|| \leq r_c\}$$

The capture condition for the pursuers is then given by

$$x_e(t) \in C(t). \tag{4}$$

To achieve this capture condition, each pursuer selects control inputs using a pursuit strategy $\mu_i(x_e, x_p^1, ..., x_p^N)$, based upon observations of the evader and pursuer positions at each time instant, resulting in the closed-loop system dynamics:

$$\begin{aligned} &\dot{x_e} = d,\ x_e(0) = x_e^0, \\ &\dot{x_p}^i = \mu_i(x_e, x_p^1, ..., x_p^N),\ x_p^i(0) = x_p^{i,0},\ i = 1, ..., N \end{aligned} \tag{5}$$

The evader may use some strategy $\gamma(x_e, x_p^1, ..., x_p^N)$ to avoid the pursuers. Any strategy μ_i which satisfies the constraints from equations (2) and (3) is referred to as an admissible pursuit strategy for pursuer i, and similarly for γ. The sets of admissible strategies for the pursuers and evaders are denoted $\mathbb{U}$ and $\mathbb{D}$, respectively.

A precise statement of the problem for the multi-agent pursuit-evasion game can now be given as the following: for any initial configuration $x_e^0,\ x_p^{i,0} \in \Omega$ satisfying $x_e^0 \notin C(0)$, find an admissible choice of pursuit strategy μ_i for each pursuer i such that, regardless of any admissible choice of evader input d, the capture condition (4) is satisfied for some time $t < \infty$.

IV. Pursuit via Safe-Reachable Area Minimization

The pursuit strategy proposed is based on the concept of the safe-reachable set as presented in [1]. The evader's safe-reachable set $\mathcal{S}_e$ is defined as the set of all points in Ω that the evader can directly move to without being captured by a pursuer. The strategy is designed so as to decrease the area of $\mathcal{S}_e$ over time. Intuitively, as this area decreases towards zero, the capture condition will be satisfied.

The definition of the safe-reachable set is briefly summarized below. Given initial conditions $x_e^0,\ x_p^{i,0}$, a point $y \in \Omega$ is *safe-reachable* if there exists some $\gamma \in \mathbb{D}$ and $t \geq 0$ such that $x_e(t) = y$ and $x_e(s) \notin C(s)$ for all $s \in [0, t]$ and all $\mu_i \in \mathbb{U}$. The *safe-reachable set* $\mathcal{S}_e$ of the evader is then defined as

$$\mathcal{S}_e = \{y \in \Omega \mid y \text{ is safe-reachable}\}.$$

Define the minimum time-to-reach function $\varphi\colon \Omega \to \mathbb{R}$ for the evader constrained to $\mathcal{S}_e$:

$$\varphi(y) = \min\{t \mid x_e(t) = y, x_e(s) \in \mathcal{S}_e, \forall s \in [0, t]\}. \tag{6}$$

Similarly, a minimum time-to-capture function $\psi^c\colon \Omega \to \mathbb{R}$ for a pursuer in Ω is defined as:

$$\psi^c(y) = \min\{t \mid y \in C(t), x_p(s) \in \Omega, \forall s \in [0, t]\}. \tag{7}$$

The minimum time-to-capture function represents, for a point y, the minimum time required for the pursuer to capture the evader if the evader were stationary at y. $\mathcal{S}_e$ relative to a single pursuer is therefore

$$\mathcal{S}_e = \{y \mid \varphi(y) < \psi^c(y)\}. \tag{8}$$

For multiple pursuers, the evader must reach a point y before *all* pursuers. Let ψ_i^c be the minimum time-to-capture function for pursuer i. The evader must reach the point y in less time than the *minimum* of all of the time-to-capture functions. Therefore

$$\mathcal{S}_e = \{y \mid \varphi(y) < \psi_i^c(y), \forall i\}.$$

Observe that the definitions of $\mathcal{S}_e$ and φ are interrelated. Computing $\mathcal{S}_e$ requires simultaneously computing the set $\mathcal{S}_e$ and the values of φ within that set. When the pursuer and evader speeds are equal, the safe-reachable set is equivalent to the generalized Voronoi decomposition of the agents [2]. In more complex situations the appropriate time-to-reach values can be computed using modified fast marching methods (FMM) as described in [1]. FMM [21], [22], [23] is a single-pass method used to numerically approximate the *Eikonal equation*, which for the agent dynamics described here can be used to compute the time-to-reach and time-to-capture functions.

It should be noted that $\mathcal{S}_e$ is defined in the open-loop sense, in that a point in $\mathcal{S}_e$ can be reached by the evader moving directly toward that point and ignoring the actions of the pursuers. Furthermore, the computational efficiency of the proposed control strategy directly results from the fact that $\mathcal{S}_e$ is defined in the open-loop sense. On the other hand, the feedback nature of the strategy comes from the receding horizon implementation of the open-loop controls. Namely, the minimization of the open-loop safe reachable is carried out at each time instant, given the current measurement of system state. The game is still played in the closed-loop sense in that the pursuers react to the movements of the evader (and the evader is presumably reacting to the pursuers). $\mathcal{S}_e$ is used by the pursuers to guide their pursuit strategy but does not necessarily affect the actions of the evader itself.

Since $\mathcal{S}_e$ depends only on the position of the agents, the area A of $\mathcal{S}_e$ depends only on the locations of the pursuers relative to the evader. The time derivative of A is given by

$$\frac{dA}{dt} = \frac{\partial A}{\partial x_e}\dot{x_e} + \sum_{i=1}^{N} \frac{\partial A}{\partial x_p^i}\dot{x_p^i} \tag{9}$$

Now consider a cooperative pursuit strategy that jointly minimizes $\frac{dA}{dt}$. According to equation (9), this joint objective can be decoupled into the individual objectives of minimizing $\frac{\partial A}{\partial x_p^i}\dot{x_p^i}$ for each pursuer i. A pursuer i is said to share an *active boundary* with the evader if there is a portion of the boundary of $\mathcal{S}_e$ where $\psi_i^c(y) < \psi_j^c(y)$ for all points y in this portion and all other pursuers j. In other words, there is a portion of the boundary of $\mathcal{S}_e$ that is defined purely by the position of pursuer i. This is analogous to having a shared Voronoi boundary with the evader in a Voronoi decomposition.

Let N_e be the set of all such pursuers that share an active boundary with the evader. The area A only depends on the pursuers in N_e, so $\frac{\partial A}{\partial x_p^i} = 0$ for all $i \notin N_e$. Any pursuer i which does not share an active boundary with the evader may simply use the pure pursuit strategy. On the other hand, for each pursuer $i \in N_e$, the choice of pursuit strategy which minimizes equation (9) is given by:

$$u_i^* = \mu_i^*(x_e, x_p^1, \ldots, x_p^N) \triangleq -v_{i,max} \frac{\frac{\partial A}{\partial x_p^i}}{||\frac{\partial A}{\partial x_p^i}||}.$$

Computing $\frac{\partial A}{\partial x_p^i}$ for each pursuer depends on the geometry of the game domain Ω and the relative speeds of the agents. An analytic solution is found below for convex domains with equal agents' speeds, and a more general algorithm based on fast marching methods is presented for non-convex domains with unequal agent speeds.

A. Convex Domains with Equal Speeds

For games played in convex domains with agents of equal speeds, the safe-reachable set of the evader is equal to the evader's Voronoi partition. Let $S(D) = \{\mathcal{S}_e, S_1, \ldots, S_N\}$ be the Voronoi partition of Ω generated by the points $\{x_e, x_p^1, \ldots, x_p^N\}$:

$$\mathcal{S}_e = \{y \in \Omega \mid \|y - x_e\| < \|y - x_p^i\|, \forall i \leq N\},$$
$$S_i = \{y \in \Omega \mid \|y - x_p^i\| \leq \min\{\|y - x_e\|, \|y - x_p^j\|\}, \forall j \neq i\}, i \leq N$$

The edge shared by $\mathcal{S}_e$ and S_i, $i \in N_e$ is called the *line of control* for pursuer i and is denoted by B_i, where L_i is the length of B_i (see Figure 1). It can be observed that by appropriate re-scaling of the dynamics in equation (5), the region Ω, and the capture radius r_c, it is sufficient to consider this problem for the case where $v_{e,max} = v_{i,max} = 1$. Thus, for the rest of the convex, equal-speed analysis, it is assumed without loss of generality that $v_{e,max} = v_{i,max} = 1$ in equation (2).

1) Convex pursuit strategy: With a convex game domain and equal agent speeds, an analytic expression for the pursuer strategies $\mu_i^*, i \in N_e$ can be found. Moreover, it will be shown that the analytic expression is nicely captured by a simple description of the control strategy, namely that pursuer i shall always head towards the midpoint of the line of control B_i if it has such a line of control.

The construction of this pursuit strategy proceeds using a particular choice of local coordinate system. First, let $\boldsymbol{\xi}_i(x_e, x_p) = x_e - x_p^i$ be the displacement vector pointing from the location of pursuer i towards the location of the evader. When there is no ambiguity, its arguments will be omitted and this vector will be denoted simply by $\boldsymbol{\xi}_i$. Denote δ_{min} to be the minimum of $||\boldsymbol{\xi}_i||$ as i ranges from 1 to N. As $\delta_{min}(0) > r_c$ until capture, at which $\delta_{min}(T) = r_c$, $||\boldsymbol{\xi}_i|| \geq r_c$ for all $i \leq N$ and $t \in [0, T]$. Define $\boldsymbol{\eta}_h^i = \frac{\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}\|_i}$ and let $\boldsymbol{\eta}_v^i \in \mathbb{R}^2$ be a unit vector orthogonal to $\boldsymbol{\eta}_h^i$, as shown in Figure 1(a). The vectors $\{\boldsymbol{\eta}_h^i, \boldsymbol{\eta}_v^i\}$ define a local coordinate system that depends on the locations of x_e and x_p^i. For any $y \in \mathbb{R}^2$ and $(x_e, x_p^1, ..., x_p^N)$ such that $y + x_p^i \in \Omega$, define

$$A_i^+(y)|_{(x_e, x_p^1, \ldots, x_p^N)} = A(x_e, x_p^1, ..., x_p^i + y, ..., x_p^N).$$

Define $D_h^i A$ and $D_v^i A$ as the directional derivatives of A along $\boldsymbol{\eta}_h^i$ and $\boldsymbol{\eta}_v^i$, then

$$\begin{cases} D_h^i A|_{(x_e, x_p^1, \ldots, x_p^N)} = \lim_{\epsilon \to 0} \frac{A_i^+(\epsilon \cdot \boldsymbol{\eta}_h^i)|_{(x_e, x_p^1, \ldots, x_p^N)} - A}{\epsilon} \\ D_v^i A|_{(x_e, x_p^1, \ldots, x_p^N)} = \lim_{\epsilon \to 0} \frac{A_i^+(\epsilon \cdot \boldsymbol{\eta}_v^i)|_{(x_e, x_p^1, \ldots, x_p^N)} - A}{\epsilon}, \end{cases} \tag{10}$$

where $A(x_e, x_p^1, ..., x_p^N)$ is denoted by A for brevity. From this expression, the partial derivative of A with respect to x_p^i is given by

$$\frac{\partial A}{\partial x_p^i} = D_h^i A \cdot \boldsymbol{\eta}_h^i + D_v^i A \cdot \boldsymbol{\eta}_v^i. \tag{11}$$

Lemma 1. *For any $i \in N_e$, it is true that*

$$D_h^i A = -\frac{L_i}{2},$$
$$D_v^i A = \frac{l_i^2 - (L_i - l_i)^2}{2||\boldsymbol{\xi}_i||},$$

where L_i is the length of the line of control B_i and l_i is the length of the segment of B_i on the side of the intersection of $\boldsymbol{\xi}_i$ with B_i opposite to $\boldsymbol{\eta}_v^i$, as shown in Figure 2.

Proof: **Perturbation along $\boldsymbol{\eta}_h^i$:** A perturbation ϵ in the pursuer's position toward the evader moves the line of control $\frac{\epsilon}{2}$ toward the evader, and generates a corresponding change in the area of the evader's Voronoi cell δA_h^i, as shown in Figure 2(a). This change in area is

$$\delta A_h^i = -\frac{L_i \epsilon}{2} + O(\epsilon^2),$$

where the $O(\epsilon^2)$ term depends on the angle of intersection between B_i and the boundaries of the Voronoi cell $\mathcal{S}_e$. From this expression, the directional derivative of A along $\boldsymbol{\eta}_h^i$ can be calculated as

$$D_h^i A = \lim_{\epsilon \to 0} \frac{\delta A_h^i}{\epsilon} = -\frac{L_i}{2}.$$

Perturbation along $\boldsymbol{\eta}_v^i$: There are two different scenarios for perturbation along $\boldsymbol{\eta}_v^i$, corresponding to the two pursuer configurations shown in Figure 1(b). In one case, as for x_p^2 in Figure 1(b), $\boldsymbol{\xi}_i$ intersects B_i. A perturbation of ϵ, as shown in Figure 2(b), will cause the evader's Voronoi cell to shrink above the new intersection by $\delta A_{v,1}^i$ and grow below it by $\delta A_{v,2}^i$. Let $\delta A_v^i = \delta A_{v,2}^i - \delta A_{v,1}^i$, with

$$\delta A_{v,1}^i = \frac{1}{2}((L_i - l_i) - \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\boldsymbol{\xi}_i\|} + O(\epsilon^2),$$
$$\delta A_{v,2}^i = \frac{1}{2}(l_i + \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\boldsymbol{\xi}_i\|} + O(\epsilon^2),$$

where the terms $O(\epsilon^2)$ again depend on the angle of intersection between B_i and the boundaries of the Voronoi cell $\mathcal{S}_e$. Thus, the resulting changes in area will be

$$\delta A_{v,1}^i = \frac{(L_i - l_i)^2 \epsilon}{2||\boldsymbol{\xi}_i||} + O(\epsilon^2)$$
$$\delta A_{v,2}^i = \frac{l_i^2 \epsilon}{2||\boldsymbol{\xi}_i||} + O(\epsilon^2)$$

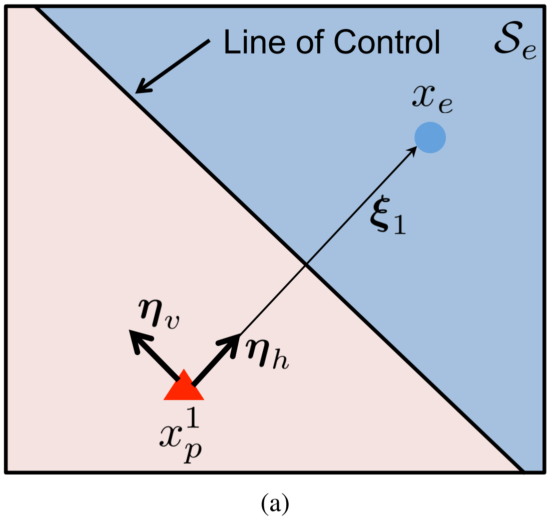


Fig. 1. **Illustrations showing the evader's safe-reachable set $\mathcal{S}_e$ in a convex domain with equal agent speeds (a) for a single pursuer and evader and (b) with an additional pursuer.**

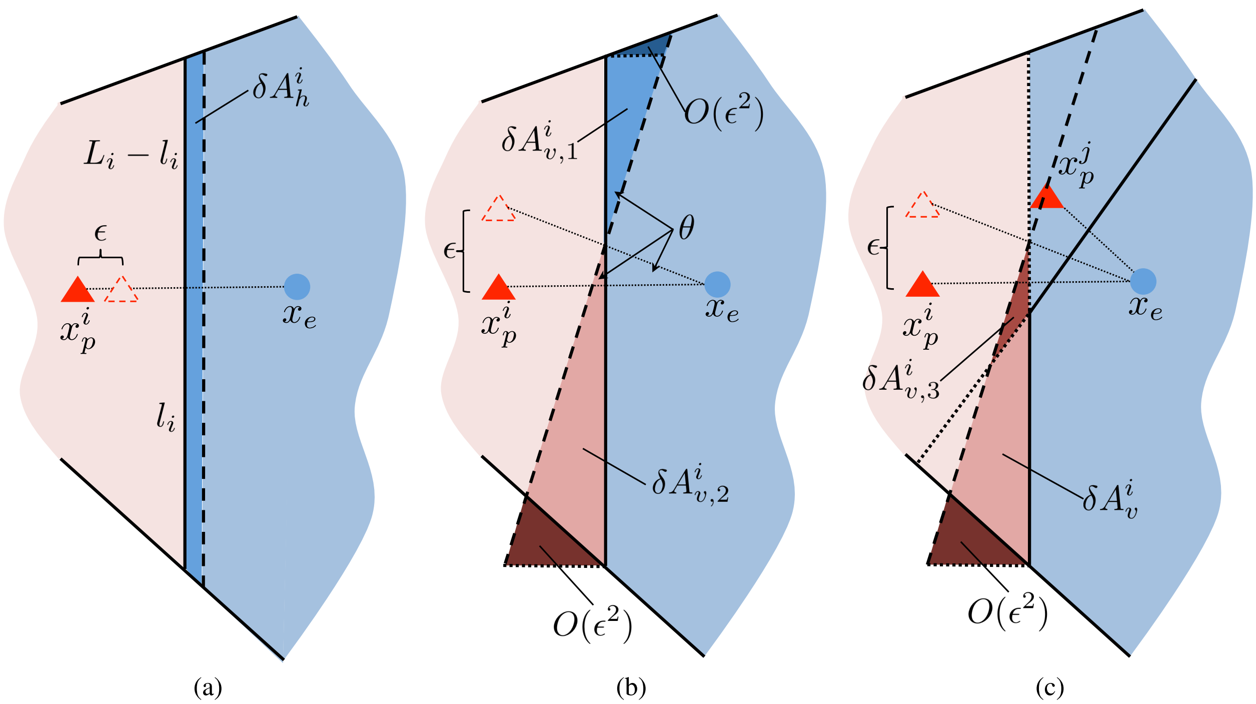


Fig. 2. **Variational change in area of the evader's safe-reachable set with respect to (a) a perturbation toward the evader, (b) perturbation parallel to the line of control, and (c) when another pursuer is present and $\boldsymbol{\xi}_i$ no longer intersects B_i, as in Figure 1(b).**

which implies

$$D_v^i A = \lim_{\epsilon \to 0} \frac{\delta A_v^i}{\epsilon} = \frac{l_i^2 - (L_i - l_i)^2}{2||\boldsymbol{\xi}_i||}.$$

The second case is that of x_p^1 in Figure 1(b), where $\boldsymbol{\xi}_i$ no longer intersects B_i due to the presence of other pursuers. As shown in Figure 2(c), the change in area is

$$\delta A_v^i = \delta A_{v,2}^i - \delta A_{v,3}^i,$$

where $\delta A_{v,2}^i$ is calculated as before and

$$\delta A_{v,3}^i = \frac{1}{2}(l_i - L_i + \frac{\epsilon}{2})^2 \frac{\epsilon}{\|\boldsymbol{\xi}_i\|} + O(\epsilon^2).$$

Note that here $l_i \geq L_i$. Letting $\epsilon \to 0$ then again it is true that

$$D_v^i A = \lim_{\epsilon \to 0} \frac{\delta A_v^i}{\epsilon} = \frac{l_i^2 - (l_i - L_i)^2}{2||\boldsymbol{\xi}_i||}.$$

■

With the above lemma, the proposed strategy μ_i^* can be rewritten in the local coordinate system as

$$\mu_i^* = -\left(\frac{\alpha_h^i}{\sqrt{|\alpha_h^i|^2 + |\alpha_v^i|^2}} \cdot \boldsymbol{\eta}_h^i + \frac{\alpha_v^i}{\sqrt{|\alpha_h^i|^2 + |\alpha_v^i|^2}} \boldsymbol{\eta}_v^i \right), \tag{12}$$

where α_h^i and α_v^i are given by

$$\alpha_h^i = -\frac{L_i}{2}, \ \alpha_v^i = \frac{l_i^2 - (L_i - l_i)^2}{2||\boldsymbol{\xi}_i||}.$$

Lemma 2. *It can be shown that under this choice of pursuit strategy, u_i always points toward the interior of Ω, thus satisfying the constraint from equation* (3)*.*

The proof is straightforward but requires some amount of algebra and is thus omitted.

Using the above results, the pursuit strategy can be shown to take the following form:

Theorem 3. *Under the proposed pursuit strategy, pursuer i should always head for the midpoint of the line of control B_i.*

Proof: From Lemma 1, the control input is the vector $(D_h^i A, D_v^i A)$ in the local coordinate system defined by $\{\boldsymbol{\eta}_h^i, \boldsymbol{\eta}_v^i\}$. Let α be the angle between the velocity input and the local horizontal axis defined by $\boldsymbol{\eta}_h^i$. It is true that

$$tan(\alpha) = \frac{D_v^i A}{D_h^i A} = \frac{\frac{l_i^2 - (L_i - l_i)^2}{2||\boldsymbol{\xi}_i||}}{-\frac{L_i}{2}} = -\frac{2l_i - L_i}{||\boldsymbol{\xi}_i||}; \quad (13)$$

Let β be the angle between the local horizontal axis and the vector from pursuer i to the midpoint of B_i. Then

$$tan(\beta) = \frac{l_i - (L_i/2)}{||\boldsymbol{\xi}_i||/2} = -\frac{2l_i - L_i}{||\boldsymbol{\xi}_i||}; \quad (14)$$

Therefore $\alpha = \beta$, thus establishing the theorem. ■

2) Proof of guaranteed capture: The pursuit strategy outlined above for the convex, equal speed game is guaranteed to capture the evader in finite time, regardless of any admissible evader input d. It can be seen that if this holds for the case of a single pursuer ($N = 1$), then the conclusion also extends to the case of multiple pursuers ($N > 1$). Indeed, for the case of $N > 1$, one can choose any pursuer i which is a Voronoi neighbor of the evader and use the arguments for the case of $N = 1$ to show that the capture condition will be satisfied. This section presents the proof for a single pursuer. Correspondingly, the notation from above will carry through without the indices i.

First, observe that as A approaches zero, the evader's Voronoi cell approaches either a line or a point. Either of the two cases clearly implies $\|x_e - x_p\|_2 \to 0$. The strategy here is then to show that, under the proposed pursuit strategy μ^* and any admissible evader control input d, either the area A or the distance δ is guaranteed to decrease until the capture condition is met.

In terms of preliminaries, by Lemma 1,

$$\frac{\partial A}{\partial x_p} = \alpha_h \boldsymbol{\eta}_h + \alpha_v \boldsymbol{\eta}_v.$$

It can be also verified in a similar manner as the proof of Lemma 1 that the partial derivative $\frac{\partial A}{\partial x_e}$ in the local coordinate system is given by

$$\frac{\partial A}{\partial x_e} = \alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v. \quad (15)$$

Also recall that the variable L in the statement of Lemma 1 depends on the spatial locations of the evader and the pursuer, as well as the geometry of the region Ω. For this proof, make the following definitions of parameters $l_{\min}$ and $l_{\max}$, which depend solely on the geometry of Ω:

$$\begin{cases} l_{\min} = \inf_{x_e \in \Omega, x_p \in \Omega} L(x_e, x_p) \\ l_{\max} = \sup_{x_e \in \Omega, x_p \in \Omega} ||x_e - x_p||. \end{cases} \quad (16)$$

Since Ω is bounded and the game ends upon capture, it is true that $l_{\min} > r_c$, $l_{\max} < \infty$, and $L \leq l_{max}$.

The following result shows that the area A is always non-increasing under the pursuit strategy μ^* for a single pursuer.

Lemma 4. *Under the proposed pursuit strategy $\mu^*(x_e, x_p)$, the area A satisfies $\frac{dA}{dt} \leq 0$ for any admissible evader control input. Furthermore, $\frac{dA}{dt} = 0$ if and only if the evader follows the following strategy:*

$$\gamma^*(x_e, x_p) = \frac{\alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v}{\sqrt{\alpha_h^2 + \alpha_v^2}}. \quad (17)$$

Proof: For an arbitrary d with $\|d\| \leq 1$:

$$\begin{aligned} \frac{dA}{dt} =& \frac{\partial A}{\partial p} \mu^*(x_e, x_p) + \frac{\partial A}{\partial e} d \\ =& -\sqrt{\alpha_h^2 + \alpha_v^2} + (\alpha_h \boldsymbol{\eta}_h - \alpha_v \boldsymbol{\eta}_v)^T d \leq 0, \end{aligned}$$

where equality holds if and only if $d(t) = \gamma^*(x_e(t), x_p(t))$. ■

To prove that the capture condition is achieved in finite time, it is necessary to show that the distance between the pursuer and the evader is strictly decreasing whenever the area A is constant. For this purpose, define

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

Clearly, the variable z is the squared Euclidean distance between the evader and pursuer. From the preceding discussions, the range of z lies in $[r_c^2, \ l_{\max}^2]$. The following result shows that $\dot{z} < 0$ whenever $\dot{A} = 0$.

Lemma 5. *If $\dot{A} = 0$, then under the pursuit strategy μ^*, the following holds:*

$$\frac{dz}{dt} = -\frac{4z}{\sqrt{z + (l_{max} - l_{min})^2}} \leq \frac{-4r_c^2}{\sqrt{r_c^2 + l_{\max}^2}}.$$

Proof: By Lemma 4, $\dot{A}(t) = 0$ if and only if $d(t) = \gamma^*(x_e(t), x_p(t))$. Thus, if the pursuer input is selected according to the strategy μ^*, then whenever $\dot{A} = 0$, then

$$\begin{aligned} \dot{z} =& 2(x_e - x_p)^T (\dot{x_e} - \dot{x_p}) \\ =& 4\boldsymbol{\xi}^T \left(\frac{\alpha_h}{\sqrt{\alpha_h^2 + \alpha_v^2}} \boldsymbol{\eta}_h \right) \\ =& \frac{-2L\|\boldsymbol{\xi}\|}{\sqrt{\frac{L^2}{4} + \frac{(l^2 - (L-l)^2)^2}{4\|\boldsymbol{\xi}\|^2}}} \\ =& -\frac{4z}{\sqrt{z + (2l - L)^2}} \\ \leq& \frac{-4r_c^2}{\sqrt{r_c^2 + l_{\max}^2}}, \end{aligned}$$

where the second equality follows from the fact that $\boldsymbol{\xi}^T\boldsymbol{\eta}_v = 0$, and the last inequality follows from the monotonicity of the function $\frac{4z}{\sqrt{z+(2l-L)^2}}$ for $z \geq 0$. ■

By this result, z is strictly decreasing whenever the area A remains constant. However, there remains the possibility that z is increasing on time intervals where A is strictly decreasing. The question then becomes whether there exists an evader control that can keep z inside $[r_c^2,\ l_{\max}^2]$ while preventing A from reaching 0. The following result proves that this is not the case, by exploiting certain properties of the proposed pursuit strategy.

Lemma 6. *Under the pursuit strategy μ^*, if $\dot{A} \geq -\beta$ for some positive constant $\beta > 0$, then $\dot{z} \leq -c(\beta)$, where the bound $c(\beta)$ is given by*

$$c(\beta) = \frac{\sqrt{2}r_c^2}{l_{\max}} - \frac{4l_{\max}}{l_{\min}}\beta. \tag{18}$$

Proof: Under strategy μ^*, the following identities hold

$$\begin{cases} \dot{A} = -\sqrt{\alpha_h^2+\alpha_v^2} + (\alpha_h\boldsymbol{\eta}_h - \alpha_v\boldsymbol{\eta}_v)^T d \\ \dot{z} = 2(x_e-x_p)^T d - \frac{2z}{\sqrt{z+(2l-L)^2}} \end{cases}$$

Now suppose $\dot{A} \geq -\beta$. From the relations $\boldsymbol{\eta}_h = \frac{x_e-x_p}{\|x_e-x_p\|}$, $\alpha_h = -\frac{L}{2}$, and $\alpha_v\boldsymbol{\eta}_v^T d \geq -|\alpha_v|$, it is true that

$$\begin{aligned}(x_e-x_p)^T d &\leq -\frac{2\|x_e-x_p\|}{L}\left[\sqrt{\alpha_h^2+\alpha_v^2} - \beta - |\alpha_v|\right] \\ &\leq \frac{2\|x_e-x_p\|\beta}{L} \leq \frac{2l_{\max}}{l_{\min}}\beta,\end{aligned}$$

which implies that

$$\dot{z} \leq \frac{4l_{\max}}{l_{\min}}\beta - \frac{\sqrt{2}r_c^2}{l_{\max}}.$$

■

Notice that this lemma also implies $\dot{A} < -\beta$ whenever $\dot{z} > -c(\beta)$. Now the previous results in this section will be combined to show that under μ^*, the area A or the distance δ decreases until the capture condition is met.

Consider an "energy" function E

$$E = kA + z$$

for a positive constant k (to be defined subsequently). Clearly, $E = 0$ if and only if $A = 0$ and $z = 0$, both of which imply that capture occurs. A proof will now be presented that E will decline to zero as t increases.

Theorem 7. *Under the pursuit strategy μ^*, if the capture condition has not been achieved before time $t > 0$, then for some positive constants $k,\ \beta > 0$*

$$E(t) \leq E(0) - c(\beta)t$$

where $E(0)$ is the initial energy and $c(\beta)$ is defined as in (18).

Proof: First, it is assumed that the β parameter in Lemma 6 is chosen such that $c(\beta) > 0$. Lemmas 5 and 6 imply that one of the following conditions must be true at any given time:

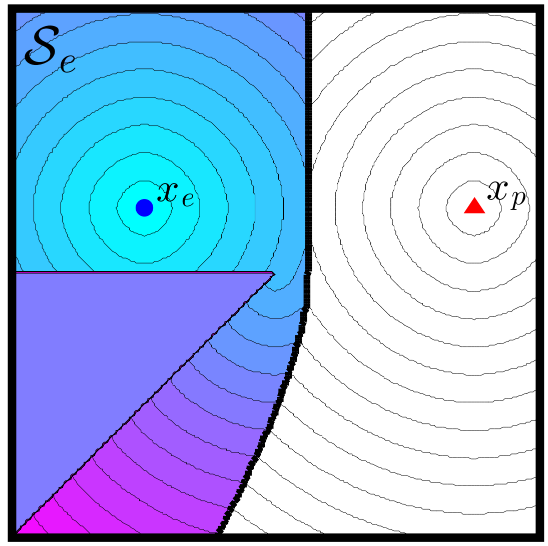


Fig. 3. **An example of safe-reachable set computed in a non-convex game domain, with contours plotted for equal time-to-reach values for each agent.**

1) $\dot{A} \geq -\beta$ and $\dot{z} \leq -c(\beta)$, or
2) $\dot{z} > c$ and $\dot{A} < -\beta$

Note that the derivative of E is

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

and $\dot{A} \leq 0$ for all time. Then, under condition 1, $\dot{A} \leq 0$ and $\dot{z} \leq -c(\beta)$, thus $\dot{E} \leq -c(\beta)$. The rate of change of z is limited by the maximum speed of the two agents and the geometry of the domain. Since $\dot{z} = 2\delta\dot{\delta}$, $\delta \leq l_{max}$, and $\dot{\delta} \leq 2$, then $\dot{z} \leq 4l_{max}$. Now, let $k = \frac{4l_{max}+c(\beta)}{\beta}$. Under condition 2, $\dot{A} < -\beta$ and of course $\dot{z} \leq 4l_{max}$, thus $\dot{E} \leq -k\beta + 4l_{max}$, therefore again $\dot{E} \leq -c(\beta)$, guaranteeing that the energy will decrease to 0 in finite time, leading to capture. ■

B. Non-Convex Domains and Unequal Speeds

For non-convex domains and unequal speeds, it is more difficult to construct the safe-reachable set geometrically. Instead, the modified fast-marching method (FMM) presented in [1] can be used to compute the safe-reachable set on a grid. The area can then be approximated using the grid and the gradient computed numerically.

The details of the modified FMM algorithm can be found in [1] and will not be repeated here. A sketch of the algorithm is as follows: first, the standard FMM algorithm is used to compute the time-to-capture $\psi_i^c(y_{j,k})$ for every node $y_{j,k}$ on the grid. The modified FMM is then used to compute the safe-reachable set $\mathcal{S}_e$, beginning with the current position of the evader x_e, and successively adding points that can be reached in time $\varphi(y_{j,k})$ with $\varphi(y_{j,k}) < \psi_i^c(y_{j,k}), \forall i$. An example of the computation performed for a non-convex region with a triangular obstacle and point capture ($r_c = 0$) is shown in Figure 3, with equal time-to-reach values plotted for the two agents. Note that the boundary of $\mathcal{S}_e$ is no longer a straight line, since the evader must first move to the corner of the

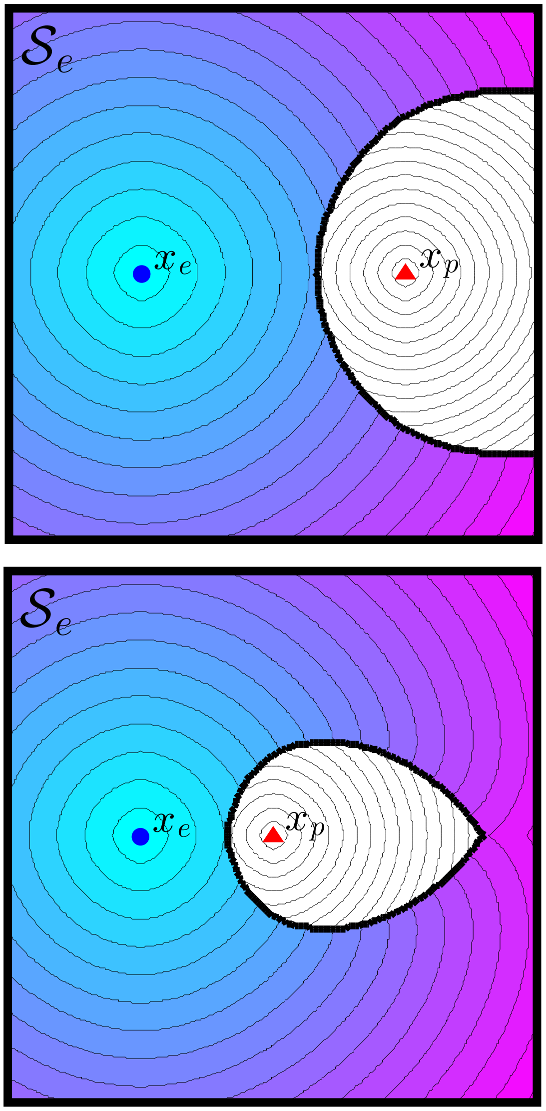


Fig. 4. **An illustration of a scenario where the pursuer (red triangle) is slower than the evader (blue circle). Note how $\mathcal{S}_e$ grows larger as the pursuer gets closer to the evader.**

obstacle in order to reach points in the lower portion of the game domain.

Let N_s be the number of grid nodes for which $\varphi(y_{j,k}) < \psi_i^c(y_{j,k}), \forall i$. The area of $\mathcal{S}_e$ can be approximated as

$$A \approx N_s h^2$$

where h is the grid spacing. The gradient with respect to pursuer movements can be numerically approximated by perturbing each pursuer by h horizontally and vertically on the grid and recomputing A. The pursuit strategy is identical, with

$$u_i^* = \mu_i^*(x_e, x_p^1, \ldots, x_p^N) \triangleq -v_{i,max} \frac{\frac{\partial A}{\partial x_p^i}}{||\frac{\partial A}{\partial x_p^i}||}.$$

There are two things to note about using safe-reachable area pursuit in non-convex domains with unequal speeds. The first is that the algorithm is only effective in cases where the pursuers are at least as fast as the evader. This is due to the fact that, if the pursuer is slower than the evader, the evader's safe-reachable set actually shrinks in area as the pursuer goes farther from the evader, as shown in Figure 4. Thus area minimization will likely not lead to capture.

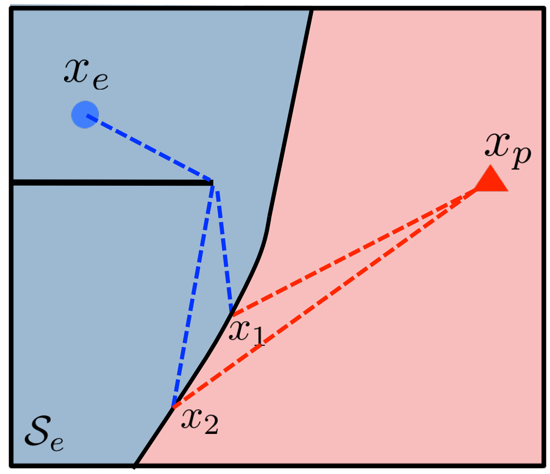
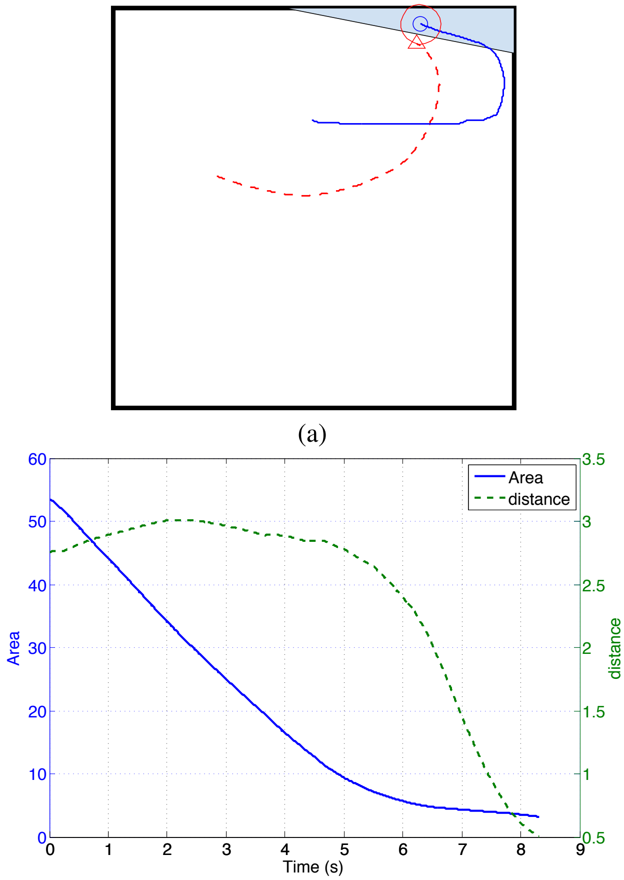


Fig. 5. **Illustration of the asymmetry between the evader and the pursuer when the domain is non-convex.**

A second point is that currently no proof has been found to guarantee capture in the non-convex case, although the area minimization strategy works well empirically. The major obstacle in this case is that the game is no longer symmetric with respect to the area of $\mathcal{S}_e$. This is illustrated in Figure 5, which shows the safe-reachable set for an evader that must turn around a non-convex obstacle. This configuration gives an advantage to the evader, as it moves in the same direction whether it is headed for the point x_1 or x_2. Thus, if the evader were to move distance ϵ toward x_1, it has also decreased its distance to x_2 by ϵ. If the pursuer moves to maintain x_1 on the boundary of the safe-reachable set, then it can move ϵ toward x_1, but it will have moved some distance less than ϵ toward x_2, and x_2 will have become part of the evader's safe-reachable set. In this manner, the evader is able to increase the area of its safe-reachable set regardless of the pursuer input.

V. Simulation Results

A number of simulations were conducted to evaluate the proposed safe-reachable area minimization pursuit strategy. The performance of the pursuit strategy was compared in a sequence of trials against two other pursuit algorithms: pure pursuit, where each pursuer instantaneously minimizes the distance between itself and the evader, and a numerical approximation of the optimal Hamilton-Jacobi-Isaacs solution on a grid. The pure-pursuit strategy was chosen for comparison purposes because it is straightforward to implement on arbitrarily defined game domains and is the optimal pursuit strategy for open domains with no boundaries. It is also guaranteed to result in capture for closed, simply connected domains [12]. Several simulation examples will be used to highlight some qualitative properties of the safe-reachable area pursuit strategy, and then the quantitative results of the numerical trials will be presented.



(b)

Fig. 6. **Simulation results for a single pursuer (triangle, dashed line) and evader (circle, solid line), showing (a) the trajectories and (b) the area of $\mathcal{S}_e$ and the distance between the agents over time.**

A. Illustrative Examples

A few simulations are presented here to highlight some qualitative aspects of the safe-area pursuit algorithm. The first set of simulations were conducted in a 10 x 10 square using the convex, analytic pursuit algorithm. The maximum speed was 1 for all agents, with capture radius of 0.5, and time steps of 0.01. In these simulations the trajectory of the evader was controlled by human input, and pursuers that did not have a line of control bordering on the evader's safe-reachable set were commanded to head straight for the evader.

An example trajectory for a game with a single pursuer is shown in Figure 6. The critical trade-off between area and distance is highlighted by this example. Note that initially the pursuer did not move directly toward the evader, and thus the distance between the agents did not decrease, but the area decreased very quickly. Near the end of the game the area decreased slowly while the distance decreased very quickly.

Figure 7 shows a comparison between the pure pursuit strategy and safe-reachable area pursuit for a scenario with 3 pursuers and highlights the cooperation in this pursuit strategy. Pure pursuit is shown in Figure 7(a), and safe-reachable area pursuit is shown in Figure 7(b). The pursuers began closely grouped, and in pure pursuit they acted independently, resulting in a prolonged chase. With the safe-reachable area pursuit strategy, the pursuers gradually separated to surround the evader. The cooperative behavior effectively contained the evader, limiting its movements until capture was achieved.

Pursuit in a non-convex environment is shown in Figure 8, which shows 2 pursuers (red triangles) pursuing a single evader (blue circle) in a simple, non-convex region with a triangular obstacle. The evader's safe-reachable set $\mathcal{S}_e$ is shown at each time, with equal time-to-reach contours plotted within $\mathcal{S}_e$.

The simulations were conducted in Matlab on a Macbook Pro laptop with a 2.2 Ghz Intel Core i7 processor with 8 GB of RAM, with computation per time-step of less than 1ms to calculate inputs for all the pursuers in the analytic, convex case, and about 100ms for each pursuer using FMM. Note that some small errors are introduced by discretization of the control scheme when distances between the evader and pursuers are comparable to the distance traveled by an agent in a single time step. Reducing the time step alleviates the problem without eliminating it entirely, and increasing the capture radius also reduces the chance of this problem occurring. It is possible that some relationship can be found between step size, velocity, and the capture radius to formally guarantee this in a discrete-time situation.

B. Comparison Tests

The results of the comparison tests conducted to evaluate the safe-reachable area pursuit strategy will now be presented. Two groups of trials were conducted. The first set of trials matched safe-reachable area pursuit, pure pursuit, and the numerical HJI strategy against each other in tests with one pursuer and one evader. For these tests, a numerical approximation to the HJI equation was computed on a 40 x 40 grid for a simple non-convex region, shown in Figure 9(b), and the pursuer and evader strategies were evaluated by numerical differentiation. The three different pursuit strategies were then evaluated against the approximate optimal evader strategy for 500 initial conditions generated randomly.

The results of this test are displayed in Figure 10. In general, the area-minimization strategy performed slightly worse than the numerical HJI pursuit strategy, and the pure pursuit strategy performed typically the worst among the three . However, it should be noted that the numerical HJI strategy depends on numerical differentiation of the approximated value function on a grid, and numerical errors can lead to sub-optimal pursuer and evader actions. For example, in 29% of trials the area minimization pursuit strategy out-performed the "optimal" HJI pursuer, and similarly in 8.5% of trials the pure pursuit strategy resulted in faster capture-times.

In addition to the numerical issues discussed above, due to computational complexity, the HJI solution can only be found for the case of a single pursuer. To evaluate the performance of the pursuit strategy with multiple pursuers, a further series of tests were conducted comparing safe-reachable area pursuit with pure pursuit. For these tests, the evader strategy was defined as the following: the evader selects as its target point y^* the farthest point from itself in $\mathcal{S}_e$, that is

$$y^* = \max_{y \in \mathcal{S}_e} \delta_g(y, x_e)$$

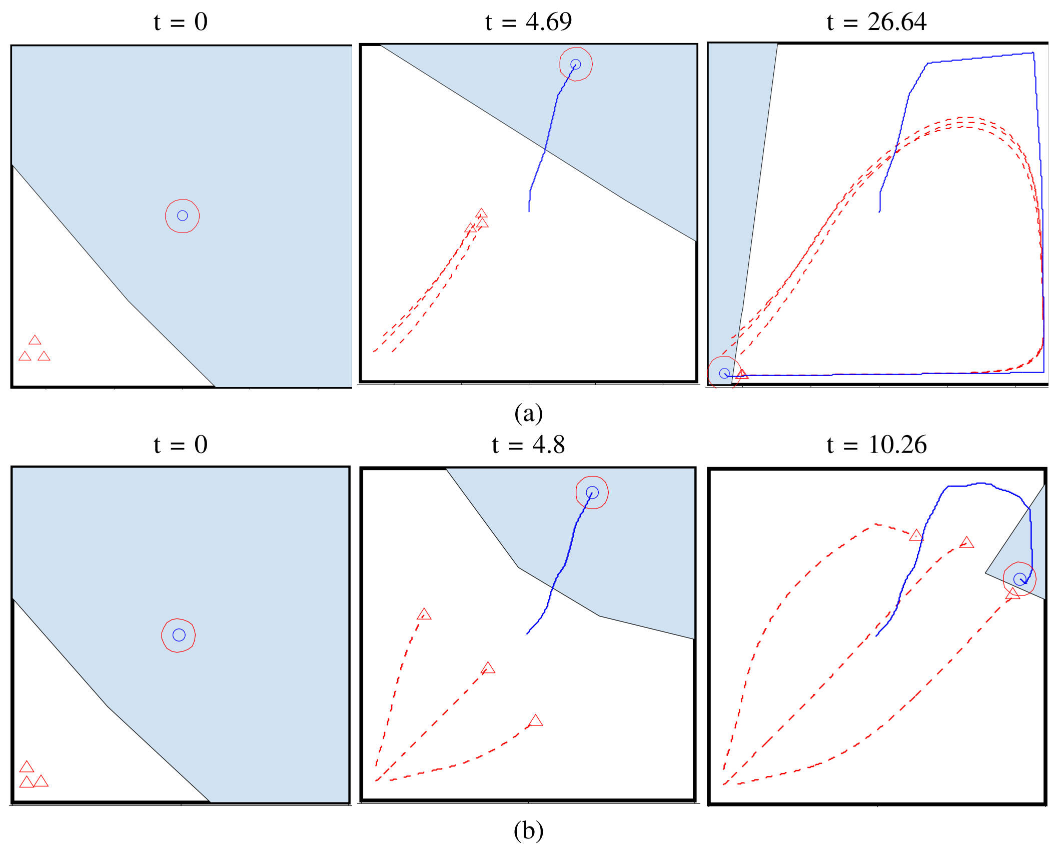


Fig. 7. **A scenario with 1 evader (blue circle) and 3 pursuers (triangles, dotted lines) using (a) pure pursuit, and (b) safe-reachable area minimization, highlighting the cooperation enforced by the safe-reachable area minimization pursuit strategy when the pursuers begin tightly spaced.**

where δ_g is the geodesic distance between y and x_e. Once the evader reaches a certain distance (set here as half of the capture radius) from y^*, a new target is selected and the evader will proceed toward this target.

Three sets of trials were conducted with 1, 2, and 3 pursuers, with one set in a square, convex region using the analytically derived pursuit formula and two others in non-convex regions using FMM, shown in Figure 9. For each set, 500 random initial conditions of pursuer and evader positions were generated. The results of the tests are summarized in Figure 11, showing histograms of the difference in time between trials, defined as the time required for the safe-reachable area strategy minus the time required for pure pursuit for each initial condition.

Table I shows the fraction of trials in each case where the area pursuit strategy resulted in faster capture times than pure pursuit. Figure 11(a) shows the distribution of results in the convex environment. In this scenario, the safe-reachable area pursuit strategy resulted in clearly superior performance, with the vast majority of trials resulting in faster capture times. The distribution of times seems somewhat bimodal in these trials, with a number of trials where safe-reachable area minimization pursuit and pure pursuit performed similarly, and then a large group where the safe-reachable area minimization pursuit strategy was clearly superior.

The results for the non-convex scenarios are shown in Figure 11(b) for the simple non-convex environment, and in Figure 11(c) for the complex non-convex environment. The simple non-convex case still resulted in a large majority of trials where the safe-reachable area minimization pursuit strategy gave faster capture, although in a smaller percentage of trials than the convex scenario. The complex non-convex case showed a decline in the performance of the safe-reachable area minimization strategy relative to the pure pursuit strategy. This is due to the fact that the obstacles create areas where the width of the free space is of similar size to the capture radius, thus the pure pursuit can still "trap" the evader, lessening the advantage conferred by the safe-reachable area strategy. In fact, it is to be expected that as the space becomes more and more similar to a single long, narrow corridor, the pure pursuit and area-pursuit strategies should have identical performance. This is especially evident in the trials with only 1 pursuer, where only about 50% of trials resulted in faster capture with safe-reachable area pursuit, with a long tail of trials where the area pursuit performed much worse than pure pursuit. These typically occurred in trials where the evader was able to escape from a confined portion of the game domain against the safe-reachable area minimizing pursuer, in part due to numerical errors in the area differentiation. Nonetheless, the safe-reachable area minimization strategy showed a clear superiority in the trials with 2 and 3 pursuers, demonstrating the benefit of cooperation.

VI. Experimental Results

Experiments were conducted using the safe-reachable area minimization pursuit strategy on the **BE**rkeley **A**utonomy and **R**obotics in **CA**pture-the-flag **T**estbed (BEARCAT).

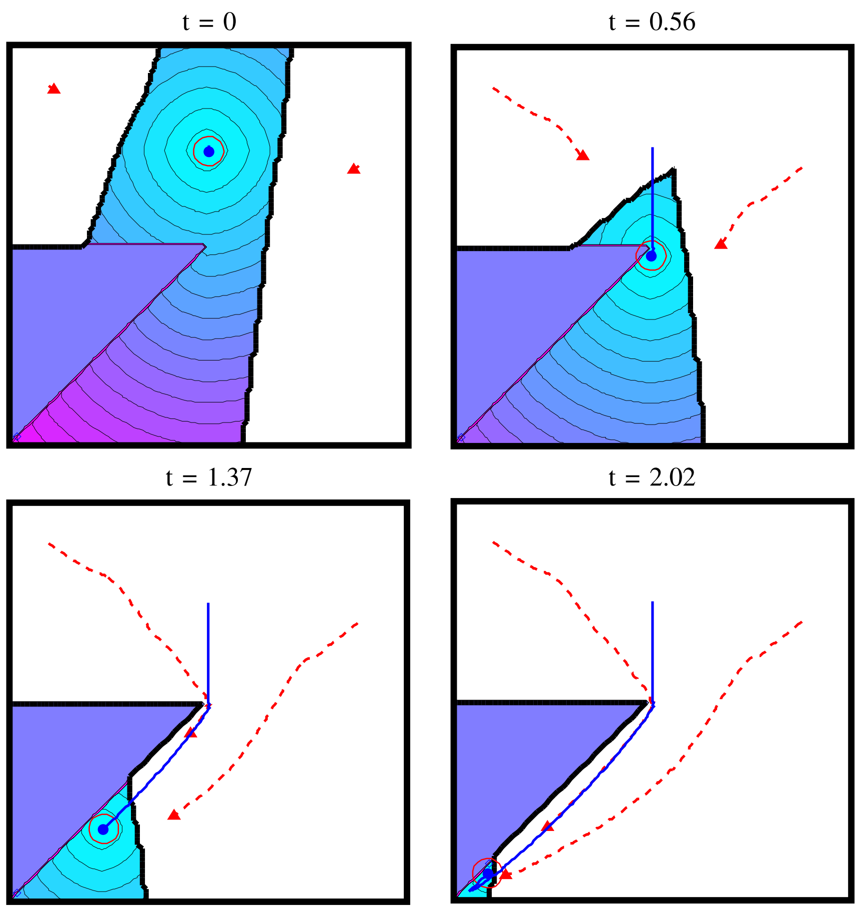


Fig. 8. **An example scenario showing 2 pursuers and 1 evader in a non-convex environment, solved using the modified FMM algorithm.**

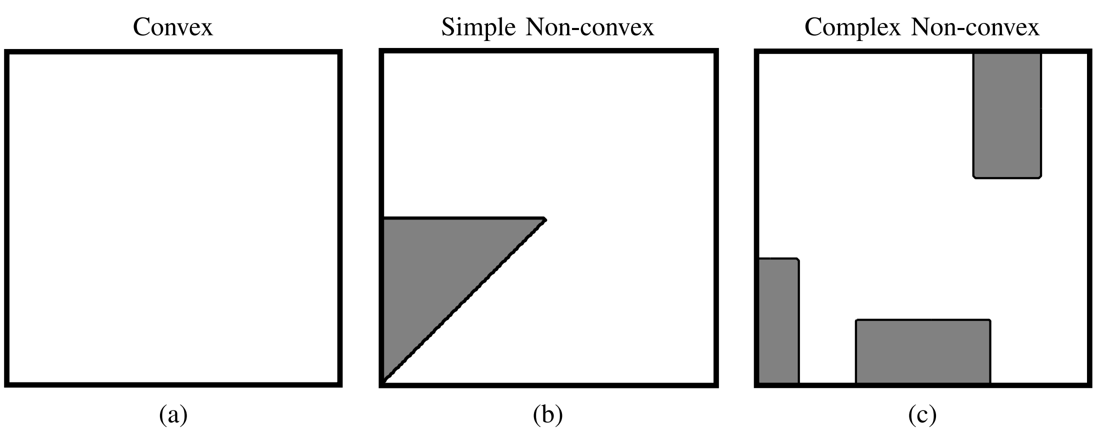


Fig. 9. **Domains used for the comparison simulations.**

	1 pursuer	2 pursuers	3 pursuers
Convex	91.6%	98.2%	96.6%
Simple Non-convex	67.2%	86.2%	86.4%
Complex Non-convex	48.2%	72.6%	76.4%

TABLE I

PERCENTAGE OF TRIALS FOR WHICH THE SAFE-REACHABLE AREA MINIMIZATION PURSUIT STRATEGY OUT-PERFORMED THE PURE PURSUIT STRATEGY.

BEARCAT is a novel testbed for research into automated assistance for human agents in adversarial games, consisting of smartphones connected to off-board computation and quadrotor UAVs. The primary purpose of BEARCAT is to provide a flexible testbed where human agents can participate in an adversarial game such as capture-the-flag, while receiving guidance from computational tools and working with autonomous agents such as UAVs. The testbed allows experiments to be performed by playing reach-avoid and pursuit-evasion games on the Berkeley campus. The major components of BEARCAT are illustrated in Figure 12, consisting of HTC Incredible smartphones [24], laptop computers, and quadrotor UAVs. The different components are networked

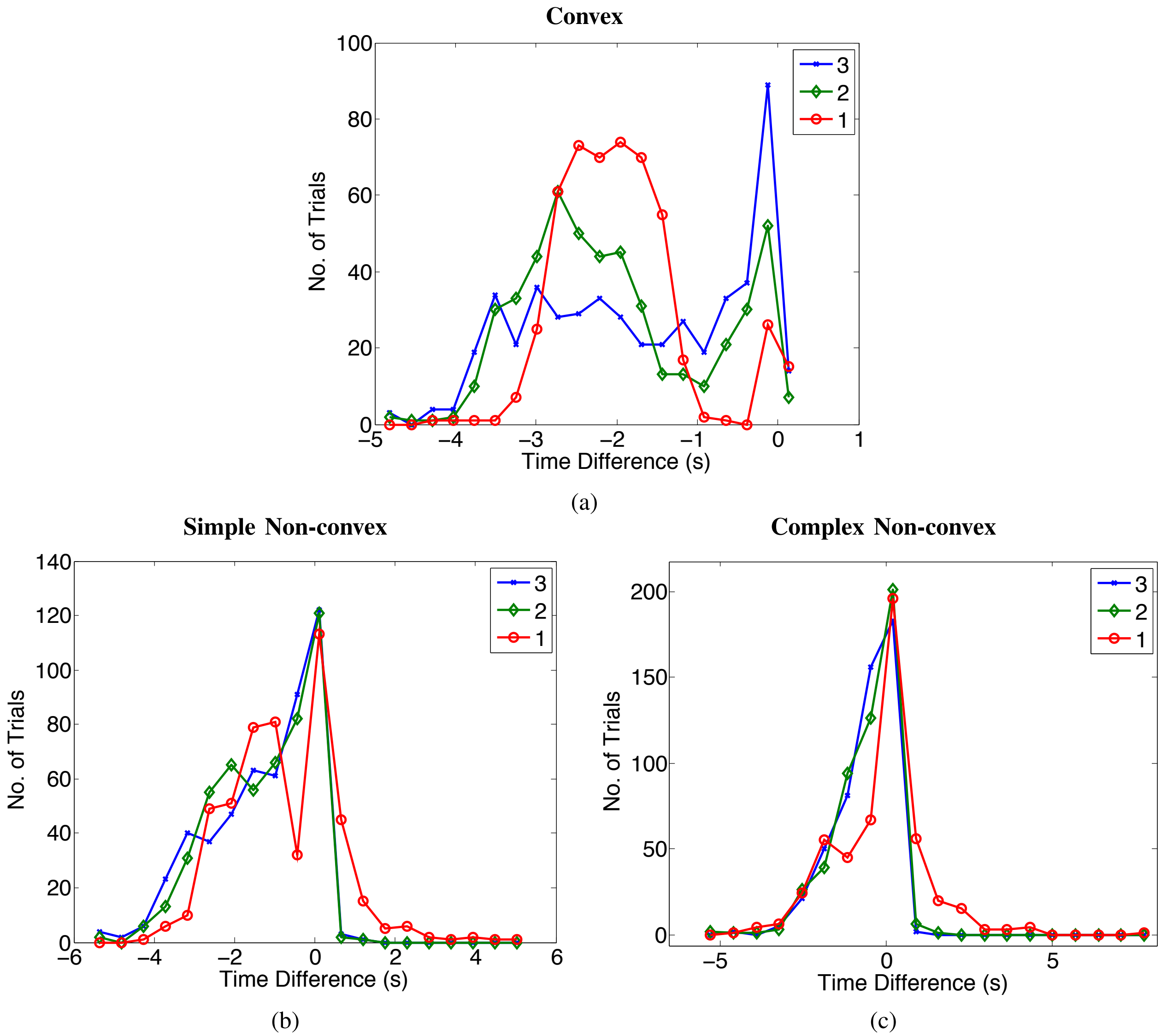


Fig. 11. **Histograms showing the number of trials versus difference in capture time between the pure pursuit and safe-reachable area minimization strategies for (a) the convex game domain, (b) the simple non-convex domain, and (c) the complex non-convex domain.**

via wireless communications, providing a complete system for both tracking agents and providing them with access to resources such as computed game solutions and UAV sensing.

The purpose of the tests was to evaluate the feasibility of the pursuit strategy as a tool for guiding human agents in a cooperative pursuit task. For these experiments, the UAV component of BEARCAT was not utilized. The game was played by two pursuers and a single evader in a small, convex field measuring about 80m x 40m. Agents were tracked using HTC Incredible smartphones, with the agent positions, game region, and the evader safe-reachable set also displayed on the phones. Pursuer strategies were computed using the analytic formulation directly on the phones, and the optimal heading direction was displayed for each pursuer, along with that pursuer's active boundary of the safe-reachable set.

Results from one of these experiments is shown in Figure 13. In this experiment, the pursuers began the game on the western side of the field, with the evader on the eastern side. The camera was placed to the north of the field, looking south. The pursuers were able to successfully trap and capture the evader.

An important point to note is that during these experiments, the GPS positions of the agents were not always perfectly reliable, and as a result the optimal headings were not always correct. Nonetheless, the pursuers were able to use the boundaries of the evader safe-reachable set to guide their actions. Instead of following the optimal headings exactly, each pursuer's active boundary of $\mathcal{S}_e$ gave an idea of that pursuer's area of responsibility, such that each pursuer could still reduce the area of $\mathcal{S}_e$ by attempting to "push" the boundary toward the evader by moving toward the perceived midpoint of the boundary. In addition, the visual display of the safe-reachable set allowed for implicit cooperation between the human pursuers, without the need for verbal communication. Thus, even though the agents were unable to always directly use the computed optimal headings, the safe-reachable set still enabled the pursuers to efficiently coordinate in capturing the evader.

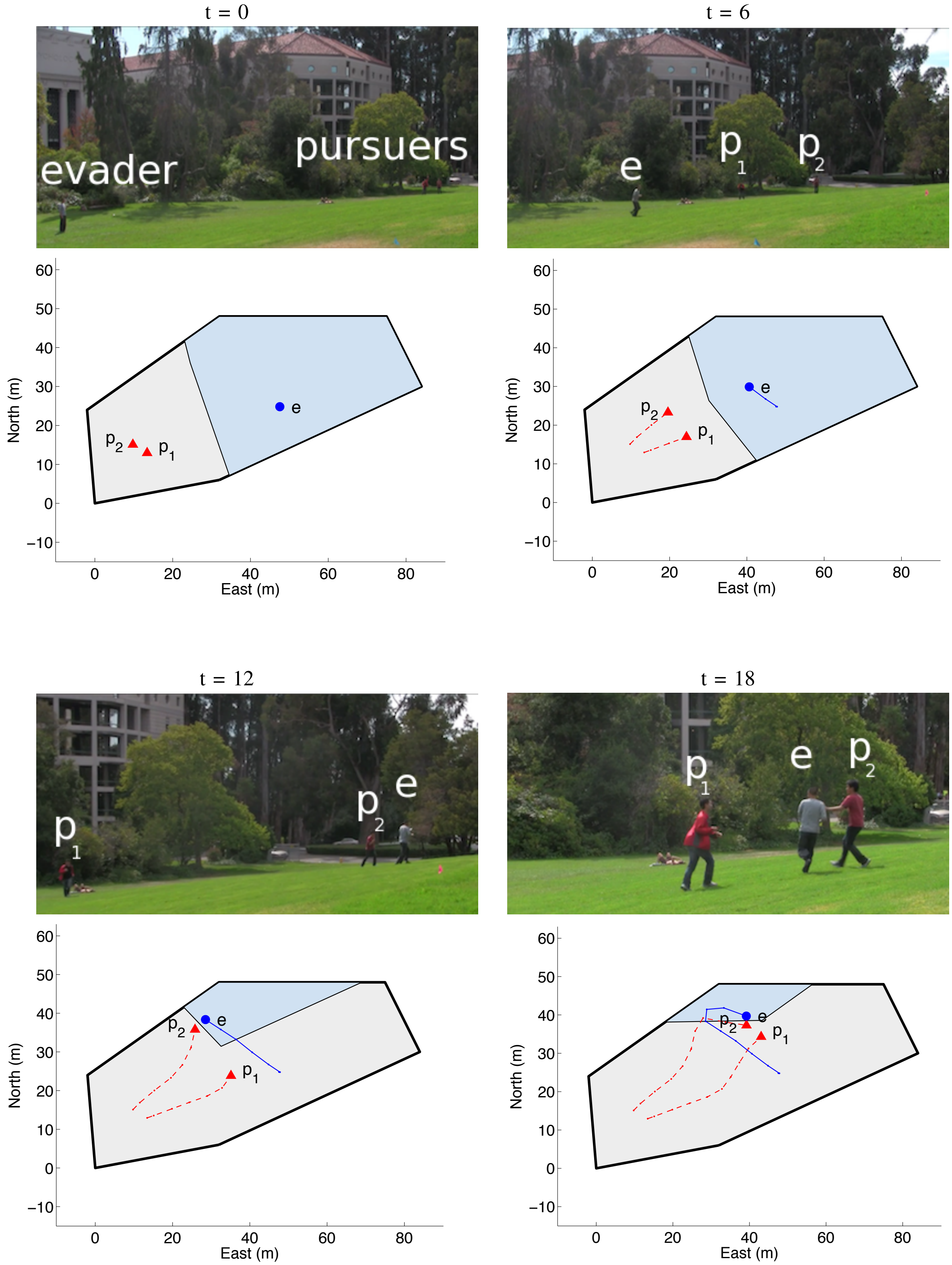


Fig. 13. **Video stills and data plotted for an experiment using the safe-reachable area minimization pursuit strategy in BEARCAT. For visual clarity, the evader is labeled e and the pursuers are labeled p_1 and p_2.**

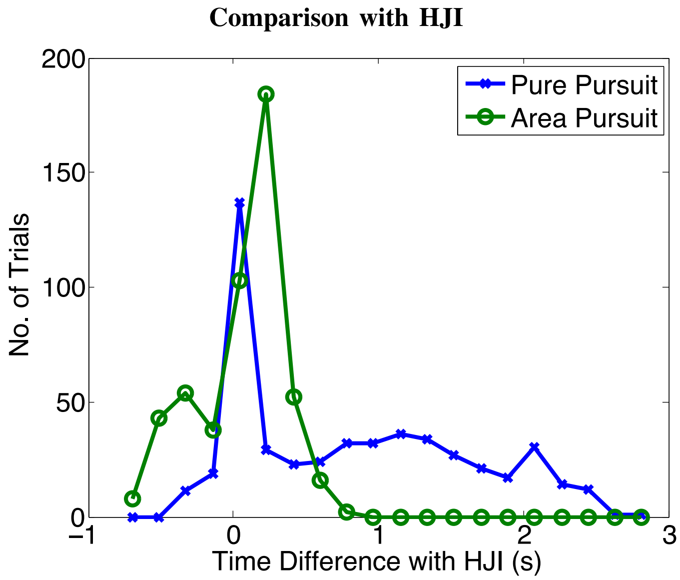


Fig. 10. **Histogram showing the difference in capture times in each single pursuer trial between the safe-reachable area minimization strategy, the pure pursuit strategy, and the numerical Hamilton-Jacobi-Isaacs pursuit strategy.**

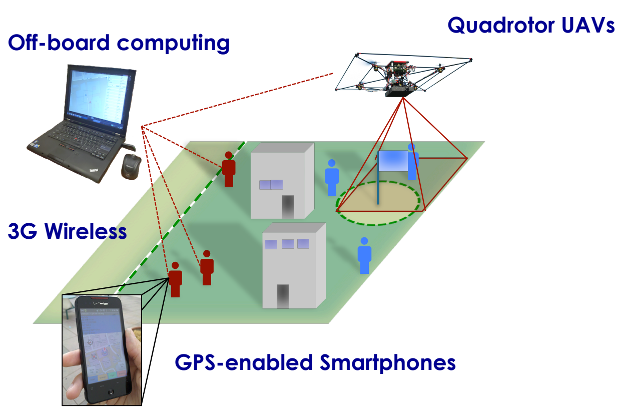


Fig. 12. **Components of the BEARCAT experimental platform.**

VII. Conclusions and Future Work

The safe-reachable area minimization strategy presented in this work has several important strengths. The construction of the safe-reachable set allows a high-dimensional problem to be reduced to lower dimensions, easing the computational burden. Additionally, each pursuer can compute its inputs independently, allowing the strategy to run efficiently in real time. Yet information is shared through the set itself, enabling cooperation between the pursuers and reducing capture time. The safe-reachable set itself also encompasses global information about the game domain, giving an advantage over a strategy like pure pursuit that ignores the presence of obstacles and boundaries.

One weakness of the strategy as presented is that safe-reachable area pursuit cannot handle more general domains that are not simply connected, where obstacles form holes in the domain. Further work is required to adapt the concepts presented here to more general domains. In addition, the numerical evaluation of the area gradient may also be improved. From the simulations and experiments it was noted that the behavior of the pursuers was typically smoother with the analytic input formula in convex domains than with the FMM computation.

Overall, safe-reachable area minimization seems to hold promise as a cooperative pursuit solution approach. As seen before in other works [4], [5], the visualization of the reachable set is a useful tool for human agents. Although GPS noise and time delay rendered the optimal headings sometimes unreliable in practice, the visualization of the set itself allowed the agents to compensate for these disturbances and head in the correct general direction to capture the evader. This result supports the idea that reachable sets are effective visual tools for assisting human agents.

Acknowledgments

The authors would like to thank Ryo Takei for his help in using the modified FMM algorithm to perform the safe-reachable set computations presented in this paper. In addition, they would like to thank Scott Hoag, Steven Xu, Ayushi Samaddar, and Catherine Pavlov for their contributions in developing the BEARCAT experimental platform and the subsequent experiments. This work would not have been possible without their assistance and support.

References

[1] R. Takei, H. Huang, J. Ding, and C. Tomlin, "Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 2012.

[2] W. Cheung, "Constrained pursuit-evasion problems in the plane," Master's thesis, Faculty of Graduate Studies, Computer Science, University of British Columbia, 2005.

[3] H. Huang, W. Zhang, J. Ding, D. Stipanović, and C. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *Proceedings of the IEEE International Conference on Decision and Control*, Orlando, Florida, 2011.

[4] M. Oishi, I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "Invariance-preserving abstractions of hybrid systems: Application to user interface design," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 229–244, 2008.

[5] H. Huang, J. Ding, W. Zhang, and C. Tomlin, "A differential game approach to planning in adversarial scenarios: a case study on capture-the-flag," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[6] R. Isaacs, *Differential Games*. New York: Wiley, 1967.

[7] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana University Mathematics Journal*, vol. 33, no. 5, pp. 773–797, 1984.

[8] T. Başar and G. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.

[9] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[10] M. Falcone and R. Ferretti, "Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods," *Journal of Computational Physics*, vol. 175, no. 2, pp. 559–575, 2002.

[11] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proceedings of the IEEE International Conference on Decision and Control*, Cancun, Mexico, 2008, pp. 3706–3712.

[12] S. Alexander, R. Bishop, and R. Ghrist, "Pursuit and evasion in non-convex domains of arbitrary dimensions," in *Proceedings of the Robotics: Science and Systems Conference*, Philadelphia, Pennsylvania, August 2006.

[13] S. Kopparty and C. Ravishankar, “A framework for pursuit evasion games in R^n,” *Information Processing Letters*, vol. 96, no. 3, pp. 114–122, 2005.

[14] T. Parsons, “Pursuit-evasion in a graph,” *Theory and Applications of Graphs*, pp. 426–441, 1978.

[15] M. Aigner and M. Fromme, “A game of cops and robbers,” *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 1–12, 1984.

[16] L. Guibas, J.-C. Latombe, S. Lavalle, D. Lin, and R. Motwani, “Visibility-based pursuit-evasion in a polygonal environment,” in *Algorithms and Data Structures*, ser. Lecture Notes in Computer Science, F. Dehne, A. Rau-Chaplin, J.-R. Sack, and R. Tamassia, Eds. Springer Berlin / Heidelberg, 1997, vol. 1272, pp. 17–30.

[17] S. LaValle and J. Hinrichsen, “Visibility-based pursuit-evasion: the case of curved environments,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 196 –202, April 2001.

[18] B. P. Gerkey, S. Thrun, and G. Gordon, “Visibility-based pursuit-evasion with limited field of view,” *International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.

[19] J. Sprinkle, J. Eklund, H. Kim, and S. Sastry, “Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller,” in *Proceedings of the IEEE International Conference on Decision and Control*, Atlantics, Bahamas, December 2004.

[20] J. McGrew, J. How, L. Bush, B. Williams, and N. Roy, “Air combat strategy using approximate dynamic programming,” *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, Aug 2008.

[21] J. A. Sethian, “Fast marching methods,” *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.

[22] ——, *Level set methods and fast marching methods*, 2nd ed., ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge, UK: Cambridge University Press, 1999, vol. 3.

[23] M. Falcone, “Fast marching methods for front propagation,” November 2007, lecture notes at “Introduction to Numerical Methods for Moving Boundaries”.

[24] HTC, “HTC Droid Incredible,” 2012, http://www.htc.com/us/products/droid-incredible-verizon.