Automation-Assisted Capture-the-Flag: A Differential Game Approach

Haomiao Huang, Jerry Ding, Wei Zhang, and Claire J. Tomlin

Abstract

Solutions to adversarial games present important challenges in robotics and control, and are made even more complex by the inclusion of human agents. Capture-the-flag serves as an excellent research environment for exploring these games: it includes complex interactions between opposing agents, and lends itself well to experiments incorporating humans and real-world considerations such as limited sensing, communications failure, and rough terrain. This work presents a formulation of a 1 vs. 1 game of capture-the-flag as a multi-stage reach-avoid game, and uses numerical solutions to the Hamilton-Jacobi-Isaacs equation to compute optimal trajectories for both agents. In addition to control inputs, these solutions determine regions of the state-space where victory is guaranteed for each agent. This allows visual tools to be created for human agents in the game allowing them to extrapolate beyond the capabilities of the automated solution. The computational method and simulations are presented, along with experiments with human agents in the BErkeley Autonomy and Robotics in CApture-the-flag Testbed (BEARCAT). These experiments demonstrate the use of the solutions in realistic conditions and highlight their utility to human agents, and also show how the results can be used to guide UAV search when full state information is unavailable.

I. Introduction

Many robotics and control applications require algorithms to control agents moving to achieve a goal in the presence of external disturbances, adversarial opposition, or moving obstacles that must be avoided. This includes piloting individual aircraft, driving a ground vehicle, air traffic control for groups of aircraft, and path planning for teams of human and robotic agents. Solutions to these problems may be found by

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, J. Ding, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA {haomiao, 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

analyzing aspects of these applications within the context of adversarial games such as pursuit-evasion games or what are referred to here to as reach-avoid games, in which the objective of one of the agents is to reach a sequence of target regions in the state-space while avoiding the capture region of an opposing agent, representing for example a moving obstacle in the environment.

The game of capture-the-flag presents an excellent framework for examining automated solutions for adversarial games. In capture-the-flag, two opposing teams each control a region with a flag that must be defended. A team achieves victory by having an agent move into the opponent's region, capture the opponent's flag, and then return to one's own region without being intercepted by an opponent. The adversarial interactions in capture-the-flag can be used as a representation for some of salient design challenges encountered in control problems such as collision avoidance for moving vehicles and coordination of multiple agents in hazardous scenarios. In addition, capture-the-flag games naturally allow for the incorporation of human agents, allowing the impact of automated solutions for such games to be examined for human as well as autonomous agents.

Although the full game is complex and multi-layered, capture-the-flag in its simplest form can be condensed into a reach-avoid game between two agents over two different stages: one agent, referred to as an attacker, seeking to reach a flag region and subsequently return to a safe region, and an opposing agent, referred to as a defender, attempting to prevent this from taking place. Finding solutions to reach-avoid games can be challenging. Unlike in pursuit-evasion games, the players must select their actions in consideration of not only the capture region, but also the target regions. In particular, the attacker must balance the competing objectives of avoiding capture by the defender while attempting to reach its target. Further complications are presented by the multi-stage nature of the problem, since the agents must select appropriate controls within each stage to ensure the feasibility of their respective winning objectives in future stages.

This paper presents the formulation of a 1 vs. 1 game of capture-the-flag as a multi-stage reach-avoid game between two agents, and describes an approach for characterizing the winning initial conditions and strategies for each agent using numerical solutions of Hamilton-Jacobi-Isaacs (HJI) equations. More specifically, the competing objectives of the attacker and the defender are encapsulated in terms of a two-stage, zero-sum differential game, with the solution to each stage being characterized by the solution of an appropriate HJI equation. Numerical methods for solving HJI equations is then leveraged to obtain a game solution in terms of the set of initial states for which an agent's control objectives can be attained, as well as a feedback strategy on the winning set. This strategy selects controls, based upon the measured positions of the agents at each time instant, to achieve the winning objectives of a given agent under the worst-case behavior of the opponent. Besides providing a complete game solution, this approach also

produces clear and intuitive tools for informing human decision-making through visualizations of the winning regions and recommended control actions.

While some algorithmic results underlying this work were first presented in [1], this paper provides a more in-depth discussion of the theoretical aspects of the HJI approach to reach-avoid games, along with a demonstration of the utility of the proposed methodology through extensive experimental results. In particular, the solutions described here have been developed into tools for human agents and evaluated through experiments with the **BE**rkeley **A**utonomy and **R**obotics in **CA**pture-the-flag **T**estbed (BEARCAT). 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 unmanned aerial vehicles (UAVs). The results demonstrate the use of the algorithms under both nominal conditions, as well as under degraded conditions in which the assumptions of perfect state knowledge and optimal opponents do not hold. For example, in the absence of information about opponent location, the solutions are used to guide the search task for a UAV assisting a human agent, and the winning region information is used by human agents to achieve desired objectives against opponents playing sub-optimal strategies.

The paper proceeds by examining related work in Section II, and then defining the 1 vs. 1 capture-the-flag problem and illustrating the desired solution approach via a 1-dimensional example in Section III. Next, the HJI method for finite horizon differential games is reviewed in Section IV, and then applied to the game of capture-the-flag in Section V. Simulation results are presented in Section VI to illustrate the solutions, and experimental validation of the solutions with human agents in the BEARCAT platform are presented in Section VII. Concluding remarks and several important future work directions are discussed in Section VIII.

II. Related Work

Reach-avoid games such as capture-the-flag parallel elements of pursuit-evasion games, with the defender playing a role analogous to a pursuer, and the attacker playing a role analogous to an evader. However, the addition of target regions necessitates considerable modification of the player strategies. In such a scenario, the attacker must consider the objective of reaching a sequence of targets in addition to evading capture, while the defender must consider the objective of guarding the target regions in addition to capturing the attacker.

An early result in pursuit-evasion games established that three pursuers are sufficient to capture an evader on any planar graph [2]. This work partitions the graph into regions that are defended and systematically cleared by a team of pursuers, and has recently been extended to games in continuous regions [3]. In these solutions, the motion planning task is simplified by the fact that the goals of the

opposing agents are either capture or avoidance of capture, as compared with more complicated objectives such as the attainment of a sequence of target regions while avoiding capture. The solutions for discrete games also inform a class of games commonly referred to as visibility-based pursuit-evasion, in which a group of searchers attempt to bring an evader into their field of view [4], [5], [6], through cooperative coverage of regions in the state space where the evader might be located. In these games, the visibility region of a searcher is commonly modeled as line-of-sight (in effect an infinite capture radius when not occluded), thus allowing these continuous games to be transformed into discrete pursuit-evasion games, and solved accordingly.

For certain motion planning applications with simple obstacle configurations, it is possible to construct optimal trajectories geometrically [7], [8]. Some proposed methods within this domain also treat moving obstacles that are potentially adversarial. They use the possible future positions of the obstacles as static obstacles in a joint state-time space, allowing static planning methods to be employed [9], [10], [11].

Another approach utilizes model-predictive control, with a model to predict opponent actions so that optimization can be performed for the controlled agents with respect to the assumed opponent behavior. This has been used in the Cornell Roboflag competition, in which two opposing teams of mobile robots play a game of capture-the-flag while directed by human supervisors [12]. Defensive strategies for intercepting multiple attackers with multiple defenders were developed using mixed-integer linear programming, with the assumption that attackers either proceed toward the goal in straight lines [13] or according to a linear feedback law [14]. Control inputs can be efficiently generated using standard optimization tools. These approaches perform well when the model is a good approximation of the actual strategy chosen by the opponent. On the other hand, guarantees of optimality or winning conditions are not provided if the opponent were to deviate from the assumed behavior.

More generally, the winning strategies for the opposing agents in a pursuit-evasion or reach-avoid game can be viewed as the solution to a zero-sum differential game [15]. Namely, a scalar value function can be used to encode the objectives of one of the agents by setting the value at a given initial condition to be negative if these objectives can be attained under an admissible strategy, and positive otherwise. The goals of the opposing agents then becomes either minimizing or maximizing this value function over admissible strategies. Under appropriate technical conditions, this value function can be computed via the solution of a Hamilton-Jacobi-Isaacs (HJI) equation, either using the method of characteristics, which integrates solution trajectories backward from a known terminal state [15], [16], or via numerical approximation on a grid representing the state space [17], [18]. While the characteristic approach provides individual solution trajectories ending in given terminal conditions, the grid-based approximation allow control strategies to be computed in feedback form over the state space of interest. The latter approach

has been successfully applied to a number of applications, including the design of conflict resolution policies for air traffic management [17] and the verification of automated aerial refueling protocols for UAVs [19]. In this work, we extend the previous methodology to account for the complex objectives and the multi-stage nature of reach-avoid games, within the context of a capture-the-flag problem.

In comparing the various approaches reviewed in this section, the solution methods for differential games in general feature trade-offs between optimality, completeness, problem complexity, and computational speed. While optimal or complete solutions typically come at the cost of high computational complexity, efficient algorithmic solutions often come at the cost of simplified problem formulations or loss of optimality and completeness. With respect to results from the literature, the pursuit-evasion work cannot be directly applied to reach-avoid games such as capture-the-flag due to the additional complexity in game objectives, and the use geometric and model-predictive methods involves assumptions on either the configuration space or the opponent behavior. The work presented here provides a Hamilton-Jacobi method for finding complete solution strategies to a reach-avoid game in complex game domains, at some expense in computational complexity. It should be noted, however, that while these solutions will not be computed in real-time, they can nonetheless be precomputed as lookup tables and recalled for use in an online setting.

III. 1 vs. 1 Capture-the-Flag

The game considered here is a simplified version of capture-the-flag with a single agent on each team. The objective of the first agent, called the attacker, is to arrive at the flag and subsequently return with it to a safe region. The objective of the second agent, called the defender, is to prevent the attacker from completing these tasks. The problem then is to characterize the sets of attacker and defender configurations for which the winning conditions are feasible, as well as to find the feedback strategies that ensure these winning conditions.

A. Problem Formulation

The game domain is assumed to be a compact set Ω in the plane $\mathbb{R}^2$. The set Ω can be regarded as the allowable free space for the agents to occupy, and in general may not be simply connected. For example, obstacles for both agents may be represented as holes in Ω. The "safe" return region for the attacker is a closed set $R \subset \Omega$, and the flag is located in a closed set $F \subset \Omega$. For simplicity of illustration, the sets Ω and R in the examples discussed below will be defined as rectangular regions, with F defined as a circular region with radius r_f (see Figure 1). However, the computational methods described here are not limited to these simple geometric shapes.

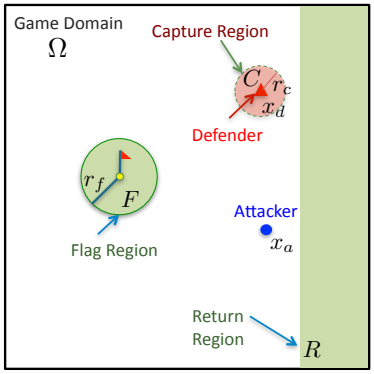

Fig. 1. **The basic configuration of the capture-the-flag game.**

The state of the game is described by the vector $\mathbf{x} = (x_a, x_d) \in \mathbb{R}^4$, where x_a and x_d are the planar positions of the attacker and defender, respectively, and $\Omega^2 \in \mathbb{R}^4$ denotes the joint domain of both agents. The equations of motion are given by

$$\begin{aligned} \dot{x}_a &= u, \qquad u \in U, \\ \dot{x}_d &= d, \qquad d \in D. \end{aligned} \tag{1}$$

where u and d are the inputs of each agent, constrained to lie within sets U and D, respectively. The initial state is described by $\mathbf{x}(0) = (x_a^0, x_d^0) = \mathbf{x}^0$. For this example, U and D are represented by speed limits $v_{a,max}$ and $v_{d,max}$, with $||u||_2 \leq v_{a,max}$ and $||d||_2 \leq v_{d,max}$. It is not necessary to assume that $v_{a,max} = v_{d,max}$, and the maximum speeds may vary throughout Ω.

As per the rules of the game, the defender is constrained to remain outside of the flag and return regions, while the attacker can move freely through either. The attacker is considered to be intercepted by the defender if the attacker comes within some capture set centered on the defender, defined here as $C = \{\mathbf{x} \mid ||x_a - x_d||_2 \leq r_c\}$, where r_c is a constant capture radius.

Victory for the attacking agent is attained by meeting all of the following conditions, assuming the game takes place over some finite time horizon $[0, T_f]$ with a time duration T_1 allocated for flag capture and $T_2 = T_f - T_1$ allocated for flag return:

- Flag capture: $x_a(t) \in F$ for some $t \in [0, T_1]$
- Flag return: $x_a(t) \in R$ for some $t \in [T_1, T_f]$
- Remaining within game domain and avoiding interception by the defender:
  $x_a(t) \in \Omega \bigwedge \mathbf{x}(t) \notin C$ for all time $t \in [0, T_f]$

In addition, the attacker also wins if at any time the defender violates the rules of the game by entering the flag or return regions. Victory for the defending agent is achieved by preventing the attacker from

achieving any of the above conditions by T_f while obeying the constraint $x_d \in \Omega \setminus (F \cup R)$. It should be remarked that when T_f is small, the options of the attacking agent may be restricted by the time limit. Correspondingly, this also leads to simplistic delaying strategies by the defending agent. Motivated by this consideration, the work presented here will be mainly concerned with scenarios in which T_f is large, namely as T_f approaches ∞.

It is assumed that the positions of both agents are fully observable to each other, and that the choice of attacker input is revealed to the defender at each time instant. In terms of agent strategies, this means that the attacker input $u(t)$ can be chosen as a function of the system state $x(t)$, while the defender input $d(t)$ can be chosen as a function of both the system state $x(t)$ and the attacker input $u(t)$. The latter assumption corresponds to a choice in the order of play to prevent the agents from second-guessing each other. Thus, the problem formulation is conservative from the point of view of the attacker, in that the defender has the advantage of knowing attacker's input selection. However, it turns out that, for the agent dynamics considered here, the opposite order of play leads to an identical solution, due to the satisfaction of a minimax condition which will be discussed subsequently.

The goal here is to provide a solution to this two-agent game of capture-the-flag in terms of both the winning regions and winning strategies for each agent. For the attacker, this means determining the subset of initial configurations $W_A \subset \mathbb{R}^4$ for which there exists a feasible attacker strategy for achieving the victory conditions, regardless of the strategy of the defending agent. Furthermore, for any configuration in W_A, it is also necessary to determine a winning strategy that ensures attainment of the attacker victory conditions. Clearly, for any permissible initial configuration outside this set, there exists some defender strategy so as to prevent the attacker from achieving victory, regardless of the attacker strategy. Thus, the defender winning set W_D is given simply as $\Omega^2 \setminus W_A$.

The game occurs over two stages: when the attacker is attempting to achieve flag capture, and then subsequently when the attacker is attempting to achieve flag return. In discussing the winning sets, it is useful to characterize some intermediate winning sets for the flag capture and flag return stages of the game as well. The sets are defined as follows:

- **Flag Capture Set** F_A: configurations from which the attacker can achieve flag capture while avoiding defender interception.
- **Stop Capture Set** F_D: $\Omega^2 \setminus F_A$, where the defender can prevent flag capture by the attacker.
- **Flag Return Set** R_A: configurations from which the attacker can achieve flag return while avoiding defender interception.
- **Stop Return Set** R_D: $\Omega^2 \setminus R_A$, where the defender can prevent flag return by the attacker.

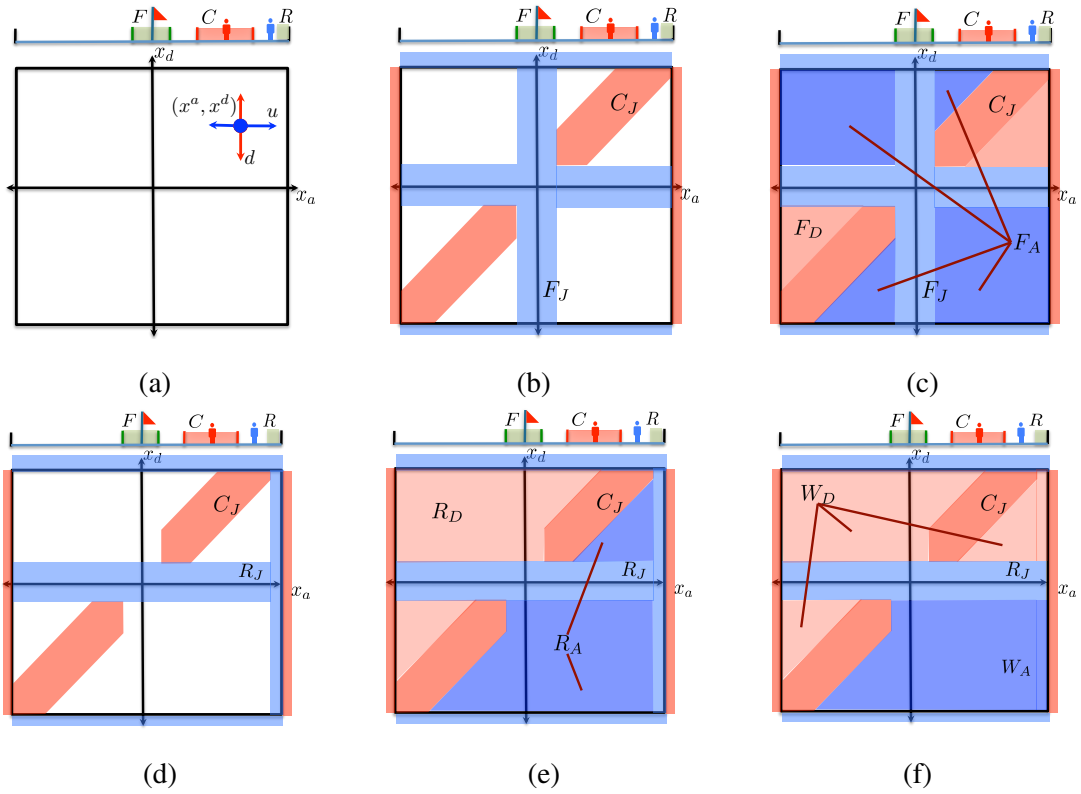

Fig. 2. **1-dimensional example illustrating (a) the joint state space and the current state of the game, along with the agents' inputs, the targets and winning configurations for the agents for (b), (c) flag capture and (d), (e) flag return, and (f) the winning configurations for the full game.**

B. 1-Dimensional Example

A 1-dimensional example game is presented in Figure 2 to clarify the region definitions and the concept of winning regions presented in the preceding text. Figure 2(a) shows the joint state space for a 1-dimensional game, with the attacker coordinate plotted on the horizontal axis and the defender coordinate plotted on the vertical axis. The current state of the game as illustrated is shown along with the possible inputs of the two agents. Note that the attacker input u moves the state along the x_a (horizontal) axis, and similarly the defender input d moves the state along the x_d (vertical) axis. It is clear from the initial configuration shown in Figure 2(a) that there is no way for the attacker to reach the flag without being intercepted by the defender.

Figure 2(b) illustrates the desired terminal configurations for each agent in the flag capture stage of the game. The target configurations for the attacker are denoted F_J and are highlighted in blue, representing the set of states where the attacker has entered F, or the defender has forfeited by either exiting the game region or entering F or R. Similarly, the defender's target configurations C_J are highlighted in red, denoting the states where the attacker has either been intercepted by entering C or has exited Ω.

From this, the winning regions for the flag capture portion of the game can be determined by inspection as those identified in Figure 2(c). Intuitively, the attacker winning configurations F_A correspond to states where the defender is not between the attacker and the flag, and F_D comprises the remainder of the states.

Similarly, the target configurations for the flag return stage are shown in Figure 2(d) with winning configurations for each agent illustrated in Figure 2(e). Note that in order for the attacker to win the two-stage game, it must first reach the flag and then return it to safety. This requires the flag capture portion of the game to end with the joint state in R_A, that is, the attacker must reach the flag in such a way that it can safely return while avoiding interception. Thus the full winning configuration for the attacker W_A will be the darker blue region as shown in Figure 2(f), with W_D as the light red region.

The 1-dimensional example discuss here provides some intuition into the winning regions in the joint state space and how the sequential nature of the game affects their construction. For this simple example, the regions can be found trivially by inspection, but the computation of such sets in $\mathbb{R}^4$ is a more complex matter. The agents are not constrained to move on a line, but may move around each other and around obstacles in the state space. In the following, a computational reachability approach is described, and then used to determine the desired sets for capture-the-flag.

IV. Computational Approach via Hamilton-Jacobi-Isaacs Equations

Capture-the-flag falls naturally under the framework of differential games [16] due to the competing objectives of the attacking and defending agents. This section will briefly review a differential game formulation of a finite horizon reachability problem which relates to a single stage of the capture-the-flag problem. A computational approach to this problem will be presented in terms of the numerical solution of a time-dependent HJI equation [17]. The specific application of this approach to the computation of winning regions and winning strategies for capture-the-flag will be demonstrated in the following section.

For the purposes of the discussions here, the opposing sides of a zero-sum differential game will be referred to as the *control* and the *disturbance*. This choice of naming reflects the fact that although one can in principle derive equilibrium solutions for both sides, a differential game is often solved for applications that require controlling a specific side. The other side is then regarded as a disturbance whose adversarial actions must be accounted for in a worst-case sense.

A. Finite Horizon Reachability

Consider a continuous time model for the evolution of a dynamic game as described by the ordinary differential equation

$$\dot{\mathbf{x}} = f(\mathbf{x}, u, d), \ \mathbf{x}(0) = \mathbf{x}^0 \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the game state, u is the input of the control, d is the input of disturbance, and $\mathbf{x}^0$ is the initial condition. The state variables could represent, for example, the joint position configurations of two opposing players, while the inputs could represent the choice of speed and heading by the two players. The input ranges of the control and disturbance are denoted by U and D, respectively.

Suppose that one is given a finite time horizon $[0, T]$. This work will assume an information pattern in which the control selects inputs $u(t), t \in [0, T]$, satisfying $u(t) \in U$, possibly as a function of the state $\mathbf{x}(t)$, according to a strategy $\mu(\mathbf{x}(t), t)$ with $u(t) = \mu(\mathbf{x}(t), t)$. On the other hand, the disturbance selects inputs $d(t), t \in [0, T]$, satisfying $d(t) \in D$, possibly as a function of both the state $\mathbf{x}(t)$ and the input u of the control, according to a strategy $\gamma(\mathbf{x}(t), u(t), t)$. This information pattern reflects a conservative view by the control that, in the worst-case, the adversary may select inputs in response to actions of the control at each time instant. The set of permissible strategies for the control and disturbance are denoted by $\mathbb{U} = \{\mu : \mathbb{R}^n \times [0, T] \to U\}$ and $\mathbb{D} = \{\gamma : \mathbb{R}^n \times U \times [0, T] \to D\}$, respectively.

Now consider a target set $\mathcal{T}$ and an undesired set $\mathcal{K}$, with $\mathcal{T}, \mathcal{K} \subset \mathbb{R}^n$. In this differential game, the objective of the control is to reach the target set $\mathcal{T}$ while avoiding the undesired set $\mathcal{K}$, and the objective of the disturbance is to prevent the control from achieving this goal, either by driving the system into the undesired set or keeping the system out of the target set. Note that by appropriately interpreting the sets $\mathcal{T}$ and $\mathcal{K}$ in terms of the flag region, return region, and capture set, one can relate this differential game to either the flag capture or flag return stage of capture-the-flag.

The problem of computing the winning regions in the differential game above can be viewed as a finite horizon reachability problem. In particular, the winning region for the control is the set of initial conditions $\mathbf{x}^0$ for which there exists some choice of control strategy $\mu \in \mathbb{U}$, such that regardless of the choice of disturbance strategy $\gamma \in \mathbb{D}$, the state trajectory $\mathbf{x}(\cdot)$ under (2) satisfies $\mathbf{x}(t) \in \mathcal{T}$ for some $t \in [0, T]$ and $\mathbf{x}(s) \notin \mathcal{K}$, $\forall s \in [0, t]$. This set is referred to as the *reach-avoid set* over $[0, T]$, and denote it by $\mathcal{RA}_T(\mathcal{T}, \mathcal{K})$. The winning region for the disturbance is simply the complement of this set.

Under suitable conditions, as discussed in [17] and [20], the reach-avoid set can be determined from the solution of a constrained Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE), using a level set representation of sets. This representation implicitly defines a set $G \subset \mathbb{R}^n$ as the zero sub-level set of a function $\phi_G : \mathbb{R}^n \to \mathbb{R}$, such that $G = \{\mathbf{x} \in \mathbb{R}^n : \ \phi_G(\mathbf{x}) \leq 0\}$. Set operations proceed in a straightforward manner using the $\max$ and $\min$ operators, with $A \cap B = \max(\phi_A, \phi_B)$ and $A \cup B = \min(\phi_A, \phi_B)$.

In this case, the reach-avoid set for $T = 0$ is simply given by $\mathcal{RA}_0(\mathcal{T}, \mathcal{K}) = \mathcal{T} \setminus \mathcal{K}$. A level set representation for this set can be computed as $\phi_{\mathcal{T} \setminus \mathcal{K}} = \max(\phi_{\mathcal{T}}, -\phi_{\mathcal{K}})$, where $\phi_{\mathcal{T}}$ and $\phi_{\mathcal{K}}$ are level set representations of the target set $\mathcal{T}$ and the undesired set $\mathcal{K}$. To compute a representation of the reach-avoid

set for $T > 0$, one can use the arguments presented in [17] and [20] to characterize the time-evolution of the reach-avoid set in terms of a terminal value HJI PDE, with the function $\phi_{\mathcal{T}\setminus\mathcal{K}}$ as the terminal condition.

$$\frac{\partial \phi}{\partial t} + \min\left[0, H\left(\mathbf{x}, \frac{\partial \phi}{\partial \mathbf{x}}\right)\right] = 0,\ \phi(\mathbf{x}, 0) = \phi_{\mathcal{T}\setminus\mathcal{K}}(\mathbf{x}), \tag{3}$$

subject to

$$\phi(\mathbf{x}, t) \geq -\phi_{\mathcal{K}}(\mathbf{x}),\ \forall t \in [-T, 0],$$

where H is the optimal Hamiltonian

$$H\left(\mathbf{x}, p\right) = \min_{u \in U} \max_{d \in D} p^T f(\mathbf{x}, u, d).$$

Let $\phi : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the viscosity solution [21] to (3), then it can be shown that the reach-avoid set is simply the zero sub-level set of the function $\phi(\mathbf{x}, -T)$:

$$\mathcal{RA}_T(\mathcal{T}, \mathcal{K}) = \left\{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}, -T) \leq 0\right\}. \tag{4}$$

In terms of practical computation, an accurate numerical approximation of the viscosity solution to (3) can be computed using the Level Set Toolbox for MATLAB [22]. As the grid required to represent the state space grows exponentially with the number of continuous states, this method may be limited computationally. Currently, systems of up to 4 dimensions can be handled easily, and solutions have been found for some systems with 5 dimensions [23]. For the interested reader, details on the theoretical and computational issues related to Hamilton-Jacobi reachability can be found in [17], [20], and a schematic description of the approach, including a dynamic programming derivation of the approach and some geometric insights, can be found in [24].

B. Synthesis of Finite Horizon Strategies

Winning strategies in regions of the state space where the solution ϕ to equation (3) is differentiable can be synthesized based upon the approach described in [25] and [26]. For zero-sum differential games, the solution to the HJI equation is typically differentiable away from some singular surfaces where the winning strategies are not uniquely defined [16]. With arbitrarily small perturbations from these surfaces, the solution becomes differentiable, allowing application of the techniques described here.

Under this setting, for $\mathbf{x} \in \mathcal{RA}_T(\mathcal{T}, \mathcal{K})$, $t \in [0, T]$, a winning strategy for the control ensuring that the state trajectory reaches the target set $\mathcal{T}$ within $[0, T]$, while avoiding the undesired set $\mathcal{K}$ can be derived as

$$\mu(\mathbf{x}, t) \in \arg\min_{u \in U} \max_{d \in D} \left(\frac{\partial \phi(\mathbf{x}, t - T)}{\partial \mathbf{x}}\right)^T f(\mathbf{x}, u, d). \tag{5}$$

Similarly, for $\mathbf{x} \notin \mathcal{RA}_T(\mathcal{T}, \mathcal{K})$, a winning strategy for the disturbance preventing the control from achieving the desired objectives within $[0, T]$ can be derived as

$$\gamma(\mathbf{x}, u, t) \in \arg\max_{d \in D} \left(\frac{\partial \phi(\mathbf{x}, t-T)}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, u, d). \tag{6}$$

Note that in the special case that the Hamiltonian of equation (3) satisfies a minimax condition, one can derive a winning disturbance strategy which is independent of the control input u. In particular, suppose that the following identity holds

$$\min_{u \in U} \max_{d \in D} p^T f(\mathbf{x}, u, d) = \max_{d \in D} \min_{u \in U} p^T f(\mathbf{x}, u, d), \tag{7}$$

for every $x, p \in \mathbb{R}^n$. Then a choice of winning strategy for the disturbance is described by

$$\tilde{\gamma}(\mathbf{x}, t) \in \arg\max_{d \in D} \min_{u \in U} \left(\frac{\partial \phi(\mathbf{x}, t-T)}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, u, d). \tag{8}$$

Given the numerical computation of the set $\mathcal{RA}_T(\mathcal{T}, \mathcal{K})$, the derivatives of ϕ are typically not available in closed form. Nonetheless, approximate strategies can be obtained by computing numerical derivatives of ϕ, where it is differentiable.

C. Approximation of Long Term Strategies

For scenarios in which the differential game formulated in Section IV-A is to be played over an extended time horizon, namely as $T \to \infty$, then it may be of interest to study an infinite horizon formulation of the reachability problem. In particular, the infinite horizon problem involves the computation of the infinite horizon reach-avoid set, and the synthesis of infinite horizon player strategies, possibly as stationary feedback strategies.

A formal treatment of the infinite horizon case requires a rigorous study of several technical issues: 1) the convergence of the viscosity solution $\phi(\mathbf{x}, -t)$ to equation (3) as $t \to \infty$; 2) the relation of the limiting function, provided that it exists, to the infinite horizon reach-avoid set; 3) the existence of stationary winning strategies for one or both players. As such, a complete mathematical treatment is beyond the scope of this paper.

For this application scenario, this work considers the approximation of long term player strategies when the solution to equation (3) does converge. More specifically, suppose that the function $\phi(\mathbf{x}, -t)$ converges as $t \to \infty$. Let ϕ^* be the limiting function, namely $\phi^*(\mathbf{x}) := \lim_{t \to \infty} \phi(\mathbf{x}, -t)$. Then, the long-term player strategies, namely the winning control strategy μ in (5) and the winning disturbance strategy γ in (6) as $T \to \infty$, can be approximated as the stationary strategies

$$\mu^*(\mathbf{x}) \in \arg\min_{u \in U} \max_{d \in D} \left(\frac{\partial \phi^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, u, d), \tag{9}$$

$$\gamma^*(\mathbf{x}, u) \in \arg\max_{d \in D} \left(\frac{\partial \phi^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T f(\mathbf{x}, u, d). \tag{10}$$

Practically speaking, the function ϕ^* can be obtained from the finite horizon reachability computation in equation (3) by checking the convergence of the level set function $\phi(\mathbf{x}, -t)$ at successive time steps. As will be discussed in the simulation results, it turns out that for reasonable choices of problem parameters, the reachability computation for each stage of capture-the-flag indeed exhibits convergence properties. Furthermore, the approximate long-term strategies, as derived using equations (9) and (10) are shown in simulation scenarios to achieve the winning conditions of the attacker and the defender.

V. Solution to Capture-the-Flag

The solution to the 1 vs. 1 capture-the-flag game can now be found by characterizing the winning regions and winning strategies for the attacker and the defender via the methodology of HJI reachable set computation, as described in the preceding section. This section will provide procedures for computing the attacker victory sets F_A, R_A, and W_A, from which the defender victory sets F_D, R_D, and W_D can be obtained via the relations described in Section III. Winning strategies for the attacker and the defender on their respective victory sets can then be derived from these reachable set computations.

With respect to the terminology introduced in the preceding section, the attacker takes on the role of the control, while the defender takes on the role of the disturbance. For the system dynamics of capture-the-flag in (1), the optimal Hamiltonian is simply given by

$$H(\mathbf{x}, p) = \min_{u \in U} \max_{d \in D} p^T u + p^T d. \tag{11}$$

A. Construction of Winning Regions

The construction of the winning sets will proceed backward, starting with the flag return stage in order to determine the terminal conditions for flag capture that could result in successful return. First, consider the flag return stage independently, as taking place over some time horizon $[0, T_2]$. The winning condition for the attacker during this stage is to arrive in the return region R within $[0, T_2]$, while avoiding interception by the defender. This portion of the game terminates at a time t with $0 \leq t \leq T_2$ if the attacker arrives in R or is intercepted by the defender. If the attacker is unable to reach R by T_2, the defender wins.

The rules require both agents to stay within the game region Ω and the defender to remain outside of F and R. The constraints for each agent are encoded as part of the winning conditions for the other, so that the attacker wins if the defender violates its constraints and vice versa.

The formal definition of the winning conditions for each agent follows. Let $\mathbf{x}$ be the joint configuration (x_a, x_d), Ω^C the complement of Ω in $\mathbb{R}^2$, and $G_A = \left\{\mathbf{x} \in \mathbb{R}^4 : \ x_a \in \Omega^C\right\}$, then the attacker winning

conditions for the flag return stage as described above are $O_R \vee O_D$, where

$$O_R = \{\exists t \in [0, T_2], x_a(t) \in R \wedge \mathbf{x}(s) \notin C \cup G_A, \forall s \in [0, t]\},$$
$$O_D = \left\{\exists t \in [0, T_2], x_d(t) \in F \cup R \cup \Omega^C \wedge \mathbf{x}(s) \notin C \cup G_A, \forall s \in [0, t]\right\}.$$

Define the sets $G_D = \left\{\mathbf{x} \in \mathbb{R}^4 : \ x_d \in F \cup R \cup \Omega^C\right\}$ and $G_R = \left\{\mathbf{x} \in \mathbb{R}^4 : \ x_a \in R\right\}$. Then the attacker target set in the flag return stage can be defined as the set $R_J = G_R \cup G_D$, and the defender target set is $C_J = C \cup G_A$. The flag return set can then be computed as

$$R_A = \mathcal{RA}_{T_2}(R_J, C_J). \tag{12}$$

Now consider the flag capture stage of the game, with a time horizon of $[0, T_1]$. If the second stage of the game is ignored, then the goal of the attacker is to simply arrive in the flag region F within $[0, T_1]$ while avoiding interception by the defender, and the goal of the defender is to intercept the attacker or prevent the attacker from reaching F. Using the same constraint encoding as before, the attacker winning condition for this stage is given by $O_F \vee O_D$, where

$$O_F = \{\exists t \in [0, T_1], x_a(t) \in F \wedge \mathbf{x}(s) \notin C \cup G_A, \forall s \in [0, t]\},$$

and O_D is modified with time horizon T_1. Define the set $G_F = \left\{\mathbf{x} \in \mathbb{R}^4 : \ x_a \in F\right\}$. Then the attacker target set in the flag capture stage can be defined as $F_J = G_F \cup G_D$, and the flag capture set can be computed as

$$F_A = \mathcal{RA}_{T_1}(F_J, C_J). \tag{13}$$

When both stages of the game are considered, it is clearly insufficient for the attacker to simply arrive in F. If the defender chooses a strategy such that the attacker arrives at the flag but the overall state $\mathbf{x}$ is in a configuration outside the flag return set R_A, then the defender can prevent the attacker from returning safely to R.

Instead, the attacker must reach the subset of the flag region for which flag return is possible. In terms of the sets introduced previously, this corresponds to the set of configurations $\tilde{R}_A = G_F \cap R_A$. The modified attacker objectives during the flag capture stage can be then written as $\tilde{O}_F \vee O_D$, where

$$\tilde{O}_F = \left\{\exists t \in [0, T_1], \mathbf{x}(t) \in \tilde{R}_A \wedge \mathbf{x}(s) \notin C \cup G_A, \forall s \in [0, t]\right\}.$$

From this, we can derive the modified attacker target set as $\tilde{F}_J = \tilde{R}_A \cup G_D$. The overall attacker winning region for capture-the-flag can be then computed as

$$W_A = \mathcal{RA}_{T_1}(\tilde{F}_J, C_J). \tag{14}$$

B. Synthesis of Winning Strategies

Over short time horizons, time-varying winning strategies can be synthesized for the flag capture and flag return stages of the game as per the methodology described in Section IV-B, using the results of the HJI reachability computations for W_A and R_A, respectively.

More specifically, let ϕ_1 and ϕ_2 be the viscosity solutions to the HJI equations used to compute W_A and R_A, respectively. Denote the partial derivatives of ϕ_i, $i = 1, 2$, with respect to the attacker and defender positions as $p_{u,i} = \frac{\partial \phi_i}{\partial x_a}$ and $p_{d,i} = \frac{\partial \phi_i}{\partial x_d}$. Given the form of the optimal Hamiltonian in equation (11), the minimax condition (7) is satisfied. Thus, the winning strategies of the attacker and defender during the flag capture and flag return phases can be derived from (5) and (8) as

$$\mu_i(\mathbf{x}, t) = -v_{a,max} \frac{p_{u,i}(\mathbf{x}, t - T_i)}{||p_{u,i}(\mathbf{x}, t - T_i)||_2}, \tag{15}$$

$$\tilde{\gamma}_i(\mathbf{x}, t) = v_{d,max} \frac{p_{d,i}(\mathbf{x}, t - T_i)}{||p_{d,i}(\mathbf{x}, t - T_i)||_2}, \tag{16}$$

for $t \in [0, T_i]$. Now consider an attacker strategy μ which initially selects controls according to μ_1, and then switches to μ_2 at the first time instant such that the trajectory $\mathbf{x}(\cdot)$ enters the set $\tilde{R}_A$. It can be verified that μ is a winning strategy for the attacker on the winning region W_A. In a similar manner, one can derive a winning strategy for the defender.

For cases in which capture-the-flag is to be played over long time horizons, namely for large values of T_1 and T_2, then it may be possible to approximate the long term strategies of the attacker and the defender using the procedures described in Section IV-C. In particular, suppose that the functions $\phi_1(\mathbf{x}, -t)$ and $\phi_2(\mathbf{x}, -t)$ converge to some functions $\phi_1^*(\mathbf{x})$ and $\phi_2^*(\mathbf{x})$ as $t \to \infty$. Denote the partial derivatives of ϕ_i^* as $p_{u,i}^* = \frac{\partial \phi_i^*}{\partial x_a}$ and $p_{d,i}^* = \frac{\partial \phi_i^*}{\partial x_d}$. Then the long term strategies for the attacker and the defender can be approximated, using equations (9) and (10), as the stationary strategies

$$\mu_i^*(\mathbf{x}) = -v_{a,max} \frac{p_{u,i}^*(\mathbf{x})}{||p_{u,i}^*(\mathbf{x})||_2}, \tag{17}$$

$$\tilde{\gamma}_i^*(\mathbf{x}) = v_{d,max} \frac{p_{d,i}^*(\mathbf{x})}{||p_{d,i}^*(\mathbf{x})||_2}. \tag{18}$$

The methods presented above are sufficient to construct a solution to the game of 1 vs. 1 capture-the-flag, giving both the initial conditions that lead to victory for each agent as well as the feedback strategies necessary to achieve victory. The following sections will illustrate the approach in simulation, and then show how these computations can be used in experimental scenarios with human agents.

VI. Simulation Results

The HJI reachability solution to capture-the-flag is illustrated here via an example with $v_{a,max} = v_{d,max} = 1$, $r_f = 1$, $r_c = 0.5$, and $T_c = T_r = 12$. The value function is computed using the Level-

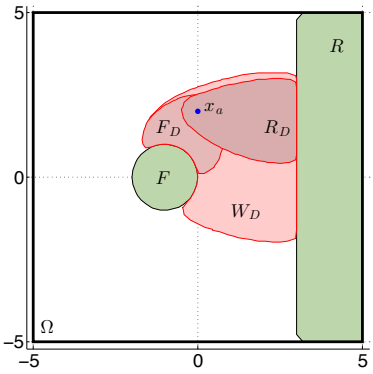

Fig. 3. **The overall winning region W_D of the defender for $x_a = (0, 2)$, with R_D and F_D super-imposed for comparison.**

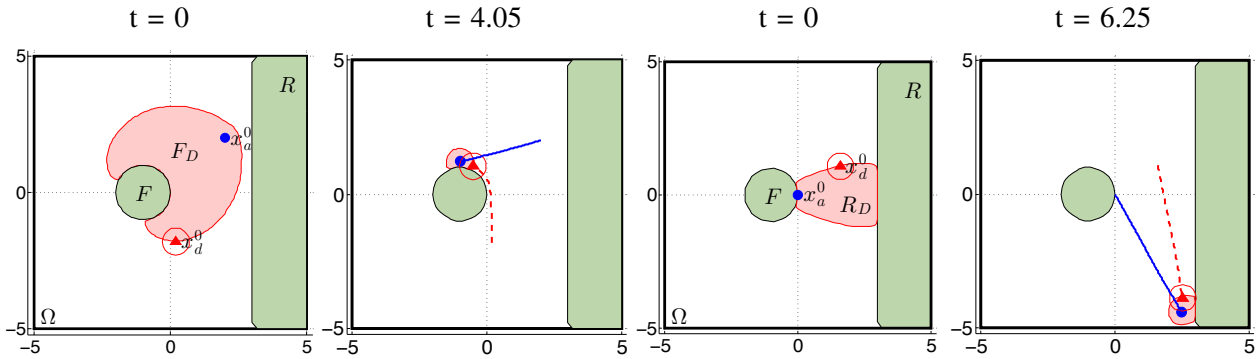

Fig. 4. **Two simulations showing cases where the defender (red triangle, dashed line) successfully intercepted the attacker (blue circle, solid line). (a) shows a scenario for flag capture only, with the resulting trajectories shown in (b). (c) and (d) show a similar scenario for flag return.**

Set Toolbox [22]. For this particular set of parameters, the value functions for the two stages of the game both converge to fixed points after about 6 time units. The actual winning regions lie in the 4-D joint configuration space $\mathbf{x}$, but for visualization purpose, slices are shown in 2-D with x_a fixed. The defender-winning sets F_D, R_D and W_D, corresponding to configurations from which the defender can always prevent the attacker from achieving the desired objectives, are shown in Figure 3, with the attacker fixed at $(0, 2)$. Observe that W_D is much larger than simply $F_D \cup R_D$, reflecting strategies where the defender uses the time during the flag capture stage to arrive at a configuration that blocks the attacker's return path.

Using the reachable sets, simulations were conducted in which the attacker and defender chose controls according to the approximate long-term strategies discussed in Section V-B. Figure 4 shows two example scenarios, one for the flag capture stage and the other for the flag return stage. In both cases, the defender

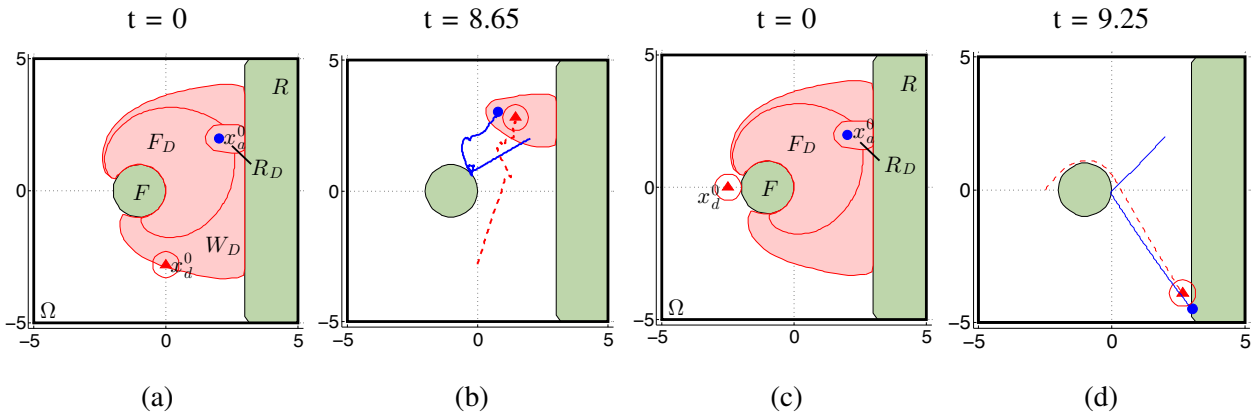

Fig. 5. **Two scenarios showing combined winning regions W_D for both game modes. In (a) the defender (red triangle, dashed line) started inside W_D and successfully prevented the attacker (blue circle, solid line) from returning with the flag, as seen in (b). (c) shows a case where the defender began outside of W_D, and was unable to prevent the attacker's successful flag capture and return, as shown in (d).**

started within the winning set and successfully intercepted the attacker. More interesting scenarios involving the interplay between the flag capture and flag return stage are shown in Figure 5. In Figure 5(a), the defender started within the overall winning set W_D, but outside the set F_D. The trajectories of the simulation in Figure 5(b) show that the defender did not try to prevent flag capture. Instead it used the time it took for the attacker to reach the flag to get into position to prevent flag return, thereby winning the game. On the other hand, when the defender began outside W_D, as in Figure 5(c), the simulated trajectory in Figure 5(d) shows that the attacker was able to successfully capture the flag and return to the safe region, without being intercepted by the defender at any point. Note how in this example the attacker did not take the shortest direct path to F, but rather reached F at a point that is closer to R, making the subsequent return path shorter and avoiding the defender.

The computation time of the reachable sets is strongly tied to the grid size and the numerical scheme for obtaining the spatial derivatives. With 45 points in each dimension, each set took approximately 1 hour to compute on an Apple Macbook Pro laptop with a 2.66 Ghz Core i7 processor and 8 GB of RAM. Sets with 25 points in each dimension can be computed in as little as 4 minutes.

It should be noted that some numerical errors are inevitable due to the necessity of solving the HJI PDE on a discrete grid. In particular, the numerical differentiation scheme is poorly equipped to handle sharp set boundaries, corresponding to discontinuities in the spatial derivative. For this reason, it was observed that near the intersection of F_D with F, the zero sub-level set sometimes slightly under-approximated the stop capture set, resulting in defender trajectories that began just outside of F_D (below the grid resolution) that nevertheless captured the attacker. More accurate solutions can be obtained with finer

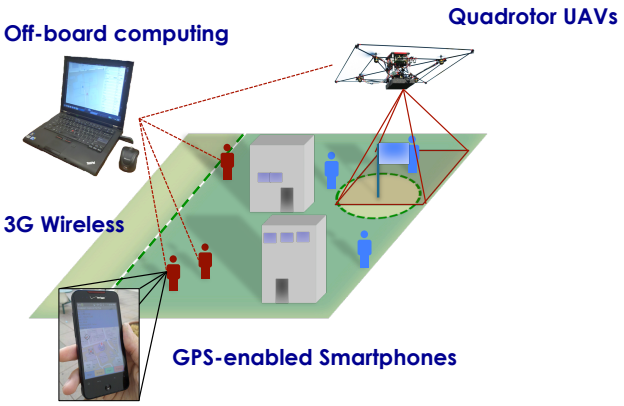

Fig. 6. **Schematic illustration of the BEARCAT testbed, showing the different components.**

spatial discretization, at the cost of higher computational load. Another possible approach is to over-bound the numerical error by some small $\epsilon > 0$ and either choose the $-\epsilon$ or $+\epsilon$ level set boundaries to ensure winning conditions for the attacker and defender, respectively.

VII. Experimental Results

The reachability method presented above was experimentally validated in a series of tests using the **BE**rkeley **A**utonomy and **R**obotics in **CA**pture-the-flag **T**estbed (BEARCAT), a system designed specifically to evaluate the use of game theory, path-planning, and control algorithms in the context of human agents working with automated systems in an adversarial game environment. BEARCAT features a group of GPS-enabled smartphones, off-board computation, and UAVs connected via WIFI and the 3G network, enabling agent movement tracking, information display, and coordination with the UAV assets. An illustration of the BEARCAT platform and its components is shown in Figure 6. The goal of the experiments was to evaluate the reachability solutions in a realistic environment, in particular the effects of their use by human agents, possibly with limited state information.

The tests were performed in various sections of the UC Berkeley campus, involving only the flag return stage of the game both with and without UAVs. As the main purpose was to test the use of the reachable sets, only a single stage of the game was considered in order to shorten testing time and simplify the testing logistics. When applicable, individual agents are equipped with HTC Incredible smartphones displaying the positions of the two agents as well as visualizations of the reachable sets and optimal inputs. Figure 7 shows a screenshot of a phone from the attacker perspective.

Tests were conducted under a range of different settings in terms of whether all of the agents receive reachability guidance, and also whether full position information is provided to each agent. When

receiving reachability guidance, each agent could view the opponent's winning set given the agent's own current position. As the optimal input is always to move at the maximum speed, the optimal input is displayed as a suggested heading. For example, in Figure 7 the reachable set displayed is the set of winning defender positions relative to the current attacker position. The optimal heading is displayed as a circular ring with an arrow in the direction of desired movement, against which the agent can compare the current heading of the phone.

The solution to the game was pre-computed, with the 4-D value function and partial derivatives stored on a PC laptop ground station that communicated with the phones via 3G wireless. Agent GPS positions and heading were transmitted to the laptop, which then performed the appropriate interpolation and transmitted the 2-D reachable set information and optimal direction to each agent as required.

Three sets of experiments were performed to validate the use of reachability information in the game. The first set of trials was played in a small, open area with no obstacles with the agents walking, and was meant to test the feasibility of the phones and the reachability computations. The second set was played on a larger area with a number of buildings serving as obstacles with running agents, and was meant to evaluate the game in a more realistic capture-the-flag game setting. The final set of trials explicitly considered the use of the reachability tools when full state information was not available. In these experiments, a quadrotor UAV performed search on behalf of the attacking agent, using the reachability information to prioritize the search. Here, sensing was simulated and state information about opposing agents was only transmitted when in line of sight.

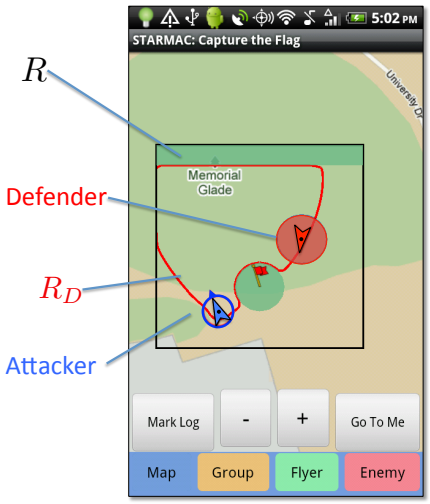

Fig. 7. **Screenshot of the attacker's phone, showing the agents, the return region R, and the defender's winning region R_D with respect to the current attacker position. The blue ring around the attacker shows the attacker's optimal input, displayed as a direction to travel at max speed.**

A. Small Area Trials

Several trials were conducted in a small, obstacle-free area to evaluate the phone technology and communications architecture. For these tests, the phones communicated with the laptop ground station via WIFI, and the agents were within visual contact of each other at all times. In these tests, the game region was defined as a 50m x 50m square, with the return region defined as a 5m deep strip running the width of the game region at the northern (upper) portion of the region (see Figure **??**). The speed of both agents was limited to $3m/s$, and the agents were instructed to maintain a walking pace. Trials were conducted where reachability guidance was provided (a) to both agents, and (b) to one agent only. These tests demonstrated that the hardware could provide reachability guidance for the agents in real time, and that the reachability information could be used to play the game.

Four trials were conducted with both agents receiving full reachability information, two with the defender in a winning initial condition and two with the attacker in a winning initial condition. In all cases, the game played out as predicted, as the agents were able to play according to their optimal inputs, resulting in victory or defeat according to the initial conditions. As the game is deterministic, when both agents play optimally the result of the game is determined completely from the initial conditions. A more interesting case is one in which one agent has access to the reachability information, and the other agent is left to play freely, without guidance. For the purposes of this discussion, the agent with reachability information will be referred to as the aided agent, and the agent without will be referred to as the unaided agent.

Several trials were carried out in this manner, varying the aided and unaided agents as well as the initial conditions. Trials where the aided agent was in a winning position proved predictably uninteresting; more illuminating results emerged from tests where the aided agent began in a losing configuration. In particular, an interesting new strategy emerged when the attacking agent was the aided agent and in a losing initial position. In the first trial, the defender chose an incorrect intercept angle and was unable to intercept the attacker. However, as the optimal headings reported by the reachability information are deterministic, on the subsequent trial the defender made an adjustment and was able to successfully intercept the attacker. This led the attacker to innovate by introducing an element of uncertainty into the game by "faking out" the defender. This is illustrated in Figure 8.

For the third trial, shown in Figure 8, the attacker did not initially follow the optimal heading and instead moved in the opposite direction, as seen in Figure 8(b). The MATLAB plot for each snapshot shows the position of the attacker (blue circle), the defender (red triangle), the flag region (green circle), and the return region (green rectangle). The attacker was attempting to reach R at the upper (northernmost) extreme of the game domain, defined by the green rectangle. The pink region shows the defender's winning

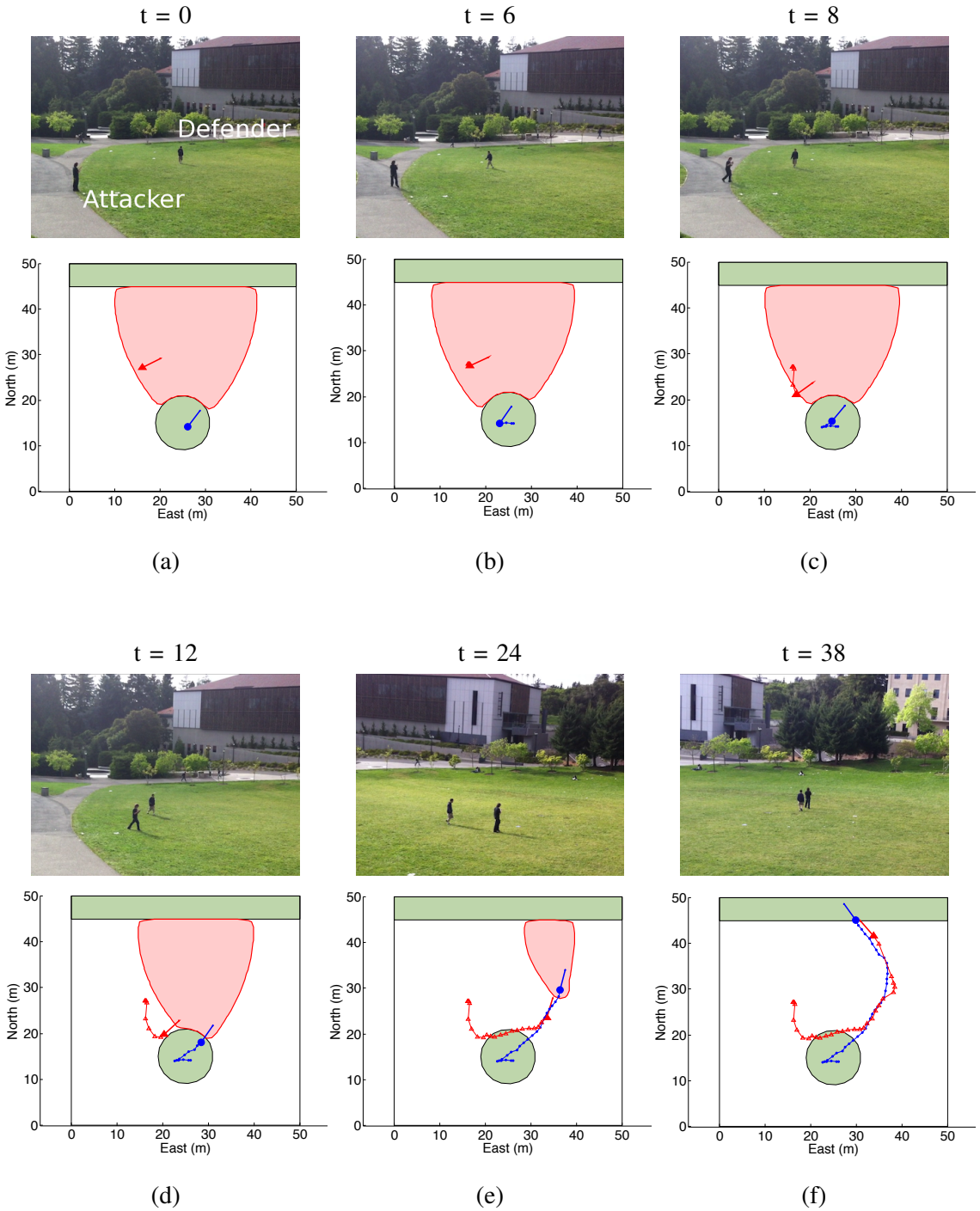

Fig. 8. **Sequence of agent positions and reachable set data where the defender (red triangle) did not have reachability information, but did begin the game within the defender winning region. The attacker (blue circle) moved unpredictably, causing the defender to move outside the winning region, resulting in an attacker victory.**

region R_D plotted for the current attacker position. This unpredictable movement of the attacker caused the defender to exit the winning region R_D, as seen in Figure 8(c), allowing the attacker to revert to following the optimal heading and subsequently winning the game. The visual display of reachability information allowed the attacker to instantly ascertain whether the "fake-out" maneuver had been successful. Once the attacker could see that the defender had exited R_D, the optimal heading could be directly followed to ensure victory. Two more trials were conducted, and in each case the attacker was able to successfully maneuver past the defender using this strategy. Note however that this maneuver was only successful with asymmetric reachability guidance.

This maneuver highlights an important aspect of the reachability tools. Although the attacker could not rely upon the reachability optimal heading when starting the game in a losing configuration, the visualization gave the agent enough information to decide when the heading recommendation was valid. The solutions not only provided optimal inputs to the agents, but the reachable set visualizations also provided contextual information that let the agent know when the automation recommendations could and could not directly provide a winning solution.

B. Large Area Trials

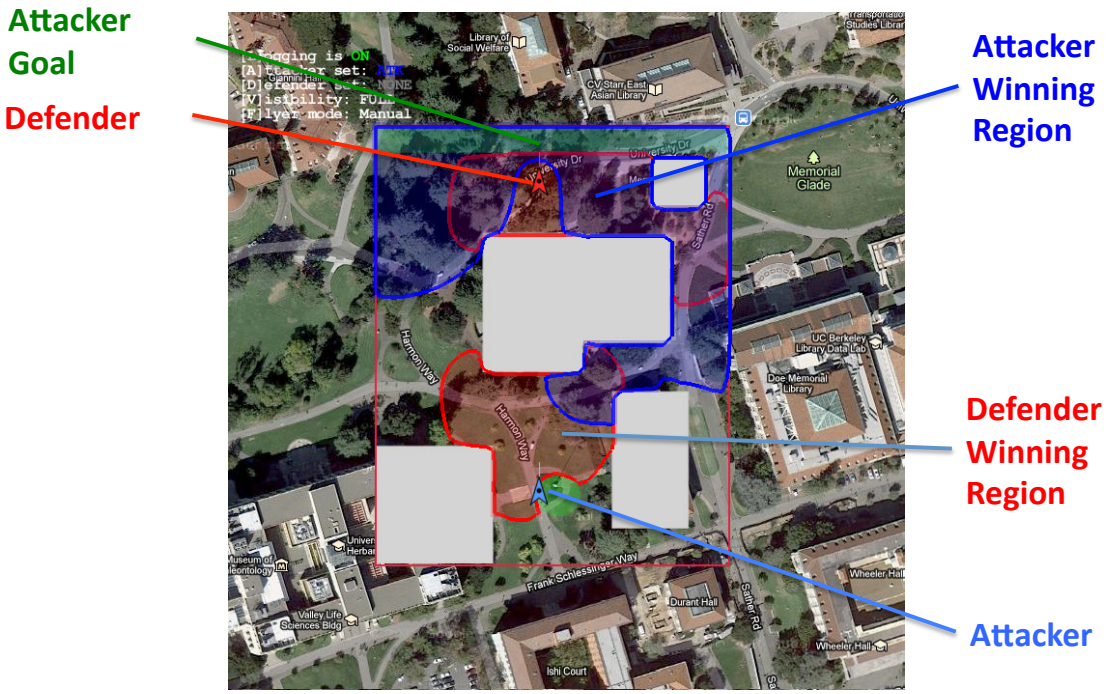

Fig. 9. **Screenshot of the laptop showing the game area with building obstacles, the agents, and their respective winning regions with respect to the other agent (map image courtesy of maps.google.com).**

The second set of trials was carried out in a larger region, with buildings serving as obstacles to movement. The main purpose of these experiments was to validate the use of the reachability information in a larger, more realistic game environment with greater agent speeds. Although agent positions were still displayed on the phones, the agents could not always see each other visually. The game area was approximately 200m x 180m, with a number of buildings serving as obstacles. Agent speeds for these trials were capped at $5m/s$, and the agents were permitted to run while playing. Figure 9 shows a screen capture from the ground station, displaying the game region with building obstacles, the positions of the two agents, and the respective winning regions. The blue region displays $R_A(x_d)$, the set of attacker positions which are winning relative to the current defender location, and the red area is $R_D(x_a)$, the set of defender positions that are winning relative to the current attacker location.

The results of the experiments demonstrated that the reachability support was still useful and valid in the presence of obstacles, and at higher speeds, given that the assumptions made in the computations held. However, the experiments also showed that complications as a result of the more realistic game conditions can introduce some uncertainty into the game, which cannot be accounted for via the current framework.

Communications delay and failure were major sources of uncertainty. In these more complex tests, phone-to-phone and phone-to-laptop communications occurred over the 3G wireless network, which was not as reliable as WIFI and sometimes subject to delays or packet loss. The terrain was another factor, as the game was played in a much more varied area than before and smaller terrain features such as fences and ledges could not be easily identified from overhead imagery. Although these difficulties combined to reduce the efficiency and accuracy of the reachability guidance, the reachability solutions nonetheless played a useful role in the game. In fact, the uncertainty introduced by these factors made for a much more exciting game with more room for human innovation.

The combination of these factors had the interesting result of making gameplay much more dependent on deception and managing opponent information, with the reachability tool used as a decision-aid rather than a primary guide. As in the “fake-out” maneuver, the attacker could win even from a defender-winning initial configuration by maneuvering the defender into a losing position. However, since both agents had reachability guidance this could only be accomplished by taking advantage of the uncertainties in the agent positions. Figure 10 shows an example of such a game. The initial condition, seen in Figure 10(a), was actually an attacker-winning configuration. However, the agents were not in line-of-sight of each other, and with the communications issues the attacker could not completely trust the information displayed. Instead, both agents moved to get better visibility, shown in Figure 10(b), bringing the two agents into direct view of each other. Note that the attacker stayed close to the southwest corner of the central

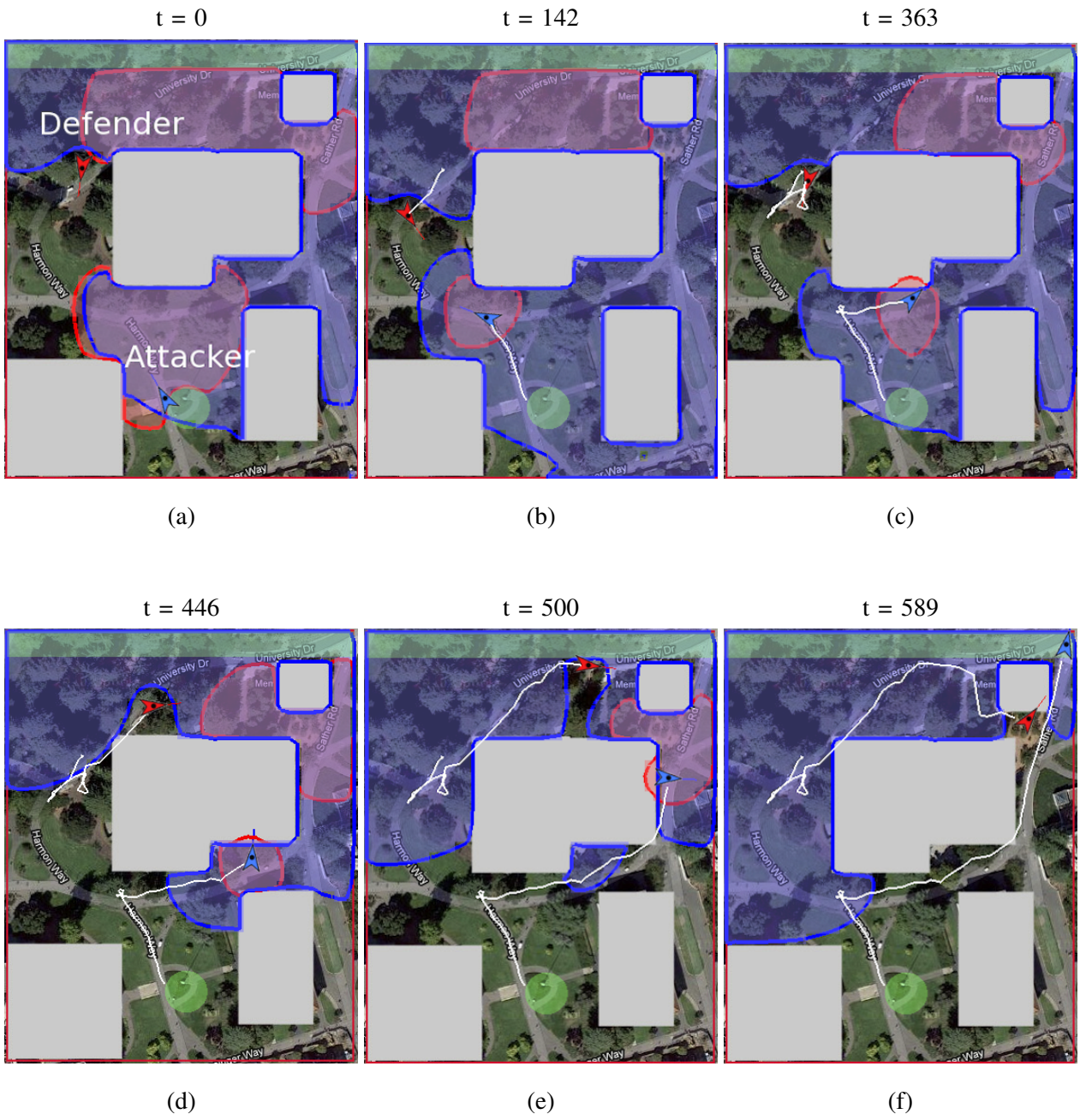

Fig. 10. **Sequence of screenshots from the ground station laptop showing the game played in a large area with obstacles, at full running speeds, with the attacker in blue, starting from the bottom of the screen, and the defender in red, at the top (map images courtesy of maps.google.com).**

building, which, although not visible from the overhead views, was a fairly cluttered area. This caused the defender to move south (down) in an attempt to better determine the location of the attacker, gambling on being able to move back into a winning configuration before the attacker could make a decision to go for the target set. Instead, this move rendered the defender more fully visible to the attacker, who was also able to visually establish that the defender was correctly reported by the phone to be well outside of $R_D(x_a)$. The attacker, now confident of victory, moved toward the target in the opposite direction, as seen in Figure 10(c), and successfully arrived at R without being intercepted, as shown in Figures 10(d)-(f).

C. UAV Search Experiment

The final set of experiments involved the use of a quadrotor UAV performing autonomous search on behalf of the attacking agent. The capture-the-flag problem presented and solved in the preceding sections assumes that both agents have full knowledge of the state of the opposing agent, a condition that does not typically hold. In an actual adversarial game, the agents would have to rely on their own sensing and any external sensing assets to locate the other agent. Indeed, as shown in the experimental results above, communications errors can result in loss of information even when the agents are broadcasting their positions as part of the game. However, the reachability information computed for the fully observed game can still serve a useful purpose when the opposing agent's position is unknown.

Consider the problem of controlling a UAV to assist an agent by searching for that agent's opponent. The opposing agent may be located anywhere within a large expanse of unobserved space, but not all unobserved positions are equally likely to be occupied. A probabilistic search method may be attempted, but it is difficult to create a distribution of opponent locations without a model of the opponent's likely strategies. Instead, the reachable sets for the fully observed game can be used to guide the search, as the reachability information can be used to determine the winning region for the opponent given the agent's current position. The opponent only poses a danger to the agent if it is found within this set, thus the opponent's winning reachable set can be used to guide the UAV in its search pattern. The search strategy is briefly presented below, along with experimental results.

1) Search Strategy: The set of positions visible to the attacker at a time t is limited by line-of-sight, and is denoted $V_A(x_a, t)$. Likewise, the set of positions visible to the defender is denoted $V_D(x_d, t)$. For notational simplicity, the dependency of the visibility regions on agent position and time will be assumed and not explicitly written. Let the position of the UAV be defined as $x_q \in \mathbb{R}^2$, with dynamics $\ddot{x_q} = a$. The visibility region V_Q of the UAV is defined as a circle centered around x_q with radius r_q. The joint visibility region V_J of the attacker and the UAV is then $V_J = V_A \cup V_Q$.

Now, let P_D represent the set of possible defender positions, from the perspective of the attacker and the UAV. The attacker and UAV will be referred to jointly as the attacking side. At any time, the boundary

of P_D grows no faster than $v_{d,max}$, representing how fast the defender can move, and any intersection with V_J is removed from P_D. The problem for the UAV is to search the area encompassed by P_D and locate the defender before the defender can intercept the attacker or successfully block the attacker from reaching its goal. To accomplish this, the reachable sets generated in the reachability solution to 1 vs. 1 capture-the-flag are used to prioritize the search areas.

The search strategy is as follows. Given an initial set P_D^0 where the defender may be, the set of all possible defender locations satisfies the following properties:

1) The boundary of P_D grows into unobserved space at a rate of $v_{d,max}$
2) The intersection $P_D \cap V_J$ is always empty

Note that unobserved in this case means unobserved by the attacking side. This set P_D may be computed and tracked using a level set representation, as presented in [27], and the visibility regions for the agents may be computed using the level set representation discussed in [28].

The search target point for the UAV is selected according to a hierarchy. The most dangerous positions for the attacker are those in the set $Z = W_D \cap V_A$, which are points where the defender has a winning input, *and* can see the attacker. From these points, the defender may follow the reachability-derived optimal input, and has a chance of successfully preventing the attacker from reaching its goal. The second set of points are points in W_D that are outside V_A. Let $\tilde{Z} = W_D \setminus V_J$ be the set of invisible points in W_D. From these points, the defender may have a winning input, but the attacker is not in sight, and thus the defender must guess at the correct input.

The priority of the UAV is to first search the set of defender positions that may reach Z without being seen. If there are no such points remaining, then the UAV should next search all positions from which the defender may reach $\tilde{Z}$. If there are no points that may reach either Z or $\tilde{Z}$, then there are no possible defender positions that may reach W_D. In this case, the UAV may search the remaining space according to distance from the UAV and the evader.

2) Search Experiments: Several experiments were carried out to evaluate the reachability-guided search algorithm using the full BEARCAT platform with a quadrotor UAV. The experiments were carried out in the same portion of the Berkeley campus as the large area trials above. Sensing was simulated, so that whenever the defender entered V_A both the defender and attacker received information about each other's locations, and when the defender entered V_Q the attacker was alerted to the defender position. Otherwise, the agents had no information about each other's positions.

The course of one of these experiments is illustrated in Figures 11(a)-(f). The initial condition for the game is shown in Figure 11(a), with the attacker and UAV beginning the game in the lower center of the game region and the defender on the opposite side of the large central building. The attacker and UAV

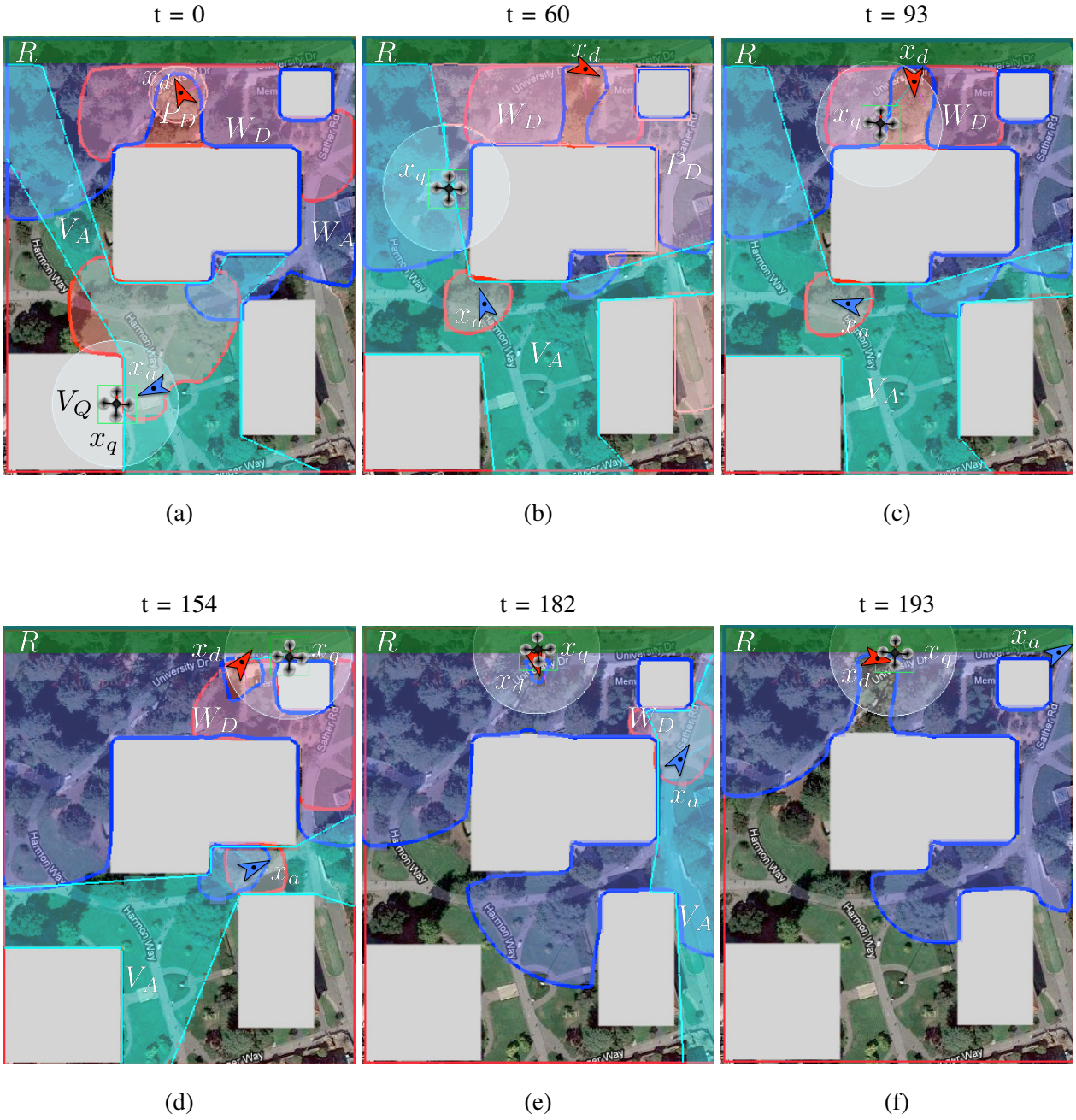

Fig. 11. **Sequence of screenshots from the ground station laptop showing a full search game with a quadrotor UAV (map images courtesy of maps.google.com).**

visual regions V_A and V_Q are illustrated in teal and white, respectively. The defender winning region W_D is the large red region centered on the attacker, and the attacker winning region W_A is the blue region centered on the defender. The set of possible defender positions P_D, from the perspective of the attacker, is the expanding pink circle centered on the defender.

Figure 11(b) shows the game 60 seconds after beginning. The attacker has moved up to the corner of

the building in an attempt to maximize visibility of the defender while still staying relatively hidden. P_D has expanded to fill up the unobserved space, and the UAV has begun its search. Note that the attacker has moved in such a way that no part of P_D can expand into the set $W_D \cap V_A$ without first passing through V_A, thus the UAV was prioritizing the area of W_D that is currently unobserved. The defender has pulled back away from the building to reduce the size of W_A, hoping to minimize the chance that the attacker will pass into W_A without being detected. At the moment, neither agent had knowledge of the other agent's position, and both were playing cautiously and unable to commit to moving toward either side of the central building.

The UAV has just detected the defender in Figure 11(c). As the UAV was programmed to track the defender once detection occurs, the defending agent was also aware of being found. Thus the defender began moving unpredictably in a patrolling motion, shifting to the eastern (right) side of the game domain, as seen in Figure 11(d). The attacker had been moving eastward (toward the right) and was temporarily blocked by the motion of the defender, but after some time the defender was forced to patrol back to the west (left) since the position of the attacker was unknown, as shown in Figure 11(e). With the UAV reporting back the position of the defender, the attacker was able to commit to moving up toward the target set from the eastern (right) side of the domain, and eventually reached the target set at the upper portion of the game domain, as shown in Figure 11(f).

This example is prototypical of the experiments that were conducted. By using the reachability information to guide the search, the UAV was able to quickly move to the most critical area and locate the defender, giving the attacker the information advantage needed to successfully win the game.

VIII. Conclusions and Future Work

The reachability-based approach presented in this paper gives a complete solution to the 1 vs. 1 game of capture-the-flag. This technique gives the set of initial conditions from which each agent can win, and also gives a method for computing optimal inputs for each agent. The primary strengths of this approach from a theoretical standpoint are that it finds complete, optimal solutions, that it can be used for arbitrary game geometries, and that it naturally handles multi-stage games.

The reachable sets generated by the solution method are also useful as visual guides for human agents and for prioritizing UAV search. For the humans, in addition to providing optimal direction of movement, the sets can give a sense of context as to when those recommendations can be relied upon to assure victory. For UAVs, the reachability results reduce the possible set of search states to only those that may allow an opponent to win the game. These results show that the solutions can be useful even when the assumption of perfect state information is relaxed in a realistic game scenario.

There are several directions in which the work can be extended. The first involves addressing computational complexity of the HJI reachability approach, arising from a grid-based value function approximation. Under a uniform grid, the number of grid nodes grows exponentially with the dimension of the joint state space. This becomes problematic as one considers scenarios with a large number of agents. Extending the reachability concepts to higher-dimensional systems and multi-agent games is currently a topic of active research.

The approach presented here can also naturally be generalized to a symmetric, two-sided game of capture-the-flag, where the two agents may choose to switch between playing the roles of the attacker and defender. This scenario would require the reachability value function to be computed for each combination of roles. Once computed, these value functions can be directly used by each agent to determine the optimal selection of roles to ensure victory.

The final direction is to expand upon the UAV experiments and formally address games with uncertainty and limited information. This could require formulations considering information states, which can greatly increase the computational demands on any solution. Further results in these directions have the potential of providing nuanced solutions to more complex scenarios.

ACKNOWLEDGMENTS

The authors would like to thank Scott Hoag, Andrew Sy, Zhengyuan Zhou, Steven Xu, and Catherine Pavlov for their assistance in the development of the BEARCAT test-bed and their participation in the experiments. In addition, 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).

REFERENCES

[1] 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.

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

[3] D. Bhadauria, K. Klein, V. Isler, and S. Suri, "Capturing an evader in polygonal environments with obstacles: The full visibility case," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1176–1189, 2012.

[4] 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.

[5] 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.

[6] 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.

[7] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[8] M. Erdmann and T. Lozano-Pérez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 477–521, 1987.

[9] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, p. 760, 1998.

[10] T. Fraichard and H. Asama, "Inevitable collision states - a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[11] J. Van Den Berg and M. Overmars, "Planning time-minimal safe paths amidst unpredictably moving obstacles," *International Journal of Robotics Research*, vol. 27, no. 11-12, p. 1274, 2008.

[12] R. D'Andrea and M. Babish, "The roboflag testbed," *Proceedings of the American Control Conference*, vol. 1, pp. 656–660, 2003.

[13] M. Earl and R. D'Andrea, "A decomposition approach to multi-vehicle cooperative control," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 276–291, 2007.

[14] G. Chasparis and J. Shamma, "Linear-programming-based multi-vehicle path planning with adversaries," *Proceedings of the IEEE American Control Conference*, pp. 1072–1077, 2005.

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

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

[17] 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.

[18] 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.

[19] 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.

[20] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Stanford University, 2002.

[21] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.

[22] I. Mitchell, *A Toolbox of Level Set Methods*, 2009, http://people.cs.ubc.ca/ mitchell/ToolboxLS/index.html.

[23] J. Sprinkle, A. D. Ames, J. M. Eklund, I. M. Mitchell, and S. Shankar Sastry, "Online safety calculations for glide-slope recapture," *Innovations in Systems and Software Engineering*, vol. 1, pp. 157–175, 2005.

[24] J. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *International Journal of Robotics Research*, Jan 2011.

[25] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349 – 370, 1999.

[26] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949 –970, Jul 2000.

[27] E. Frew, "Combining area patrol, perimeter surveillance, and target tracking using ordered upwind methods," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[28] Y. Tsai, L. Cheng, S. Osher, P. Burchard, and G. Sapiro, "Visibility and its dynamics in a PDE based implicit framework," *Journal of Computational Physics*, vol. 199, no. 1, pp. 260–290, 2004.