

A Differential Game Approach to Planning in Adversarial Scenarios: A Case Study on Capture-the-Flag

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

Abstract—Capture-the-flag is a complex, challenging game that is a useful proxy for many problems in robotics and other application areas. The game is adversarial, with multiple, potentially competing, objectives. This interplay between different factors makes the problem complex, even in the case of only two players. To make analysis tractable, previous approaches often make various limiting assumptions upon player actions. In this paper, we present a framework for analyzing and solving a two-player capture-the-flag game as a zero-sum differential game. Our problem formulation allows each player to make decisions rationally based upon the current player positions, assuming only an upper bound on the movement speeds. Using Hamilton-Jacobi reachability analysis, we compute winning regions for each player as subsets of the joint configuration space and derive the corresponding winning strategies. Simulation results are presented along with implications of the work as a tool for automation-aided decision-making for humans and mixed human-robot teams.

I. INTRODUCTION

Modern tools in robotics and automation offer great promise in a number of application areas ranging from the military [1] to air traffic control [2] to commercial logistics [3]. Many of these scenarios will involve robots or complex automation coordinating with human agents and operators, requiring tools that not only generate solutions for difficult automation tasks, but also make those solutions comprehensible and useable to the humans who will be responsible for the system [4], [5]. The game of capture-the-flag is complex and adversarial, with opposing players and competing objectives that require decisions to be made at multiple levels of play. It is an excellent proxy for studying the tools needed to make advanced automation systems a reality.

Capture-the-flag is a two-sided game played by teams of mobile players constrained to remain within a finite game area divided into two regions allocated to each side. Each team owns a flag located in their territory that can be captured by an opposing player moving to within some small distance of the flag. However, while inside the opposing team's territory, a player can be intercepted if an opponent moves within some distance of that player. The objective

of the game is to capture the opponent's flag and return to safety while protecting one's own from the enemy. Each flag is also typically surrounded by a forbidden region to prevent a defending player from simply staying at the flag location, making flag capture impossible.

Capture-the-flag in robotics and automation has been explored in a variety of ways, most notably in the Cornell RoboFlag competition, where two opposing teams of mobile robots play a game of capture-the-flag while directed by human players [6]. The competition has produced a number of results dealing with robot control and planning [7], [8], [9] as well as issues relating to human-robot interaction [10].

The complexity of a game like capture-the-flag defies easy analysis. Unlike standard pursuit-evasion games, the goal of each player is not simply to evade or pursue, but also to attack and defend the flag. Many of the previous results focus on the multiple-agent coordination aspect of the game while making simplifying assumptions about the adversarial aspect. A defensive strategy for intercepting multiple attackers with multiple defenders was developed in [7] using mixed-integer linear programming, with the assumption that attackers proceed toward the goal in straight lines. Another approach is [11] to solve for optimal defender control actions using linear programming under an assumed attacker strategy in the form of a linear feedback law.

A similar approach is often taken in more complex pursuit-evasion games such as air combat, where the roles of the players may change over time. Here control strategies are usually generated for one side while assuming some model that predicts the actions of the opposing side. One approach uses nonlinear model-predictive control to generate control inputs for an unmanned aerial vehicle [12], while another uses approximate dynamic programming for a similar application [13]. In both cases, adversary actions are predicted based upon expected opponent strategies, with feedback and re-planning used to adjust for deviations from the predicted actions at run-time.

In this work, we formulate the two-player capture-the-flag problem as a differential game and describe an approach for characterizing the winning regions and winning strategies of each player using Hamilton-Jacobi reachability analysis. This formulation allows each player to choose their control actions rationally based upon the current positions of the two players, without the use of explicit prediction models. The application of Hamilton-Jacobi reachability under a differential game setting has found previous successes in aircraft collision avoidance [14], automated inflight-refueling [15], and autonomous quadrotor aerobatics [16]. In the case of

This work is supported in part by NASA (NNA06CN22A), AFOSR under MURI grant FA9550-06-1-0312, NSF under grant CNS-0931843, and ONR under MURI grant N00014-09-1-1051

H. Huang is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA haomiao@stanford.edu

J. Ding, W. Zhang, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA [{jding, weizhang, tomlin}@EECS.Berkeley.Edu">{jding, weizhang, tomlin}@EECS.Berkeley.Edu](mailto)

[†]These authors contributed equally to this work.

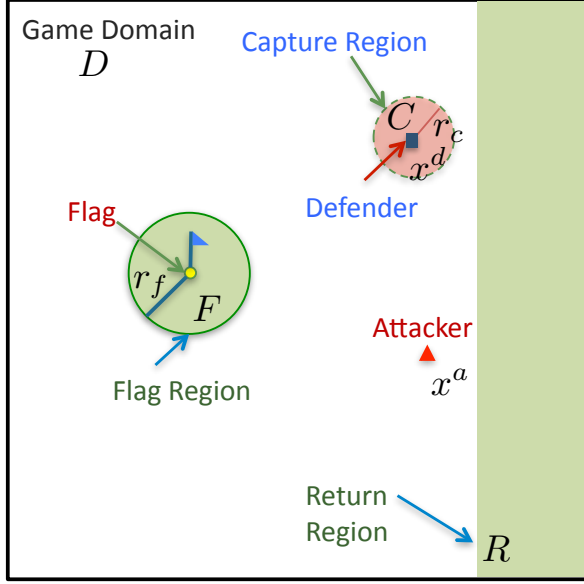


Fig. 1. The basic configuration of the capture-the-flag game.

capture-the-flag, this technique allows us to compute the boundaries of the winning regions for both players in the joint configuration space, as well as derive approximate winning strategies associated with these regions. This approach not only allows us to examine the game in a flexible and complete manner, but also produces clear and intuitive tools for informing human decision-making through visualizations of the winning regions and the suggested control actions [17].

The structure of this paper is as follows. We describe our problem formulation and give some geometric insights in Section II. The Hamilton-Jacobi solution is described in Section III, with computational and simulation results presented in Section IV. We conclude with discussions and future work in Section V.

II. PROBLEM FORMULATION

The problem considered in this paper is a simplified version of capture-the-flag with a single player on each team and full observability, where the objective of one player, called the attacker, is to capture a flag and return to a safe region, while the objective of the other player, called the defender, is to prevent the attacker from completing this task. The focus is on characterizing the sets of attacker and defender configurations for which there exists a winning strategy for either player, as well as finding the strategies that ensure those winning conditions.

The playing domain is assumed to be some rectangular set $D \subset \mathbb{R}^2$. The “safe” return region for the attacker is a rectangular strip $R \subset D$, and the flag is located in a circular region $F \subset D$ with radius r_f (see Figure 1).

A configuration of the game is described by the vector $(x^a, x^d) \in \mathbb{R}^4$, where x^a and x^d are the planar positions of the attacker and defender, respectively. 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. Thus, the set of permissible configurations is given by $\hat{D} = D \times D \setminus (F \cup R) \subset \mathbb{R}^4$. The equations of motion are given by $\dot{x}^a = u$ and $\dot{x}^d = d$, where u and d are the velocity inputs, constrained by speed limits $V_{a,max}$ and $V_{d,max}$ according to $\|u\|_2 \leq V_{a,max}$ and $\|d\|_2 \leq V_{d,max}$. For fairness, we assume that $V_{a,max} = V_{d,max}$. The attacker is considered to be tagged or intercepted by the defender if the defender comes within distance r_c of the attacker. This corresponds to a capture region $C = \{(x^a, x^d) \in \mathbb{R}^4 : \|x^a - x^d\|_2 \leq r_c\}$.

Victory for the attacking player is attained by meeting all of the following conditions, assuming the game takes place over some finite time horizon $[0, T_f]$:

- Flag capture: $x^a(T_c) \in F$ for some finite time T_c where $0 \leq T_c \leq T_f$
- Flag return: $x^a(T_r) \in R$ for some finite time T_r where $T_c \leq T_r \leq T_f$
- Avoiding defender capture: $x^a(t) \in D \wedge (x^a(t), x^d(t)) \notin C$ for all time $t \in [0, T_r]$

Victory for the defending player is achieved by preventing the attacker from achieving any of the above conditions by T_f while obeying the constraint $x^d \in D \setminus (F \cup R)$. It should be remarked that when T_f is small, the options of the attacking player may be severely restricted by the time limit. Correspondingly, this also leads to simplistic delaying strategies by the defending player. Motivated by this consideration, we will be specifically interested in scenarios where T_f is large, namely as T_f approaches ∞ .

It is assumed that the positions of both players are fully observable to each other, and that the defender input $d(t)$ may be a function of the attacker input $u(t)$ at each time t . The latter assumption corresponds to a choice in the order of play to prevent an infinite regression of second-guessing between the players. For the simple dynamics considered here, the opposite order of play leads to an equivalent problem formulation and identical solutions.

The goal of this work is to provide a solution to this two-player game of capture-the-flag in terms of both the winning regions and winning strategies for each player. For the attacker, we would like to determine a subset of initial configurations $W_A \subset \mathbb{R}^4$ for which there exists a feasible attacker control strategy for achieving the victory conditions, regardless of the control strategy of the defending player. Furthermore, for any configuration in W_A , we would like to determine a winning strategy that ensures the attacker objectives. 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 $\hat{D} \setminus W_A$.

In building up towards the full winning sets, we will also characterize the partial winning sets for the flag capture and flag return phases of the game. Some useful terminologies are summarized below.

- **Flag Capture Set F_A** : configurations from which the attacker can achieve flag capture while avoiding defender interception.

- **Stop Capture Set** F_D : $\hat{D} \setminus F_A$, 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 : $\hat{D} \setminus R_A$, where the defender can prevent flag return by the attacker.

III. HAMILTON-JACOBI REACHABILITY IN CAPTURE-THE-FLAG

Capture-the-flag falls naturally under the framework of differential games due to the competing objectives of the attacking and defending players [18]. This allows us to characterize the winning regions of the attacker and defender from the results of numerical Hamilton-Jacobi reachable set calculations [14], [19].

The numerical Hamilton-Jacobi approach has a number of advantages. Although it is possible to use geometric analysis to construct some solutions to the game for simple configurations of the game regions, the numerical approach allows arbitrarily complex geometries to be addressed, including obstacles. Moreover, this framework naturally allows for the sequential aspect of the game, where the attacker must first reach the flag zone and then subsequently return to safety. Such sequencing is very difficult to analyze otherwise.

As a further benefit, we can derive approximate winning control strategies for the respective players from the solutions of these computations. For details on the theoretical background behind the application of Hamilton-Jacobi equations to problems in optimal control and differential games, the interested reader is referred to [20].

A. Hamilton-Jacobi Reachability

We refer to the two players in a general differential game as Player I and Player II. We assume that the continuous dynamics are modeled by the ordinary differential equation

$$\dot{x} = f(x, u, d), \quad x(0) = x_0 \quad (1)$$

where $x \in \mathbb{R}^n$ is the system state, u is the control input of Player I, d is the control input of Player II, and x_0 is the initial condition. The input ranges of Player I and Player II will be denoted by \mathbb{U} and \mathbb{D} , respectively.

In terms of control strategies, Player I selects controls $u(t)$, $t \in [0, T]$, satisfying $u(t) \in \mathbb{U}$, possibly as a function of the state $x(t)$. Similarly, Player II selects controls $d(t)$, $t \in [0, T]$, satisfying $d(t) \in \mathbb{D}$. However, we allow this selection to be a function of both the state $x(t)$ and the control $u(t)$ of Player I. The set of permissible control strategies for Players I and II are denoted by $P(T)$ and $\Gamma(T)$, respectively.

Now consider sets $R, A \subset \mathbb{R}^n$ and time horizon $[0, T]$, we are interested in computing the set of initial conditions $x_0 \in \mathbb{R}^n$ for which there exists some choice of Player I control policy $\pi(T) \in P(T)$, such that regardless of the choice of Player II control strategy $\gamma(T) \in \Gamma(T)$, the state trajectory $x(\cdot)$ under model (1) satisfies $x(t) \in R$ for some $t \in [0, T]$ and $x(s) \notin A$, $\forall s \in [0, t]$. In other words, this is the set of initial conditions that can be controlled into R

within finite time by Player I, while avoiding A at all times, regardless of the control strategy of Player II. We denote this set by $\mathcal{RA}_T(R, A)$, and refer to it as the *reach-avoid* set over $[0, T]$.

Under suitable conditions, given in [14] and [19], the reach-avoid set can be obtained from the solution of a constrained Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE), using level set representation of sets. Under this representation, a set $G \subset \mathbb{R}^n$ is defined implicitly as the sublevel set of a function $\phi_G : \mathbb{R}^n \rightarrow \mathbb{R}$, such that $G = \{x \in \mathbb{R}^n, \phi_G(x) \leq 0\}$.

Let $\phi : \mathbb{R}^n \times [-T, 0] \rightarrow \mathbb{R}$ be the viscosity solution [21] of the following constrained terminal value HJI PDE,

$$\frac{\partial \phi}{\partial t} + \min \left[0, H \left(x, \frac{\partial \phi}{\partial x} \right) \right] = 0, \quad \phi(x, 0) = \phi_R(x) \quad (2)$$

subject to

$$\phi(x, t) \geq -\phi_A(x)$$

where the optimal Hamiltonian is given by

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

and ϕ_R, ϕ_A are the level set representations of R and A , respectively. Then by the argument presented in [14] and [19], the reach-avoid set is given by

$$\mathcal{RA}_T(R, A) = \{x \in \mathbb{R}^n, \phi(x, -T) \leq 0\}$$

A highly accurate numerical solution to equation (2) can be computed using the Level Set Toolbox for MATLAB [22].

B. Reachability for Capture-the-Flag

In this section, we characterize the attacker victory sets F_A , R_A , and W_A using the reach-avoid operator. The corresponding defender victory sets F_D , R_D , and W_D can be obtained via the relations described in Section II. Under the differential game setting, the attacker takes on the role of Player I, while defender takes on the role of Player II.

We first consider the flag return phase of the game, taking place over some time horizon $[0, T_r]$. The goal of the attacker during this phase is to arrive in the return region R within $[0, T_r]$, while avoiding capture by the defender. The rules require both players to stay within the game region D and the defender to remain outside of F and R . In the framework of a differential game, we encode the constraints for each player as part of the winning conditions for the other, so that the attacker wins if the defender violates its constraints and vice versa. Note that this formulation can also be used to encode other state constraints such as obstacles.

Let x be the joint configuration (x^a, x^d) , D^C the complement of D in \mathbb{R}^2 , and $G_A = \{x \in \mathbb{R}^4 : x^a \in D^C\}$, then the attacker winning condition are $O_R \vee O_D$, where

$$\begin{aligned} O_R &= \{\exists t \in [0, T_r], x^a(t) \in R \wedge x(s) \notin C \cup G_A, \forall s \in [0, t]\} \\ O_D &= \{\exists t \in [0, T_r], x^d(t) \in F \cup R \cup D^C \wedge x(s) \notin C \cup G_A, \forall s \in [0, t]\} \end{aligned}$$

Define the sets $G_D = \{x \in \mathbb{R}^4 : x^d \in F \cup R \cup D^C\}$ and $G_R = \{x \in \mathbb{R}^4 : x^a \in R\}$. Then from the above conditions, we can compute the flag return set as

$$R_A = \mathcal{RA}_{T_r}(G_R \cup G_D, C \cup G_A) \quad (3)$$

Now consider the flag capture phase of the game, with a time horizon of $[0, T_c]$. If the second phase of the game is ignored, then the goal of the attacker is to simply arrive in the flag region F within $[0, T_c]$ 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 phase is given by $O_F \vee O_D$, where

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

Define the set $G_F = \{x \in \mathbb{R}^4 : x^a \in F\}$. Then the flag capture set can be computed as

$$F_A = \mathcal{RA}_{T_c}(G_F \cup G_D, C \cup G_A) \quad (4)$$

However, when both phases 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 (x^a, x^d) is in a configuration outside the flag return set R_A , then the defender can prevent the attacker from returning safely to R . In order to ensure flag return, the attacker must reach the set of configurations $\tilde{R}_A = G_F \cap R_A$ during the first phase of the game. This corresponds to the modified attacker objectives during the flag capture phase $\tilde{O}_F \vee O_D$, where

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

From this, we can compute the attacker winning set as

$$W_A = \mathcal{RA}_{T_c}(\tilde{R}_A \cup G_D, C \cup G_A) \quad (5)$$

C. Control Strategies

Based upon the approach described in [23] and [24], we describe here a method for the synthesis of winning strategies in regions of the state space where the solution ϕ to (2) is differentiable. For pursuit-evasion games, the solution to the HJI equation is typically differentiable away from some singular surfaces where the winning strategies are not uniquely defined [18]. With arbitrarily small perturbations from these surfaces, the solution becomes differentiable, allowing application of the techniques described here.

Under this setting, for $x \in \mathcal{RA}_T(R, A)$, the control strategy that allows Player I to achieve the desired objectives regardless of the control strategy of Player II is described by

$$u^*(x, t) = \arg \min_{u \in \mathbb{U}} \max_{d \in \mathbb{D}} p(x, -t)^T f(x, u, d), \quad t \in [0, T] \quad (6)$$

where $p = \frac{\partial \phi}{\partial x}$. Similarly, for $x \notin \mathcal{RA}_T(R, A)$, the Player II control strategy that is guaranteed to prevent Player I from achieving the desired objectives is given by

$$d^*(x, t) = \arg \max_{d \in \mathbb{D}} p(x, -t)^T f(x, u^*, d), \quad t \in [0, T] \quad (7)$$

Given the numerical computation of the set $\mathcal{RA}_T(R, A)$, the derivatives of ϕ are not available in closed form. Nonetheless, approximate control laws can be obtained by temporal discretization, for example at sampling instants $kT, k = 0, 1, \dots$, and computing numerical derivatives of $\phi(x, -kT)$.

For the capture-the-flag problem, the optimal Hamiltonian is given by

$$H(x, p) = \min_{u \in \mathbb{U}} \max_{d \in \mathbb{D}} p_u^T u + p_d^T d$$

where $p_u = (p_1, p_2)$ and $p_d = (p_3, p_4)$. From this we can deduce the explicit winning strategies¹

$$u^*(x, t) = -V_{a, \max} \frac{p_u(x, -t)}{\|p_u(x, -t)\|_2} \quad (8)$$

$$d^*(x, t) = V_{d, \max} \frac{p_d(x, -t)}{\|p_d(x, -t)\|_2} \quad (9)$$

Over short time horizons, time varying strategies can be synthesized for the flag capture and flag return phases of the game as per equations (8) and (9), using W_A and R_A , respectively. Under the attacker control strategy constructed from W_A , the trajectory $x = (x^a, x^d)$ is guaranteed to enter \tilde{R}_A within $[0, T_c]$, satisfying the flag capture objective. At the time instant when $x(t) \in \tilde{R}_A$, the attacker can switch to the control strategy constructed from R_A and safely return to R , thus winning the game.

In certain cases where the solution to (2) converges, namely $\lim_{t \rightarrow -\infty} \phi(x, t) = \phi_\infty(x)$ for some $\phi_\infty : \mathbb{R}^n \rightarrow \mathbb{R}$, time-invariant control strategies approximating (6) and (7) can be constructed. Specifically, suppose ϕ is computed numerically at time steps $-kT_s, k = 0, 1, \dots$, we can select an index k_0 at which $\|\phi(\cdot, -k_0T_s) - \phi(\cdot, -(k_0+1)T_s)\|_\infty < \epsilon$, for some tolerance $\epsilon > 0$. If ϵ is chosen sufficiently small, we can reasonably assume that $\phi(\cdot, -t) \approx \phi(\cdot, -k_0T_s), \forall t > k_0T_s$. This allows us to approximate the long-term strategies of the two players by

$$u^*(x) \approx \arg \min_{u \in \mathbb{U}} \max_{d \in \mathbb{D}} p(x, -k_0T_s)^T f(x, u, d) \quad (10)$$

$$d^*(x) \approx \arg \max_{d \in \mathbb{D}} p(x, -k_0T_s)^T f(x, u^*, d) \quad (11)$$

For our application, the sets R_A and W_A indeed converge numerically for sufficiently large T_r and T_c , leading to approximate long-term strategies for the attacker and defender.

IV. COMPUTATIONAL RESULTS

The proposed reachability-based approach is illustrated through an example with $V_{a, \max} = V_{d, \max} = 1$, $r_f = 1$, $r_c = 0.5$, and $T_f = 12$. The Hamilton-Jacobi reachable sets are calculated using the Level-Set Toolbox [22] developed at the University of British Columbia. For the particular set of parameters, the sets F_D , R_D and W_D all converge to fixed points after about 6 time units, corresponding to configurations from which the defender can always prevent the attacker from achieving the desired objectives, regardless

¹The choice of player order does not affect the player strategies in this problem because the optimal defender and attacker inputs are independent of each other.

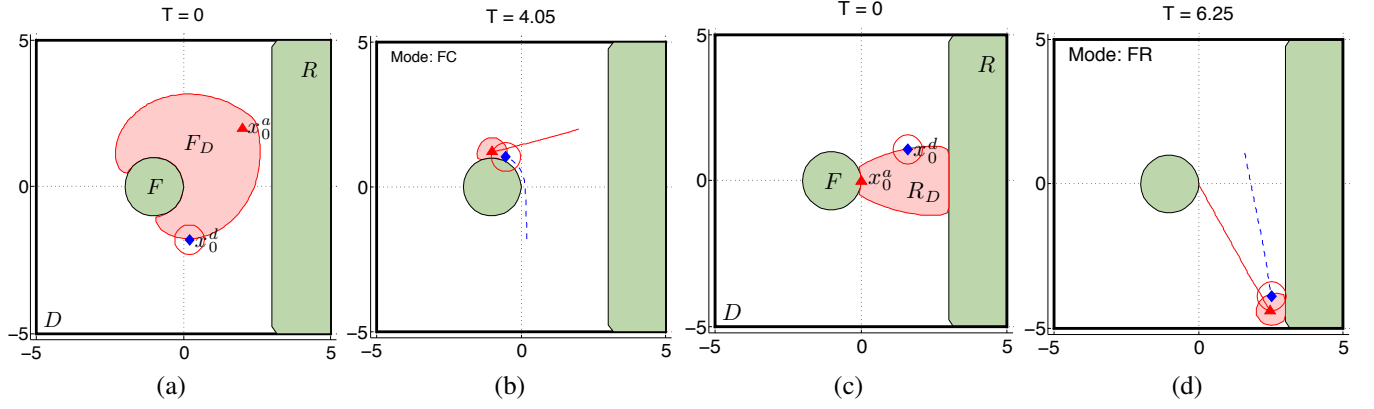


Fig. 3. Two simulations showing cases where the defender (blue diamond) successfully intercepts the attacker (red triangle). (a) shows a scenario for flag capture only, with the resulting trajectories shown in (b). (c) and (d) show a similar scenario for flag return.

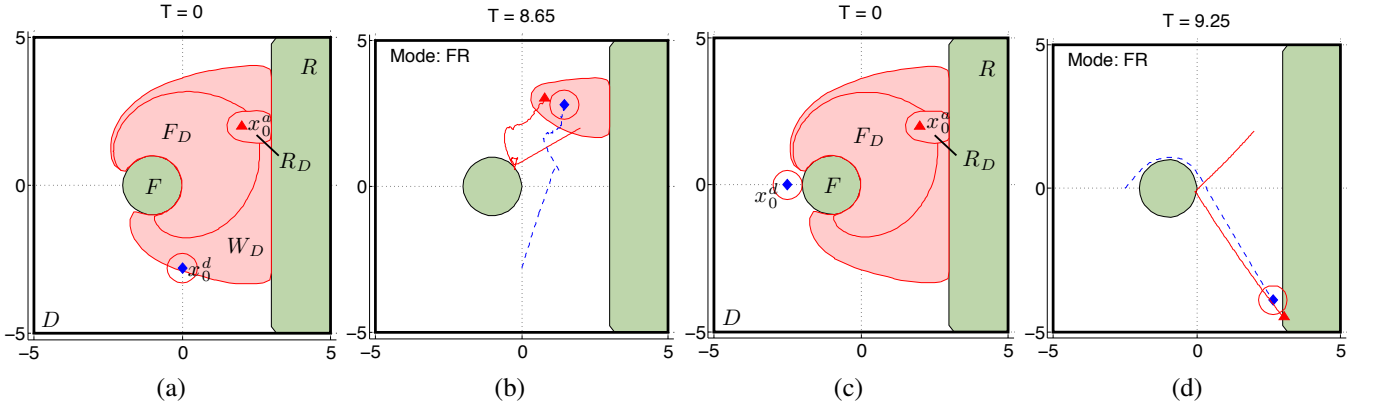


Fig. 4. Two scenarios showing combined winning regions W_D for both game modes. In (a) the defender starts inside W_D and successfully prevents the attacker from returning with the flag, as seen in (b). (c) shows a case where the defender is outside of W_D , and is unable to prevent the attacker's successful flag capture and return, as shown in (d).

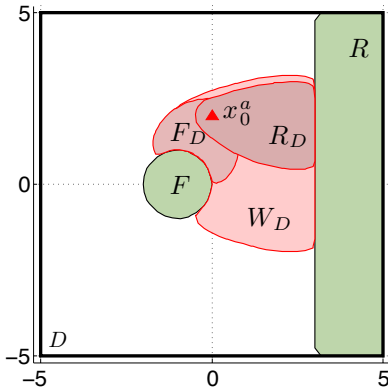


Fig. 2. The overall winning region W_D of the defender for $x^a = (2, 2)$, with R_D and F_D super-imposed for comparison.

of the time horizon. The actual sets lie in the 4-D joint configuration space (x^a, x^d) , but for visualization purpose, slices are shown in 2-D with x^a fixed. For example, the converged sets F_D , R_D and W_D are shown in Figure 2 with the attacker fixed at $(2, 2)$. It is observed 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 phase to arrive

at a configuration that blocks the attacker's return path.

Using the reachable sets, simulations are conducted where the attacker and defender choose controls according to the strategies discussed in Section III-C. Figure 3 shows two example scenarios, one for the flag capture phase and the other for the flag return phase. In both cases, the defender starts within the winning set and successfully intercepts the attacker. More interesting scenarios involving the interplay between the flag capture and flag return phase are demonstrated in Figure 4. In Figure 4a, the defender starts within the overall winning set W_D , but outside the set F_D . The trajectory simulation in Figure 4b shows that the defender does not try to prevent flag capture. Instead it uses the time it takes 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 begins outside W_D , as in Figure 4c, the simulated trajectory in Figure 4d shows that the attacker is able to successfully capture the flag and return to the safe region, without being intercepted by the defender at any point.

The computation time of the reachable sets is strongly tied to the numerical scheme for obtaining the spatial derivatives and the grid size. With 45 points in each dimension and at high numerical accuracy, each set took approximately 1 hour



Fig. 5. Smartphone application for playing automation-assisted capture-the-flag.

to compute on an Apple Macbook Pro laptop with a 2.66 Ghz Core i7 processor and 8 GB of RAM. Sets with 30 points in each dimension can be computed in as little as 4 minutes at lower numerical accuracy.

It should be noted that some numerical errors are inevitable due to the necessity of solving the Hamilton-Jacobi 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. Due to this reason, it is observed that near the intersection of F_D with F , the zero sub-level set sometimes slightly under-approximates the stop capture set, resulting in defender trajectories that begin just outside of F_D (below the grid resolution) that nevertheless capture the attacker. More accurate solutions can be obtained with finer 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.

V. CONCLUSION AND FUTURE WORK

Hamilton-Jacobi reachability has allowed us to compute winning sets to characterize the two-player capture-the-flag game without making limiting assumptions on the player actions. Moreover, the reachable sets provide a simple and intuitive visualization tool, and the method used to enforce domain constraints for the game rules can be used to account for static obstacles in the game domain. We are currently implementing software to use reachable sets computed offline to generate online visualization and movement recommendations for a human game of capture-the-flag. The players will use GPS-equipped smartphones for localization and to display the reachability information, as seen in Figure 5.

In the future, we hope to extend this work to scenarios with partial observability and teams of multiple players.

ACKNOWLEDGMENTS

The authors would like to thank Scott Hoag and Andrew Sy for their work on developing the capture-the-flag software.

REFERENCES

- [1] D. of the Air Force, "United states air force unmanned aircraft systems flight plan 2009-2047," *oai.dtic.mil*, Jan 2009.
- [2] H. Erzberger, "Automated conflict resolution for air traffic control," *25 th International Congress of the Aeronautical Sciences*, Jan 2006.
- [3] A. Madrigal, "Autonomous robots invade retail warehouses," <http://www.wired.com/wiredscience/2009/01/retailrobots/>.
- [4] D. Axe, "Building trust: Human fears limit autonomy and capability of robots," *Defense Technology International*, Nov 2007.
- [5] U. Metzger and R. Parasuraman, "The role of the air traffic controller in future air traffic management: An empirical study of active control versus passive monitoring," *Human Factors*, Jan 2001.
- [6] R. D'Andrea and M. Babish, "The roboflag testbed," *Proceedings of the American Control Conference*, vol. 1, pp. 656–660, 2003.
- [7] 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.
- [8] M. Campbell, R. D'Andrea, D. Schneider, A. Chaudhry, S. Waydo, J. Sullivan, J. Veverka, and A. Klochko, "Roboflag games using systems based, hierarchical control," *Proceedings of the American Control Conference*, vol. 1, pp. 661–666, 2003.
- [9] S. Waydo and R. Murray, "Vehicle motion planning using stream functions," *IEEE International Conference on Robotics and Automation*, 2003. *Proceedings. ICRA'03*, pp. 2484–2491, 2003.
- [10] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, "A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with roboflag," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 35, no. 4, p. 481, 2005.
- [11] G. Chasparis and J. Shamma, "Linear-programming-based multi-vehicle path planning with adversaries," *American Control Conference*, 2005. *Proceedings of the 2005*, pp. 1072–1077, 2005.
- [12] 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," *Decision and Control, 43rd IEEE Conference on*, 2004.
- [13] J. McGrew, J. How, L. Bush, B. Williams, and N. Roy, "Air combat strategy using approximate dynamic programming," *AIAA Guidance, Navigation, and Control Conference*, Aug 2008.
- [14] 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, July 2005.
- [15] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.
- [16] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. 2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010.
- [17] M. Oishi, I. Mitchell, A. Bayen, and C. 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.
- [18] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.
- [19] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Stanford University, 2002.
- [20] M. Bardi and I. Capuzzo Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhauser, 1997.
- [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. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.
- [24] 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.