A General, Open-Loop Formulation for Reach-Avoid Games

Zhengyuan Zhou, Ryo Takei, Haomiao Huang, and Claire J. Tomlin

***Abstract*— A reach-avoid game is one in which an agent attempts to reach a predefined goal, while avoiding some adversarial circumstance induced by an opposing agent or disturbance. Their analysis plays an important role in problems such as safe motion planning and obstacle avoidance, yet computing solutions is often computationally expensive due to the need to consider adversarial inputs. In this work, we present an open-loop formulation of a two-player reach-avoid game whereby the players define their control inputs prior to the start of the game. We define two open-loop games, each of which is conservative towards one player, show how the solutions to these games are related to the optimal feedback strategy for the closed-loop game, and demonstrate a modified Fast Marching Method to efficiently compute those solutions.**

I. Introduction

A reach-avoid game is one in which one player attempts to arrive at a goal set in the state-space while avoiding some other, undesired set of states. The goal of the opposing player is to prevent the first player from arriving at its goal, possibly by driving the first player into the undesired set. Such games encompass a large number of robotics and control applications. For example, many safe motion-planning problems may be formulated as reach-avoid games, where the goal is to control an agent into some desired state or configuration while avoiding a set of obstacles or otherwise unsafe configurations. Computing solutions to such games is complicated by the adversarial interaction between the two players.

There have been many approaches to safe-motion planning and reach-avoid problems using probabilistic formulations, where a probability distribution is assigned to the actions of the moving obstacles and a solution is computed that minimizes an expected cost with respect to the probability of collision [1]–[5].

In safety-critical applications, a worst-case approach can be taken by treating the two players as adversarial agents each attempting to act in some optimal manner. In certain cases, geometric methods can be used to compute the set of states under which collision with moving obstacles is inevitable [6], [7]. The most complete approach to the reach-avoid game is to formulate the problem as a differential game and solve the related Hamilton-Jacobi-Isaacs(HJI) equation [8]–[11]. The player attempting to reach the goal is treated as an attacker whose objective is to arrive at the goal, and the opposing player is treated as a defender attempting to prevent this by intercepting the attacker. The value of the game is defined as the time required for the attacking player to reach the goal, and the objectives of the attacking and defending players are minimizing and maximizing this value, respectively. This value can be approximated numerically on a grid, and control inputs extracted via numerical differentiation [9], [12].

Unfortunately, computing solutions to HJI equations is computationally infeasible for large problems, as the grid required for approximating the value function grows exponentially as size of the state-space increases. Even smaller problems such as 2-player, kinematic games, typically cannot be solved in real-time, requiring pre-computation.

A previous publication presented an *open-loop formulation* to the reach-avoid game, in which the players select their inputs prior to the start of the game and then follow those inputs without modification during the game [11]. The information pattern was conservative from the attacker's perspective, in that the defender was allowed full knowledge of the attacker's input choice. In addition to the open-loop reach-avoid game, a modified Fast Marching Method (FMM) algorithm for efficiently computing attacker-conservative solutions was also presented.

In this paper, we present a general formulation of the open-loop reach-avoid game as a pair of differential games, and relate them to the optimal, feedback solution, with the formulation and information patterns presented in Section II. Each game is played conservatively from the perspective of one player, with the upper (maximum) value of the game representing conservatism on the part of the attacker, and the lower (minimum) value representing conservatism on the part of the defender. We then expand upon the results presented in a previous publication [11] on computing the upper value to computing a bound for the lower value in Section III. We can then relate the optimal control inputs of the upper and lower games to the optimal, closed-loop, feedback control input derived by solving the HJI equation, and state a condition under which the controls are equivalent, as shown in Section IV. Finally, we demonstrate a modified FMM algorithm to compute this lower bound and extract the related player control inputs, as shown in Section V.

II. Game Formulation and Information Patterns

Suppose there are two players P_1 and P_2, whose states are confined in a bounded, open domain Ω. The domain Ω can be further partitioned as follows: $\Omega = \Omega_{free} \cup \Omega_{obs}$, where Ω_{free} represents the free space that the two players can move, while Ω_{obs} are impenetrable obstacles for both players. Define $\Sigma =$

This work is supported in part by ONR under HUNT Award Number 550740, by AFOSR MURI FA9550-10-1-0567, and NSF under grant CNS-0931843

Z. Zhou, R. Takei, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA {zhengyuan, rrtakei, tomlin}@EECS.Berkeley.edu

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

$\{\sigma\colon [0,\infty) \to \mathcal{S}^1\}$ as the set of controls for each player, where $\mathcal{S}^1$ is the set of all vectors of unit length. We assume that the dynamics of the players are defined by the decoupled system for $t > 0$:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), a(t))a(t), \quad x_1(0) = x_1^0, \\ \dot{x}_2(t) &= f_2(x_2(t), b(t))b(t), \quad x_2(0) = x_2^0. \end{aligned} \tag{1}$$

Here, the functions $f_1, f_2\colon \Omega \times \mathcal{S}^1 \to \mathbb{R}$ are assumed to be positive and Lipschitz continuous. Throughout the article, we shall use a to denote a control for P_1 and b to denote a control for P_2, with $a, b \in \Sigma$. We note that by the definition of Σ, $a(t)$ and $b(t)$ represent the directions in which P_1 and P_2 are moving respectively at time t. And f_1, f_2 represent the speeds of P_1 and P_2 respectively in (1). Given $a, b \in \Sigma$, we call the solution $x_1(\cdot), x_2(\cdot)$ to the initial value problem (1) as *paths* for P_1 and P_2 respectively. We shall use the notation $\mathbf{x}(\cdot) = (x_1(\cdot), x_2(\cdot))$ for the joint path, $\mathbf{x^0} = (x_1^0, x_2^0)$ for the joint initial state and $\mathbf{x}(t) = (x_1(t), x_2(t))$ for the joint state at time t. For notational simplicity, we shall drop the explicit dependence of the paths on the controls and initial states.

We now define the payoff for the *reach-avoid game*. We first define two sets: $\mathcal{T} \subset \Omega$, the *target set*, and $\mathcal{A} \subset \Omega^2$, the *avoid set*. The goal for P_1 is to reach $\mathcal{T}$ as quickly as possible, while avoiding entering a joint state in $\mathcal{A}$ with P_2. The payoff function is then defined as:

$$\mathcal{J}(\mathbf{x^0}, a, b) = \inf\{t \geq 0 \mid x_1(t) \in \mathcal{T}, \mathbf{x}(s) \notin \mathcal{A}, \forall s \in [0,t]\}, \tag{2}$$

with the convention that the infimum of the empty set is infinity. We say P_1 is captured by P_2 if their joint state enters $\mathcal{A}$.

For convenience, we define the optimal time-to-reach functions for P_1 and P_2 respectively:

$$u_1(x) = \inf_{a\in\Sigma}\{t \geq 0 \mid x_1(t) = x\}, \tag{3}$$

$$u_2(x) = \inf_{b\in\Sigma}\{t \geq 0 \mid x_2(t) = x\}. \tag{4}$$

For $t \geq 0$, we also characterize the reachability of P_1 and P_2 respectively as follows:

$$\mathcal{R}_1(t) = \{x \mid u_1(x) \leq t\}, \tag{5}$$

$$\mathcal{R}_2(t) = \{x \mid u_2(x) \leq t\}. \tag{6}$$

Note that in defining the t-reachable set for both players, we are not concerned with the avoid set $\mathcal{A}$.

A. Types of value functions

Given the payoff of a differential game, the *value* of the game depends on the information pattern employed by each player. In this section, we introduce several types of value functions that are relevant here.

Definition 1: The *open-loop upper value* and the *open-loop lower value* of the reach avoid game are defined respectively as follows:

$$\begin{aligned} \overline{v}(\mathbf{x^0}) &= \inf_{a\in\Sigma}\sup_{b\in\Sigma}\mathcal{J}(\mathbf{x^0}, a, b), \\ \underline{v}(\mathbf{x^0}) &= \sup_{b\in\Sigma}\inf_{a\in\Sigma}\mathcal{J}(\mathbf{x^0}, a, b). \end{aligned} \tag{7}$$

The open-loop values can be seen as the most conservative information pattern, where one player chooses the optimal control knowing the opponent's choice of control. The upper value is defined conservatively towards P_1 since P_2 will choose the control in response to the control already chosen by P_1, with P_1 assuming the worst case. Conversely, the lower value is conservative towards P_2. Due to this conservative bias towards each player, it can be shown that $\underline{v}(\mathbf{x^0}) \leq \overline{v}(\mathbf{x^0})$ for all $\mathbf{x^0}$. Corresponding to the open-loop upper and lower values, we have the optimal open-loop controls defined for both players in both information patterns. For the upper value, we denote the optimal controls (not necessarily unique) of P_1 and P_2 as $\bar{a}$ and $\bar{b}$ respectively, formally written as:

$$\begin{aligned} \bar{a} &\in \arg\inf_{a\in\Sigma}\sup_{b\in\Sigma}\mathcal{J}(\mathbf{x^0}, a, b) \\ \bar{b} &\in \arg\sup_{b\in\Sigma}\mathcal{J}(\mathbf{x^0}, \bar{a}, b) \end{aligned} \tag{8}$$

Similarly, for the lower value, we denote we denote the optimal controls (not necessarily unique) of P_1 and P_2 as $\underline{a}$ and $\underline{b}$ respectively, formally written as:

$$\begin{aligned} \underline{b} &\in \arg\sup_{b\in\Sigma}\inf_{a\in\Sigma}\mathcal{J}(\mathbf{x^0}, a, b) \\ \underline{a} &\in \arg\inf_{a\in\Sigma}\mathcal{J}(\mathbf{x^0}, a, \underline{b}) \end{aligned} \tag{9}$$

Another more well-known information pattern is that of the *non-anticipative strategy game*, otherwise referred to as the feedback strategy game. A feedback strategy $\gamma[\sigma]$ maps a opposing input sequence (and its corresponding state) into a control. The set of admissible feedback strategies is defined in the following manner: let σ_p, $\hat{\sigma}_p \in \Sigma$ be two controls, then the set of admissible feedback strategies $\mathcal{F}$ is:

$$\begin{aligned} \mathcal{F} :=& \\ &\{\gamma\colon \Sigma \to \Sigma \mid \forall t > 0, \sigma_p|_{s=0}^t = \hat{\sigma}_p|_{s=0}^t \\ &\text{implies } \gamma[\sigma_p]|_{s=0}^t = \gamma[\hat{\sigma}_p]|_{s=0}^t\}. \end{aligned}$$

Then the value of this game is defined as:

$$U(\mathbf{x^0}) = \inf_{\gamma\in\mathcal{F}}\sup_{\sigma_p\in\Sigma}\mathcal{J}(\mathbf{x^0}, \gamma[\sigma_p], \sigma_p). \tag{10}$$

There is a corresponding "sup inf" definition, but they coincide provided the value is continuous and that Isaacs' condition [17] holds, which is trivially true due to decoupled dynamics. The key property distinguishing the feedback strategy value from the open-loop values is that the former satisfies the dynamic programming principle (DPP). The DPP yields a necessary and sufficient condition that $U(x, y)$ can be characterized by the Hamilton-Jacobi-Isaacs (hereafter referred to as HJI) equation, where x and y are the states for P_1 and P_2 respectively:

$$H(\nabla_x U, \nabla_y U, x, y) = 1, \tag{11}$$

where

$$H(p, q, x, y) = \max_{\alpha\in\mathcal{S}^1}\{p \cdot \alpha f_1(x, \alpha)\} + \min_{\beta\in\mathcal{S}^1}\{q \cdot \beta f_2(y, \beta)\}. \tag{12}$$

For the above Hamiltonian, it can be shown that U is the viscosity solution to the PDE (11), provided it is continuous. Furthermore, it is be related to the open-loop values as

$$\underline{v}(\mathbf{x^0}) \leq U(\mathbf{x^0}) \leq \overline{v}(\mathbf{x^0}). \tag{13}$$

While standard methods for solving (11) on a Cartesian grid exist, they generally suffer from the curse of dimensionality. The complexity increases exponentially in the number of players, which makes it computationally intractable even for a modestly sized grid. In [11] and [14], novel methods for computing the open-loop values are presented. The main advantage of these methods is that all computations are performed on the underlying domain, thereby avoiding the curse of dimensions. In [11], we have presented an efficient way to compute $\overline{v}$ for a reach-avoid game known as capture-the-flag (CTF), which can be easily extended to computing $\overline{v}$ for the general reach-avoid game.

In the next section, we provide a way to compute a lower bound on $\underline{v}$. This leads to the fact that, for a class of initial states where the open-loop upper and lower values coincide, the feedback strategy value is $U(\mathbf{x^0}) = \underline{v}(\mathbf{x^0}) = \overline{v}(\mathbf{x^0})$. We then show that in those cases, the equality of values implies the equality of the controls. In other words, for those cases we can verify that we have found the optimal controls without actually solving the HJI equations. For these initial value problems, we can have both fast computation and the optimality of the controls obtained.

III. Computation of Lower bounds on $\underline{v}$

In general, computing $\underline{v}$ is not a trivial task. In this section, we seek to find a straightforwardly computable lower bound to $\underline{v}$. Furthermore, we show that, in some cases, the lower bound is equal to $\underline{v}$.

Definition 2: For each $y \in \Omega$, we define the partial avoid set with respect to y to be

$$\mathcal{A}_1(y) = \{x \in \Omega \mid (x, y) \in \mathcal{A}\}.$$

That is, $\mathcal{A}_1(y)$ is the set of all states of P_1 that are captured with P_2's position fixed at y.

We first state a naive lower bound without proof.

Proposition 1: $\underline{v}(\mathbf{x^0}) \geq \inf_{x \in \mathcal{T}} u_1(x)$.

Intuitively, $\underline{v}$ must be at least as large as the shortest time required for P_1 to reach the target in the absence of P_2. In fact, if we restrict our attention to the reach-avoid game with point capture(i.e. capture achieved when $x_1 = x_2$), then we have strict equality between the two quantities. Induction shall furnish a simple argument that we have

$$\underline{v}(\mathbf{x^0}) = \inf_{x \in \mathcal{T}} u_1(x) \tag{14}$$

if for each y, the cardinality of $\mathcal{A}_1(y)$ is finite.

Definition 3: Given $x \in \Omega$ define

$$t^*(x; x_1^0) = \inf_{a \in \Sigma} \{t \geq 0 \mid x_1(t) \in \mathcal{T}, x_1(0) = x_1^0, \\ (x_1(s), x) \notin \mathcal{A}, \forall s \in [0, t]\}.$$

That is, given that P_2 remains at the point x throughout, $t^*(x; x_1^0)$ is the shortest time for P_1, starting at x_1^0, to reach the target without ever getting "captured" by P_2.

Remark 1: This definition together with Proposition 1 naturally leads to

$$\underline{v}(\mathbf{x^0}) \geq t^*(x_2^0; x_1^0). \tag{15}$$

Definition 4: For any subset $R \subset \Omega$ with $x_2^0 \in R$, we define the following function:

$$w_R(x; x_2^0) = \inf_{b \in \Sigma} \{t \geq 0 \mid \quad x_2(t) = x, x_2(0) = x_2^0, \\ x_2(s) \in R, \forall s \in [0, t]\}.$$

Thus, $w_R(x; x_2^0)$ is the minimum time for P_2 to reach x from x_2^0 by traveling along a path that is contained in R. Note that the set R may serve as a restriction on how fast P_2 can get to a point of interest since the path must be contained entirely in R. Therefore, it is possible that P_2 may not be able to take the optimal path in the original domain. In some sense, $w_R(x; x_2^0)$ characterizes a reach time for P_2 and $t^*(x; x_1^0)$ characterizes a reach time for P_1. The key to computing the lower bound for $\underline{v}$ lies in connecting those two quantities via the following set R^*.

Definition 5: Let $R^* \subset \Omega$ be the maximal set containing x_2^0 such that for all $y \in R^*$ the following holds:

$$w_{R^*}(y; x_2^0) < t^*(y; x_1^0). \tag{16}$$

Therefore, R^* is the set of all points that P_2 may move in, such that P_2 can arrive at a point x before P_1 can end the game. To see that R^* exists, we note that $\{x_2^0\}$ is certainly a set that satisfies the inequality (16). We therefore take the collection of all sets that satisfy the inequality. We take the union of the collection, which certainly still satisfies the inequality since for every fixed x and x_2^0, the function value will never decrease when we enlarge a set R to a larger set $\tilde{R}$ such that $R \subset \tilde{R}$. Therefore the union is maximal, and the maximality implies its uniqueness.

The following lemma characterizes an important property of R^* that shall prove useful in the computations.

Lemma 1: Given a point $x \in \Omega$, we have $w_{R^*}(x; x_2^0) < t^*(x; x_1^0)$ if and only if $x \in R^*$.

Proof: If $x \in R^*$, then by definition,$w_{R^*}(x; x_2^0) < t^*(x; x_1^0) \leq \infty$.

If x is not in R^*, then since we require in the definition of the function $w_{\mathbb{R}}$ that the path being taken should be contained entirely in R^*, we conclude that there is no path for P_2 that starts at x_2^0 and ends at x while still satisfying the requirement that whole path is contained in R^*. Therefore, x can never be reached, hence $w_{R^*}(x; x_2^0) = \infty$. ■

Remark 1 gives us a naive bound on the open loop lower value. Using t^* and R^*, we can construct a much better lower bound. To this end, we provide a general lower bound on $\underline{v}$.

Definition 6: Given a joint initial state $\mathbf{x^0}$, we define $\underline{\underline{v}}(\mathbf{x^0})$ as follows:

$$\underline{\underline{v}}(\mathbf{x^0}) = \sup_{x \in R^{**}} t^*(x),$$

where

$$R^{**} = \{x \in R^* \mid \mathcal{A}_1(x) \cap \mathcal{R}_1(T(x)) = \emptyset, \\ T(x) = w_{R^*}(x; x_2^0)\}.$$

Essentially, R^{**} is a set that contains all the useful points in R^*, that is, points x at which P_2 can arrive at before P_1, and along a path that ensures that P_1 cannot end the game before P_2 reaches x. This is important as $\mathcal{A}_1(x)$ is only an obstacle for P_1 after P_2 has reached x. This leads to the following result.

Theorem 1: Given a joint initial state $\mathbf{x^0}$, we have $\underline{\underline{v}}(\mathbf{x^0}) \leq \underline{v}(\mathbf{x^0})$.

Proof: Take any $x \in R^*$. By definition, $w_{R^*}(x; x_2^0) < t^*(x; x_1^0)$. Thus, P_2 can reach the point x in time $T = w(x; x_2^0)$. Let $\sigma_b \in \Sigma$ be the control for P_2 that reaches x at time T and remains stationary thereafter. That is, P_2 can control itself so that $\mathcal{A}_1(x)$ serves as a stationary obstacle to P_1 for all time greater than or equal to T. But the assumption $\mathcal{A}_1(x_2) \cap \mathcal{R}_1(T) = \emptyset$ implies that P_1 cannot distinguish whether $\mathcal{A}_1(x)$ is an avoid set that has been present for all $t \geq 0$ or just for $t \geq T$. Thus, it would take P_1 at least $t^*(x; x_1^0)$ time to reach $\mathcal{T}$, since if P_2 were to start at x and remains stationary throughout, then it would by definition take the first player $t^*(x; x_1^0)$ to reach the target. As a result, we have $\mathcal{J}(\mathbf{x^0}, a, \sigma_b) \geq t^*(x; x_1^0)$ for any choice $a \in \Sigma$, which implies $\inf_{a\in\Sigma} \mathcal{J}(\mathbf{x^0}, a, b^*) \geq t^*(x_2^*)$. Finally, since $\underline{v}(\mathbf{x^0}) = \sup_{b\in\Sigma} \inf_{a\in\Sigma} \mathcal{J}(\mathbf{x^0}, a, b) \geq \inf_{a\in\Sigma} \mathcal{J}(\mathbf{x^0}, a, \sigma_b) \geq t^*(x; x_1^0)$. But since x is any point in R^*, we have $\underline{v}(\mathbf{x^0}) \geq \sup_{x\in R^*} t^*(x; x_1^0)$, and hence the desired inequality. ■

With this understanding of R^{**}, we may select the control input for P_2 as moving to a point $x_2^* \in \arg\sup_{x\in R^{**}} t^*(x; x_1^0)$ and remaining stationary at x_2^* thereafter. Since P_2 is guaranteed to arrive at x_2^* before P_1, $\mathcal{A}_1(x_2^*)$ appears to P_1 as a permanent obstacle, forcing it to take at least $t^*(x_2^*, x_1^0)$ to reach the goal.

IV. Optimality of Controls

We note that since $\underline{\underline{v}}$ is a lower bound for $\underline{v}$,by invoking Ineqaulity (13),we see that $\underline{\underline{v}} = \overline{v}$ implies the following:

$$\underline{v} = U = \overline{v}. \tag{17}$$

This provides us with an easy way to test if the open loop values coincide with the closed loop value U. In the case that they are indeed equal, we shall show that the controls obtained from the open loop computations also coincide with controls from solving the HJI equation.

To this end,we begin with a useful characterization. Let a^* and b^* be the controls extracted from the Hamilton-Jacobi-Isaacs formulation, i.e. $U(\mathbf{x^0}) = \mathcal{J}(\mathbf{x^0}, a^*, b^*)$.

Lemma 2: $\mathcal{J}(\mathbf{x^0}, a^*, b) \leq \mathcal{J}(\mathbf{x^0}, a^*, b^*) \leq \mathcal{J}(\mathbf{x^0}, a, b^*)$ for all $a, b \in \Sigma$.

Proof: We will only prove the second inequality, using Equation (10); the first inequality can be proved similarly by considering the "$\sup\inf$" definition of the value.

Take any $a \in \Sigma$, and consider $\bar{\gamma} \in \mathcal{F}$ such that $\bar{\gamma}(b^*) = a$.

$$\begin{aligned}\mathcal{J}(\mathbf{x^0}, a^*, b^*) &= \inf_{\gamma\in\mathcal{F}} \mathcal{J}(\mathbf{x^0}, \gamma(b^*), b^*)\\ &\leq \mathcal{J}(\mathbf{x^0}, \bar{\gamma}(b^*), b^*)\\ &= \mathcal{J}(\mathbf{x^0}, a, b^*).\end{aligned}$$

■

This lemma is a restatement of the optimality condition for the feedback control: when both players are playing according to the optimal controls extracted from the solutions to the HJI equation, neither player has any incentive to deviate from this best control, as any deviation only makes that player worse off.

Theorem 2: Denote $\Sigma_{P_1}^*$ and $\Sigma_{P_2}^*$ to be the set of optimal controls for P_1 and P_2 respectively extracted from the solution to the HJI equation.

1) Let $\bar{a}$ and $\bar{b}$, defined per (8), be two controls extracted from computing $\overline{v}(\mathbf{x^0})$ for P_1 and P_2 repectively, not necessarily unique . If $\underline{\underline{v}}(\mathbf{x^0}) = \overline{v}(\mathbf{x^0})$, then $\bar{a} \in \Sigma_{P_1}^*$, $\bar{b} \in \Sigma_{P_2}^*$.
2) Let $\underline{a}$ and $\underline{b}$,defined per (9), be two controls extracted from computing $\underline{v}(\mathbf{x^0})$ for P_1 and P_2 repectively, not necessarily unique. If $\underline{\underline{v}}(\mathbf{x^0}) = \overline{v}(\mathbf{x^0})$, then $\underline{a} \in \Sigma_{P_1}^*$, $\underline{b} \in \Sigma_{P_2}^*$.

Proof: Take $\bar{a} \in \inf_{a\in\Sigma} \sup_{b\in\Sigma} \mathcal{J}(\mathbf{x^0}, a, b)$, we have for any P_2's control $\sigma_{P_2} \in \Sigma$

$$\begin{aligned}\overline{v}(\mathbf{x^0}) &= \inf_{a\in\Sigma} \sup_{b\in\Sigma} \mathcal{J}(\mathbf{x^0}, a, b)\\ &= \sup_{b\in\Sigma} \mathcal{J}(\mathbf{x^0}, \bar{a}, b) \geq \mathcal{J}(\mathbf{x^0}, \bar{a}, \sigma_{P_2})\end{aligned}$$

In particular, if we take $\sigma_{P_2} = b^*$, where $b^* \in \Sigma_{P_2}^*$, we have

$$U(\mathbf{x^0}) = \mathcal{J}(\mathbf{x^0}, a^*, b^*) = \overline{v}(\mathbf{x^0}) \geq \mathcal{J}(\mathbf{x^0}, \bar{a}, b^*)$$

But by Lemma 2, we also have $\mathcal{J}(\mathbf{x^0}, a^*, b^*) \leq \mathcal{J}(\mathbf{x^0}, \bar{a}, b^*)$. So it must be that $\mathcal{J}(\mathbf{x^0}, a^*, b^*) = \mathcal{J}(\mathbf{x^0}, \bar{a}, b^*)$. Therefore we have $\bar{a} \in \Sigma_{P_1}^*$.

By (8), $\bar{b} \in \sup_{b\in\Sigma} \mathcal{J}(\mathbf{x^0}, \bar{a}, b) = \sup_{b\in\Sigma} \mathcal{J}(\mathbf{x^0}, a^*, b) \geq \mathcal{J}(\mathbf{x^0}, a^*, b)$, for any $b \in \Sigma$. By Lemma 2, we have $\bar{b} \in \Sigma_{P_2}^*$.

The second case can be proved analogously by noting that in this case $\underline{\underline{v}} = \underline{v}$ and using the other inequality in Lemma 2. ■

Remark 2: Those two statements together state that if the two open-loop values coincide, then $\bar{a}$ produces the same value as $\underline{a}$, and both yield the same value as the control computed for P_1 by the solver of HJI equations. The same is true for $\bar{b}$, $\underline{b}$ and the control computed for P_2 by the solver of HJI equations. Under the conditions stated in the above theorem, this indeed provides us with the efficient computation of the optimal controls without the cost incurred by solving HJI. In the case $\underline{\underline{v}}(\mathbf{x^0}) = \overline{v}(\mathbf{x^0}) = \infty$, even though P_1's controls extracted from the open-loop lower or upper value are optimal in the HJI sense, they are trivial in the sense under such a condition any control of P_1 is equally bad. Since $\underline{\underline{v}}(\mathbf{x^0}) = \infty$, P_2 can always prevent P_1 from reaching the target even though P_2 is being extremely conservative. Thus there are no inputs for P_1 such that P_1 can ever arrive at the target.

Practically speaking, the equivalence presented above gives a useful way of verifying that the open-loop control computations may be followed without any loss of optimality. Note that the conditions under which this equivalence holds are not particularly restrictive ones, and we are not limited to initial conditions where player actions are entirely independent of each other. It is certainly true that if the players begin the game in a way such that neither can affect the other's paths, for example if the players begin the game far from each other and relevant goals, then the open-loop and close-loop values will be equal. However, this is not a necessary condition, and as the computational example in Section VI shows, there are configurations where the open-loop values produce optimal results that requires both the attacker and defender to modify their actions to account for the presence of the other player.

V. The Modified Fast Marching Method for Computing $\underline{\underline{v}}$

Having established a bound $\underline{\underline{v}}$ based on $t^*(x; x_1^0)$ and $w_R(x; x_2^0)$, as well as the set R^*, we would like to have efficient methods to compute those quantities. The FMM [15], [16], with appropriate modifications, lends itself well to the current context. As FMM has been described in detail elsewhere, we shall mainly be concerned with the specific modifications employed to compute the quantities at hand, while referring the reader to the general references for the details of the basic FMM.

We use a N by N uniform 2-D cartesian grid to approximate the domain Ω. We shall use $t^*{}_{(i,j)}$ to denote the function value $t^*(x; x_1^0)$, with x approximated by the grid node (i, j). We also use $w_R^{(i,j)}$ to denote the function value $w_R(x; x_2^0)$. Finally, $\mathcal{A}_1^{(i,j)}$ denotes the partial avoid set $\mathcal{A}_1(y)$ with y approximated by the node (i, j). We use Accepted to represent the the collection of all computed grid nodes (i, j) whose t^* are computed, we use NarrowBand to denote the collection of the grid nodes which are to be added to Accepted. Finally FarAway represents the grid nodes that are neither in Accepted nor NarrowBand. W and T are 2 N by N arrays which are used to store values for $w_R^{(i,j)}$ and $t^*{}_{(i,j)}$ respectively for all grid nodes (i,j). $W_{(i,j)}$ and $T_{(i,j)}$ represent the values stored for node (i, j) in the array W and T respectively. The algorithm then proceeds as follows:

1) $W_{(i,j)} = \infty$, if $(i, j) \in \Omega_{obs}$, and place those grid nodes into FarAway.

2) $W_{(i,j)} = 0$, if (i, j) is the initial position of P_2, and set (i, j) to be in Accepted.

3) For each node (i, j) adjacent to a node in Accepted, run the Eikonal update as described in [11]. After running the update, we have $W_{(i,j)} = w_R^{(i,j)}$ for all (i, j) adjacent to a node in Accepted and place these nodes in NarrowBand

4) Take the (i, j) that has the smallest $W_{(i,j)}$. If it is equal to ∞ return W and T and continue to Step 5, otherwise compute $t^*{}_{(i,j)}$ by doing the following: Treating $\mathcal{A}_1^{(i,j)}$ as an obstacle, compute $t^*{}_{(i,j)}$ using standard FMM. Set $T_{(i,j)}$ to be $t^*{}_{(i,j)}$ and put (i, j) into Accepted. If $W_{(i,j)} < t^*{}_{(i,j)}$, set $W_{(i,j)}$ to $t^*{}_{(i,j)}$, otherwise set $W_{(i,j)}$ to be ∞. Now return to step 3.

5) Find the node (i, j) in W that maximizes $T_{(i,j)}$, record this value in a variable M and set $T_{(i,j)}$ to be -∞. Use the FMM presented in [11] to compute the $W_{(i,j)}$-reachable set of P_1 and test if it has nonempty intersection with $\mathcal{A}_1^{(i,j)}$. If so, then return to step 5. Otherwise, set $\underline{\underline{v}} = M$ and return $\underline{\underline{v}}$.

Remark 3: Note that instead of computing t^* for every point, we compute it "on the fly" in the sense that we stop immediately when the smallest $W_{(i,j)}$ in Narrowband is equal to ∞. This saves a considerable amount of computational time. We also note that the justification for Step 4 is Lemma 1, since if $W_{(i,j)} \geq t^*{}_{(i,j)}$,then (i, j) is not in R^*, which by the lemma, means $W_{(i,j)}$ should be ∞. Later, if we want to extract R^*, we need only look at the nodes that have finite values.

VI. Numerical Results

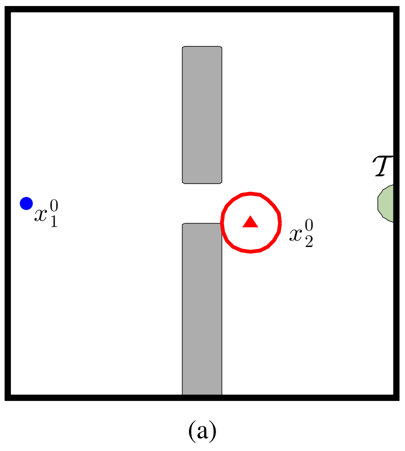
We now show numerical results for an example scenario to illustrate the concepts presented in this paper and to demonstrate the algorithm presented in Section V. Figure 1(a) shows the initial condition for the scenario considered. The gray, rectangular regions in the middle are the obstacles, and the green, semi-circular region to the right is the target set. The initial positions of both players are plotted, along with the capture radius around P_2. In this scenario, P_2 has a maximum speed of 0.25 and P_1 has a maximum speed of 1. Figure 1(b) shows R^* and contour plots for t^* within R^*. Obviously t^* increases as P_2 moves toward the opening in the obstacles, forcing P_1 to move up around the obstacles in order to reach the goal. The final paths of the two players are shown in Figure 1(c): as expected P_2 moves to block the opening, as that results in the maximum t^*.

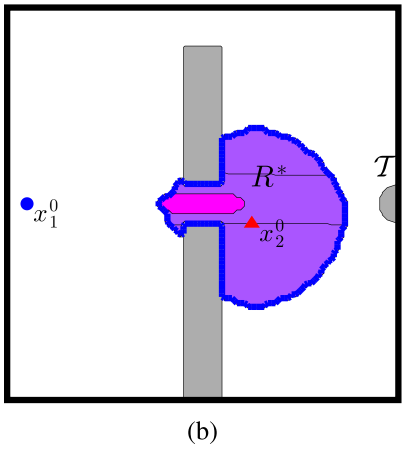
The same scenario computed using the upper value solution discussed in [11] is shown in Figure 2. In this case, P_1 is playing conservatively. $\mathcal{S}_1$ denotes the set of points that P_1 can safely reach before P_2, given that P_2 has knowledge of P_1's inputs. Again, P_1 is forced to move up over the obstacles as opposed to passing through the central opening. The control inputs in the two cases are the identical, implying that the solutions found are equivalent to the optimal, feedback strategy results.

The computations are performed on a 100x100 grid using compiled C++ code in Matlab, on a Macbook Pro laptop with a 2.4 GHz Intel Core i7 processor and 8 GB RAM. Total elapsed time for computing R^* and associated t^* was 1.25 seconds.

VII. Conclusions and Future Work

In this work, we have presented a general formulation of the open-loop reach-avoid game as a pair of games conservative for the attacking and defending players, respectively. By computing a lower bound on the open-loop lower value game, we are able to show the relationship between the values for the two open-loop games to the optimal, feedback





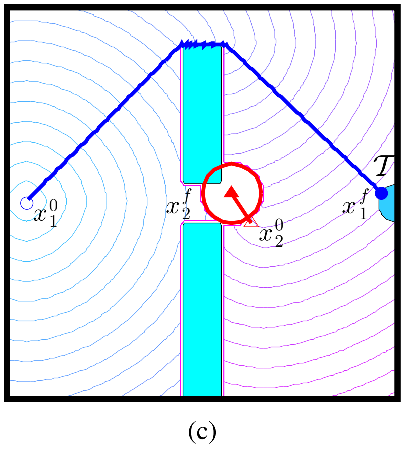


Fig. 1. **Example scenario showing (a) the initial conditions of the players, the target set $\mathcal{T}$, and game domain, (b) the set R^* with contours plotted for t^* within, and (c) the trajectories taken for each player, with equal time-to-reach contours for player 1 plotted.**

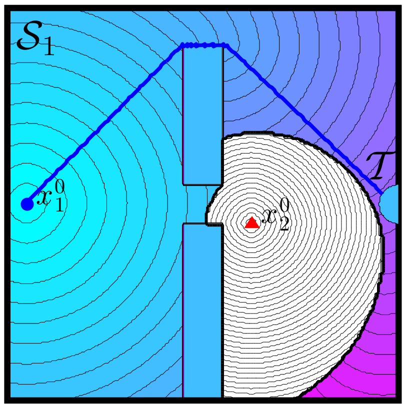


Fig. 2. **Solution to the upper value problem, showing the safe reachable set $\mathcal{S}_1$ for P_1 and the resultant path to the target set.**

value. This has also allowed us to determine when the open-loop game inputs are equal to the feedback optimal.

In addition to the theoretical results, we have presented an algorithm for quickly computing the lower bound to the lower value game, generating fast, feasible control inputs for the defending player. Taken all together, the open-loop formulation allows a control system to quickly evaluate a given initial condition and potentially generate a feasible path. If no feasible open-loop solution is found, the system can then fall back on more complex, potentially time-intensive solution methods.

For future work, we are currently extending the results to vehicles with non-holonomic dynamics as well as implementing the open-loop control in an iterative, model predictive control-like method.

References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Ed.* Englewood Cliffs, NJ: Prentice Hall, 2002.
[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
[3] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, Gaussian systems," in *ICRA*, Shanghai, China, May 2011.
[4] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *J. of Artificial Intelligence Research*, vol. 23, pp. 1–40, 2005.
[5] L. Blackmore, "Robust path planning and feedback design under stochastic uncertainty," in *Proc. of the AIAA Conf. on Guidance, Navigation and Control*, Honolulu, HI, August 2008.
[6] T. Fraichard and H. Asama, "Inevitable collision states-a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.
[7] J. Van Den Berg and M. Overmars, "Planning time-minimal safe paths amidst unpredictably moving obstacles," *Int'l. J. of Robotics Research*, vol. 27, no. 11-12, p. 1274, 2008.
[8] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana Univ. Math. J.*, vol. 33, no. 5, pp. 773–797, 1984. [Online]. Available: http://dx.doi.org/10.1512/iumj.1984.33.33040
[9] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.
[10] 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 *ICRA*, Shanghai, China, 2011.
[11] R. Takei, H. Huang, J. Ding, and C. Tomlin, "Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation,"in *IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, Minnesota, 2012.
[12] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
[13] M. Bardi, "Some applications of viscosity solutions to optimal control and differential games," in *Viscosity solutions and applications (Montecatini Terme, 1995)*, ser. Lecture Notes in Math. Berlin: Springer, 1997, vol. 1660, pp. 44–97.
[14] R. Takei, R. Tsai, Z. Zhou, and Y. Landa, "An efficient algorithm for a visibility-based surveillance-evasion game," *Communications and Mathematical Sciences, submitted*, 2012.
[15] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.
[16] J. A. Sethian, *Level set methods and fast marching methods*, 2nd ed., ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 1999, vol. 3.
[17] R. Isaacs, *Differential Games*. New York: Wiley, 1967.