Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation

Ryo Takei, Haomiao Huang, Jerry Ding, Claire J. Tomlin

***Abstract*— We present an efficient algorithm which computes, for a kinematic point mass moving in the plane, a time-optimal path that visits a sequence of target sets while conservatively avoiding collision with moving obstacles, also modelled as kinematic point masses, but whose trajectories are unknown. The problem is formulated as a pursuit-evasion differential game, and the underlying construction is based on optimal control. The algorithm, which is a variant of the fast marching method for shortest path problems, can handle general dynamical constraints on the players and arbitrary domain geometry (e.g. obstacles, non-polygonal boundaries). Applications to a two-stage game, capture-the-flag, is presented.**

I. INTRODUCTION

The central problem considered in this paper is that of *safe motion planning*, in which the objective is to plan the motion of a kinematic object moving in the plane to visit a sequence of target waypoints in minimum time, while avoiding collision with moving obstacles whose trajectories are unknown. This has many important applications, including aircraft conflict resolution and warehouse automation, where an autonomous vehicle must navigate in an environment populated by other vehicles whose intentions may not be known ahead of time. Planning in a dynamic environment has often been addressed under the assumption that the trajectories of the dynamic elements are known *a priori* [1]–[4] or can be predicted deterministically over short time horizons, with re-planning to account for deviations [5]–[8]. Relaxing this assumption requires considering the range of future motions that the moving objects can take.

Existing approaches to the problem with unpredictable obstacle motion can be generally classified as either probabilistic or worst-case. In a stochastic motion planning framework, one typically assigns some probability distribution to the actions of the moving obstacles and then computes a solution that minimizes an expected cost or with respect to a bound on the probability of collision [9]–[13].

Another class of methods takes a worst-case approach by identifying the set of the agent states which can result in a collision with the moving obstacles, and then plans paths that avoid these states. They are well suited for scenarios in which the application requirements place hard constraints on safety. In such cases, one needs to account for the worst-case possibility that the moving obstacles are adversarial. For simple domains with uniform speed profiles, geometric methods [14]-[15] are computationally efficient. In a complicated planning environment, one may consider a differential game formulation of the problem in which a collision between the agent and a moving obstacle is interpreted as a capture condition in a pursuit-evasion game [16]. The optimal controls of the agent can be then computed numerically from the solution to a Hamilton-Jacobi-Issacs (HJI) partial differential equation [17] using methods described in [18].

In this paper, we present an algorithm which computes a time-optimal path for an agent through a sequence of goal sets, while avoiding collisions with one or more unpredictably moving point obstacles. First, we consider a one stage problem with a single goal set and a single obstacle and pose the problem as an open-loop pursuit-evasion differential game (also known as a *static game* [19]). The upper value of the game (conservative from the perspective of the agent) is shown to be related to the solution of a minimum time-to-reach problem subject to state constraints. This allows us to compute the solution to the motion planning problem from the solution of a Hamilton-Jacobi-Bellman (HJB) equation. From the HJB solution, we also derive a "safe-reachable set', namely the set of states reachable by the agent from a given initial condition, under trajectories which conservatively avoid collisions with the point obstacle. Furthermore, we discuss how our formulation extends naturally to *multi-stage* problems, in which the agent is given a sequence of target (waypoint) sets to be visited in order, as well as scenarios with multiple moving obstacles.

In addition to [16], previous applications of Hamilton-Jacobi (HJ) methods to safe motion planning include [20], [21], [22]. In comparison, the simpler information pattern of the open-loop game (the choice of controls only depends on the initial condition) reduces the computational task to solving low dimensional HJB equations on the configuration space of each player, rather than a high dimensional HJI equation on the joint configuration space of all the players (required for continuous state feedback). This reduction is particularly effective for games with many players, as the computational complexity of numerical algorithms solving HJI equations scales exponentially in the number of state dimensions. Furthermore, the HJB formulation allows us to take advantage of efficient and robust numerical algorithms, such as the fast marching method (FMM). In comparison with geometric planning methods, our grid-based numerical optimal control approach allows us to handle complicated

This work is supported by the AFOSR CHASE MURI under grant FA9550-10-1-0567, HUNT Project under grant 550740, and the NSF Large CPS ActionWebs Project under grant CNS-0931843. R.T., J.D. and C.J.T. are with the Electrical Engineering and Computer Sciences department at UC Berkeley. {rrtakei,jding,tomlin}@eecs.berkeley.edu. H.H. is with the Aeronautics and Astronautics department at Stanford University. haomiao@stanford.edu

domain geometry, as well as inhomogeneous speed profiles (possibly due to variations in the physical terrain).

The remainder of the paper is organized as follows. Section II describes the open-loop game formulation of the safe motion planning problem and its solution technique using a "modified" fast marching method. Section III shows how the method can be generalized to multi-stage settings. Numerical results are presented in Section IV. Section V concludes the paper and discusses future directions.

II. Open-loop pursuit-evasion game

In this section, we present the open-loop pursuit-evasion game formulation of the safe motion planning problem. We also describe an algorithm which computes the set of points reachable by the agent while avoiding "capture" (or collision) with the adversarial point obstacle.

A. Open-loop game formulation

Suppose the agent, A, and the moving point obstacle, B, are both confined to an open domain $\Omega \subset \mathbb{R}^n$. Let $\mathbf{x}(t) = (x_a(t), x_b(t)) \in \Omega^2$ represent the joint states of A and B, respectively, at a given time $t \geq 0$, with initial states $\mathbf{x}(0) = \mathbf{x}^0 = (x_a^0, x_b^0)$. Let U, W be the compact sets of control values for A, B, respectively, and $\mathbb{U} = \{u \colon [0,\infty) \to U, \text{ measurable}\}, \mathbb{W} = \{w \colon [0,\infty) \to W, \text{ measurable}\}$, be the sets of admissible controls. The dynamics of the two players are assumed to be holonomic and decoupled:

$$\begin{aligned} \dot{x}_a(t) &= f_a(x_a(t), u(t)), \qquad u \in \mathbb{U}, \\ \dot{x}_b(t) &= f_b(x_b(t), w(t)), \qquad w \in \mathbb{W}. \end{aligned} \tag{1}$$

Furthermore, we assume f_a and f_b are Lipschitz continuous and that for all $y \in \Omega$, there exists $\tilde{w} \in W$ such that $f_b(y, \tilde{w}) = 0$. This allows B the option to stop moving.

We shall say a *path* $x_a(\cdot)$ (resp. $x_b(\cdot)$) is *admissible* if it is a solution to (1) for some $u \in \mathbb{U}$ (resp. $w \in \mathbb{W}$), such that $x_a(t) \in \Omega$ (resp. $x_b(t) \in \Omega$), $\forall t \geq 0$. To simplify notation, we will not explicitly write the dependency of the path $x_a(\cdot)$ (resp. $x_b(\cdot)$) on the control input u (resp. w) and the initial condition x_a^0 (resp. x_b^0).

Let $\mathcal{T}_1 \subset \Omega$ denote a closed *target set*. A's objective is to reach $\mathcal{T}_1$ as quickly as possible, while avoiding capture by B. We define capture as the event in which the positions of both players coincide: $x_a(t) = x_b(t)$ for some time $t \geq 0$. The corresponding cost function for A is then a time-to-reach function, subject to a safety constraint:

$$\mathcal{J}_{\mathcal{T}_1}(\mathbf{x}^0, u, w) = \inf\{t \mid x_a(t) \in \mathcal{T}_1, x_a(s) \neq x_b(s), \forall s \leq t\}. \tag{2}$$

Throughout this article, we assume that the infimum of the empty set is infinity.

Given $\mathbf{x}^0$, the value function of a differential game depends on the type of control strategy invoked by each player. For an *open-loop game*, the control strategies are as follows: At time $t = 0$, A chooses its complete control over the time horizon of interest, assuming that B will always counter with its worst-case control. Thus, the upper value $v_1(\mathbf{x})$ for the open-loop game is given by

$$v_1(\mathbf{x}^0) = \inf_{u \in \mathbb{U}} \sup_{w \in \mathbb{W}} \mathcal{J}_{\mathcal{T}_1}(\mathbf{x}^0, u, w). \tag{3}$$

The definition (3) is a conservative estimate of the value from A's perspective: If $v_1(\mathbf{x}^0) < \infty$, A is guaranteed a control $u^* \in \mathbb{U}$ that will reach $\mathcal{T}_1$ at time $t = v_1(\mathbf{x}^0)$ while avoiding capture, *for any* control of B (hence ensuring collision avoidance). On the other hand, $v_1(\mathbf{x}^0) = \infty$ only means that *knowing A's choice of control*, B has a choice of controls resulting in collision with A.

In what follows, we describe a method to compute $v_1(\mathbf{x}^0)$ and the time-optimal control/path from a given $\mathbf{x}^0$ to $\mathcal{T}_1$.

B. Safe-reachable set

Given $\mathbf{x}^0$, we say that a point $y \in \Omega$ is *safe-reachable* if there exists $u \in \mathbb{U}$ and $t \geq 0$ such that $x_a(t) = y$ and $x_a(s) \neq x_b(s)$ for all $s \in [0, t]$ and $w \in \mathbb{W}$; we refer to such $x_a(\cdot)$ as a *safe-reachable path*. The *safe-reachable set* of A for the system (1) is defined as

$$\mathcal{S}_1 = \{y \in \Omega \mid y \text{ is safe-reachable}\}. \tag{4}$$

Clearly, $v_1(\mathbf{x}^0) < \infty$ if and only if $\mathcal{S}_1 \cap \mathcal{T}_1 \neq \emptyset$.

Let us define the minimum time-to-reach function $\varphi_1 \colon \Omega \to \mathbb{R}$ for A, constrained to the set $\mathcal{S}_1$:

$$\varphi_1(y) = \min\{t \mid x_a(t) = y, x_a(s) \in \mathcal{S}_1, \forall s \leq t\}. \tag{5}$$

Similarly, we define the minimum time-to-reach function $\psi \colon \Omega \to \mathbb{R}$ for B in Ω:

$$\psi(y) = \min\{t \mid x_b(t) = y, x_b(s) \in \Omega, \forall s \leq t\}. \tag{6}$$

It can be verified that $\mathcal{S}_1$ is the maximal set satisfying

$$\varphi_1(y) < \psi(y), \; \forall y \in \mathcal{S}_1. \tag{7}$$

Indeed, suppose there exists $\mathcal{S} \supset \mathcal{S}_1$ which satisfies both (5) and (7), with $\mathcal{S}_1$ replaced by $\mathcal{S}$. Then by (7), for any $z \in \mathcal{S} \backslash \mathcal{S}_1$, the optimal path $x_a^*(\cdot)$ from x_a^0 to z in $\mathcal{S}$ has the property that $\varphi_1(x_a^*(t)) < \psi(x_a^*(t)), \forall t \in [0, \varphi_1(z)]$. But this implies $x_a^*(\cdot)$ is a safe-reachable path to z, and so $z \in \mathcal{S}_1$, which results in a contradiction.

Fig. 1 illustrates $\mathcal{S}_1$, φ_1 and ψ for $\Omega \subset \mathbb{R}^2$ where A is twice as fast as B. Note that in general the safe-reachable set is not necessarily the same as the equal time-to-reach points for the two players, as there may be points that are only reachable for A if it moves past B. For comparison, we have overlaid the locus representing the equal time-to-reach points of A and B.

The main result relates φ_1 to the value function v_1:

Theorem 1: $v_1(\mathbf{x}^0) = \min\{\varphi_1(y) \mid y \in \mathcal{T}_1\}$.

Proof: First, consider the case in which $v_1(\mathbf{x}^0) = \infty$. According to definition (3), for any $u \in \mathbb{U}$, there exists a $w \in \mathbb{W}$ such that B captures A before the target set $\mathcal{T}_1$ is reached. Therefore, $\mathcal{S}_1 \cap \mathcal{T}_1 = \emptyset$ and from (5), $\varphi_1(y) = \infty$ for all $y \in \mathcal{T}_1$, as desired.

If $v_1(\mathbf{x}^0) < \infty$, let $x_a^*(\cdot)$ be the path corresponding to the control $u^* = \arg\min_{u \in \mathbb{U}} \max_{w \in \mathbb{W}} \mathcal{J}_{\mathcal{T}_1}(\mathbf{x}^0, u, w)$. Note that necessarily $x_a^*(t) \in \mathcal{S}_1$ for all $t \leq v_1(\mathbf{x}^0)$; indeed, if $x_a^*(s) \notin \mathcal{S}_1$ for some $s < v_1(\mathbf{x}^0)$, there then exists some $w \in \mathbb{W}$ such that $x_a^*(s) = x_b(s)$ thereby implying the contradiction $v_1(\mathbf{x}^0) = \infty$. Thus, $v_1(\mathbf{x}^0)$ is the minimum time-to-reach to

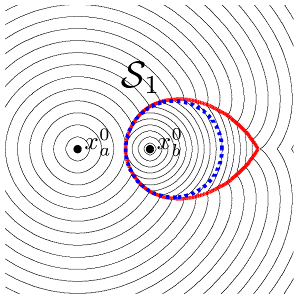


Fig. 1. $\mathcal{S}_1$ is the subset partitioned by the solid red curve containing x_a^0. Superimposed are level lines of φ_1 on $\mathcal{S}_1$ and ψ on $\Omega\backslash\mathcal{S}_1$. The dotted blue circle is the locus of equal time-to-reach points of A and B.

the target set $\mathcal{T}_1$ along admissible paths contained in the set $\mathcal{S}_1$, which is precisely $\min\{\varphi_1(y) \mid y \in \mathcal{T}_1\}$. ■

In light of Theorem 1, the problem (and efficiency) of evaluating $v_1(\mathbf{x}^0)$ hinges on the construction method for φ_1. In optimal control theory, it is well known that the minimum time-to-reach function is the *viscosity solution* to a Hamilton-Jacobi-Bellman (HJB) equation [23]. In particular, φ_1 defined as per (5) satisfies the HJB equation

$$-\min_{\tilde{u}\in U}\{\nabla\varphi_1(y)\cdot f_a(y,\tilde{u})\} = 1,\ \forall y \in \mathcal{S}_1\backslash\{x_a^0\}, \quad (8)$$

with boundary conditions

$$\varphi_1(x_a^0) = 0, \quad (9)$$

$$\varphi_1(y) = \infty, \quad y \in \Omega\backslash\mathcal{S}_1. \quad (10)$$

While problems of the form (8)-(10) can be solved in many ways when $\mathcal{S}_1$ is known a priori, the difficulty here lies in the fact that $\mathcal{S}_1$ itself depends on φ_1 (i.e. $\mathcal{S}_1$ is the maximal set satisfying (7)). To solve this coupled problem, we propose an algorithm inspired by the fast marching method (FMM).

C. The modified fast marching method

The FMM [24], [25], [26], is a single-pass method used to numerically approximate the *Eikonal equation*

$$\alpha(y)\|\nabla\varphi(y)\| = 1, \quad y \in \Omega\backslash\mathcal{T}, \quad (11)$$

equipped with some Dirichlet boundary conditions $\varphi = \infty$ on $\partial\Omega$ and $\varphi(y) = g(y), \forall y \in \mathcal{T}$, for some closed $\mathcal{T} \subset \Omega$. Note that the equation in (11) is equivalent to that in (8) if $f_a(y,\tilde{u}) = \alpha(y)\tilde{u}, \forall y \in \Omega, \tilde{u} \in U$, where U is a unit disc and α is a positive scalar function representing the speed.

Suppose we are to approximate φ in (11) by a grid function $\varphi_{i,j}$ on a uniform 2-D Cartesian grid $\mathcal{G}$, where $\varphi_{i,j} \approx \varphi(x_{i,j})$, $x_{i,j} = (ih, jh) \in \mathcal{G}$ and h is the grid spacing. To simplify the treatment of the boundary conditions, we assume that $\partial\Omega$ is well-discretized by the grid points $\partial\mathcal{G} \subset \mathcal{G}$. Consider the finite difference approximation of (11),

$$\alpha(x_{i,j})/h \begin{bmatrix} (\varphi_{i,j} - \min\{\varphi_{i\pm1,j}, \varphi_{i,j}\})^2 + \\ (\varphi_{i,j} - \min\{\varphi_{i,j\pm1}, \varphi_{i,j}\})^2 \end{bmatrix}^{1/2} = 1 \quad (12)$$

The scheme (12) is consistent and stable, and converges to the viscosity solution of (11) as $h \to 0$ [27]. Let us write $\varphi^*_{i,j} = \varphi^*_{i,j}(\varphi_{i\pm1,j}, \varphi_{i,j\pm1}, \alpha(x_{i,j}))$ as the solution to (12) for $\varphi_{i,j}$ (using the quadratic formula).

At each iteration, FMM partitions $\mathcal{G}$ into Accepted (where the approximation is obtained), Narrow-Band (candidates to be added to Accepted) and Far Away (neither Accepted nor Narrow-Band). The key principle exploited by the FMM is that of "causality" [24], which states that the solution at a node only depend on adjacent nodes that have smaller values. This defines an ordering of the nodes, in increasing values of $\varphi_{i,j}$; this ordering is realized by adding the smallest valued candidate in Narrow-Band into Accepted at each step, until all nodes are in Accepted. The FMM is executed as follows:

1) Set $\varphi_{i,j} = \infty, \forall x_{i,j} \in \mathcal{G}$ and label them as Far Away.
2) Set $\varphi_{i,j} = g(x_{i,j}), \forall x_{i,j} \in \partial\mathcal{G}$, and label them as Accepted.
3) For all nodes $x_{i,j}$ adjacent to a node in Accepted, set $\varphi_{i,j} = \varphi^*_{i,j}$ and label them as Narrow-Band.
4) Choose a node $x^{\min}$ in Narrow-Band with the smallest $\varphi_{i,j}$ and label it as Accepted.
5) Set $\varphi_{i,j} = \varphi^*_{i,j}$ for all non-Accepted nodes adjacent to $x^{\min}$ from the previous step, and (re-)label them as Narrow-Band.
6) Repeat from step 4, until all nodes are labeled Accepted.

The algorithm terminates in a finite number of iterations, since the total number of nodes is finite. On a grid with N nodes, the complexity is $O(N\log N)$ and the algorithm naturally extends to three or higher dimensions [25]. We note, however, that the number of nodes N required for accurate approximation of φ would in general scale exponentially with the number of state dimensions.

Remark 2: Strictly speaking, in the general setting (8) where the dynamics are not isotropic, the FMM is not applicable. Instead, a similar causality-ordering procedure is possible via the Ordered Upwind Method (OUM) [28]. To simplify the presentation, we will refer to all causality-ordering methods as FMM, with the caveat that OUM should be applied for anisotropic dynamics.

We now describe a modified FMM to compute the solution to (8)-(10) with the maximal $\mathcal{S}_1$ that satisfies (7). Assume that (6) is approximated by $\psi_{i,j} \approx \psi(x_{i,j})$ at each $x_{i,j} \in \mathcal{G}$. The key observation is, *any node $x_{i,j}$ reachable by A in time $t = \varphi_{i,j} \geq \psi_{i,j}$ must not belong in $\mathcal{S}_1$*. This calls for a simple modification to the FMM; namely, implement the method with boundary condition (9) and insert immediately after step 4:

4.5) At the node $x^{\min}_{i,j}$, if $\varphi_{i,j} \geq \psi_{i,j}$, then set $\varphi_{i,j} = \infty$.

To see why this modification is sufficient, suppose that, at the start of step 3 of the current iteration, all Accepted nodes have the correct $\varphi_{i,j}$ value. Suppose also that $x_{i,j}$ is the smallest element in Narrow Band and $\varphi^*_{i,j} \geq \psi_{i,j}$. Since $\varphi^*_{i,j}$ is computed using neighboring Accepted nodes (which we are assuming to have correct values), this implies that an optimal path in $\mathcal{S}_1$ would take longer than $\psi_{i,j}$ to reach $x_{i,j}$. Then, $\varphi_{i,j} = \infty$, since $x_{i,j}$ cannot be safe-reachable. On the other hand, if $\varphi^*_{i,j} < \psi_{i,j}$, there is a safe-reachable path in $\mathcal{S}_1$ that takes less than $\psi_{i,j}$ time to reach $x_{i,j}$. Thus, after step 4.5, $\varphi_{i,j} < \infty$ if and only if $x_{i,j} \in \mathcal{S}_1$. Since all Accepted nodes are initially correct (step 2), the above argument holds inductively until all Accepted nodes are computed correctly.

D. Extracting A's optimal control and path

If $v_1(\mathbf{x}^0) < \infty$, the next step is to identify A's optimal control $u^* \in \arg\min_{u\in\mathbb{U}} \max_{w\in\mathbb{W}} \mathcal{J}_{\mathcal{T}_1}(\mathbf{x}^0, u, w)$. Note that u^* is not necessarily unique, but all such controls yield the same value $v_1(\mathbf{x}^0)$. The function φ_1, which was used in the evaluation of $v_1(\mathbf{x}^0)$ is particularly convenient for this purpose.

Note that u^* is precisely the control for A corresponding to a time-optimal safe-reachable path from x_a^0 to a point $x_a^f \in \arg\min_{y\in\mathcal{T}_1} \varphi_1(y)$. Thus, φ_1 can be used to extract the optimal feedback control $u^* = u^*(y)$ where it is smooth. For example, in the case where $f_a(y, \tilde{u}) = \tilde{u}$ and U is a unit disc, it can be shown that $u^*(y) = -\nabla\varphi_1(y)/\|\nabla\varphi_1(y)\|$. In general, $\nabla\varphi_1(y)$ may not exist at some $y \in \Omega$; these are precisely the points where no unique optimal control exists.

The optimal path $x_a^*(\cdot)$ of A can be computed by solving the dynamical system

$$\dot{x}_a^*(t) = -f_a(x_a^*(t), u^*(x_a^*(t))) \tag{13}$$

from $t = \varphi(x_a^f)$ to $t = 0$ (backward in time), with a terminal condition $x_a^*(\varphi(x_a^f)) = x_a^f$. Note that $x_a^*(0) = x_a^0$. This can be approximated by any standard numerical ODE solver.

III. Multi-stage games

In the previous section, we presented an algorithm for computing the minimum time-to-reach to a single target set, subject to a collision avoidance constraint with respect to a moving point obstacle. In this section, we consider an extension of this algorithm to the case in which A's objective includes a collection of target sets which must be visited in sequence.

A. The open-loop value of a two-stage game

For simplicity of presentation, we discuss our approach for a two-stage safe motion planning problem (see for example the capture-the-flag game considered in [16]). Generalization to arbitrary number of stages is briefly described in section III-C.

Let $\mathcal{T}_1, \mathcal{T}_2 \subset \Omega$ be disjoint closed target sets, such that A's objective is to first reach $\mathcal{T}_1$, and then reach $\mathcal{T}_2$. As before, A's goal is to accomplish this in minimum time, while avoiding all possible locations where it may be captured by B. We shall assume that φ_1 and $\mathcal{S}_1$ have been computed using the method discussed in Section II-C, and that $\min_{y\in\mathcal{T}_1} \varphi_1(y) < \infty$, i.e. $\mathcal{T}_1$ is safe-reachable.

Suppose A travels along a time-optimal safe-reachable path to $z \in \mathcal{T}_1$ in the first stage; this will take (at most) $\varphi_1(z)$ time. Next, A's goal is to find a time-optimal safe-reachable path from z to $\mathcal{T}_2$. The open-loop value function is the minimum of the total time spent in the two stages, over all possible intermediate points $z \in \mathcal{T}_1$:

$$v_2(\mathbf{x}^0) = \min_{z\in\mathcal{T}_1}\Big[\varphi_1(z) + \min_{u\in\mathbb{U}}\max_{w\in\mathbb{W}} \mathcal{J}_{\mathcal{T}_2}\big(z, x_b(\varphi_1(z)), u, w_{\varphi_1(z)}\big)\Big], \tag{14}$$

where, $w_{\varphi_1(z)}(t) = w(t + \varphi_1(z)), \forall t \geq 0$. This time-shift in B's control indicates that, at the beginning of the second stage, B could be at any point that is reachable in time $\varphi_1(z)$.

As with the single-stage game, we shall characterize the open-loop value for the two-stage game (14) using a minimum time-to-reach function. Given $\mathbf{x}^0$ and $\mathcal{T}_1$, we say $y \in \Omega$ is *second-stage safe-reachable* if it is reachable by an admissible path of A that first visits an intermediate point in $\mathcal{T}_1$, and avoids capture by B for all $w \in \mathbb{W}$. Then, the second-stage safe-reachable set is defined as $\mathcal{S}_2 = \{y \in \Omega \mid y \text{ is second-stage safe-reachable}\}$. The analog of (5) on $\mathcal{S}_2$ is:

$$\begin{aligned}\varphi_2(y) = \min\big\{t + \varphi_1(z) \mid\ & z \in \mathcal{T}_1, t \geq 0, x_a(t) = y,\\ & x_a(0) = z, x_a(s) \in \mathcal{S}_2, \forall s \in [0, t)\big\}.\end{aligned} \tag{15}$$

In the above definition, as with (14), we assume that A takes a time-optimal safe-reachable path to the intermediate point $z \in \mathcal{T}_1$.

The main result for the two-stage game is analogous to Theorem 1. We omit the proof, as it is a direct application of the definitions given above, using a similar line of argument as in the proof of Theorem 1.

Theorem 3: $v_2(\mathbf{x}^0) = \min\{\varphi_2(y) \mid y \in \mathcal{T}_2\}$.

The maximal $\mathcal{S}_2$ and φ_2 satisfy the coupled relation:

$$\mathcal{S}_2 = \{y \in \Omega \mid \varphi_2(y) < \psi(y)\}, \tag{16}$$

$$\begin{cases} -\min\limits_{\tilde{u}\in U}\{\nabla\varphi_2(y) \cdot f_a(y, \tilde{u})\} = 1, & y \in \mathcal{S}_2 \backslash \mathcal{T}_1,\\ \varphi_2(y) = \varphi_1(y), & y \in \mathcal{T}_1,\\ \varphi_2(y) = \infty, & y \in \Omega\backslash\mathcal{S}_2. \end{cases} \tag{17}$$

Thus, $(\varphi_2)_{i,j}$ (the approximation of φ_2 on a grid $\mathcal{G}$) can be computed by performing the modified FMM described in section II-C with the boundary condition in (17).

B. Extracting A's optimal control and path

Suppose that we have computed φ_1 and φ_2, and that $v_2(\mathbf{x}^0) < \infty$. Let $x_a^f \in \arg\min\{\varphi_2(y) \mid y \in \mathcal{T}_2\}$. Then, the optimal path $x_a^*(\cdot)$ for each stage of the two-stage open-loop game can be computed sequentially. For all $y \in \mathcal{S}_2$, let $u_2^* = u_2^*(y)$ be the optimal control extracted from φ_2 (see section II-D). In the second stage, $x_a^*(\cdot)$ can be computed by solving (13), with u^* replaced by u_2^*, backward in time from $t = \varphi_2(x_a^f)$, with the terminal condition x_a^f. The ODE is solved (backward in time) until $x_a^*(t) \in \mathcal{T}_1$. From this point onwards, the optimal control $u^* = u^*(y)$ for $y \in \mathcal{S}_1$ is extracted from φ_1, until $t = 0$, when $x_a^*(0) = x_a^0$.

C. Algorithm for a multi-stage game

We now present an algorithm which computes the open-loop value function and the optimal path of A, for a K-stage game: Suppose A's objective is to find a time-optimal path that visits (closed) target sets $\mathcal{T}_1, \ldots, \mathcal{T}_K \subset \Omega$ consecutively, while avoiding capture by B. For each $k = 1, 2, \ldots, K$, the coupled relation for the k-th-stage safe-reachable set $\mathcal{S}_k$, and minimum time-to-reach function φ_k is given by

$$\mathcal{S}_k = \{y \in \Omega \mid \varphi_k(y) < \psi(y)\},$$
$$\begin{cases} -\min_{\tilde{u}\in U}\{\nabla\varphi_k(y)\cdot f_a(y,\tilde{u})\} = 1, & y \in \mathcal{S}_k\backslash\mathcal{T}_{k-1}, \\ \varphi_k(y) = \varphi_{k-1}(y), & y \in \mathcal{T}_{k-1}, \\ \varphi_k(y) = \infty, & y \in \Omega\backslash\mathcal{S}_k. \end{cases} \tag{18}$$

Furthermore, in an analogous fashion as Theorems 1 and 3, the k-th open-loop value function can be shown to satisfy $v_k(\mathbf{x}^0) = \min\{\varphi_k(y) \mid y \in \mathcal{T}_k\}$.

Algorithm 1 outlines the methodology to compute the open-loop value function for an K-stage game. Note how the k-th safe-reachable set $\mathcal{S}_k$ provides a condition on whether $\mathcal{T}_k$ contains a safe-reachable point. Also, the algorithm can easily compute all open-loop value functions for the intermediate targets (labeled as "optional output" in Algorithm 1); the computation is terminated prematurely if an intermediate target (and hence, all subsequent target(s)) is found to be not safe-reachable.

Input: Targets (waysets) $\mathcal{T}_1, \ldots, \mathcal{T}_K$; Dynamics f_a, f_b;
Initial states $\mathbf{x}^0 = (x_a^0, x_b^0) \in \Omega^2$.
Output: φ_k, $\mathcal{S}_k$ (and $v_k(\mathbf{x}^0)$) for $k = 1, \ldots, K$.

Compute ψ (see (6)) using FMM;
for $k = 1, 2, \ldots, K$ *(stages)* **do**
  Compute φ_k using the modified FMM;
  Set $\mathcal{S}_k = \{y \in \Omega \mid \varphi_k(y) < \infty\}$;
  if $\mathcal{S}_k \cap \mathcal{T}_k = \emptyset$ **then**
    $v_k(\mathbf{x}^0) = v_{k+1}(\mathbf{x}^0) = \cdots = v_K(\mathbf{x}^0) = \infty$;
    Output("$\mathcal{T}_k, \ldots, \mathcal{T}_K$ cannot be reached.");
    Return;
  end
  $v_k(\mathbf{x}^0) = \min\{\varphi_k(y) \mid y \in \mathcal{T}_k\}$; (optional output)
end

Algorithm 1: Computing the open-loop value function.

In Algorithm 2, we outline a procedure to compute A's optimal path, assuming that φ_k are known for all $k = 0, 1, \ldots, K$ (computed using Algorithm 1) and $v_K(\mathbf{x}^0) < \infty$. If Algorithm 1 was terminated at some $k' < K$ (i.e. $v_m(\mathbf{x}^0) < \infty$ for $m < k'$ and $v_m(\mathbf{x}^0) = \infty$ for $m \geq k'$), Algorithm 2 can still be invoked for $k = 1, 2, \ldots, k'-1$ thus computing an optimal safe-reachable path from x_a^0 to $\mathcal{T}_{k'-1}$. Since each stage involves one instance of the modified FMM, the total complexity for a K-stage game on a grid with N nodes is $O(KN \log N)$.

Remark 4: Suppose one "concatenates" the optimal paths of single-stage games as follows: Compute the (single-stage) optimal safe-reachable path from x_a^0 with optimal final point $y_1 \in \mathcal{T}_1$, then compute the (single-stage) optimal safe-reachable path from y_1 with optimal final point $y_2 \in \mathcal{T}_2$, etc. In general, this procedure will not yield a globally optimal path for the corresponding multi-stage game with targets $\mathcal{T}_1, \mathcal{T}_2, \ldots$. We illustrate this with a two-stage example in the absence of any obstacles, shown in Fig. 2. Indeed, the boundary condition on $\mathcal{T}_k$ in (18) is crucial in computing a globally optimal path.

Input: Targets (waysets) $\mathcal{T}_1, \ldots, \mathcal{T}_K$; Dynamics f_a;
Initial state $x_a^0 \in \Omega$; $\varphi_1, \ldots, \varphi_K$.
Output: Optimal path $x_a^*(\cdot)$ for A.

Choose $x_a^f \in \arg\min\{\varphi_K(y) \mid y \in \mathcal{T}_K\}$;
for $k = K, K-1, \ldots, 1$ **do**
  Extract optimal control $u_k^* = u_k^*(y)$ from φ_k;
  Solve $\dot{x}_a^* = f_a(x_a^*, u^*(x_a))$ backwards in time from $t_k^f = \varphi_k(x_a^f)$ with $x_a^*(t_k^f) = x_a^f$, until $x_a^*(t) \in \mathcal{T}_{k-1}$;
  Set $x_a^f = x_a^*(t)$;
end

Algorithm 2: Computing an optimal path of A.

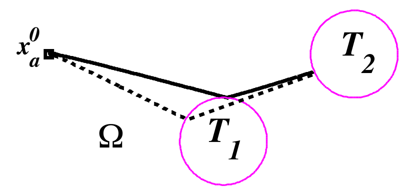


Fig. 2. The concatenation of two single-stage optimal paths (dotted) does not coincide with the two-stage optimal path (solid).

Remark 5: In our implementation of Algorithm 1, we perform the FMM (for ψ) and the modified FMM (for φ_k) on the same grid $\mathcal{G}$. The boundary condition in (18) can be implemented as $(\varphi_k)_{i,j} = (\varphi_{k-1})_{i,j}, \forall x_{i,j} \in \mathcal{T}_{k-1} \cap \mathcal{G}$. From this, we can approximate the safe-reachable sets and the value function on $\mathcal{G}$ by evaluating $\mathcal{S}_k \leftarrow \{y \in \mathcal{G} \mid (\varphi_k)_{i,j} < \infty\}$ and $v_k(\mathbf{x}^0) \leftarrow \min\{(\varphi_k)_{i,j} \mid x_{i,j} \in \mathcal{T}_k \cap \mathcal{G}\}$.

We note two immediate generalizations that could be handled in our formulation above. First, suppose A's dynamics is different at each stage, i.e. $f_a = f_a^{(k)}$ upon exiting $\mathcal{T}_{k-1}$ in the k-th stage, for $k = 1, 2, \ldots, K$ (assume that $\mathcal{T}_0 = \{x_a^0\}$). In this case, one can simply replace f_a in (18) by $f_a^{(k)}$. This could be interpreted as a hybrid system in which the discrete modes correspond to the different stages of the game. In this context, the problem of designing a safe maneuver sequence over several modes has been studied in [22].

Second, consider the presence of multiple moving point obstacles $B_1, \ldots B_p$, with distinct initial states and dynamics. The objective for A is to find a path to the target(s) that avoid possible capture by *any* B_i. In this setting, if ψ_i is the minimum time-to-reach function for each B_i, the aforementioned method applies with $\psi(y) = \min_{i=1,\ldots,p} \psi_i(y)$ for all $y \in \Omega$.

IV. Numerical results

We tested our algorithm for the capture-the-flag game on an actual 2-D map of size 400^2 pixels, shown in Fig. 3. The map data $f_{i,j}$, $i, j \in 1, \ldots, 400$ have values 0 in the obstacles (buildings) and 1 in the walkways; other regions have intermediate speeds in $(0, 1)$.

We chose the control sets U, W to be the unit disc, and the dynamics to be isotropic: $f_a(y, \tilde{u}) = \alpha(y)\tilde{u}, \forall y \in \Omega, \tilde{u} \in U$ for some scalar (speed) function α, and similarly for f_b. A's

Fig. 3. **A segment of the UC Berkeley campus used for the computational tests. (© 2011 Google)**

speed at each grid node $x_{i,j}$ was set to $5f_{i,j}$ and $2f_{i,j}$, in the first and second stages, respectively. Fig. 4 shows the case in which B's speed was set to $f_{i,j}$ at each grid node $x_{i,j}$. Fig. 5 shows the case in which a second point obstacle B_2 with speed $0.5f_{i,j}$ was added to the previous test. Note how the presence of B_2 causes A to make a detour in its path to $\mathcal{T}_2$. Since the optimal static game path for A is guaranteed to not be captured regardless of B's choice of control, we omit plotting particular paths for B.

All computations were performed on a desktop computer with 3.33 GHz Intel Core-II duo processors. The code was implemented in C using the mex compiler, running within a Matlab platform. Both tests were completed (including the computation of optimal paths) in less than 0.5 seconds each. We emphasize that the implementation and the computational efficiency are oblivious to the arbitrariness of the speed function. Also, the presence of multiple moving obstacles poses no major difficulties.

V. Concluding Remarks

We have presented an algorithm for computing a guaranteed collision-free path for an agent on a domain containing an unpredictably moving point obstacle. Our approach is to formulate the problem as an open-loop pursuit-evasion game, and characterizing the value function as the solution to a HJB equation with constraints. We solve this constrained equation by means of a novel modification of the fast marching method. The resulting algorithm is efficient, accurate, and readily adaptable to complicated domain geometry and inhomogeneous player dynamics. Generalizations to multi-stage safe motion planning (multi-stage games) and multiple moving obstacles are presented. In particular, we tested the algorithm for the capture-the-flag game on a sample map data. While our present implementation can only handle isotropic dynamics, it can be extended to anisotropic [28] and non-holonomic [29] dynamics. Furthermore, the algorithm can be implemented in higher dimensions, as well as incorporate a positive capture radius [16]. We note that the notion of an open-loop formulation was inspired by a visibility based surveillance-evasion game solver introduced in [30].

The efficiency, accuracy and versatility of the algorithm makes it suitable for a variety of practical applications. We plan to use this path planning in a smartphone-based platform that we have developed for human agents playing the game of capture-the-flag. This will allow us to evaluate the algorithm in a realistic, adversarial planning scenario, and explore the use of robust path-planning as a tool to assist humans.

References

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

[2] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 4, 1987.

[3] K. Fujimura and H. Samet, "Planning a time-minimal motion among moving obstacles," *Algorithmica*, vol. 10, pp. 41–63, 1993.

[4] T. Fraichard, "Trajectory planning in a dynamic workspace: a 'state-time space' approach," *Advanced Robotics*, vol. 13, no. 6-8, pp. 75–94, 1999.

[5] L. Jaillet and T. Siméon, "A prm-based motion planner for dynamically changing environments," *Int'l. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 1606–1611 vol. 2, 2004.

[6] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite rrts for rapid replanning in dynamic environments," in *ICRA*, April 2007, pp. 1603 –1609.

[7] K. Bekris and L. Kavraki, "Greedy but safe replanning under kinodynamic constraints," in *ICRA*. IEEE, 2007, pp. 704–710.

[8] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int'l. J. on Robotics Research*, vol. 21, no. 3, p. 233, 2002.

[9] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Ed.* Englewood Cliffs, NJ: Prentice Hall, 2002.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[11] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *ICRA*, Shanghai, China, May 2011.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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

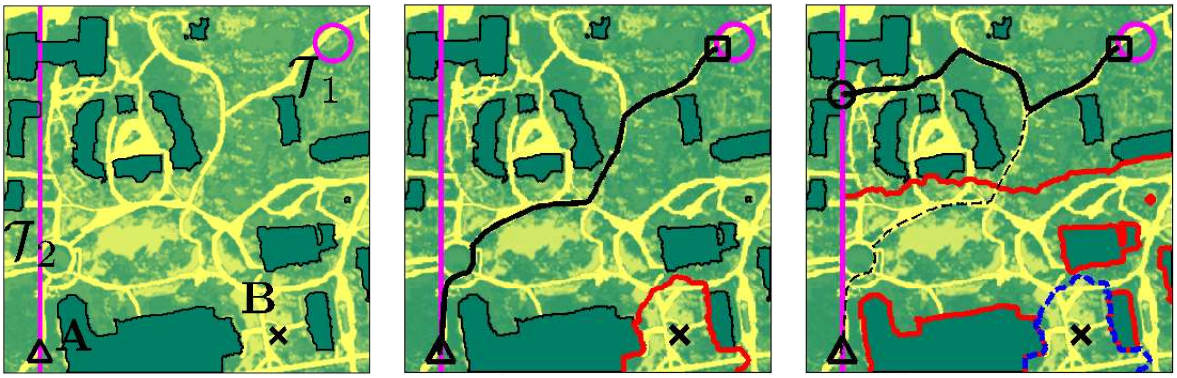
[18] I. Mitchell and J. Templeton, "A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems," *Hybrid Systems: Computation and Control*, pp. 480–494, 2005.

[19] 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.

[20] K. Alton and I. Mitchell, "Optimal path planning under defferent norms in continuous state spaces," in *ICRA*. Orlando, FL: IEEE, 2006, pp. 866–872.

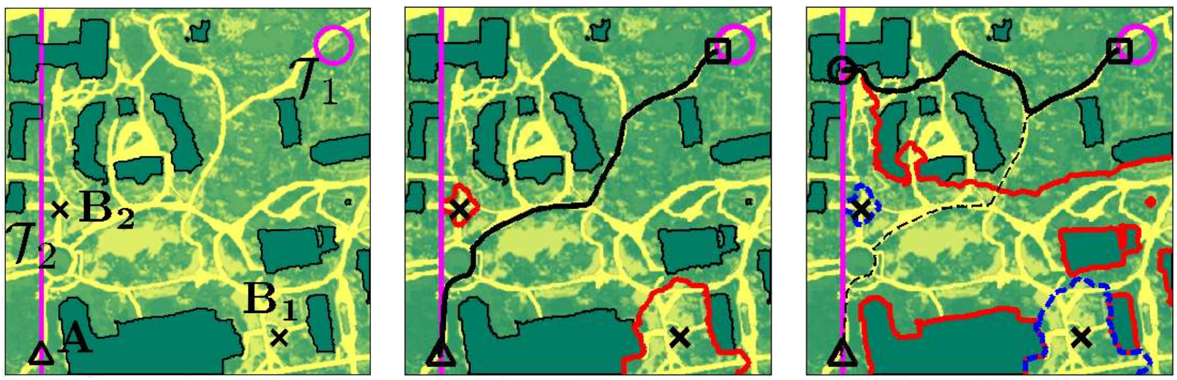
[21] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[22] J. Gillula, G. Hoffmann *et al.*, "Applications of hybrid reachability analysis to robotic aerial vehicles," *International Journal of Robotics Research*, vol. 30, no. 3, pp. 335–354, 2011.



(a) (b) (c)

Fig. 4. **Numerical results showing (a) the target sets, obstacle boundaries (black contours), and initial states of A and B, (b) $\mathcal{S}_1$ (red curve) and the safe reachable path to $\mathcal{T}_1$, and (c) $\mathcal{S}_2$ (red curve) and the safe reachable path (black curve) from $\mathcal{T}_1$ to $\mathcal{T}_2$. As a comparison, we have plotted A's optimal path (black dotted curve) and B's reachable set from the first stage (blue dotted curve).**



(a) (b) (c)

Fig. 5. **Numerical results for the same setting as in Fig. 5 (with $B = B_1$), but with an additional moving point obstacle B_2.**

[23] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, ser. Systems & Control: Foundations & Applications. Boston, MA: Birkhäuser, 1997, with appendices by Maurizio Falcone and Pierpaolo Soravia.

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

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

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

[27] E. Rouy and A. Tourin, "A viscosity solutions approach to shape-from-shading," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 867–884, 1992.

[28] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms," *SIAM J. Numer. Anal.*, vol. 41, no. 1, pp. 325–363, 2003.

[29] R. Takei, R. Tsai, H. Shen, and Y. Landa, "A practical path-planning algorithm for a vehicle with a constrained turning radius: a Hamilton-Jacobi approach," in *Amer. Control Conf.*, July 2010.

[30] R. Takei, R. Tsai, Z. Zhou, and Y. Landa, "An efficient algorithm for a visibility based surveillance-evasion game," UCLA, ftp://ftp.math.ucla.edu/pub/camreport/cam12-10.pdf, Tech. Rep., January 2012.