A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games

Ian M. Mitchell, *Member, IEEE,* Alexandre M. Bayen, *Member, IEEE,* and Claire J. Tomlin, *Member, IEEE*

***Abstract*— We describe and implement an algorithm for computing the set of reachable states of a continuous dynamic game. The algorithm is based on a proof that the reachable set is the zero sublevel set of the viscosity solution of a particular time-dependent Hamilton-Jacobi-Isaacs partial differential equation. While alternative techniques for computing the reachable set have been proposed, the differential game formulation allows treatment of nonlinear systems with inputs and uncertain parameters. Because the time-dependent equation's solution is continuous and defined throughout the state space, methods from the level set literature can be used to generate more accurate approximations than are possible for formulations with potentially discontinuous solutions. A numerical implementation of our formulation is described and has been released on the web. Its correctness is verified through a two vehicle, three dimensional collision avoidance example for which an analytic solution is available.**

***Index Terms*— Reachability, Hamilton-Jacobi equations, differential games, verification.**

I. Introduction

As THE systems we design grow more complex, it becomes more difficult to determine whether they work correctly. Consequently, verification and validation have received major attention in many fields of engineering. The simplest form of computational validation is simulation, but unfortunately it can only check a single trajectory of the system at a time. For systems with many different state values and/or many input signals, it would be prohibitively expensive to check the safety of every possible system trajectory by simulation alone. One avenue that researchers have followed in their quest to catch every potential failure mode is the computation of reachable sets, which capture the behavior of entire groups of trajectories at once.

In this paper we describe a method for computing the *backwards reachable set* of a continuous system: the set of states from which trajectories start that can reach some given target set (see figure 1). If the target set consists of those states that are known to be unsafe, then the backwards reachable set contains states which are potentially unsafe and should therefore be avoided. As an example, consider collision avoidance protocols for two aircraft. The target set would contain those states already "in collision," such as those states where the aircraft are within the five mile horizonal separation distance mandated by the Federal Aviation Administration. The backwards reachable set contains those states which will lead to a collision. In this case, the backwards reachable set will extend many miles in front of the aircraft but not significantly behind it, since the aircraft is always moving forward. If another aircraft enters this reachable set, there is cause for alarm. We examine this scenario further in section III-B and [1].

We previously explored the idea of using a time-dependent Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE) for computing reachable sets in [2], [3], although the formulation used in those papers was not well suited for numerical implementation. The modified formulation in [4], [5] worked numerically, but it has been shown to be unsound for some nonconvex problems. The novel theoretical contribution of this paper is a proof (by a reduction to a terminal cost differential game) that the viscosity solution of a particular time-dependent HJI PDE provides an implicit surface representation of the continuous backwards reachable set.

Several features set this formulation apart from the alternatives described in the literature. Compared to algorithms that impose a polygonal or ellipsoidal shape upon the reachable set, HJI based methods can represent nonconvex reachable sets and handle nonlinear dynamics with two sets of adversarially opposed input parameters—typically one set is considered to be optimally chosen control inputs and the other to be a worst case, and therefore robust, treatment of uncertain time-varying model parameters and disturbances. The time-dependent formulation has several advantages over existing stationary HJI and viability based approaches. The most fundamental is the fact that the PDE's viscosity solution is everywhere continuous and well defined. Consequently, we can apply level set methods [6] to numerically approximate its solution with subgrid accuracy and avoid the high cost of grid refinement common to these other methods. Finally, because the solution is defined both inside and outside the reachable set, it can be used to synthesize control signals to avoid the reachable set and hence the unsafe target set. For more details, see section II-C.

A MATLAB implementation of the algorithms discussed

Manuscript received January 16, 2002; revised March 18, 2004. Research supported by DARPA under the Software Enabled Control Program (AFRL contract F33615-99-C-3014), by the DoD Multidisciplinary University Research Initiative (MURI) program CoMotion (Computational Methods for Collaborative Motion) administered by the Office of Naval Research (Grant N00014-02-1-0720), by a Frederick E. Terman Faculty Award, and by a graduate fellowship provided by the Délégation Générale pour l'Armement in France.

I. M. Mitchell is with the Department of Computer Science, University of British Columbia; 2366 Main Mall, Vancouver, BC, V6T 1Z4 CANADA (email: mitchell@cs.ubc.ca).

A. M. Bayen is with the Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (email: bayen@ce.berkeley.edu).

C. J. Tomlin is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (email: tomlin@stanford.edu).

here is available from [7], and includes documentation [8] and source code for the example in section III-B and several other reach sets. While implemented in a dimensionally independent manner, the solution is approximated on a Cartesian grid of the state space, and hence memory and computational time requirements rise exponentially with dimension. In practice, systems of dimensions 1–3 can be examined interactively, while dimensions 4–5 are slow but feasible on computers with sufficient memory.

The remainder of this paper describes our time-dependent formulation, proves its correctness, and presents an example that demonstrates its accuracy and its application to air traffic control. We note that a completely different proof of the single player version of this formulation was developed in [9].

II. How to Compute the Reachable Set

In this section we formally define the reachable set for a system, discuss a few of its properties, and formulate a terminal value HJI PDE whose solution describes it. Figure 1 illustrates the set we seek to compute.

A. The Reachable Set

We model our system with the ordinary differential equation

$$\frac{dx}{dt} = \dot{x} = f(x, a, b), \tag{1}$$

where x is our state, $a(\cdot)$ is the input for player I and $b(\cdot)$ is the input for player II. In the discussion below we will assume $x \in \mathbb{R}^n$, although the methods outlined can also be applied to periodic state spaces; for example, the state space in section III-B includes a periodic angular dimension.

Assumption 1: The input signals are drawn from the following sets

$$\begin{aligned} a(\cdot) \in \mathfrak{A}(t) &\triangleq \{\phi : [t, 0] \to \mathcal{A} \mid \phi(\cdot) \text{ is measurable}\} \\ b(\cdot) \in \mathfrak{B}(t) &\triangleq \{\phi : [t, 0] \to \mathcal{B} \mid \phi(\cdot) \text{ is measurable}\} \end{aligned}$$

where $\mathcal{A} \subset \mathbb{R}^{n_a}$ and $\mathcal{B} \subset \mathbb{R}^{n_b}$ are compact and $t \in [-T, 0]$ for some $T > 0$. We will consider two input signals to be identical if they agree almost everywhere.

Assumption 2: The flow field $f : \mathbb{R}^n \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}^n$ is uniformly continuous, bounded, and Lipschitz continuous in x for fixed a and b. Consequently, given a fixed $a(\cdot) \in \mathfrak{A}(t)$, $b(\cdot) \in \mathfrak{B}(t)$ and initial point, there exists a unique trajectory solving (1) [10].

Solutions of (1) are trajectories of our system and will be denoted by

$$\xi_f(s; x, t, a(\cdot), b(\cdot)) : [t, 0] \to \mathbb{R}^n$$

which satisfies the initial conditions $\xi_f(t; x, t, a(\cdot), b(\cdot)) = x$ and the differential equation almost everywhere

$$\begin{aligned} &\frac{d}{ds}\xi_f(s; x, t, a(\cdot), b(\cdot)) \\ &\quad = f(\xi_f(s; x, t, a(\cdot), b(\cdot)), a(s), b(s)). \end{aligned}$$

Note that we employ a semi-colon to distinguish between the argument s of ξ_f and the trajectory parameters x, t, $a(\cdot)$ and $b(\cdot)$.

Assumption 3: The *target set* $\mathcal{G}_0 \subset \mathbb{R}^n$ for our reachability problem is closed and can be represented as the zero sublevel set of a bounded and Lipschitz continuous function $g : \mathbb{R}^n \to \mathbb{R}$

$$\mathcal{G}_0 = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}. \tag{2}$$

We assume that player I will try to steer the system away from the target with her input $a(\cdot)$, and player II will try to steer the system towards the target with her input $b(\cdot)$. For readers who prefer a more intuitive understanding of the inputs, consider that in our example the target set will represent the capture set in a pursuit-evasion game. Our evader (control input) will then be player I and the pursuer (adversarial disturbance) will be player II.

In a differential game setting, it is important to address what information the players know about each other's decisions. To specify our information pattern, define first a *strategy* for the second player as a map $\gamma : \mathfrak{A}(t) \to \mathfrak{B}(t)$ which specifies an input signal for player II as a function of the input signal that player I chooses. We will allow player II to use only *nonanticipative strategies*; that is strategies

$$\begin{aligned} \gamma \in \Gamma(t) \triangleq \{\vartheta : \mathfrak{A}(t) \to &\mathfrak{B}(t) \mid \\ &a(r) = \hat{a}(r) \text{ for almost every } r \in [t, s] \\ &\implies \vartheta[a](r) = \vartheta[\hat{a}](r) \\ &\qquad\qquad \text{for almost every } r \in [t, s]\}. \end{aligned}$$

Informally, this restriction means that if player II cannot distinguish between input signals $a(\cdot)$ and $\hat{a}(\cdot)$ of player I until after time s, then player II cannot respond differently to those signals until after time s. It will turn out that this choice gives an advantage to player II over player I, but we postpone further discussion of whether this information pattern is the correct one for our reachability purposes until section II-D.

Note that in our formulation of the problem, a trajectory starts at some initial time $t < 0$ and we would like to know if it has passed into or through the target set by time zero. We will sometimes want to discuss the length of time that a trajectory has had to evolve; we adopt the differential game notation $\tau = -t$ to denote this positive quantity. We use the free variables s and r to denote times in the range $[t, 0]$.

To solve the backwards reachability problem, we want to determine the *backwards reachable set* $\mathcal{G}(\tau)$ for $\tau \in [0, T]$. Remembering that $t = -\tau$, we define this set as

$$\begin{aligned} \mathcal{G}(\tau) \triangleq \{x \in \mathbb{R}^n \mid \exists \gamma \in \Gamma(t), \forall a(\cdot) \in \mathfrak{A}(t), \exists s \in [t, 0], \\ \xi_f(s; x, t, a(\cdot), \gamma[a](\cdot)) \in \mathcal{G}_0\}. \end{aligned} \tag{3}$$

Informally, $\mathcal{G}(\tau)$ is the set of states from which there exists strategies for player II that for all inputs of player I will generate trajectories which lead to the target set within time τ. A proof of the following remark can be found in [11, section 2.1].

Remark 1: If $\mathcal{G}_0$ is closed, then $\mathcal{G}(\tau)$ is closed.

B. A Time-Dependent Hamilton-Jacobi-Isaacs Equation for the Reachable Set

In this section we state the main theoretical result of this paper—that the reachable set can be determined by solving for the viscosity solution [12] of a time-dependent HJI equation.

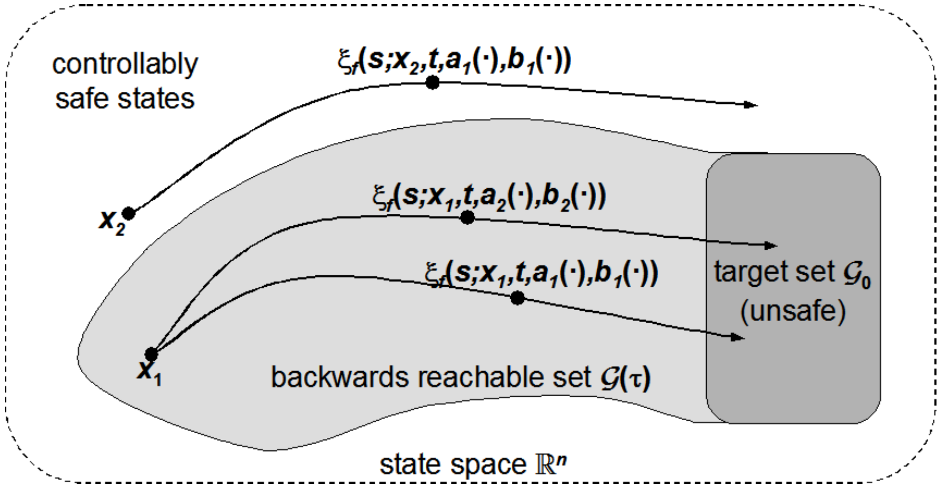

Fig. 1. Target set and backwards reachable set. Several trajectories are shown starting at the same time t but from different states x and subject to different input signals $a(\cdot)$ and $b(\cdot)$. Input signal $a(\cdot)$ is chosen to drive the trajectory away from the target set, while input signal $b(\cdot)$ is chosen to drive the trajectory toward the target. The labeling of safe and unsafe states shown is the most common (but not the only possible) interpretation of these sets.

Theorem 2: Let $v : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the viscosity solution of the terminal value HJI PDE

$$\begin{aligned} D_t v(x,t) + \min[0, H(x, D_x v(x,t))] &= 0, \\ v(x,0) &= g(x), \end{aligned} \tag{4}$$

where

$$H(x,p) = \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} p^T f(x,a,b). \tag{5}$$

Then the zero sublevel set of v describes $\mathcal{G}(\tau)$

$$\mathcal{G}(\tau) = \{x \in \mathbb{R}^n \mid v(x,t) \leq 0\}. \tag{6}$$

The proof of this theorem is given in the appendix. The goal of the proof is to show that reachability is equivalent to a terminal cost differential game. Unfortunately, such a game can only determine whether a trajectory is in the target set at exactly time zero. If such a game were used on the original system, player I could “avoid” the target by driving a trajectory into the target and then out the other side before time zero. To avoid this situation, we introduce an augmented system which allows player II to freeze the evolution of trajectories should player I attempt to drive a trajectory out of the target set. The augmented system’s trajectories are shown to be equivalent to those of the original system, and the HJI PDE of the resulting differential game is shown to be equivalent to (4).

Remark 3: Under Assumptions 1–3, it can be shown that $v(x,t)$ is bounded and Lipschitz continuous in both x and t [10, Theorem 3.2].

The significance of this theorem and the subsequent remark is that we can harness well developed numerical schemes from the level set literature to compute accurate approximations of $v(x,t)$, and therefore accurate approximations of $\mathcal{G}(\tau)$, for even complicated nonlinear dynamics. In previous papers we have presented alternative HJI PDE formulations for computing the backwards reachable set. In [3], the Hamiltonian was restricted to negative values only within the target set; unfortunately, the resulting potential for discontinuities in the solution makes accurate numerical implementation difficult. In [4], minimization was performed as a separate, postprocessing step. While this formulation is more efficient, reasoning about it is more challenging and its use is restricted to problems in which the Hamiltonian and target set are convex. Consequently, we advocate using the formulation in Theorem 2 to determine reachable sets.

C. Alternative Algorithms for Computing the Reachable Set

The verification of complex systems has received much attention lately; consequently, so has the design of efficient methods for computing reachable sets. In this section we review a variety of alternative algorithms. Further discussion can be found in [11, section 2.3].

The search for methods of computing the reachable sets of purely discrete systems, such as those modeled by finite automata, has met with considerable success and has led to the development of powerful tools for automatic verification; for example, the binary decision diagram [13]. Most engineering systems, however, are not purely discrete. Continuous dynamics are the norm in control engineering problems, and in many modern systems important behaviors arise from the interaction between discrete and continuous components. The primary challenge of computing reachable sets for these *hybrid systems* lies on the continuous side, where sets containing uncountable numbers of states must be represented and evolved.

Our approach to this challenge has its roots in the work of Isaacs [14], who used his calculations to derive capture regions in pursuit-evasion games. The modern methods most closely related to ours are based on a minimum time to reach function which is the viscosity solution of a stationary (time-independent) HJI PDE. Numerical approximations of this function can be computed iteratively for general nonlinear dynamics with two competing inputs [15], or for restricted classes of dynamics by fast iterative [16] and even noniterative [17], [18] schemes. The value of this function at each point in the state space is the minimax time required (over inputs $b(\cdot)$ and $a(\cdot)$ respectively) to go from that state to any point in the target set, and so the sublevel sets of this function can be used directly as representations of finite horizon backwards reachable sets. Furthermore, the gradient of this function can be used to deduce optimal strategies for both players' inputs for states inside the backwards reachable set.

When used in a reachability setting, the stationary HJI formulation has several disadvantages compared to the time-dependent formulation described in section II-B. Unless the system is small time controllable,[1] the minimum time to reach function may not be continuous [20]. In the neighborhood of these discontinuities—which often occur on the boundary of the reachable set—it is difficult to construct a numerical approximation that achieves even grid level accuracy. Achieving a given level of accuracy therefore requires computationally expensive grid refinement (halving the grid size multiplies the execution time by at least eight in three dimensions). In contrast, the continuous solution afforded by the time-dependent formulation makes it possible to achieve subgrid level accuracy, as demonstrated in section III-B and [21]. A more subtle disadvantage is the total lack of information that the minimum time to reach function provides outside the reachable set, where its value is a constant. In safety verification applications, we are at least as interested in choosing optimal input strategies for states outside the reachable set (states that are currently safe but may become unsafe without appropriate input intervention) as for states inside it (states that may already be doomed).

A second equivalent formulation of backwards reachable sets are the victory domains or discriminating kernels from viability theory [22], [23]. These techniques are more general than either time-dependent or stationary HJI approaches because they can directly handle reachability under state constraints as well as certain classes of discontinuous dynamics. Numerical schemes guaranteeing under or over approximation of the reachable set have been designed [24], features that are not yet available in numerical approximations of either HJI formulation. The disadvantages of this approach are similar to those of the stationary HJI: grid level resolution of the reachable set's boundary and a lack of information outside of it.

The stationary HJI and viability based approaches might seem to benefit dimensionally because they do not require a time variable; however, in practice this observation is misleading, because the iterations required to solve these formulations are comparable in time and memory cost to the effort needed to handle the time dimension in the time-dependent formulation.

The three approaches discussed thus far are closely related in that they calculate backwards reachable sets and work in a fixed frame of reference (Eulerian schemes) rather than following individual trajectories of the system. A second broad class of reachability algorithms finds forward reachable sets by following system trajectories (Lagrangian schemes). These include the tools *CheckMate* [25], *d/dt* [26], *HyTech* [27] and *Coho* [28], as well as the algorithms in [29], [30]. These techniques all generate polyhedral overapproximations of the reachable set and cannot treat systems with adversarial inputs.

[1] A system is *small-time controllable* if for every state, the system can remain near that state for all times and can reach nearby states in arbitrarily small amounts of time [19]. Many nonlinear dynamic systems, including the example in section III-B, are not small time controllable.

Algorithms for ellipsoidal approximation of forward reachable sets which allow for competing inputs but are restricted to linear dynamics have also been developed [31]. As a class, these techniques may scale better to high dimension (although we are not aware of any published analysis in dimension higher than six), but most find nonlinear systems difficult to analyze.

A final group of reachability algorithms partitions the state space into subsets in advance, and then builds a discrete transition system over the elements of this partition. Partitions may be rectangular [32], [33] or may be constructed from a cylindrical algebraic decomposition of polynomials appearing in the dynamics [34].

Reachable sets are also closely related to invariant sets, which are often represented as sublevel sets of Lyapunov functions. Finding such functions for systems with nonlinear dynamics is difficult in general, but algorithms for restricted classes of dynamics do exist; for example, dynamics which are piecewise linear [35], are linear with sector bounded nonlinearities [36] or are piecewise polynomial [37].

D. Discussion of Information Patterns in the Two Player Game

Throughout this paper, we have chosen to let player II select a nonanticipative strategy that can respond to the input choices of player I. In this section we discuss some possible alternatives to this information pattern. We consider four basic types of controls for the game players—open loop, state feedback, nonanticipative strategies, and anticipative strategies.

Because our reachable sets generally represent "unsafe" portions of the state space, we usually prefer to overapproximate them rather than underapproximate them. Therefore, whenever a choice must be made between giving player I or player II an advantage, we choose to give it to player II, who is trying to make the reachable set larger. If in another context player I should be given the advantage, it is straightforward to modify the Hamiltonian (5) and apply the same numerical approximation methods, although there are some technical details regarding the modified definition of the reach set [22].

An *open loop strategy* requires that both players decide their entire input signals $a(s)$ and $b(s)$ for all $s \in [t, 0]$ without any knowledge of the other players' decisions. *State feedback* allows players I and II to choose $a(s)$ and $b(s)$ respectively based on the current value of $\xi_f(s; x, t, a(\cdot), b(\cdot))$. We defined nonanticipative strategies in section II-A. Our system dynamics are deterministic, so by allowing player II to make decisions about $b(s)$ with full knowledge of $a(r)$ for $r \in [t, s]$, a nonanticipative strategy gives player II all the information of state feedback, plus player I's current input $a(s)$. While player I is at a slight disadvantage under this information pattern, at a minimum she has access to sufficient information to use state feedback, because player II must declare her strategy before player I chooses a specific input and thus player I can determine the response of player II to any input signal. An *anticipative strategy* would be equivalent to allowing player II to choose $b(s)$ based on knowledge of $a(r)$ for all $r \in [t, 0]$; in other words, player I would have to reveal her entire input signal in advance to player II.

The systems in which we are interested use state feedback controllers. Clearly the open loop pattern of information is unsuitable for verifying such systems, and the anticipative strategy model is inappropriate as well because it allows player II knowledge of the future. While state feedback might be a more appropriate model of our systems than nonanticipative strategies, it is not so easily turned into a HJI PDE. We have therefore chosen to use nonanticipative strategies, and give whatever advantage they confer to player II. It can be proven that the value of the differential game (4)–(5) under nonanticipative strategies is always less than the value under state feedback [38], and consequently our choice cannot lead to underapproximation of the state feedback reachable set.

The input of player II can be used not only for the control signal of an adversary, but also for bounded disturbance inputs and uncertain model parameters. Allowing these signals to act in an optimal nonanticipative manner is a robust, if potentially pessimistic, treatment of their effects on the system. It should be noted, however, that this treatment implicitly allows the parameter values to vary discontinuously with time; it may be excessively pessimistic for time invariant or slowly varying parameters.

III. A Collision Avoidance Example

In this section we briefly describe how to solve (4) computationally, and then apply our algorithm to a classical collision avoidance differential game. While the dynamics of this game are relatively simple, the results have been applied to the design of a collision alert system for high altitude air traffic control. Additional examples can be found in [8], [9], [21], [39].

A. Implementing a Level Set Algorithm

The HJI PDE (4) is a nonlinear PDE that in many cases lacks a classical solution. It can be shown [10] under appropriate conditions for zero sum differential games that there exists a suitable unique weak solution to (4) called the viscosity solution [12]. This solution is continuous and bounded but may develop kinks—places where the derivatives become discontinuous—even with smooth terminal conditions $g(x)$ and dynamics $f(x, a, b)$. A family of algorithms called *level set methods* has been designed specifically to compute approximations to the viscosity solution for time-dependent HJI PDEs with continuous initial conditions and Hamiltonians such as (4); for overviews of these methods, see [40], [41].

To calculate backwards reachable sets, we have written state-of-the-art MATLAB and C++ implementations of high resolution level set methods. Source code for the former is available from [7]. We describe only the key features of those implementations here, but more details are available in [8], [21]. Note that these numerical schemes are not designed to guarantee overapproximation, but rather to compute the most accurate approximation possible; the error may be of either sign.

We assume that the human modeler of the system can provide a way to compute the optimum over inputs a and b necessary to find $H(x, p)$ for fixed x and p in (5). This static local optimization may be difficult in general, but it is always easier than dynamic optimization over the full input signal histories $a(\cdot)$ and $b(\cdot)$ in the underlying dynamic game. In some common cases it is trivial; for example, $f(x, a, b) = f_1(x) + f_2(x)a + f_3(x)b$, which is the form taken in (7) below.

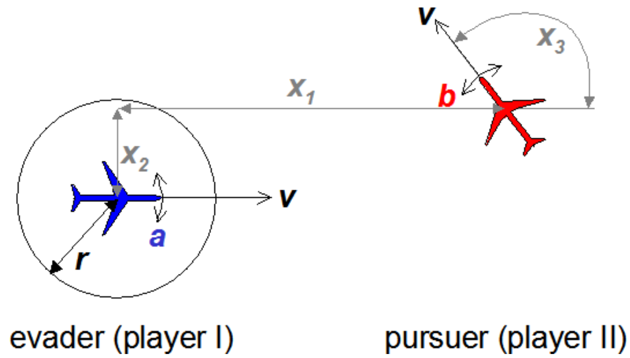

Fig. 2. Relative coordinate system for collision example.

A Lax-Friedrichs approximation to the Hamiltonian [42] ensures stability. The spatial derivative $D_x v(x, t)$ is computed with a fifth order accurate weighted essentially non-oscillatory approximation [43]. Time integration is accomplished with a second or third order total variation diminishing explicit Runge-Kutta scheme [44]. For most target sets $\mathcal{G}_0$, implicit surface functions $g(x)$ for the terminal conditions are straightforward to construct using intersection, union and set complement on basic shapes whose analytic implicit surface functions are known, such as spheres, cylinders and hyperplanes. We apply linear extrapolation away from zero along non-periodic edges of the necessarily finite computational domain.

B. The Collision Avoidance Differential Game

In this section we use the algorithms described above to determine reachability for a three dimensional kinematic model of two adversarial vehicles: the *pursuer* wishes to get within a certain distance of the *evader*. In the dynamic game literature this problem is called *the game of two identical cars* [45], and the reachable set corresponds to the set within which the pursuer can capture the evader. Our previous publications [3]–[5] have called this problem the *three dimensional aircraft collision avoidance example*. MATLAB code to generate this reachable set is available at [7].

Each vehicle is modeled as a simple kinematic point object with planar position and heading, fixed linear velocity and controllable angular velocity.

We say that a *collision* has occurred if the two vehicles come within distance r of one another (also sometimes called a *loss of separation*). Our goal is to determine the set of states from which the pursuer can cause a collision to occur. Translating into reachability terms, $\mathcal{G}_0$ is the set of all states where the two vehicles are within r units of one another, the evader is player I (angular velocity input a), the pursuer is player II (angular velocity input b), and the capture set is $\mathcal{G}(\tau)$. Because $\mathcal{G}_0$ depends only on the relative positions of the vehicles, we can simplify the system down to three dimensions by working in relative coordinates $x \in \mathbb{R}^2 \times [0, 2\pi[$. As shown in figure 2,

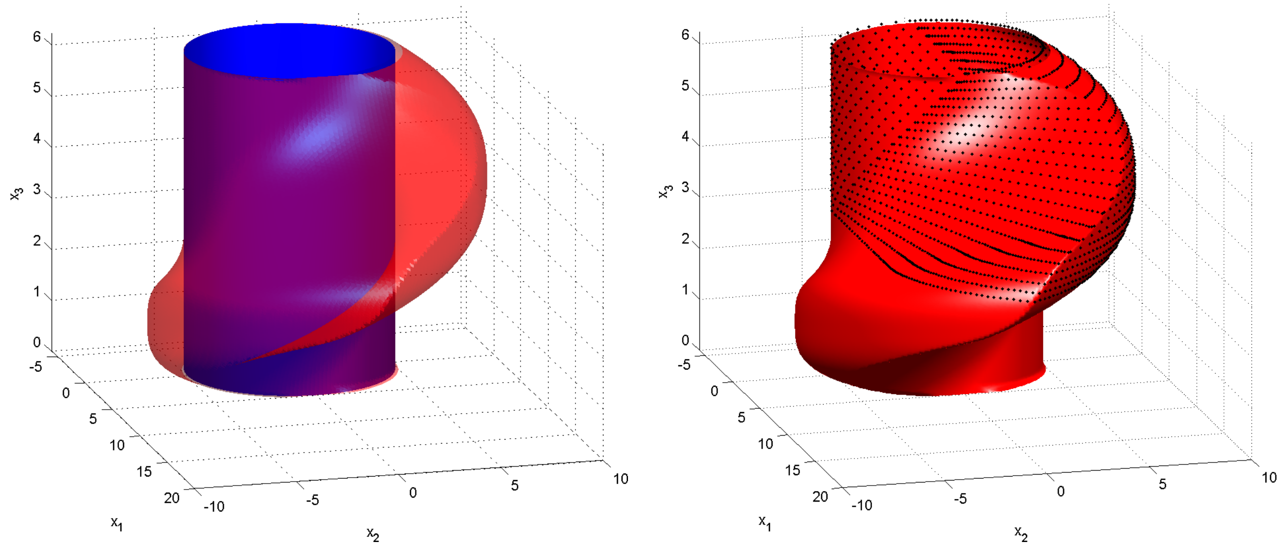

Fig. 3. Left: target set (solid cylinder) and backwards reachable set (transparent) for the collision avoidance example in three dimensions on the $N = 126$ grid. Right: backwards reachable set (solid) and 2612 analytically determined points on the reachable set's boundary (dots).

we fix the evader at the origin and facing along the positive x_1 axis. Then the pursuer's relative location and heading are described by the flow field

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v_a + v_b \cos x_3 + a x_2 \\ v_b \sin x_3 - a x_1 \\ b - a \end{bmatrix} = f(x, a, b). \quad (7)$$

The resulting Hamiltonian is

$$\begin{aligned} H(x,p) &= \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} \left[p^T f(x,a,b) \right], \\ &= \begin{pmatrix} -p_1 v_a + p_1 v_b \cos x_3 + p_2 v_b \sin x_3 \\ + \alpha |p_1 x_2 - p_2 x_1 - p_3| - \beta |p_3| \end{pmatrix}, \end{aligned} \quad (8)$$

where the bounds on inputs are $a \in \mathcal{A} = [-\alpha, +\alpha]$ and $b \in \mathcal{B} = [-\beta, +\beta]$. Since a collision can occur at any relative heading, the target set $\mathcal{G}_0$ depends only on x_1 and x_2 and includes any state within distance r of the planar origin

$$\begin{aligned} \mathcal{G}_0 &= \{x \in R^3 | x_1^2 + x_2^2 \leq r^2\}, \\ g(x) &= \sqrt{x_1^2 + x_2^2} - r, \end{aligned} \quad (9)$$

for our HJI PDE's terminal conditions (the bounded computational domain ensures that this $g(x)$ is bounded). We wish to determine $\mathcal{G}(\tau)$ in the limit $\tau \to \infty$, so we compute (4) backward from $t = 0$ until we find $H(x, D_x v) \approx 0$. For the parameters considered below, this occurs beyond $t = -2.6$.

If the vehicles are identical ($v_a = v_b$ and $\mathcal{A} = \mathcal{B}$) then it is possible using differential game theory to determine the optimal inputs for both pursuer and evader and thereby find points lying on the surface of the reachable set. This game was solved with the pursuer at the origin in [45]; we have recreated these original results, and then modified them to solve the game with the evader at the origin [46] (it turns out that the two cases are not symmetric). We use these analytic results to validate the correctness of our calculations for the parameters

$$\begin{aligned} r &= 5, \\ v_a = v_b &= 5, \\ \mathcal{A} = \mathcal{B} &= [-1, +1]. \end{aligned}$$

Figure 3 shows the target set, backwards reachable set, and some analytically determined points lying on its surface (animations showing the growth of this set are available at [7], [47]). Using a much larger set of points (approximately 240 000), we evaluated the accuracy of our reachable set algorithm. Figure 4 shows that we achieve better than first order accuracy in average and root mean square norms, and nearly first order accuracy in maximum norm. The maximum error is approximately a grid cell, and the average error is less than 3% of a grid cell. We have achieved similar or better results for a number of other examples in two and three dimensions; for more details, see [21].

While this system may seem simplistic, the dynamics and choice of parameters are flexible enough that it can be successfully applied to detect potential conflicts in en route air traffic control [1]; we summarize those results here. Given the monitoring accuracy in the current system, it has been shown previously that kinematic equations are sufficiently predictive of high altitude traffic (see [48] and references therein), which makes equation (7) an accurate enough model of the relative motion of two aircraft.

The main task of Air Traffic Controllers is to prevent *losses of separation*: to keep the aircraft separated by more than five nautical miles. Two aircraft with potentially conflicting flight paths are modeled as two differential games in which each aircraft is a threat to the other; consequently, in each game one aircraft fulfills the role of pursuer (input b) and one the role of evader (input a). We use *Enhanced Traffic Management System* (ETMS) data [1] to extract the relevant values for v_a, v_b and the corresponding ranges $\mathcal{A}$ and $\mathcal{B}$, which are aircraft dependent. ETMS data also contains recorded flight plans for all aircraft in the National Airspace System; in other words, for

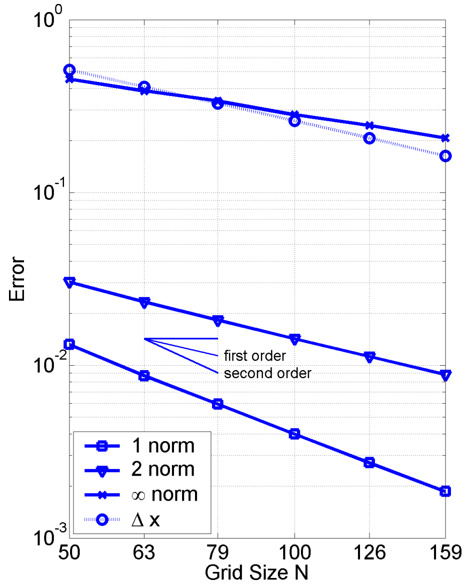

Fig. 4. Experimental convergence rates for the collision avoidance example. Average (□), root mean square (▽) and maximum (×) errors are shown. For comparison the grid cell spacing Δx (○) is included, as are lines equivalent to first order and second order convergence rates.

each aircraft the successive radar measurements of its location. We use a collection of precomputed reachable sets (run with a variety of realistic parameters v_a, v_b, $\mathcal{A}$ and $\mathcal{B}$) to check for each pair of aircraft whether one is in the other's reachable set. If this happens, we classify the situation as a conflict.

An example of a conflict is shown in Figure 5. In the first subplot, the aircraft flying from Anchorage to Los Angeles is treated as pursuer while the aircraft flying from Philadelphia to San Francisco is the evader. In the second subplot, the roles are reversed. In each subplot, slices of the reachable set (for the appropriate relative heading) are shown around the evader. A conflict is detected, and based on the subsequent ETMS data we can deduce that it was resolved by an altitude change initiated by the Air Traffic Controller.

We ran approximately 1600 such examples with data from the Oakland Air Route Traffic Control Center, which is in charge of northern California and Nevada. We showed that our method caught all potential conflicts but generated three false alarms, and is therefore a little conservative. We have thus proposed this method as an accurate technique for automated conflict identification in en route high altitude traffic.

IV. Conclusions and Future Work

We have presented an algorithm which can numerically compute the backwards reachable set for a two player, nonlinear differential game with a general target set. The algorithm is based on a formulation of reachability in terms of the viscosity solution of a time-dependent Hamilton-Jacobi-Isaacs PDE, and we have proven that the analytic solution of this equation is the exact reachable set as defined by (3). Unlike related techniques based on stationary HJI PDEs or viability theory, the solution of the time-dependent HJI PDE is continuous and defined throughout the state space, so this formulation can achieve the same level of accuracy on a coarser (and hence computationally less expensive) grid, and provides information about optimal control choices both inside and outside the reachable set. Our implementation is based on level set methods, and its accuracy and convergence have been demonstrated by a three dimensional pursuit evasion example.

The primary weakness of this formulation of reachable sets, and many others, is the exponential growth of memory and computational cost as the system dimension increases. One way to mitigate these costs is to project the reachable set of a high dimensional system into a collection of lower dimensional subspaces [49]. Any error introduced by the projection yields an overapproximation of the reachable set. We are currently determining for what types of systems such an overapproximation yields useful results.

We are extending our time-dependent HJI method in several directions. Many problems of interest are modeled by hybrid systems, and the general reachability algorithm for such systems [3] requires the introduction of state constraints to the continuous portion of the reachable set. We are also investigating probabilistic definitions of reachability for stochastic systems. Finally, we are examining how to use the results of the time-dependent solution to synthesize provably safe control laws that avoid the reachable set.

Acknowledgements

We would like to thank Professors Ronald Fedkiw and Stanley Osher for extensive discussions about the details of numerical schemes for solving the Hamilton-Jacobi PDE. In addition, we would like to thank Professors John Lygeros, L. C. Evans, Shankar Sastry and Alexander Kurzhanski for discussions about the previous and current time-dependent HJI formulations. Professors L. C. Evans and Doron Levy aided us with some technical details in the proof of Theorem 2.

References

[1] A. Bayen, S. Santhanam, I. Mitchell, and C. Tomlin, "A differential game formulation of alert levels in ETMS data for high altitude traffic," in *Proceedings of the AIAA Conference on Guidance, Navigation and Control*, Austin, TX, 2003, AIAA Paper 2003-5341.

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

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

[4] I. Mitchell and C. Tomlin, "Level set methods for computation in hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, B. Krogh and N. Lynch, Eds. Springer Verlag, 2000, no. 1790, pp. 310–323.

[5] I. Mitchell, A. Bayen, and C. J. Tomlin, "Validating a Hamilton-Jacobi approximation to hybrid system reachable sets," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. D. D. Benedetto and A. Sangiovanni-Vincentelli, Eds. Springer Verlag, 2001, no. 2034, pp. 418–432.

[6] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[7] [Online]. Available: http://www.cs.ubc.ca/~mitchell/ToolboxLS

[8] I. M. Mitchell, "A toolbox of level set methods," Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2004-09, July 2004. [Online]. Available: http://www.cs.ubc.ca/~mitchell/ToolboxLS/toolboxLS.pdf

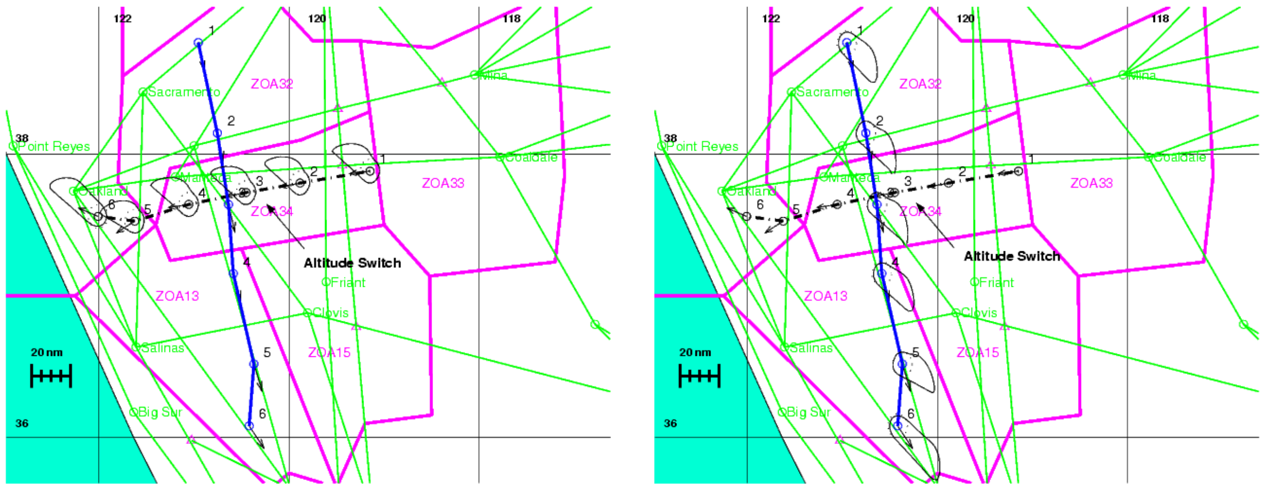

Fig. 5. Aircraft 1 (dash dotted), arriving at San Francisco Airport (SFO) from Philadelphia Airport; aircraft 2 (solid), en route from Ted Stevens Anchorage Airport to Los Angeles Airport. Aircraft 1 is using the Modesto 2 arrival to SFO. Interpolation of the positions of both aircraft between the labels 2 and 3 shows that aircraft 2 is in the $\mathcal{G}(\tau)$ for aircraft 1. ATC avoids the conflict by commanding aircraft 1 to descend to 24,000ft (which initiates the descent into SFO). For more details, see [1]. ETMS data courtesy of NASA Ames.

[9] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

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

[11] I. M. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Scientific Computing and Computational Mathematics Program, Stanford University, August 2002.

[12] M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 282, no. 2, pp. 487–502, 1984.

[13] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, August 1986.

[14] R. Isaacs, *Differential Games*. John Wiley, 1967.

[15] M. Bardi, M. Falcone, and P. Soravia, "Numerical methods for pursuit-evasion games and viscosity solutions," in *Stochastic and Differential Games: Theory and Numerical Methods*, ser. Annals of International Society of Dynamic Games, M. Bardi, T. Parthasarathy, and T. E. S. Raghavan, Eds. Birkhäuser, 1999, vol. 4.

[16] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao, "Fast sweeping methods for a class of Hamilton-Jacobi equations," *SIAM Journal on Numerical Analysis*, vol. 41, no. 2, pp. 673–694, 2003.

[17] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static Hamilton-Jacobi equations: Theory and algorithms," *SIAM Journal on Numerical Analysis*, vol. 41, no. 1, pp. 325–363, 2003.

[18] M. Broucke, M. D. D. Benedetto, S. D. Gennaro, and A. Sangiovanni-Vincentelli, "Optimal control using bisimulations: Implementation," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. D. D. Benedetto and A. Sangiovanni-Vincentelli, Eds. Springer Verlag, 2001, no. 2034, pp. 175–188.

[19] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. New York: CRC Press, 1993.

[20] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Boston: Birkhäuser, 1997.

[21] I. M. Mitchell, "Demonstrating numerical convergence to the analytic solution of some backwards reachable sets with sharp features," Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2004-01, January 2004.

[22] P. Cardaliaguet, "A differential game with two players and one target," *SIAM Journal of Control and Optimization*, vol. 34, no. 4, pp. 1441–1460, 1996.

[23] J. Aubin, *Viability Theory*, ser. Systems & Control: Foundations & Applications. Birkhäuser, 1991.

[24] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre, "Set-valued numerical analysis for optimal control and differential games," in *Stochastic and Differential Games: Theory and Numerical Methods*, ser. Annals of International Society of Dynamic Games, M. Bardi, T. E. S. Raghavan, and T. Parthasarathy, Eds. Birkhäuser, 1999, vol. 4, pp. 177–247.

[25] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid system verification," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 64–75, 2003.

[26] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, O. Maler and A. Pnueli, Eds. Springer Verlag, 2003, no. 2623, pp. 20–35.

[27] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 540–554, 1998.

[28] M. Greenstreet and I. Mitchell, "Reachability analysis using polygonal projections," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, F. Vaandrager and J. van Schuppen, Eds. Springer Verlag, 1999, no. 1569, pp. 103–116.

[29] S. Raczynski, "Continuous simulation, differential inclusions, uncertainty, and traveling in time," *Simulation*, vol. 80, no. 2, pp. 87–100, 2004.

[30] A. Bemporad, F. D. Torrisi, and M. Morari, "Optimization-based verification and stability characterization of piecewise affine and hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, B. Krogh and N. Lynch, Eds. Springer Verlag, 2000, no. 1790, pp. 45–59.

[31] A. B. Kurzhanski and P. Varaiya, "Reachability analysis for uncertain systems—the ellipsoidal technique," *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, vol. 9, no. 3, pp. 347–367, 2002.

[32] J. Preußig, O. Stursberg, and S. Kowalewski, "Reachability analysis of a class of switched continuous systems by integrating rectangular approximation and rectangular analysis," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, F. Vaandrager and J. van Schuppen, Eds. Springer Verlag, 1999, no. 1569, pp. 209–222.

[33] R. P. Kurshan and K. L. McMillan, "Analysis of digital circuits through symbolic reduction," *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 11, pp. 1356–1371, 1991.

[34] A. Tiwari and G. Khanna, "Series of abstractions for hybrid automata," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, C. J. Tomlin and M. R. Greenstreet, Eds. Springer Verlag, 2002, no. 2289, pp. 465–478.

[35] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.

[36] H. Hindi and S. Boyd, "Analysis of linear systems with saturation using convex optimization," in *Proceedings of the IEEE Conference on Decision and Control*, Tampa, FL, 1998, pp. 903–908.

[37] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*, ser.

Lecture Notes in Computer Science, R. Alur and G. J. Pappas, Eds. Springer Verlag, 2004, no. 2993, pp. 477–492.
[38] M. Bardi, “Some applications of viscosity solutions to optimal control and differential games,” in *Viscosity Solutions and Applications*, ser. Lecture Notes in Mathematics, I. Capuzzo-Dolcetta and P. L. Lions, Eds. Springer, 1995, no. 1660, pp. 44–97.
[39] J. Sprinkle, J. M. Eklund, and S. S. Sastry, “Deciding to land a UAV safely in real time,” in *Proceedings of the American Control Conference*, Portland, OR, 2005, to Appear.
[40] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.
[41] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. New York: Cambridge University Press, 1999.
[42] M. G. Crandall and P.-L. Lions, “Two approximations of solutions of Hamilton-Jacobi equations,” *Mathematics of Computation*, vol. 43, no. 167, pp. 1–19, 1984.
[43] S. Osher and C.-W. Shu, “High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations,” *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 907–922, 1991.
[44] C.-W. Shu and S. Osher, “Efficient implementation of essentially non-oscillatory shock-capturing schemes,” *Journal of Computational Physics*, vol. 77, pp. 439–471, 1988.
[45] A. W. Merz, “The game of two identical cars,” *Journal of Optimization Theory and Applications*, vol. 9, no. 5, pp. 324–343, 1972.
[46] I. Mitchell, “Games of two identical vehicles,” Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, Tech. Rep. SUDAAR 740, July 2001.
[47] [Online]. Available: http://cherokee.stanford.edu/~mitchell
[48] A. M. Bayen, P. Grieder, G. Meyer, and C. J. Tomlin, “Lagrangian delay predictive model for sector-based air traffic flow,” to appear in the *AIAA Journal of Guidance, Control and Dynamics*.
[49] I. Mitchell and C. J. Tomlin, “Overapproximating reachable sets by Hamilton-Jacobi projections,” *Journal of Scientific Computing*, vol. 19, no. 1–3, pp. 323–346, 2003.
[50] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations and Boundary Value Problems*, 4th ed. John Wiley & Sons, 1986.
[51] R. B. Ash, *Real Analysis and Probability*, ser. Probability and Mathematical Statistics. Academic Press, 1972, vol. 11.

APPENDIX
PROOF OF THE TIME-DEPENDENT FORMULATION

At the end of this section we prove Theorem 2. The proof depends on some results from the literature of viscosity solutions and differential games, and on the definition of a new system which has an augmented set of inputs for player II.

A. Augmenting the Dynamics

In the proof, we will use a modified set of system dynamics in which we augment player II's inputs with the scalar

$$\underline{b}(\cdot) \in \underline{\mathfrak{B}}(t) \triangleq \{\phi : [t, 0] \to [0, 1] \mid \phi(\cdot) \text{ is measurable}\}.$$

Define the augmented input for player II as

$$\tilde{b} = \begin{bmatrix} b & \underline{b} \end{bmatrix} \in \mathcal{B} \times [0, 1],$$

and similarly define $\tilde{\mathcal{B}}$, $\tilde{\mathfrak{B}}(t)$ and $\tilde{\Gamma}(t)$. The differential game referred to in the remainder of this section will be played with dynamics

$$\tilde{f}(x, a, \tilde{b}) \triangleq \underline{b} f(x, a, b), \tag{10}$$

and its trajectories will be denoted by $\xi_{\tilde{f}}(s; x, t, a(\cdot), \tilde{b}(\cdot))$.

From (10), we see that player II may choose to play the game with normal dynamics by taking $\underline{b} = 1$, may choose slowed dynamics with $\underline{b} \in]0, 1[$, or may choose to freeze the dynamics entirely by taking $\underline{b} = 0$. Because the latter case proves important, we will call this additional scalar $\underline{b}$ the *freezing input*. By choosing $\underline{b} = 0$, player II can stop the evolution of a trajectory of the augmented system and hence keep it from ever leaving the target set once it enters.

Clearly, there is a close connection between trajectories of the augmented system (10) and trajectories of the original system (1). We can formalize the connection through the pseudo-time variable $\sigma : [t, 0] \to [t, 0]$, which for any $\underline{b}(\cdot) \in \underline{\mathfrak{B}}(t)$ is given by

$$\sigma(s) \triangleq t + \int_t^s \underline{b}(\lambda)\, d\lambda \tag{11}$$

Since $\underline{b}(\cdot)$ is measurable, σ is absolutely continuous and is differentiable almost everywhere. There exists no true inverse of σ, because its monotonic increase is not strict; however, in the proof below we will use a function $\sigma^\dagger$, which is almost an inverse of σ in the sense that $\sigma(\sigma^\dagger(s)) = s$. The proof of Lemma 6 defines $\sigma^\dagger$ as the limit of a convergent sequence of functions.

Lemma 4 (Equivalence of Trajectories): For any $a(\cdot) \in \mathfrak{A}(t)$ and $\tilde{b}(\cdot) = \begin{bmatrix} b(\cdot) & \underline{b}(\cdot) \end{bmatrix} \in \tilde{\mathfrak{B}}(t)$, define σ as in (11) and $\sigma^\dagger$ as in Lemma 6. Then for every trajectory of the original system (1), there is a trajectory of the augmented system (10) related through the pseudo-time variable σ

$$\xi_f(\sigma(s); x, t, a(\sigma^\dagger(\cdot)), b(\sigma^\dagger(\cdot))) = \xi_{\tilde{f}}(s; x, t, a(\cdot), \tilde{b}(\cdot))$$

for any $s \in [t, 0]$.

Proof: This proof is an adaptation of a classical ODE uniqueness proof [50, section 2.12]. Define the shorthand

$$\begin{aligned} \xi_f(s) &\triangleq \xi_f(s; x, t, a(\sigma^\dagger(\cdot)), b(\sigma^\dagger(\cdot))), \\ \xi_{\tilde{f}}(s) &\triangleq \xi_{\tilde{f}}(s; x, t, a(\cdot), \tilde{b}(\cdot)). \end{aligned} \tag{12}$$

Then we can write

$$\begin{aligned} \xi_{\tilde{f}}(s) &= \xi_{\tilde{f}}(t) + \int_t^s \frac{d\xi_{\tilde{f}}(\lambda)}{d\lambda}\, d\lambda, \\ &= x + \int_t^s f(\xi_{\tilde{f}}(\lambda), a(\lambda), b(\lambda))\underline{b}(\lambda)\, d\lambda, \end{aligned}$$

and

$$\begin{aligned} &\xi_f(\sigma(s)) \\ &\quad= \xi_f(t) + \int_t^{\sigma(s)} \frac{d\xi_f(\rho)}{d\rho}\, d\rho, \\ &\quad= x + \int_t^{\sigma(s)} f(\xi_f(\rho), a(\sigma^\dagger(\rho)), b(\sigma^\dagger(\rho)))\, d\rho, \\ &\quad= x + \int_t^s f(\xi_f(\sigma(\lambda)), a(\lambda), b(\lambda))\underline{b}(\lambda)\, d\lambda, \end{aligned} \tag{13}$$

where the change of variables $\rho = \sigma(\lambda)$ after the second step is justified in Lemma 6. From these two equations and the fact that $\underline{b}(\lambda) \in [0, 1]$,

$$\begin{aligned} &\|\xi_f(\sigma(s)) - \xi_{\tilde{f}}(s)\| \\ &\quad\le \int_t^s \left\| \begin{pmatrix} f(\xi_f(\sigma(\lambda)), a(\lambda), b(\lambda)) \\ \quad - f(\xi_{\tilde{f}}(\lambda), a(\lambda), b(\lambda)) \end{pmatrix} \underline{b}(\lambda) \right\| d\lambda, \\ &\quad\le \int_t^s \left\| \begin{matrix} f(\xi_f(\sigma(\lambda)), a(\lambda), b(\lambda)) \\ \quad - f(\xi_{\tilde{f}}(\lambda), a(\lambda), b(\lambda)) \end{matrix} \right\| d\lambda, \\ &\quad\le K \int_t^s \|\xi_f(\sigma(\lambda)) - \xi_{\tilde{f}}(\lambda)\|\, d\lambda, \end{aligned} \tag{14}$$

where K is the Lipschitz constant for the flow field f. Letting

$$\psi(s) = \int_t^s \|\xi_f(\sigma(\lambda)) - \xi_{\tilde{f}}(\lambda)\| \, d\lambda,$$

we see that $\psi(t) = 0$, $\psi(s) \geq 0$, and $\dot{\psi}(s) = \|\xi_f(\sigma(s)) - \xi_{\tilde{f}}(s)\|$. Rewriting (14) in terms of ψ we get the differential inequality

$$\dot{\psi}(s) - K\psi(s) \leq 0,$$

whose only solution is $\psi(s) \equiv 0$ [50]. Therefore $\xi_f(\sigma(s)) = \xi_{\tilde{f}}(s)$. ■

While more general than is required for the proof of Theorem 2, the formal connection established by this lemma between the two systems makes it useful in its own right. For the purposes of the theorem, we need only a straightforward corollary of Lemma 4.

Corollary 5: Using the definitions (12), the augmented system's trajectory $\xi_{\tilde{f}}(\cdot)$ visits only a subset of the states visited by the original system's trajectory $\xi_f(\cdot)$; specifically, those visited in the time interval $[t, \sigma(s)]$.

The following short lemma justifies the change of variables in (13), and its proof defines $\sigma^\dagger$.

Lemma 6:

$$\begin{aligned} &\int_t^{\sigma(s)} f(\xi_f(\rho), a(\sigma^\dagger(\rho)), b(\sigma^\dagger(\rho))) \, d\rho \\ &\quad = \int_t^s f(\xi_f(\sigma(\lambda)), a(\lambda), b(\lambda)) \underline{b}(\lambda) \, d\lambda, \end{aligned} \tag{15}$$

Proof: For $k = \{1, 2, 3, \ldots\}$, let $\underline{b}_k$ be a sequence of smooth functions such that $\underline{b}_k(s) \in \left[\frac{1}{k}, 1 + \frac{1}{k}\right]$ for all s and $\underline{b}_k \to \underline{b}$ pointwise as $k \to \infty$ (such a sequence can be constructed). Define the smooth functions

$$\sigma_k(s) \triangleq t + \int_t^s \underline{b}_k(\lambda) \, d\lambda.$$

Then $\sigma_k \to \sigma$ uniformly as $k \to \infty$. Because of the bounds on s and $\underline{b}_k$, σ_k is bounded and strictly monotonically increasing, and a smooth σ_k^{-1} exists for each k. Then by Helly's Theorem [51, Theorem 8.2.1] there exists a monotonically increasing right-continuous $\sigma^\dagger$ and a subsequence $\sigma_{k_n}^{-1}$ such that $\sigma^\dagger = \lim_{k_n \to \infty} \sigma_{k_n}^{-1}$. With smooth σ_{k_n} and $\sigma_{k_n}^{-1}$ and Lipschitz continuous f, the change of variables $\rho = \sigma_{k_n}(\lambda)$ is rigorously defensible for each k_n:

$$\begin{aligned} &\int_t^{\sigma_{k_n}(s)} f(\xi_f(\rho), a(\sigma_{k_n}^{-1}(\rho)), b(\sigma_{k_n}^{-1}(\rho))) \, d\rho \\ &\quad = \int_t^s f(\xi_f(\sigma_{k_n}(\lambda)), a(\lambda), b(\lambda)) \underline{b}_{k_n}(\lambda) \, d\lambda. \end{aligned}$$

By the Dominated Convergence Theorem [51, Theorem 1.6.9], we can take the limit as $k_n \to \infty$ inside the integral to arrive at (15). ■

B. The Differential Game and its Solution

We will work with a finite horizon differential game played over time horizon $[-T, 0]$ whose dynamics are governed by the flow field (10). A trajectory in this game has a terminal cost

$$C(x, t, a(\cdot), \tilde{b}(\cdot)) = g(\xi_{\tilde{f}}(0; x, t, a(\cdot), \tilde{b}(\cdot))).$$

and no running cost. The goal of player I will be to maximize this cost, while player II will try to minimize it. Consequently, the value of our differential game will be

$$\begin{aligned} v(x, t) &= \inf_{\tilde{\gamma} \in \tilde{\Gamma}(t)} \sup_{a(\cdot) \in \mathfrak{A}(t)} C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) \\ &= \inf_{\tilde{\gamma} \in \tilde{\Gamma}(t)} \sup_{a(\cdot) \in \mathfrak{A}(t)} g(\xi_{\tilde{f}}(0; x, t, a(\cdot), \tilde{\gamma}[a](\cdot))) \end{aligned} \tag{16}$$

Lemma 7: The value function $v(x, t)$ of our game is the viscosity solution of the Hamilton-Jacobi-Isaacs terminal value PDE

$$\begin{aligned} D_t v(x, t) + \tilde{H}(x, D_x v(x, t)) &= 0, \\ v(x, 0) &= g(x), \end{aligned} \tag{17}$$

where

$$\tilde{H}(x, p) = \max_{a \in \mathcal{A}} \min_{\tilde{b} \in \tilde{\mathcal{B}}} p^T \tilde{f}(x, a, \tilde{b}). \tag{18}$$

Proof: This lemma is just a special case of Theorem 4.1 in [10]. ■

C. The Proof of Theorem 2

We need one more intermediate result before proving Theorem 2.

Lemma 8: For $t \in [-T, 0]$, the value function $v(x, t)$ given by (16) describes the reachable set $\mathcal{G}(\tau)$ according to (6).

Proof: We show that

$$x \in \mathcal{G}(\tau) \implies v(x, t) \leq 0, \tag{19}$$

$$v(x, t) \leq 0 \implies x \in \mathcal{G}(\tau). \tag{20}$$

Case 1: We will assume that $x \in \mathcal{G}(\tau)$ and $v(x, t) > 0$ and derive a contradiction. Consider first the implications of (16).

$$\begin{aligned} &v(x, t) = \inf_{\tilde{\gamma} \in \tilde{\Gamma}(t)} \sup_{a(\cdot) \in \mathfrak{A}(t)} C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) > 0, \\ &\quad \implies \exists \epsilon > 0, \forall \tilde{\gamma} \in \tilde{\Gamma}(t), \\ &\quad\quad \sup_{a(\cdot) \in \mathfrak{A}(t)} C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) > 2\epsilon > 0, \\ &\quad \implies \exists \epsilon > 0, \forall \tilde{\gamma} \in \tilde{\Gamma}(t), \exists \hat{a}(\cdot) \in \mathfrak{A}(t), \\ &\quad\quad C(x, t, \hat{a}(\cdot), \tilde{\gamma}[\hat{a}](\cdot)) > \epsilon > 0. \end{aligned} \tag{21}$$

Now consider the implications of $x \in \mathcal{G}(\tau)$. By (3) there is a $\gamma \in \Gamma(t)$ such that for the $\hat{a}(\cdot)$ from (21) and $b(\cdot) = \gamma[\hat{a}](\cdot)$ there exists $s \in [t, 0]$ such that $\xi_f(s; x, t, \hat{a}(\cdot), b(\cdot)) \in \mathcal{G}_0$. By (2), $g(\xi_f(s; x, t, \hat{a}(\cdot), b(\cdot))) \leq 0$. Choose freezing input signal

$$\underline{b}(r) = \begin{cases} 1, & \text{for } r \in [t, s[; \\ 0, & \text{for } r \in [s, 0[. \end{cases}$$

Combine this $\underline{b}(\cdot)$ with the $b(\cdot)$ chosen above to get $\tilde{b}(\cdot)$, an input which will generate a trajectory

$$\xi_{\tilde{f}}(r; x, t, \hat{a}(\cdot), \tilde{b}(\cdot)) = \begin{cases} \xi_f(r; x, t, \hat{a}(\cdot), b(\cdot)), & \text{for } r \in [t, s]; \\ \xi_f(s; x, t, \hat{a}(\cdot), b(\cdot)), & \text{for } r \in [s, 0]. \end{cases}$$

In particular,

$$\begin{aligned} &\xi_{\tilde{f}}(0; x, t, \hat{a}(\cdot), \tilde{b}(\cdot)) = \xi_f(s; x, t, \hat{a}(\cdot), b(\cdot)) \\ &\quad \implies g(\xi_{\tilde{f}}(0; x, t, \hat{a}(\cdot), \tilde{b}(\cdot))) = C(x, t, \hat{a}(\cdot), \tilde{b}(\cdot)) \leq 0. \end{aligned}$$

A nonanticipative strategy for $\underline{b}(\cdot)$ can be designed (with additional constraints on s) and $b(\cdot) = \gamma[\hat{a}](\cdot)$ is already nonanticipative, so $\tilde{b}(\cdot)$ is nonanticipative and we have a contradiction of (21). Therefore we have proved (19).

Case 2: We assume that $v(x,t) \leq 0$ and $x \notin \mathcal{G}(\tau)$ and derive a contradiction. First, consider the implications of $x \notin \mathcal{G}(\tau)$. Negating (3)

$$\begin{aligned} &\forall \gamma \in \Gamma(t), \exists \hat{a}(\cdot) \in \mathfrak{A}(t), \forall s \in [t,0], \\ &\qquad \xi_f(s; x, t, a(\cdot), \gamma[a](\cdot)) \notin \mathcal{G}_0. \end{aligned} \tag{22}$$

Now consider any strategy $\tilde{\gamma} \in \tilde{\Gamma}(t)$ of player II in the augmented system and extract from it the response $\gamma \in \Gamma(t)$ of player II in the original system by omitting the final component of its output (elements of $\tilde{\Gamma}(t)$ and $\Gamma(t)$ accept the same input function drawn from $\mathfrak{A}(t)$, but the former's output function includes the scalar freezing signal). Choose $\hat{a}(\cdot)$ from (22) which corresponds to that γ. By Corollary 5, the set of states visited by the augmented trajectory is a subset of the states visited by the original trajectory, so in combination with (22)

$$\begin{aligned} &\xi_f(s; x, t, \hat{a}(\cdot), \gamma[\hat{a}](\cdot)) \notin \mathcal{G}_0 \ \forall s \in [t,0] \\ &\quad \implies \xi_{\tilde{f}}(s; x, t, \hat{a}(\cdot), \tilde{\gamma}[\hat{a}](\cdot)) \notin \mathcal{G}_0 \ \forall s \in [t,0]. \end{aligned} \tag{23}$$

The composition of g with $\xi_{\tilde{f}}$ is a continuous function from the compact interval $[t,0]$ to $\mathbb{R}$, and so it achieves its extrema. By (2) and (23) we can conclude that there exists $\delta > 0$ such that

$$\begin{aligned} &g(\xi_{\tilde{f}}(s; x, t, \hat{a}(\cdot), \tilde{\gamma}[\hat{a}](\cdot))) \geq \delta \ \forall s \in [t,0] \\ &\quad \implies C(x, t, \hat{a}(\cdot), \tilde{\gamma}[\hat{a}](\cdot)) \\ &\qquad\qquad = g(\xi_{\tilde{f}}(0; x, t, \hat{a}(\cdot), \tilde{\gamma}[\hat{a}](\cdot))) \geq \delta. \end{aligned} \tag{24}$$

Now return to (16)

$$\begin{aligned} v(x,t) &= \inf_{\tilde{\gamma} \in \tilde{\Gamma}(t)} \sup_{a(\cdot) \in \mathfrak{A}(t)} C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) \leq 0, \\ &\implies \forall \epsilon > 0, \exists \tilde{\gamma} \in \tilde{\Gamma}(t), \\ &\qquad \sup_{a(\cdot) \in \mathfrak{A}(t)} C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) \leq \epsilon, \\ &\implies \forall \epsilon > 0, \exists \tilde{\gamma} \in \tilde{\Gamma}(t), \forall a(\cdot) \in \mathfrak{A}(t), \\ &\qquad C(x, t, a(\cdot), \tilde{\gamma}[a](\cdot)) \leq \epsilon. \end{aligned}$$

Choose $\epsilon = \delta/2$ to derive a contradiction of (24) and hence prove (20). ∎

The proof of Theorem 2 is now straightforward.

Proof: From Lemma 7 we know that the value function v for the differential game (16) is the viscosity solution to the HJI PDE (17). From Lemma 8 we know that the reachable set is characterized by (6).

For the final step of the proof, start with $\tilde{H}$ from (18) and H from (5). Then we see that

$$\begin{aligned} \tilde{H}(x,p) &= \max_{a \in \mathcal{A}} \min_{\tilde{b} \in \tilde{\mathcal{B}}} p^T \tilde{f}(x, a, \tilde{b}), \\ &= \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} \min_{\underline{b} \in [0,1]} p^T (\underline{b} f(x, a, b)), \\ &= \min_{\underline{b} \in [0,1]} \underline{b} \left(\max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} p^T f(x, a, b) \right), \\ &= \min[0, H(x,p)]. \end{aligned}$$

Consequently, the two HJI PDEs (4) and (17) are equivalent, and so v is also the solution of (4). ∎

Ian M. Mitchell (S'97–M'03) received a B. A. Sc. in Engineering Physics and an M. Sc. in Computer Science from the University of British Columbia, Canada in 1994 and 1997 respectively, and a Ph. D. in Scientific Computing and Computational Mathematics from Stanford University in 2002. After spending a year as a postdoctoral researcher in the Department of Electrical Engineering and Computer Science at the University of California, Berkeley and the Department of Computer Science at Stanford, Dr. Mitchell joined the faculty in the Department of Computer Science at the University of British Columbia as an Assistant Professor in August, 2003. He is the recipient of a 1999 SIAM/AAAS Mass Media Fellowship and a 1997-98 Stanford School of Engineering Graduate Fellowship. His research interests include scientific computing, hybrid systems, verification and robotics.

Alexandre M. Bayen (S'02–M'04) received the B.S. degree in applied mathematics from the Ecole Polytechnique, Palaiseau, France, in 1998, the M.S. degree in aeronautics and astronautics from Stanford University, Stanford, CA, in 1999, and a Ph.D. in aeronautics and astronautics from Stanford University, Stanford, CA, in 2004. He was a visiting researcher at NASA Ames from 2000 to 2003. From 2004 to 2005, he worked for the Department of Defense in France, where he held the rank of Major. During that time, he was the Research Director of the Laboratoire de Navigation Autonome, within the Laboratoire de Recherche Balistiques et Aerodynamiques in Vernon, France. Since March 2005, he has been an Assistant Professor in the Department of Civil and Environmental Engineering at the University ot California at Berkeley. His research interests include combinatorial optimization, hybrid systems, air traffic automation, and control of distributed systems. Mr. Bayen is the recipient of the Graduate Fellowship of the Delegation Generale pour l'Armement (1998-2002) from France, and of the Ballhaus Prize for best doctoral thesis from the Department of Aeronautics and Astronautics at Stanford University (2004).

Claire J. Tomlin (S'93–M'99) is an Associate Professor in the Department of Aeronautics and Astronautics at Stanford University, and a Courtesy Associate Professor in the Department of Electrical Engineering. She received a Ph.D. in Electrical Engineering from the University of California at Berkeley in 1998, and joined Stanford in September 1998. She received the M.Sc. from Imperial College, London, in 1993, and the B.A.Sc. from the University of Waterloo, Canada, in 1992, both in Electrical Engineering. She has held visiting research positions at NASA Ames, Honeywell Labs, and the University of British Columbia. Claire Tomlin is a recipient of the Eckman Award of the American Automatic Control Council (2003), MIT Technology Review's Top 100 Young Innovators Award (2003), the AIAA Outstanding Teacher Award (2001), an NSF Career Award (1999), a Terman Fellowship (1998), and the Bernard Friedman Memorial Prize in Applied Mathematics (1998). Her research interests are in control systems, specifically hybrid control theory, and she works on air traffic control automation, flight management system analysis and design, and modeling and analysis of biological cell networks.