A Minimum Discounted Reward Hamilton–Jacobi Formulation for Computing Reachable Sets

Anayo K. Akametalu, Shromona Ghosh, Jaime F. Fisac, Vicenc Rubies-Royo, and Claire J. Tomlin, *Fellow, IEEE*

***Abstract*—We propose a novel formulation for approximating reachable sets through a minimum discounted reward optimal control problem. The formulation yields a continuous solution that can be obtained by solving a Hamilton–Jacobi equation. Furthermore, the numerical approximation to this solution is the unique fixed-point to a contraction mapping. This allows for more efficient solution methods that are not applicable under traditional formulations for solving reachable sets. Lastly, this formulation provides a link between reinforcement learning and learning reachable sets for systems with unknown dynamics, allowing algorithms from the former to be applied to the latter. We use two benchmark examples, double integrator, and pursuit-evasion games, to show the correctness of the formulation as well as its strengths in comparison to previous work.**

***Index Terms*—Approximate reachability, machine learning, reachability analysis, safety analysis.**

I. Introduction

Computing reachable sets has been a popular approach for verifying and validating the behavior of dynamical systems. In the reachability problem, one specifies a target set in the state space, and then aims to find the set of initial states admitting trajectories that can eventually enter the target within a specified time horizon. The target may represent a region we desire to drive the system towards, or to keep the system away from. Due to its generality, Hamilton–Jacobi (HJ) reachability analysis, which casts the problem in the optimal control framework, has seen broad usage in many safety-critical applications including emergency landing of unmanned aerial vehicles [1], vehicle platooning [2], safe learning [3], [4], collision-avoidance [5], [6], and many others [7], [8].

The standard HJ formulation for reachability solves a *minimum reward* (MR) optimal control problem, by computing the viscosity solution of a particular time-dependent HJ partial differential equation [6]. The solution can be approximated on a grid via value iteration (VI), which recursively applies a *backup operator* until convergence. One downside of MR optimal control problems is that the backup operator in this setting is not a contraction mapping, thus a specific initialization of VI is required for convergence to the correct solution. A contraction mapping allows for arbitrary initialization, and with a good initialization convergence can be accelerated. This is particularly useful when computing reachable sets for multiple systems with similar dynamics allowing for the solution of one problem to be used as an initialization for the other.

On the other hand, infinite horizon *total discounted reward* (TDR) problems yield backup operators that are contraction mappings [9]. This allows for more efficient solution methods, like policy iteration [10], [11] and multigrid approaches [12], [13].

Drawing inspiration from TDR problems, we propose a *minimum discounted reward* (MDR) formulation for computing tight over- and underapproximations of infinite horizon reachable sets. This new formulation yields a contraction mapping, making it possible to extend policy iteration and multigrid approaches to this setting. In addition, it also provides a way forward for learning reachable sets when dynamics are unknown or difficult to model, a topic that has garnered some attention [14], [15]. This draws greatly from reinforcement learning (RL), which attempts to solve the infinite horizon TDR problem when the system model is unknown.

The rest of this article is organized as follows. In Section II we briefly go over the MR formulation for HJ reachability analysis, and TDR problems. In Section III, we describe the MDR formulation. In Section IV, we talk about the implications of this new formulation for computing reachable sets. Section V contains examples demonstrating the ideas developed throughout this article. Finally, Section VI concludes the article.

Here, both the one-player and adversarial two-player settings are considered, and we opt to use the terminology from optimal control for consistency. For example *optimal control* will be used to refer to both settings, and we use terms like *reward* instead of *payoff* throughout.[1]

II. Background

We consider a two-player optimal control problem where the first player (control) wants to keep the system away from the target set for a given time horizon, and the second player (disturbance) has the opposite objective.[2] Our goal is to characterize the *infinite-horizon backward reachable set*, which is the set of initial states for which the disturbance wins the game. Going forward all mentions of reachable set refer to the infinite horizon backwards reachable set.

Manuscript received 24 May 2021; revised 20 July 2022 and 18 September 2022; accepted 7 May 2023. Date of publication 24 October 2023; date of current version 30 January 2024. This work was supported by the NSF through the CPS FORCES and VehiCal projects, in part by the UC-Philippine-California Advanced Research Institute, in part by the ONR MURI Embedded Humans, in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-2-0196, and in part by the Toyota under the iCyPhy center. The work of Anayo. K. Akametalu was supported by the National GEM Consortium Fellowship. Recommended by Associate Editor T. Faulwasser. *(Corresponding author: Vicenc Rubies-Royo.)*

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: kakametalu@eecs.berkeley.edu; shromona.ghosh@eecs.berkeley.edu; jfisac@eecs.berkeley.edu; vrubies@berkeley.edu; tomlin@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TAC.2023.3327159

[1]Typically the terms *optimal control* and *reward* are reserved for the one player setting and *zero sum differential game* and *payoff* are the analogous terms in the adversarial two-player setting.

[2]We can also consider the case where the objectives are switched, and the control wants to drive the system towards the target, by making minor modifications to the material presented here.

A. System Model

The analysis in this article considers a fully observable system, whose underlying dynamics may be nondeterministic, but bounded. We can formalize this as a dynamical system with state $x \in \mathbb{R}^n$, and two inputs, $u \in \mathcal{U} \subset \mathbb{R}^{n_u}, d \in \mathcal{D} \subset \mathbb{R}^{n_d}$ (with $\mathcal{U}$ and $\mathcal{D}$ compact) which we will refer to as the *controller* and the *disturbance*

$$\dot{x} = f(x, u, d). \tag{1}$$

The flow field $f : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \to \mathbb{R}^n$ is assumed to be Lipschitz continuous and bounded. In the single-player case we drop the disturbance input, and just have $f(x, u) : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^n$.

Letting $\mathbb{U}$ and $\mathbb{D}$ denote the collections of measurable[3] functions $\boldsymbol{u} : [0, \infty) \to \mathcal{U}$ and $\boldsymbol{d} : [0, \infty) \to \mathcal{D}$, respectively, and allowing the controller and disturbance to choose any such signals, the evolution of the system from any initial state x is determined (see for example [16], Ch. 2, Th. 1.1 and 2.1) by the unique continuous trajectory $\xi : [0, \infty) \to \mathbb{R}^n$ solving

$$\begin{aligned} \dot{\xi}(s) &= f(\xi(s), \boldsymbol{u}(s), \boldsymbol{d}(s)), \text{ a.e. } s \geqslant 0 \\ \xi(0) &= x. \end{aligned} \tag{2}$$

Note that this is a solution in Carathéodory's *extended sense*, that is, it satisfies the differential equation *almost everywhere* (i.e., except on a subset of Lebesgue measure zero).

Throughout our analysis, we will use the notation $\xi_x^{\boldsymbol{u},\boldsymbol{d}}(\cdot)$ to denote the state trajectory $t \mapsto x$ corresponding to the initial condition $x \in \mathbb{R}^n$, the control signal $\boldsymbol{u} \in \mathbb{U}$ and the disturbance signal $\boldsymbol{d} \in \mathbb{D}$.

B. HJ Reachability: Minimum Distance to Target

The target set $\mathcal{T}$ can be implicitly characterized as the subzero level set of a Lipschitz *surface function* $l : \mathbb{R}^n \to \mathbb{R}$

$$x \in \mathcal{T} \iff l(x) < 0. \tag{3}$$

This function always exists, since we can simply choose the *signed distance function* to $\mathcal{T}$, $s_{\mathcal{T}}(x)$, which is Lipschitz continuous by construction.[4] We use a clipped signed distance since the problem is solved over a fixed domain in practice, $l(x) = \min(\max(s_{\mathcal{T}}(x), -L), L)$ with $L > 0$, where L is usually taken to be the largest value (in magnitude) on the domain.

To express whether a given trajectory *ever* enters the target set, let the functional $\mathcal{V} : \mathbb{R}^n \times \mathbb{U} \times \mathbb{D} \to \mathbb{R}$ assign to each initial state x and input signals $\boldsymbol{u}(\cdot)$, $\boldsymbol{d}(\cdot)$ the lowest value of $l(\cdot)$ achieved by trajectory $\xi_x^{\boldsymbol{u},\boldsymbol{d}}(\cdot)$ over all times $t \geqslant 0$

$$\mathcal{V}\left(x, \boldsymbol{u}(\cdot), \boldsymbol{d}(\cdot)\right) := \inf_{t \geqslant 0} l\left(\xi_x^{\boldsymbol{u},\boldsymbol{d}}(t)\right). \tag{4}$$

This outcome $\mathcal{V}$, also referred to here as the *minimum of rewards*,[5] will be nonpositive if there exists any $t \in [0, \infty)$ at which the trajectory enters the target set, and will be strictly positive if the system avoids the target for all $t \geqslant 0$.

The reachable set can then be obtained by solving a game between the controller and disturbance, where the disturbance can use *nonanticipative strategies* to respond to the controller's signal. The disturbance's set of nonanticipative strategies is $\mathcal{B} = \{\boldsymbol{\beta} : \mathbb{U} \to \mathbb{D} \mid \forall t \geqslant 0\ \forall \boldsymbol{u}(\cdot), \hat{\boldsymbol{u}}(\cdot) \in \mathbb{U}, (\boldsymbol{u}(\tau) = \hat{\boldsymbol{u}}(\tau) \text{ a.e.} \tau \geqslant 0) \Rightarrow (\boldsymbol{\beta}[\boldsymbol{u}](\tau) = \boldsymbol{\beta}[\hat{\boldsymbol{u}}](\tau) \text{ a.e.} \tau \geqslant 0)\}$. Intuitively, the disturbance is given an instantaneous informational advantage in the game, but its actions at any given time can only depend on what the controller has done up to that time. With this in place we can define the *value function* $V(x)$ and ultimately the reachable set $\mathcal{R}(\mathcal{T})$

$$V(x) = \inf_{\beta[\boldsymbol{u}](\cdot) \in \mathcal{B}} \sup_{\boldsymbol{u} \in \mathbb{U}} \mathcal{V}\left(x, \boldsymbol{u}(\cdot), \beta[\boldsymbol{u}](\cdot)\right) \tag{5}$$

$$\mathcal{R}(\mathcal{T}) = \{x \mid V(x) \leqslant 0\}. \tag{6}$$

The optimization in (5) is referred to here as the MR optimal control problem. It has been shown that the value function for problems with outcome given by (4) can be characterized as the unique viscosity solution to a particular variational inequality [17], [18], [19]. When the problem is solved over a finite time interval $[0, T]$, the *finite-horizon value function* $V(x, t)$ can be computed by solving the HJ equation

$$\begin{aligned} 0 &= \min\left\{l(x) - V(x,t), \frac{\partial V}{\partial t}(x,t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \frac{\partial V}{\partial x}(x,t) f(x,u,d)\right\} \\ V(x,T) &= l(x). \end{aligned} \tag{7}$$

As $T \to \infty$, $V(x, t)$ becomes independent of t. We accordingly drop the dependence on t and recover $V(x)$ as defined in (5).

C. Computing the Value Function

Several approximation schemes have been proposed for solving (7) and similar HJ equations on a fixed grid G [6], [20], [21], [22], [23]. Here, we will use a semi-Lagrangian approximation based on a discrete time dynamic programming (DP) principle

$$V_{\Delta t}^{k+1}(x) = \min\{l(x), \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} V_{\Delta t}^k(x + \Delta t f(x, u, d))\} \tag{8a}$$

$$V_{\Delta t}^0(x) = l(x) \tag{8b}$$

$$V_{\Delta t} = \lim_{k \to \infty} V_{\Delta t}^k \tag{8c}$$

where $V_{\Delta t}(x)$ converges to $V(x)$ as the discrete time step $\Delta t \to 0$. With the semi-Lagrangian approximation, the value function is solved on a discrete grid. Representing the approximation in vectorized form, $\vec{V} \in \mathbb{R}^{n_G}$, the semi-Lagrangian approach yields

$$\vec{V}_i^0 = l(x_i) \tag{9a}$$

$$\vec{V}_i^{k+1} = \min\{l(x_i), \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} I[\vec{V}^k](x_i + \Delta t f(x_i, u, d))\} \tag{9b}$$

$$\vec{V} = \lim_{k \to \infty} \vec{V}^k \tag{9c}$$

for $i = 1, \ldots, n_G$, where $\{x_i\}_{i=1}^{n_G}$ are the grid nodes, n_G is the number of grid nodes, $\vec{V}_i^k$ is the approximate value for $V(x_i, k\Delta t)$, and $I[\vec{A}] : \mathbb{R}^n \to \mathbb{R}$ represents an interpolation operator defining, for every point x, the polynomial reconstruction based on the values $\vec{A}$. Unless stated otherwise G is taken to be a regular equidistant array of points with mesh spacing Δx_j along the jth axis, $j = 1, \ldots, n$.[6] We use a multilinear interpolator for the interpolation scheme, thus the interpolation function $I[\vec{A}](\cdot)$ is given by a convex combination over the elements of $\vec{A}$

$$I[\vec{A}](x) = \phi(x) \cdot \vec{A} \tag{10}$$

[3]A function $f : X \to Y$ between two measurable spaces (X, Σ_X) and (Y, Σ_Y) is said to be measurable if the preimage of a measurable set in Y is a measurable set in X, that is: $\forall V \in \Sigma_Y, f^{-1}(V) \in \Sigma_X$, with Σ_X, Σ_Y σ-algebras on X, Y.

[4]For any nonempty set $\mathcal{M} \subset \mathbb{R}^m$, the signed distance function $s_{\mathcal{M}} : \mathbb{R}^m \to \mathbb{R}$ is defined as $\inf_{y \in \mathcal{M}} |z - y|$ for points outside of $\mathcal{M}$ and $-\inf_{y \in \mathbb{R}^m \setminus \mathcal{M}} |z - y|$ for points inside $\mathcal{M}$, where $|\cdot|$ denotes a norm on $\mathbb{R}^m$.

[5]In this context reward refers to $l(x)$, but in general it can be any real-valued function.

[6]In the most general case the control and disturbance sets are also discretized, $\mathcal{U} = \{u_i\}_{i=1}^{n_{\mathcal{U}}}$ and $\mathcal{D} = \{d_i\}_{i=1}^{n_{\mathcal{D}}}$, and the minimax game is approximated.

where $\phi(\cdot):\mathbb{R}^n \to \mathbb{R}^{n_G}$ and $\forall x \in \mathcal{R}^n$ the elements of $\phi(x)$ are nonnegative and sum to 1.

For conciseness we introduce *control policies* $\pi(\cdot):\mathbb{R}^n \to \mathcal{U}$ and *disturbance policies* $\rho(\cdot):\mathbb{R}^n \to \mathcal{D}$, which map from state to control and state to disturbance, respectively. Note that the individual minimax games being played at each grid point can be collectively thought of as a game over policies. We also introduce the *backup operator* $B[\cdot]:\mathbb{R}^{n_G} \to \mathbb{R}^{n_G}$, which maps vectorized value functions onto themselves. Define the backup operator as

$$B[\vec{A}] := \min\{\vec{l}, \max_{\pi}\min_{\rho}\Phi_{\pi,\rho}\vec{A}\} \tag{11}$$

where $\vec{l} \in \mathbb{R}^{n_G}$ with $\vec{l}_i = l(x_i)$ for $i = 1,\ldots,n_G$, $\Phi_{\pi,\rho} \in \mathbb{R}^{n_G \times n_G}$ is a policy-dependent stochastic matrix,[7] and row i of $\Phi_{\pi,\rho}$ is $\phi(x_i + \Delta t f(x,\pi(x_i),\rho(x_i)))$. For the one player setting this matrix becomes Φ_π. We can now express (9) more concisely as

$$\vec{V}^0 = \vec{l} \tag{12a}$$

$$\vec{V}_i^{k+1} = B[\vec{V}^k] \tag{12b}$$

$$\vec{V} = \lim_{k\to\infty} \vec{V}^k. \tag{12c}$$

This recursive procedure (12) is referred to as *VI*, and here $\vec{V}$ is the *vectorized value function*, which is used to approximate the value function $V(x)$ as $I[\vec{V}](x)$.

The VI algorithm in (12) can be used to solve other optimal control problems, albeit with a different backup operator and initialization $\vec{V}^0$. As other optimal control problems are presented we redefine the backup operator and specify the initialization required for the VI procedure.

D. TDR and Contraction Mappings

Consider the infinite horizon TDR optimal control problem

$$V(x) := \sup_{\boldsymbol{u}\in\mathbb{U}} \int_0^\infty r\left(\xi_x^{\boldsymbol{u}}(t)\right)\exp(-\lambda t)dt, \quad \lambda > 0 \tag{13}$$

where $r(\cdot):\mathbb{R}^n \to \mathbb{R}$ is a state-dependent reward function and λ is a *discount rate*.

It is known that this value function is the solution to the time-independent HJ equation [24]

$$\lambda V(x) = \max_{u\in\mathcal{U}} \frac{\partial V}{\partial x}(x)f(x,u) + g(x). \tag{14}$$

In this instance, $V(x)$ can be approximated through VI with the following backup operator:

$$B[\vec{A}] := \vec{r} + \max_{\pi}\gamma\Phi_\pi\vec{A} \tag{15}$$

where $\vec{r}_i = \Delta t \cdot r(x_i)$, and $\gamma = \exp(-\lambda\Delta t)$ is the *discount factor*.

It is a standard result that (15) is a *contraction mapping*[8] with Lipschitz constant γ, and thus by the *Banach fixed point theorem* VI converges to a unique solution, independent of the initialization.

The independence to initialization allows for the value function to be approximated using other methods, such as policy iteration or multigrid techniques, which can exhibit faster convergence (more on this in Section IV). The operator given by (11) is not a contraction mapping,[9] and thus these methods are not available for MR optimal control problems.

In the next section, we show that applying discounting to MR problems yields a backup operator that is a contraction mapping, and in Section IV we look at the benefits that follow.

III. Minimum of Discounted Rewards

We now present the MDR optimal control problem. Discounting the rewards, allows the reachable set to be approximated as a fixed point of a contraction mapping. Contraction mappings have more flexibility in how they can be solved, and this can allow for faster convergence. Moving forward any mention of value function $V(x)$ refers to the MR setting, and is defined by (4), (5), and (7).

A. What to Discount?

Consider the following outcome for the MDR problem:

$$\mathcal{Z}\left(x,\boldsymbol{u}(\cdot),\boldsymbol{d}(\cdot)\right) := L + \inf_{t\geqslant 0}\left(l\left(\xi_x^{\boldsymbol{u},\boldsymbol{d}}(t)\right) - L\right)\exp(-\lambda t). \tag{16}$$

The quantity in the infimum is always nonpositive since L upperbounds $l(x)$ by construction. Note that if $\lambda = 0$, i.e., no discounting, then we have the MR outcome (4).

We define the MDR value function as

$$Z(x) := \inf_{\beta[\boldsymbol{u}](\cdot)\in\mathcal{B}} \sup_{\boldsymbol{u}\in\mathbb{U}} \mathcal{Z}\left(x,\boldsymbol{u}(\cdot),\beta[\boldsymbol{u}](\cdot)\right). \tag{17}$$

Later we will show that $Z(x)$ can be used to construct over- and underapproximations of the reachable set. Furthermore, the tightness of the approximations can be tuned.

To compute $Z(x)$, we first define two convenience functions

$$h(x) := l(x) - L \tag{18}$$

$$U(x) := Z(x) - L. \tag{19}$$

Clearly, $Z(x)$ is trivially recovered from $U(x)$, and $U(x)$ is a solution to a HJ equation.

Theorem 1: The function $U(x)$ is the unique viscosity solution to the following time-independent HJ equation:

$$0 = \min\left\{h(x) - U(x), \max_{u\in\mathcal{U}}\min_{d\in\mathcal{D}}\frac{\partial U}{\partial x}(x)f(x,u,d) - \lambda U(x)\right\}. \tag{20}$$

The structure of the proof is analogous to well-known classical viscosity solution approaches, such as the ones found in [19] and [25]. The only minor difference is that the positive multiplicative discount term in (16) has to be propagated through all the steps of the proof. More generally, the proofs first show that a proposed solution to a HJ equation satisfies a particular *DP principle*, and then shows that the DP principle is contradicted if the proposed solution is not also the unique viscosity solution for the HJ equation. The DP principle for $U(x)$ is stated below.

Proposition 1 (Dynamic programming principle): For $\delta > 0$

$$U(x) = \inf_{\beta[\boldsymbol{u}](\cdot)\in\mathcal{B}} \sup_{\boldsymbol{u}\in\mathbb{U}_\delta}\left[\min\left\{\inf_{t\in[0,\delta]} h\left(\xi_x^{\boldsymbol{u},\beta[\boldsymbol{u}]}(t)\right)\exp(-\lambda t),\right.\right.$$
$$\left.\left.\exp(-\lambda\delta)U(\xi_x^{\boldsymbol{u},\beta[\boldsymbol{u}]}(\delta))\right\}\right] \tag{21}$$

[7] $\Phi_{\pi,\rho}$ is effectively the probability transition matrix for a Markov Decision process over a finite state space given by the grid nodes $\{x_i\}_{i=1}^{n_G}$.

[8] In particular, $||B[\vec{A}_1] - B[\vec{A}_2]||_\infty \leqslant \gamma||\vec{A}_1 - \vec{A}_2||_\infty$ for any two vectors $\vec{A}_1, \vec{A}_2$.

[9] Any vector $\alpha\vec{1} \in \mathbb{R}^{n_G}$ is a fixed point of (11) for $\alpha < -L$.

where $\mathbb{U}_\delta$ consists of measurable functions on the interval $[0, \delta]$. The expression in Proposition 1, represents the main differences between the proofs of Theorem 1 and those in [19] and [25].

Next, we show how to solve (20), and thus solve for $Z(x)$.

B. Computing the Discounted Value Function

The discrete approximation of (20) is given by

$$U_{\Delta t}(x) = \min\left\{h(x), \max_{u\in\mathcal{U}}\min_{d\in\mathcal{D}} \gamma U_{\Delta t}(x+\Delta t f(x,u,d))\right\} \tag{22}$$

which can be solved on a grid G via VI

$$\vec{U}^0 \in \mathbb{R}^{n_G} \tag{23a}$$

$$\vec{U}^{k+1} = B[\vec{U}^k] \tag{23b}$$

$$\vec{U} = \lim_{k\to\infty} \vec{U}^k \tag{23c}$$

with the backup operator defined as

$$B[\vec{A}] := \min\left\{\vec{h}, \max_{\pi}\min_{\rho} \gamma\Phi_{\pi,\rho}\vec{A}\right\} \tag{24}$$

where $\vec{h}_i = h(x_i)$, and the VI procedure can be initiated arbitrarily.

The MDR value function $Z(x)$ is then approximated by $I[\vec{U}](x) + L$, where again $I[\vec{U}](\cdot)$ is the interpolation operator. We now prove the most significant result of this article, that is (24) is a contraction.

Lemma 1 (For any two functions q, g): A $\times B \to \mathbb{R}$, with A, B bounded sets

$$|\max_a\min_b q(a,b) - \max_a\min_b g(a,b)| \leqslant \max_a\max_b |q(a,b)-g(a,b)|. \tag{25}$$

Proof: Define any minimax optimizers for q as the pair (a_q, b_q), and minimax optimizers of g as the pair (a_g, b_g). Without loss of generality we assume that $q(a_q, b_q) \geqslant g(a_g, b_g)$. We then have the following inequalities:

$$|\max_a\min_b q(a,b) - \max_a\min_b g(a,b)| \leqslant |q(a_q,b_q) - \min_b g(a_q,b)|$$
$$\leqslant |q(a_q,b_{gg}) - g(a_q,b_{gg})| \leqslant \max_a\max_b |q(a,b)-g(a,b)|$$

with $b_{gg} := \arg\min_b g(a_q, b)$. ■

Theorem 2: The operator given by (24) is a contraction mapping in the infinity norm $||\cdot||_\infty$ on the space $\mathbb{R}^{n_G}$.

Proof: Defining $B[\cdot]$ as in (24), take $\vec{A}_1, \vec{A}_2 \in \mathbb{R}^{n_G}$

$$||B[\vec{A}_1] - B[\vec{A}_2]||_\infty =$$

$$||\min\left\{\vec{h}, \max_\pi\min_\rho \gamma\Phi_{\pi,\rho}\vec{A}_1\right\} - \min\left\{\vec{h}, \max_\pi\min_\rho \gamma\Phi_{\pi,\rho}\vec{A}_2\right\}||_\infty.$$

Leveraging the identity $\min\{\vec{a},\vec{b}\} = \frac{1}{2}((\vec{a}+\vec{b}) - |\vec{a}-\vec{b}|)$ and using the shorthand $\Pi[\vec{A}] = \max_\pi\min_\rho \gamma\Phi_{\pi,\rho}\vec{A}$, the above is equal to

$$\frac{1}{2}||(\Pi[\vec{A}_2] - \Pi[\vec{A}_1]) - (|\Pi[\vec{A}_1] - \vec{h}| - |\Pi[\vec{A}_2] - \vec{h}|)||_\infty$$

which by the triangle inequality, is upper bounded by

$$\frac{1}{2}||(\Pi[\vec{A}_1] - \Pi[\vec{A}_2])||_\infty + \frac{1}{2}||(|\Pi[\vec{A}_1] - \vec{h}| - |\Pi[\vec{A}_2] - \vec{h}|)||_\infty.$$

Given the inequality $||(|\vec{a}| - |\vec{b}|)||_\infty \leqslant ||\vec{a}-\vec{b}||_\infty$, this has upper bound

$$||(\Pi[\vec{A}_1] - \Pi[\vec{A}_2])||_\infty.$$

Finally from Lemma 1, the last upper bound is

$$||\max_\pi\min_\rho \gamma\Phi_{\pi,\rho}\vec{A}_1 - \max_\pi\min_\rho \gamma\Phi_{\pi,\rho}\vec{A}_2||_\infty$$

$$= \max_\pi\max_\rho ||\gamma\Phi_{\pi,\rho}(\vec{A}_1 - \vec{A}_2)||_\infty \leqslant \gamma||\vec{A}_1 - \vec{A}_2||_\infty$$

where the last inequality comes from the fact that $\Phi_{\pi,\rho}$ is a stochastic matrix for all policies, thus $||\Phi_{\pi,\rho}||_\infty = 1$. ■

Theorem (2) shows that the value function $Z(x)$ can be approximated as the fixed point solution to a contraction mapping.

In the next section, we show that the reachable set can be approximated from $Z(x)$, thus implying the reachability problem can be approximately solved by solving for the fixed-point of a contraction mapping. To the best of the authors knowledge using a contraction mapping, like (24), to compute reachable sets has never been done before. In Section IV, we discuss the computational advantages and new opportunities that come with this approach for computing reachable sets.

C. Under- and Overapproximating the Reachable Set

With MDR formulation there is no particular level curve of the value function that characterizes the reachable set. However, it is possible to find level curves that correspond to over- and underapproximations of the reachable set.

We have the inequality $Z(x) \geqslant V(x)$ because the discounted terms in the outcome are nonpositive. It immediately follows that:

$$\{x \mid Z(x) \leqslant 0\} \subseteq \mathcal{R}(\mathcal{T}). \tag{26}$$

To get an overapproximation of the reachable set, start by defining $\tau(x)$ as the time when the minimum distance to the target is achieved for a trajectory starting at state x under the optimal control and disturbance signals $\boldsymbol{u}, \boldsymbol{d}$ for (5). Let $\boldsymbol{u}_\lambda, \boldsymbol{d}_\lambda$ represent the optimal signals for (17), then we have the following bound:

$$Z(x) - V(x) = \inf_{t\geqslant 0}(l(\xi_x^{\boldsymbol{u}_\lambda,\boldsymbol{d}_\lambda}(t)) - L)e^{-\lambda t} - (\inf_{t\geqslant 0} l(\xi_x^{\boldsymbol{u},\boldsymbol{d}}(t)) - L)$$
$$\leqslant \inf_{t\geqslant 0}(l(\xi_x^{\boldsymbol{u}_\lambda,\boldsymbol{d}}(t)) - L)e^{-\lambda t} - (\inf_{t\geqslant 0} l(\xi_x^{\boldsymbol{u}_\lambda,\boldsymbol{d}}(t)) - L)$$
$$\leqslant (L - l(\xi_x^{\boldsymbol{u},\boldsymbol{d}}(\tau(x))))(1-\exp(-\lambda\tau(x))). \tag{27}$$

Noting that $V(x) = l(\xi_x^{\boldsymbol{u},\boldsymbol{d}}(\tau(x)))$, we get

$$Z(x) - L(1-\exp(-\lambda\tau(x))) \leqslant V(x)\exp(-\lambda\tau(x)). \tag{28}$$

Clearly, if $Z(x) > L(1-\exp(-\lambda\tau(x)))$, then $V(x) > 0$ and $x \notin \mathcal{R}(\mathcal{T})$. Assuming an upper bound $\bar{\tau} \geqslant \tau(x)$, we have the following overapproximation for the reachable set:

$$\mathcal{R}(\mathcal{T}) \subseteq \{x \mid Z(x) \leqslant L(1-\exp(-\lambda\bar{\tau}))\}. \tag{29}$$

Given $V(x) \leqslant Z(x)$ and (28), we have

$$\frac{Z(x) - L(1-\exp(-\lambda\tau(x)))}{\exp(-\lambda\tau(x))} \leqslant V(x) \leqslant Z(x) \tag{30}$$

and from this expression we see the tightness of the bounds (and thus the reachable set approximations) is tunable by λ.

IV. Implications of Discounting

So far we have presented a novel approach for computing reachable through the fixed-point of a contraction mapping. This has two main implications: one, iterative methods that empirically have better convergence than VI can now be employed, and two, this approach also paves the way for learning reachable sets in a manner that is analogous to RL. It is important to note that many of the results presented in this section are standard for the TDR setting, and the main insight is that they also apply to the newly formulated MDR setting.

A. Improving Convergence

VI can be used to solve for the fixed-point, but there are also other approaches that may offer faster convergence like policy iteration and multigrid approaches.

Multigrid approaches manage the tradeoff between computation and accuracy that comes from selecting the grid resolution. The general approach is to first run VI on a coarse grid (e.g., grid spacing $2\Delta x_j$), and then to use the result as the initialization for VI on a finer grid. Using the coarse solution (CS) as an initialization is possible because of the contraction mapping. These approaches have been applied extensively for TDR problems, where empirical and theoretical improvements have been shown [12], [13]. We compare multigrid approaches to VI in Section V.

In the one player setting (control only), policy iteration is another alternative to VI. Like the name suggests, the method iterates through policies until the optimal policy is obtained. In practice, policy iteration is typically recommended over VI because policies can converge faster than values resulting in faster convergence of the algorithm [26]. A more detailed analysis of policy iteration (as it pertains to TDR) can be found in [10], [11], and [27].

In order to present the policy iteration algorithm for MDR problems, we first define the *policy backup operator* $B^\pi[\cdot] : \mathbb{R}^{n_G} \to \mathbb{R}^{n_G}$

$$B^\pi[\vec{A}] = \min\left\{\vec{h}, \gamma\Phi_\pi\vec{A}\right\}. \tag{31}$$

This operator is a contraction mapping. The policy iteration algorithm generates the sequence $\{\vec{U}^{\pi^k}\}_k$ according to

$$\vec{U}^{\pi^k} = B^{\pi^k}[\vec{U}^{\pi^k}] \tag{32a}$$

$$\pi^{k+1} = \arg\max_\pi B^\pi[\vec{U}^{\pi^k}] \tag{32b}$$

$$\vec{U} = \lim_{k\to\infty}\vec{U}^{\pi^k}. \tag{32c}$$

Note that $\vec{U}^{\pi^k}$, the fixed-point of (32a), is obtained through VI with the policy backup. Furthermore, policy iteration is possible because the convergence of each policy backup is agnostic to the initialization.

B. Learning Reachable Sets

When the system model is unknown or complex, the reachable set must be learned from data. There are two approaches for this: model-based and model-free.

In the model-based approach the model is assumed to be parameterized by a vector μ. The model parameters are fitted using data from the system. The value function is then computed using this model. As more data are collected, the process can be repeated. Here, we are intentionally vague about the data and the fitting process, and we focus our attention on how to obtain the value function given the fitted model.

The data collection and fitting produce a sequence of parameters $\{\mu_k\}_k$, which in turn corresponds to a sequence of models $\{f_{\mu_k}\}_k$ and vectorized value functions $\{\vec{U}_{\mu_k}\}_k$. With the MDR formulation $\vec{U}_{\mu_k}$ can be used as the initialization when computing $\vec{U}_{\mu_{k+1}}$. Assuming regularity in the dynamics (with respect to μ), if the parameters only deviate slightly between iterations then $\vec{U}_{\mu_k}$ should be a good approximation of $\vec{U}_{\mu_{k+1}}$, resulting in faster convergence. By contrast, for the MR formulation the VI algorithm must be initialized with $\vec{l}$ every time a new value function is computed.

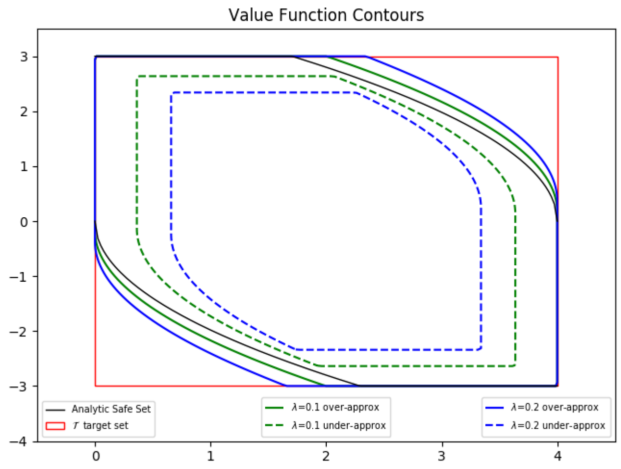

Fig. 1. Analytic safe set and target set $\mathcal{T}$ are shown in black (interior) and red (exterior), respectively. The over and under approximated Z are shown in bold green and dotted green for $\lambda = 0.1$; and bold blue and dotted blue line for $\lambda = 0.2$ (all interior).

Model-free approaches attempt to compute the value function directly from the data. This is the approach taken in many RL algorithms that attempt to approximate the value function for TDR problems with unknown models.

Temporal difference (TD) learning is at the heart of many of these methods, which includes TD-lambda [28], Q-learning [29], deep Q Networks [30], and actor–critic methods [31]. The key idea is to represent the value function with a parametric function approximator,[10] define a loss function on the parameters that is minimized when the value function is the fixed-point of the TDR backup operator, and to update the parameters by performing stochastic gradient descent on the loss function with samples from a real system or simulator. The typical loss function in the TDR setting is

$$\mathcal{L}(\theta) = \frac{1}{2}\left(V_\theta(x) - (r(x) + \gamma V_\theta(x^+))\right)^2 \tag{33}$$

where x^+ refers to the state transitioned to from x in the dataset. All of these techniques can be used in the MDR by changing the loss function to

$$\mathcal{L}(\theta) = \frac{1}{2}\left(U_\theta(x) - \min\{h(x), \gamma U_\theta(x^+)\}\right)^2. \tag{34}$$

We do not explore the model-free techniques any further in this paper, but leave it open to future work.

V. Numerical Examples

This section uses two benchmark models, double integrator and pursuit-evasion game, to exemplify the numerical properties of the MDR formulation. The double integrator model will be used to display the overapproximation/underapproximation of the reachable set, as well as to compare policy iteration with VI. The pursuit-evasion game will be used to demonstrate the advantages of multigridding, and initializing VI with precomputed solutions to similar problems.

Unless stated otherwise, all algorithms are initialized with $\vec{h} = \vec{l} - L$, and are considered converged when the distance (in the infinity norm) between consecutive iterates falls below $\epsilon = .001$. All

[10]A simple function approximator would be interpolation on a grid where the parameters are the grid node values.

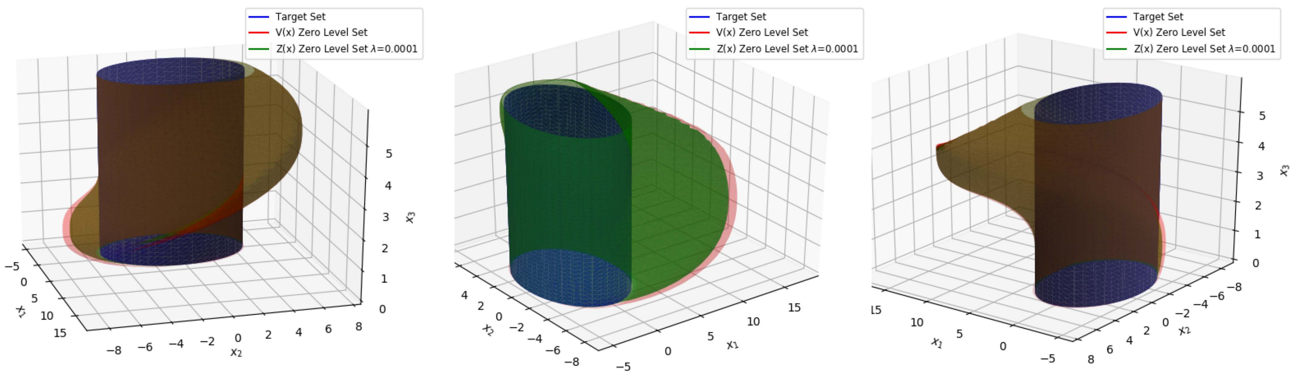

Fig. 2. Target set $\mathcal{T}$ (blue cylinder), zero sublevel sets of V (red), and Z (green) shown from three different perspectives. The discount rate for Z is $\lambda = 0.01$. Note that the zero sublevel set of Z is a subset of the zero sublevel set of V.

TABLE I
VI VERSUS POLICY ITERATION

CPU time in seconds

# actions	VI	Policy Iteration	
	T_{total}	T_{total}	$T_{\text{total}} - T_{\Phi_{\pi_u}}$
2	1.468	78.197	0.102
50	8.753	302.456	1.007
250	36.973	308.565	4.318
500	65.305	326.280	9.760

experiments were run on a 2016 MacBook Pro with Core i7 processor and 16 GB RAM.

A. Double Integrator

The double integrator consists of two states (x_1, x_2), and control $u \in [-u_{\max}, u_{\max}]$ with dynamics

$$\dot{x_1} = x_2$$
$$\dot{x_2} = u. \qquad (35)$$

The state space is discretized into a 161×161 grid on the domain $[-1, 5] \times [-5, 5]$, and $u_{\max} = 2$.

The task is to keep the state trajectory inside the box $\mathcal{K} = [0, 4] \times [-3, 3]$, thus the target is taken to be its complement $\mathcal{T} = \mathcal{K}^C$. For ease of exposition we define the *safe set* $\Omega(\mathcal{T}) := \mathcal{R}(\mathcal{T})^C$.

We first show, in Fig. 1, that different level curves of Z overapproximate (bold line), and underapproximate (dotted lines) the analytic safe set (shown in black) for two values of $\lambda = 0.1$ and 0.2, with $\bar{\tau} = 2$. As expected, smaller values of λ yield tighter approximations.

To verify the convergence properties of the MDR formulation, we initialize VI with the zero vector $\vec{0} \in \mathbb{R}^{N_G}$, and set $\lambda = 0.1$. The error (in the infinity norm) between the converged solution and the one visualized in Fig. 1 (which was initialized with $\vec{h}$) is 0.000299, suggesting convergence to the same fixed point. Under the MR setting, VI fails to converge with this particular initialization.

Moving on to Table I, we compare VI and policy iteration with increasing number of discrete actions. In the table, we see that the runtime of VI increases linearly with the number of actions, and policy iteration scales much better. In our particular implementation, the overall runtime favors VI, but it is important to note that the majority of the time in policy iteration is spent constructing Φ_{π_u}, which is denoted by $T_{\Phi_{\pi_u}}$.[11] Excluding this cost, policy iteration becomes more attractive.

[11]The data structure used to represent the interpolation is very efficient for sparse matrix multiplication, but is not ideal for indexing, which is necessary to create Φ_{π_u}, and results in a relatively large $T_{\Phi_{\pi_u}}$.

TABLE II
PURSUIT EVASION: VI WITH MULTIGRID

CPU time in seconds

# nodes	Coarse grid	Fine grid	Fine grid-CS	Multigrid
40^3	2.068	29.684	24.320	26.388
80^3	33.166	352.099	317.444	350.610

B. Pursuit-Evasion Game

We now consider the pursuit-evasion game described in [6]. In the game, player I (the control) tries to avoid being captured by player II (the disturbance) on a two dimensional plane. Each player is modeled as a simple kinematic point object with planar position and heading, fixed linear velocity, and controllable angular velocity. Taking player I to be at the origin, the states (x_1, x_2, x_3) are the relative position and heading of player II. The dynamics are

$$\dot{x_1} = -v_u + v_d \cos x_3 + u x_2$$
$$\dot{x_2} = v_d \sin x_3 - u x_1$$
$$\dot{x_3} = d - u. \qquad (36)$$

The state space is over the domain $[-6, 20] \times [-10, 10] \times [0, 2\pi[$ with $\mathcal{U} = [-u_{\max}, u_{\max}]$ and $\mathcal{D} = [-d_{\max}, d_{\max}]$.

Player I is considered captured when the relative distance (in position) between both players is less than $R > 0$, thus the target set is given by

$$\mathcal{T} = \{x | x_1^2 + x_2^2 < R^2\}. \qquad (37)$$

We first compute the value functions for the MR and MDR on a $41 \times 41 \times 41$ grid, setting the model parameters to $v_u = v_d = 5$, $u_{\max} = d_{\max} = 1$, and $R = 5$. This will be referred to as the nominal model M_n. A visualization of the zero sublevel set, for both $V(x)$ and $Z(x)$ for $\lambda = 0.001$, is shown in Fig. 2.

In the first experiment, we compare a multigrid approach to VI. For the multigrid, we first run VI on a coarse grid, which we construct to have half the resolution per dimension of the nominal grid, e.g., if the nominal grid has 41^3 nodes then the coarse grid has 21^3 nodes. The results are given in Table II. We first run VI with the standard initialization on both the coarse and fine grid. This produces the values in columns two and three. We then run VI on the fine grid initialized with the CS, which makes up column four. Column five (multigrid) is obtained by adding columns two and four. The multigrid approach outperformed VI for the pursuit-evasion game.

We now construct two other models by tweaking M_n: setting $u_{\max} = 1.5$, which gives the evader an advantage, we get model M_e, and setting $d_{\max} = 1.5$, which gives the pursuer an advantage, we get

TABLE III
PURSUIT EVASION: VI WITH WS

CPU time in seconds

# nodes	M_n	M_e	M_e-WS	M_p	M_p-WS
40^3	35.043	32.242	21.483	26.319	23.366
80^3	439.751	416.965	308.821	300.847	296.568

model M_p. In the final experiment, we look at the impact of initializing VI with a solution from a similar model. Just like in the previous benchmark example, this experiment is motivated by Section IV-B, where now M_e and M_p represent two possible models inferred from the system observations. In this context, we have just "learned" that the evader/pursuer is more maneuverable (M_e/M_p). We compute both value functions with and without setting the initialization to the solution for M_n. Again, we refer to this initialization as a warm start (WS). The results are given in Table III.

VI. Conclusions and Future Work

We presented a novel MDR HJ formulation for approximating reachable sets. The main advantage of this new formulation, over previous work, is that the solution can be obtained as the unique fixed point to a contraction mapping. We also showed that policy iteration and multigrid approaches can be used to yield faster convergence.

As learning and data-driven approaches become more powerful and pervasive, perhaps the greatest contribution of this work is that it can lie somewhere in between traditional control and model-free RL. The approach is certainly model-based, but because of its agnosticism to initialization it also has the flexibility to incorporate data and build towards solutions iteratively. We foreshadowed how this can be done with TD learning, and in the future it can be applied to other RL algorithms.

References

[1] J. Ding, C. J. Tomlin, L. R. Hook, and J. Fuller, "Initial designs for an automatic forced landing system for safer inclusion of small unmanned air vehicles into the national airspace," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf.*, 2016, pp. 1–12.

[2] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," in *Proc. Conf. Decis. Control*, 2015, pp. 4695–4701.

[3] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," in *Proc. IEEE 53rd Conf. Decis. Control*, 2014, pp. 1424–1431. [Online]. Available: http://www.ece.ubc.ca/kaynama/papers/CDC2014_safelearning.pdfhttps://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7039601

[4] J. H. Gillula and C. J. Tomlin, "Guaranteed safe online learning via reachability: Tracking a ground target using a quadrotor," in *Proc. Int. Conf. Robot. Autom.*, May 2012, pp. 2723–2730. [Online]. Available: https://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6225136

[5] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *Proc. IEEE 47th Conf. Decis. Control*, 2008, pp. 4357–4363. [Online]. Available: https://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4739434

[6] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[7] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *Proc. Int. Conf. Robot. Autom.*, 2011, pp. 2160–2165. [Online]. Available: https://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980268

[8] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proc. Int. Conf. Robot. Autom.*, 2011, pp. 1451–1456. [Online]. Available: https://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980264

[9] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic Program. and Optimal Control*. Belmont, MA, USA: Athena Scientific, 1995.

[10] R. A. Howard, *Dynamic Program. and Markov Processes*. Cambridge, MA, USA: MIT Press, 1964.

[11] M. L. Puterman and S. L. Brumelle, "On the convergence of policy iteration in stationary dynamic programming," *Math. Operations Res.*, vol. 4, no. 1, pp. 60–69, 1979.

[12] A. Alla, M. Falcone, and D. Kalise, "An efficient policy iteration algorithm for dynamic programming equations," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. A181–A200, 2015.

[13] C. Chow and J. N. Tsitsiklis, "An optimal one-way multigrid algorithm for discrete-time stochastic control," *IEEE Trans. Autom. Control*, vol. 36, no. 8, pp. 898–914, Aug. 1991.

[14] A. K. Akametalu and C. J. Tomlin, "Temporal-difference learning for online reachability analysis," in *Proc. Control Conf.*, 2015, pp. 2508–2513.

[15] B. Djeridane and J. Lygeros, "Neural approximation of PDE solutions: An application to reachability computations," in *Proc. IEEE 45th Conf. Decis. Control*, 2006, pp. 3034–3039.

[16] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. New York, NY, USA: Tata McGraw-Hill, 1955.

[17] E. Barron and H. Ishii, "The Bellman equation for minimizing the maximum cost," *Nonlinear Anal.: Theory, Methods Appl.*, vol. 13, no. 9, pp. 1067–1089, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0362546X89900965

[18] E. Barron, "Differential games with maximum cost" *Nonlinear Anal.: Theory, Methods Appl.*, vol. 14, no. 11, pp. 971–989, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0362546X9090113U

[19] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. 18th Int. Conf. Hybrid Syst.: Comput. Control*, 2015, pp. 11–20.

[20] M. Bardi, M. Falcone, and P. Soravia, "Numerical methods for pursuit-evasion games via viscosity solutions," in *Stochastic And Differential Games*. Berlin, Germany: Springer, 1999, pp. 105–175.

[21] M. Falcone and R. Ferretti, "Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations," *Numerische Mathematik*, vol. 67, no. 3, pp. 315–344, 1994.

[22] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Berlin, Germany: Springer Science & Business Media, 2003, vol. 153.

[23] J. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=39986&tool=pmcentrez&rendertype=abstract

[24] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Berlin, Germany: Springer Science & Business Media, 2008.

[25] 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://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA127758

[26] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial Intelligence: A Modern Approach*, vol. 2, no. 9. Upper Saddle River, NJ, USA: Prentice hall, 2003.

[27] O. Bokanowski, S. Maroso, and H. Zidani, "Some convergence results for Howard's algorithm," *SIAM J. Numer. Anal.*, vol. 47, no. 4, pp. 3001–3026, 2009.

[28] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[29] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 279–292, 1992.

[30] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[31] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1008–1014.