



MIT Open Access Articles

Continuous Tensor Train-Based Dynamic Programming for High-Dimensional Zero-Sum Differential Games

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation	Tal, Ezra, Gorodetsky, Alex and Karaman, Sertac. 2018. "Continuous Tensor Train-Based Dynamic Programming for High-Dimensional Zero-Sum Differential Games."
As Published	10.23919/acc.2018.8431472
Publisher	IEEE
Version	Author's final manuscript
Citable link	https://hdl.handle.net/1721.1/137851
Terms of Use	Creative Commons Attribution-Noncommercial-Share Alike
Detailed Terms	http://creativecommons.org/licenses/by-nc-sa/4.0/

Continuous Tensor Train-Based Dynamic Programming for High-Dimensional Zero-Sum Differential Games

Ezra Tal, Alex Gorodetsky, and Sertac Karaman

Abstract—Zero-sum differential games constitute a prominent research topic in several fields ranging from economics to motion planning. Unfortunately, analytical techniques for differential games can address only simple, illustrative problem instances, and most existing computational methods suffer from the curse of dimensionality, *i.e.*, the computational requirements grow exponentially with the dimensionality of the state space. In order to alleviate the curse of dimensionality for a certain class of two-player pursuit-evasion games, we propose a novel dynamic-programming-based algorithm that uses a continuous tensor-train approximation to represent the value function. In this way, the algorithm can represent high-dimensional tensors using computational resources that grow only polynomially with dimensionality of the state space and with the rank of the value function. The proposed algorithm is shown to converge to optimal solutions. It is demonstrated in several problem instances; in case of a seven-dimensional game, the value function representation was obtained with seven orders of magnitude savings in computational and memory cost, when compared to standard value iteration.

I. INTRODUCTION

The field of differential games has been the subject of extensive research since the seminal work by Isaacs in the 1950s [1]. Differential game theory is relevant to diverse fields, ranging from robust control to economics [2].

We consider two-player differential games with zero-sum utilities. This class of games is studied in the context of optimal control, for example concerning pursuit-evasion scenarios in collision avoidance [1], [2]. If linear dynamics and quadratic cost are considered, solutions can be obtained using the Game Riccati differential equation. However, these results do not hold for a more general class of games, including those that involve nonlinear dynamics [2].

Numerical schemes that obtain the game’s value function by finding a solution to the Hamilton-Jacobi-Isaacs partial differential equation are commonly utilized in order to address games for which analytical solutions cannot be obtained. The dynamic programming (DP) approach specifically does so by applying a backwards analysis based on Bellman’s principle of optimality. Methods based on DP are successfully applied to differential games in the literature, *e.g.*, [3], [4]. Yet, DP methods based on naive grid discretization are intractable for games represented by high-dimensional vector differential equations. Strictly speaking, memory and computation requirements grow exponentially

with increasing dimensionality of the state space. This phenomenon is called the *curse of dimensionality*. As a result, existing numerical methods based on grid discretization can handle only up to a few dimensions (*e.g.*, no more than three), and do not scale to realistic problem instances.

Existing literature addresses the curse of dimensionality in various ways, *e.g.*, using system decompositions [5] or by forgoing spatial discretization [6]. In this paper, we propose a new representation for the value function in order to alleviate the curse of dimensionality in low-rank problems. The tensor-train (TT) decomposition enables representation of a high-dimensional problem in a manner that does not inherently suffer from the curse of dimensionality [7]. Specifically, we utilize a continuous analogue of the TT called the functional tensor-train (FT). While the TT represents an n -dimensional tensor, the FT represents an n -variate function. A software package for FT-based calculation is freely available [8], and was used for implementations in this work.

The application of FT-based approximation to represent the cost function has produced promising outcomes in one-sided optimal control [9]. The algorithm avoids the curse of dimensionality and remains feasible even for high-dimensional problems. This is especially relevant to the present work, as differential games often need to account for state variables corresponding to both players.

In this paper we present a dynamic programming algorithm for high-dimensional pursuit-evasion games. The algorithm employs FT-based representations and computational methods to approximate Bellman’s DP operator. By virtue of the approximation, the computational cost and memory requirement grow polynomially with increasing dimension. Therefore the algorithm remains tractable for high-dimensional problems that admit a low-rank value function.

We prove that the proposed algorithm converges to the exact value function, for a certain class of problems. Along the way, we provide a complexity analysis and worst-case convergence bounds based on the contraction property of value iteration. Finally, we present computational results for several challenging problem instances, including a seven-dimensional aerial pursuit scenario.

II. PROBLEM DEFINITION

In this section, the problem formulation is presented, and conditions for existence and continuity of the value function are provided. These results are used to show the consistency of discrete game approximations in the next section.

We consider a general vector-valued differential equation

$$\dot{x}(t) = f(t, x(t), u(t), v(t)), \quad x(0) = x_0,$$

E. Tal and S. Karaman are with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA. {eatal, sertac}@mit.edu

A. Gorodetsky is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA. goroda@umich.edu

where $x \in X$ is the state, $u \in \mathcal{U}$ is the control by player i (the minimizer), and $v \in \mathcal{V}$ is the control by player ii (the maximizer).

- Assumption 1** (i) The function $f : [0, t_f] \times X \times \mathcal{U} \times \mathcal{V} \rightarrow \mathbb{R}^n$ is bounded and continuous in x ,
(ii) the state space X is a compact Riemannian manifold,
(iii) the control inputs $u(\cdot) : [0, t_f] \rightarrow \mathcal{U}$, $v(\cdot) : [0, t_f] \rightarrow \mathcal{V}$ are measurable for $t \in [0, t_f]$,
(iv) the control sets $\mathcal{U} \subset \mathbb{R}^{n_u}$, $\mathcal{V} \subset \mathbb{R}^{n_v}$ are compact.

We consider three distinct cost structures:

- (IH) Infinite-horizon games with discount rate $\beta > 0$ and cost function:

$$J_{\mu_u, \mu_v}(x_0) = \int_0^\infty e^{-\beta t} g(x(t), \mu_u(x(t)), \mu_v(x(t))) dt,$$

where $\mu_u : X \rightarrow \mathcal{U}$ is a feedback strategy for player i and $\mu_v : X \rightarrow \mathcal{V}$ for player ii.

- (PT) Games with predetermined finite termination time t_f and cost function:

$$J_{\mu_u, \mu_v}(x_0) = \int_0^{t_f} g(t, x(t), \mu_u(x(t)), \mu_v(x(t))) dt + q(x(t_f)).$$

- (TC) Games with a terminal constraint set $X_f \subset X$, such that $t_f = \inf\{t \in \mathbb{R}^+ | x(t) \in X_f\}$ and cost function:

$$J_{\mu_u, \mu_v}(x_0) = \int_0^{t_f} g(x(t), \mu_u(x(t)), \mu_v(x(t))) dt + q(x(t_f)).$$

For (TC), the termination time from $x(0) = x_0$ under the policy pair (μ_u, μ_v) is denoted by $t_f(x, \mu_u, \mu_v)$.

We assume that in all games both players have knowledge of the full state, the opponent control action, and the dynamics and cost structure of the game. For the sake of brevity, we will omit time indices from this point onward. Note that in case of (IH) and (TC), the running cost function is independent of time. In case of (PT), a general time independent expression can be recovered by setting the augmented state $x \leftarrow (t, x)$, which is possible as t_f is finite. This allows us to unify the cost structures of (PT) and (TC) by defining a terminal set as $X_f = \{(t, x) | t \geq t_f\}$ for (PT).

In this work, we focus on memoryless pure strategies. Strategies with memory need not be considered, given that a saddle point in feedback strategies exists [2]. Mixed strategies could be considered; however, they are technically distributions over the multi-dimensional continuous control spaces, which makes their synthesis fairly challenging.

The solution of a game is the value function, which can then be used as an implicit representation of the optimal strategies. Specifically, the value function $V : X \rightarrow \mathbb{R}$ gives the cost of the game from an initial condition $x \in X$. Its upper value is given by

$$\bar{V}(x) = \min_{\mu_u \in \mathcal{M}_u} \max_{\mu_v \in \mathcal{M}_v} J_{\mu_u, \mu_v}(x), \quad (1)$$

where \mathcal{M}_u and \mathcal{M}_v are the sets of admissible strategies for players i and ii. These sets consist of all functions $\mu_u : X \rightarrow \mathcal{U}$ and $\mu_v : X \rightarrow \mathcal{V}$, respectively. The reciprocal lower value is obtained by swapping the min and max operators.

- Assumption 2** The differential equation f and the running cost g are separable into terms containing u and v , i.e.,

$$\begin{aligned} f(x, u, v) &= f_u(x, u) + f_v(x, v), \\ g(x, u, v) &= g_u(x, u) + g_v(x, v). \end{aligned}$$

Furthermore, f_u , f_v , g_u , and g_v are continuous and bounded.

- Lemma 3** [10] Under Assumptions 1 and 2, the differential game admits a saddle point equilibrium, such that there exists a value function V and $V = \bar{V} = \underline{V}$.

Necessary conditions for existence of a value function are hard to formulate [11]. A sufficient condition is given by Assumption 2 and used in Lemma 3. Assumption 4 below can be interpreted as a controllability condition in proximity of X_f , so that the termination set can be entered through any point on its boundary. This results in Lipschitz continuity of the value function along the boundary of the termination set [12]. Finally, Lemma 5 below gives the global continuity result for the value function.

- Assumption 4** For (PT) and (TC), the terminal cost function q is bounded and Lipschitz continuous. Additionally, in case of (TC), X_f is closed and its boundary ∂X_f is Lipschitz. Furthermore, $\forall x \in \partial X_f$, $t_f(x, \mu_u^*, \mu_v^*) -$ with μ_u^*, μ_v^* the saddle point policies - is continuous.

- Lemma 5** [13] Under Assumptions 1, 2, and 4, the value function $V(x)$ is Lipschitz continuous in x .

III. PRELIMINARIES

This section is devoted to the description of two essential building blocks of our algorithm: Markov Game (MG) approximations and low-rank tensor approximations. The MG approximation enables the application of the discrete DP paradigm. It is a game-theory analogue of the well-known Markov Decision Process (MDP) from control theory. The low-rank tensor approximation is used to represent the value function of the MG in compressed form. Its purpose is to mitigate the curse of dimensionality that naive DP algorithms suffer from, and thereby enable the application of our algorithm to high-dimensional differential games.

A. Markov Game approximation

The MG approximation consists of discretization in both time and space. The state trajectory and control inputs are approximated by piece-wise constant functions. The MG is defined by the tuple $\mathcal{G}^h = \langle Z^h, Z_f^h, \mathcal{U}, \mathcal{V}, p^h, r^h, q^h, \gamma^h \rangle$, where $Z^h \subset X$ is a finite set of states, $Z_f^h \subset Z^h$ is the terminal constraint set for (PT) and (TC), $p^h(z, u, v, z')$ is the transition probability from state z to z' under control actions u and v , $r^h(z, u, v)$ is the stage cost for state z under

control actions u and v , q^h – with $q^h(z) = q(z)$, $\forall z \in Z^h$ – is the terminal cost function for (PT) and (TC), and γ^h is the discount factor for future cost for (IH). The parameter h is an approximation parameter that represents a measure of the fineness of the discretization. The discrete domain Z^h is typically chosen such that its convex hull is equal to X .

A pure feedback strategy is given by a map $\mu_u^h : Z^h \rightarrow \mathcal{U}$ for player i and $\mu_v^h : Z^h \rightarrow \mathcal{V}$ for player ii. For (PT) and (TC), the expected cost associated with a strategy pair and corresponding trajectory of finite length is given by

$$J_{\mu_u^h, \mu_v^h}^h(z_0) = \mathbb{E} \left[\sum_{i=0}^{N-1} r^h(z_i, \mu_u^h(z_i), \mu_v^h(z_i)) + q(z_N) \right]$$

with z_i the state at time step i , so that $z_N \in Z_f^h$ is the termination state. The expected cost for (IH) is

$$J_{\mu_u^h, \mu_v^h}^h(z_0) = \mathbb{E} \left[\sum_{i=0}^{\infty} \gamma^h r^h(z_i, \mu_u^h(z_i), \mu_v^h(z_i)) \right].$$

For convenience we assume a regular grid, but the methods can trivially be extended to any rectilinear grid. We denote the nodal interval along dimension i by $h[i]$ and define $h = \max_i h[i]$. The transition probabilities are obtained using barycentric interpolation over the standard simplex in the orthant corresponding to the direction of $f(z, u, v)$, leading to an upwind differencing scheme. Let $e[i]$ be the vector with i -th element equal to $h[i]$ and all other elements equal to zero. The transition probabilities are then given by

$$\begin{aligned} p^h(z, u, v, z + e[i]) &= \frac{\Delta t^h}{h[i]} f^+(z, u, v)[i] \\ p^h(z, u, v, z - e[i]) &= \frac{\Delta t^h}{h[i]} f^-(z, u, v)[i] \end{aligned}$$

for $i = 1, 2, \dots, n$ and $p^h(z, u, v, z)$

$$= 1 - \sum_{i=1}^n (p^h(z, u, v, z + e[i]) + p^h(z, u, v, z - e[i]))$$

with

$$\begin{aligned} f^+(z, u, v)[i] &= \begin{cases} f(z, u, v)[i] & \text{if } f(z, u, v)[i] > 0 \\ 0 & \text{otherwise} \end{cases}, \\ f^-(z, u, v)[i] &= \begin{cases} -f(z, u, v)[i] & \text{if } f(z, u, v)[i] < 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

In this work, transition probabilities along the boundaries of Z^h are specified by a reflective boundary condition. Consider a state z at an extreme point of Z^h , such that $z' = z + e[i] \notin Z^h$ or $z' = z - e[i] \notin Z^h$, then the transition probability to z' is disregarded and its value is added to the self-transition probability, as follows:

$$p^h(z, u, v, z) \leftarrow p^h(z, u, v, z) + p^h(z, u, v, z').$$

The temporal discretization interval Δt^h may be set specifically for the problem, and must satisfy

$$0 < \Delta t^h \leq Q^h(z, u, v)^{-1} \quad (2)$$

with $Q^h(z, u, v) = \sum_{i=1}^n \frac{|f(z, u, v)[i]|}{h[i]}$, in order to comply with the Courant–Friedrichs–Lewy (CFL) condition [14].

The stage cost of the MG is

$$r^h(z, u, v) = \frac{1 - \gamma^h}{\beta} g(z, u, v)$$

with $\gamma^h = e^{-\Delta t^h \beta}$ for (IH) and

$$r^h(z, u, v) = \Delta t^h g(z, u, v)$$

for (PT) and (TC), in which case $\gamma^h = 1$.

This scheme satisfies local consistency conditions provided below. The first two conditions can straightforwardly be verified by symbolic evaluation, the third condition holds by (2) and suitable selection of Δt^h in case of $f(z, u, v) = 0$, and the fourth condition holds trivially by setting $K_1 = \sqrt{n}$ with n the number of state dimensions.

Definition 6 [10] *The sequence of MGs \mathcal{G}^h satisfies local consistency conditions if $\Delta t^h(z, u, v) > 0$ such that*

- (i) $\mathbb{E}[z_{i+1}^h - z_i^h] = f(z, u, v) \Delta t^h(z, u, v) + o(\Delta t^h(z, u, v))$,
- (ii) $\text{Cov}[z_{i+1}^h - z_i^h] = o(\Delta t^h(z, u, v))$,
- (iii) $\limsup_{h \rightarrow 0} \Delta t^h(z, u, v) \rightarrow 0$,
- (iv) $\|z_{i+1}^h - z_i^h\|_2 \leq K_1 h$

for some real K_1 . In (ii), $o(\cdot)$ refers to little- o notation, i.e., we write $f(h) = o(g(h))$ to mean that $\lim_{h \rightarrow 0} \frac{f(h)}{g(h)} = 0$.

The upper value function of \mathcal{G}^h ,

$$V^{h+}(z) = \min_{\mu_u^h \in \mathcal{M}_u^h} \max_{\mu_v^h \in \mathcal{M}_v^h} J_{\mu_u^h, \mu_v^h}^h(z),$$

is the analogue of (1). Similar to the continuous differential game, the sets \mathcal{M}_u^h and \mathcal{M}_v^h contain all $\mu_u^h : Z^h \rightarrow \mathcal{U}$ and $\mu_v^h : Z^h \rightarrow \mathcal{V}$, respectively. The lower value function V^{h-} is given by swapping the min and max operators. Theorem 8 gives the MG consistency result, which is based on local consistency and continuity of the value function by Lemma 5. Note that V^{h+} (as well as V^{h-}) is a tensor that can be seen as a function $Z^h \rightarrow \mathbb{R}$, while V is a function $X \rightarrow \mathbb{R}$.

Remark 7 *To enable the comparison in Theorem 8 and onward, we implicitly consider a tensor consisting of evaluations of V on the set $Z^h \subset X$.*

Theorem 8 [10] *Under Assumptions 1, 2, 4 and satisfaction of local consistency conditions, $V^{h+}(x) \rightarrow \bar{V}$ and $V^{h-}(x) \rightarrow \underline{V}$ as $h \rightarrow 0$. Consequently, $V^{h+}(x) \rightarrow V$ and $V^{h-}(x) \rightarrow V$ as $h \rightarrow 0$, i.e., the upper and lower value functions of the MG converge to the value function of the continuous differential game.*

B. Low-rank tensor approximation

The value function of a MG on a regular grid with l nodes along each of its n dimensions imposes $\mathcal{O}(l^n)$ storage cost. Moreover, a naive approach to DP has computational complexity of the same order. Hence, it will become intractable as the dimension n increases. To mitigate this curse of dimensionality, we propose to exploit low-rank separable structure, i.e., that certain functions in the problem can be

written in terms of a small number of products of univariate functions. We exploit such separability by leveraging recent advances in the area of tensor decompositions [15].

In this work, we utilize the tensor-train representation of low-rank tensors [7] as an ansatz for representing low-multilinear-rank functions [16], [17]. Suppose $X \subset \mathbb{R}^n$ is a tensor-product domain, functional tensor-train (FT) representation of a multivariate function $f : X \rightarrow \mathbb{R}$ is

$$f(x_1, \dots, x_n) = \sum_{i_0=1}^{r_0} \cdots \sum_{i_n=1}^{r_n} f_{i_0, i_1}^{(1)}(x_1) \cdots f_{i_{n-1}, i_n}^{(n)}(x_n), \quad (3)$$

with r_i called the FT ranks. The univariate functions $f_{k,l}^{(i)}$ are scalar-valued functions of the i th variable and can take various forms [17] including expansions of polynomials, piecewise polynomials, piecewise linear elements, piecewise constant elements, etc. In the current work, piecewise linear functions are used according to the description provided in [9]. For clarity, Equation (3) can be written as the product of matrices with sizes $r_i \times r_{i+1}$ with $i = 0, 1, \dots, (n-1)$:

$$f(x_1, \dots, x_n) = F_1(x_1) \dots F_n(x_n)$$

with

$$F_i(x_i) = \begin{bmatrix} f_{1,1}^{(i)}(x_i) & \cdots & f_{1,r_i}^{(i)}(x_i) \\ \vdots & \ddots & \vdots \\ f_{r_{i-1},1}^{(i)}(x_i) & \cdots & f_{r_{i-1},r_i}^{(i)}(x_i) \end{bmatrix}. \quad (4)$$

Since f is a scalar-valued function, the FT imposes that $r_0 = r_n = 1$. As the ranks increase (and thereby the size of the cores), the ability to represent interactions between variables increases. The TT ranks are bounded by the singular value decomposition (SVD) ranks of the unfoldings of f [7], [18], [17], *i.e.*, $r_k \leq \text{SVD-RANK} f(x_{\leq k}; x_{>k})$ where $x_{\leq k} = (x_1, \dots, x_k)$ and $x_{>k} = (x_{k+1}, \dots, x_n)$. Note that in the functional case described here, the functional SVD represents f as a Hilbert-Schmidt kernel, and we refer the reader to further details about how the ranks of this kernel are related to the FT ranks in [18].

In this work we utilize the rank-adaptive cross-approximation scheme described in [17] as implemented in [9] to obtain a functional representation of the cost function. Similar to the discrete case, it only requires function evaluation at nodal locations. Hence, in this work the discrete value function V^h defined on the discrete domain Z^h is approximated by a piecewise linear continuous function \tilde{V}^h defined on the convex hull of Z^h . Akin Remark 7, we implicitly consider a tensor consisting of evaluations of \tilde{V}^h at Z^h when comparing \tilde{V}^h and V^h .

The algorithm aims to render an δ -accurate approximation as defined in Assumption 9. It requires $\mathcal{O}(nlr^2)$ function evaluations and $\mathcal{O}(nlr^3)$ operations, where we have assumed that $r_i < r$ for $i = 0, \dots, n$. The required memory for storage of the FT approximation is $\mathcal{O}(nlr^2)$. Evaluation of the FT at a single location in the state space requires $\mathcal{O}(nlr^2)$ operations. Note that a high FT rank may be required to obtain an accurate representation of some functions, and the

larger the FT rank the smaller the compression ratio we obtain. Thus our proposed representation format gains its full advantage when the cost functions have the low-rank separable structure exploited by the FT.

Assumption 9 *The rank-adaptive cross-approximation operator Π_δ with accuracy parameter $\delta \in (0, 1)$ provides an approximation $\tilde{V}^h = \Pi_\delta V^h$, such that*

$$\|\tilde{V}^h - V^h\|_F \leq \delta \|V^h\|_F$$

with $\|\cdot\|_F$ a multidimensional generalization of the Frobenius norm. Note that this is a tensor-norm, since we consider evaluations of \tilde{V}^h on Z^h .

IV. FT-BASED MINMAX DYNAMIC PROGRAMMING

In this section we propose the FT-based DP algorithm. The aim of the algorithm is to obtain the game's value function and thereby an implicit representation of the optimal strategy for each player. The first step is to obtain a consistent discretization of the differential game according to the method given in Section III-A. Next, the value function of the resulting MG is iteratively computed. Each iterate is computed and stored in compressed form to reduce computation time and storage requirements. Furthermore, a multigrid scheme is used to obtain convergence on a fine grid through successive refinement. First, we describe the approximation of the minmax value iteration in Section IV-A. Next, we describe a best-response algorithm that computes a solution to the MG in Section IV-B. Finally, we discuss multigrid in Section IV-C.

A. Approximate minmax value iteration

Value iteration performs an iterative calculation of the value function based on the Bellman equation

$$V^{h+}(z) = \min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} r^h(z, u, v) + \gamma^h \sum_{z'} p^h(z, u, v, z') V^{h+}(z'). \quad (5)$$

From this point, we omit some of the superscript modifiers to improve readability. Properties of the upper value function that hold under minmax DP, typically also hold for the lower value function under minmax DP.

We now define two minmax DP operators. The exact minmax operator T^h performs value iteration on the value function V^h in its full uncompressed form by solving the minmax optimization problem for all nodes $z \in Z^h$ at each iteration. Its approximate counterpart \tilde{T}_δ^h is a composite operator that performs value iteration and compression.

Definition 10 *Let $B(Z^h)$ be the set of real-valued value functions over Z^h , *i.e.*, $V^h \in B(Z^h)$. The minmax Bellman operator T^h is a mapping from $B(Z^h)$ to itself, defined as*

$$(T^h V^h)(z) = \min_{\mu_u^h \in \mathcal{M}_u^h} \max_{\mu_v^h \in \mathcal{M}_v^h} (T^{\mu_u, \mu_v} V^h)(z)$$

$$\text{with } (T_{\mu_u^h, \mu_v^h}^h V^h)(z) = \begin{cases} q^h(z) & \text{if } z \in Z_f^h \\ r^h(z, \mu_u^h(z), \mu_v^h(z)) \\ \quad + \gamma^h \sum_{z'} p^h(z, \mu_u^h(z), \mu_v^h(z), z') V^h(z') & \text{otherwise.} \end{cases}$$

Definition 11 The approximate maxmin DP operator \tilde{T}_δ^h is the combined δ -accurate rank-adaptive cross-approximation Π_δ and Bellman operator T^h , and it is defined as

$$\tilde{T}_\delta^h \tilde{V}^h = \Pi_\delta \left(T^h \tilde{V}^h \right).$$

The approximate operator has two main advantages: Firstly, it is applied directly to the previous value function iterate in FT-format. Hence, the value function never needs to be stored in its explicit full form, which results in a reduction of the required storage. Secondly, the cross-approximation algorithm requires only $\mathcal{O}(nlr^2)$ function evaluations. Consequently, the minmax optimization problem only needs to be solved for a subset of the nodes in Z^h , leading to a reduction in computational cost. This subset is selected by the cross-approximation algorithm at each iteration.

B. Best response iteration

At each value iteration the minmax optimization problem (5) must be solved for a subset of the nodes in Z^h . We apply a method called best response (BR) iteration by which updated control actions are iteratively calculated. The method is equivalent to best response dynamics with unit discretization step [21]. By Definition 12, any fixed point of BR iteration is a pure strategy equilibrium. Damped Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization is applied to solve the one-sided optimization problems in (6), see e.g., [23], for details.

Definition 12 The best response (also called optimal or rational response) mappings are as follows:

$$\begin{aligned} BR_u(z, v) &:= \operatorname{argmin}_{u \in \mathcal{U}} \bar{J}(z, u, v), \\ BR_v(z, u) &:= \operatorname{argmax}_{v \in \mathcal{V}} \bar{J}(z, u, v) \end{aligned} \quad (6)$$

with $\bar{J}(z, u, v) = r^h(z, u, v) + \gamma^h \sum_{z'} p^h(z, u, v, z') \tilde{V}^h(z')$. Note that these mappings may be set-valued.

C. Multigrid implementation

The value iteration algorithm is run iteratively in a grid refinement scheme. The multigrid algorithm progresses over a sequence of discretizations $h_i, i = 0, 1, \dots, N$, with $h_i > h_{i+1}$. Accompanying sequences of accuracy parameters δ^{h_i} and convergence criteria $\delta_V^{h_i}$ must be provided. For $h_i, i = 1, 2, \dots, N$, the initial value function is obtained based on the final iterate at the previous discretization level by means of interpolation operator \mathcal{I}_{i-1}^i , such that $\tilde{V}_0^{h_i} = \mathcal{I}_{i-1}^i \tilde{V}^{h_{i-1}}$ with $\tilde{V}^{h_{i-1}}$ the final iterate at h_{i-1} . The definition of the interpolation operator follows naturally from the piecewise linear uni-variate basis functions of the FT that are used in this work. The operator works directly on the matrix-valued

functions that comprise each core, as given in (4), and thus has computation cost linear in the number of dimensions. An overview of the algorithm is given below, where $\|\cdot\|_2$ indicates the function L2-norm.

Algorithm 1 FT-Based Minmax Value Iteration

Input: Initial value function in FT format $\tilde{V}_0^{h_0}$,
FT-cross accuracy parameters $\delta^{h_i}, i = 0, 1, \dots, N$,
Convergence criteria $\delta_V^{h_i}, i = 0, 1, \dots, N$,
1: **for** $i = 0, 1, \dots, N$ **do**
2: $k = 0$
3: **while** $\|\tilde{V}_k^{h_i} - \tilde{V}_{k-1}^{h_i}\|_2 > \delta_V^{h_i}$ **do**
4: $\tilde{V}_{k+1}^{h_i} = \tilde{T}_{\delta^{h_i}}^{h_i} \tilde{V}_k^{h_i}$
5: $k \leftarrow k + 1$
6: **end while**
7: **if** $i < N$ **then**
8: $\tilde{V}_0^{h_{i+1}} = \mathcal{I}_i^{i+1} \tilde{V}_k^{h_i}$
9: **end if**
10: **end for**
11: **return** $\tilde{V}_k^{h_N}$

V. ANALYSIS

In this section, we prove two major results: sup-norm error bounds for approximation of the value function, and the computational complexity of the algorithm per iteration.

A. Convergence

Before convergence of the approximate algorithm is addressed, we treat the sup-norm convergence of the exact Bellman operator T^h for the various differential game formulations.

Definition 13 Let $B(Z^h)$ indicate the set of all real-valued functions on Z^h , and let $\|\cdot\|_\infty$ indicate the supremum-norm, then a mapping $T^h : B(Z^h) \rightarrow B(Z^h)$ is a sup-norm contraction with modulus $\alpha \in (0, 1)$ if

$$\|T^h V^h - T^h V^{h'}\|_\infty \leq \alpha \|V^h - V^{h'}\|_\infty, \quad \forall V^h, V^{h'} \in B(Z^h).$$

For (TC), a policy pair μ_u, μ_v for which termination of the game occurs in finite time is considered proper, as opposed to improper policy pairs for which no termination occurs in finite time. We define the notion of *unilateral termination capability* as the capability by a player to terminate the game within finite time under any admissible adversarial policy.

Assumption 14 For (TC), we assume that either

- (i) all admissible policy pairs are proper, or
- (ii) at least one player has unilateral termination capability and incurs infinite cost under any improper policy.

Theorem 15 Minmax value iteration by mapping T^h is a sup-norm contraction for (IH) and (ST), and for (TC) under Assumption 14 (i).

Proof: The discount factor $\gamma^h \in (0, 1)$ is applied in (IH) under any adversarial policy. Therefore the standard discounted DP contraction proof holds and T^h is a sup-norm contraction mapping with modulus γ^h (see for example [24]).

In case of (ST) and (TC) under Assumption 14 (i), any policy is proper under any adversarial policy and thus the standard proof for undiscounted shortest path problems holds [24]. ■

Lemma 16 [24] *If the mapping T^h is a sup-norm contraction with modulus $\alpha \in (0, 1)$ that maps $B(Z^h)$ into itself, the following holds:*

- (i) V^{h*} is a unique fixed point of T^h such that $V^{h*} = T^h V^{h*}$,
- (ii) $T^{hk} V^h \rightarrow V^{h*}$ as $k \rightarrow \infty$ for any initial $V^h \in B(Z^h)$,
- (iii) $\|T^h V^h - V^{h*}\|_\infty \leq \alpha \|V^h - V^{h*}\|_\infty$ for any $V^h \in B(Z^h)$.

Lemma 16 presents a convergence rate for exact Markov game value iteration in case of (IH), (ST), and (TC) under Assumption 14 (i). In case only Assumption 14 (ii) holds for (TC), convergence of exact VI is still guaranteed, but the DP operation is not a contraction mapping [25].

The following lemma will prove useful in the analysis of the contractive properties of the FT-based Bellman operator \tilde{T}^h . Our major result regarding the convergence of the FT-based value iteration algorithm is given by Theorem 18.

Lemma 17 *Under Assumption 9, for Π_δ using accuracy parameter $\delta \in (0, l^{-2n}\epsilon)$ with $\epsilon \in (0, 1]$ the following holds:*

$$\|\Pi_\delta V^h - V^h\|_\infty \leq \epsilon \|V^h\|_\infty.$$

Proof: For a tensor V^h of l^n elements $\|V^h\|_\infty \leq \|V^h\|_F \leq \sqrt{nl} \|V^h\|_\infty$ by Cauchy-Schwartz. Using Assumption 9, we can now write: $\|\Pi_\delta V^h - V^h\|_\infty \leq \|\Pi_\delta V^h - V^h\|_F \leq \delta \|V^h\|_F \leq \delta \sqrt{l^n} \|V^h\|_\infty$. ■

Theorem 18 *Under Assumptions 1, 2, 4, 14(i) and 9, approximate value iteration \tilde{T}_δ^h on the MG value function \tilde{V}^h is able to converge to within arbitrary bounds of the exact value function V^* of the continuous differential game as $h \rightarrow 0$. The sup-norm error of its iterates is bounded by*

$$\begin{aligned} \|\tilde{V}_k^h - V^{h*}\|_\infty &\leq \epsilon \frac{1 - (\alpha + \alpha\epsilon)^k}{1 - (\alpha + \alpha\epsilon)} \|V^{h*}\|_\infty \\ &\quad + (\alpha + \alpha\epsilon)^k \|\tilde{V}_0^h - V^{h*}\|_\infty \end{aligned}$$

with V^{h*} the exact value function of the MG, ϵ the FT-cross accuracy parameter in the sup-norm, and α the sup-norm contraction modulus of the exact Bellman mapping T^h .

Proof: By Lemma 3, V^* exists and is unique. The mapping T^h is a sup-norm contraction with modulus $\alpha < 1$ by Theorem 15. We now use that by Lemma 16 under the exact mapping T^h $V_k^h \rightarrow V^{h*}$ as $k \rightarrow \infty$, which in turn converges to V^* as $h \rightarrow 0$ by Theorem 8. We also need that by Lemma 17 the parameter ϵ can be set arbitrarily small.

First we formulate a recursive bound:

$$\begin{aligned} &\|\tilde{V}_k^h - V^{h*}\|_\infty \\ &\leq \|\tilde{V}_k^h - T^h \tilde{V}_{k-1}^h\|_\infty + \|T^h \tilde{V}_{k-1}^h - V^{h*}\|_\infty \\ &\leq \epsilon \|T^h \tilde{V}_{k-1}^h\|_\infty + \|T^h \tilde{V}_{k-1}^h - V^{h*}\|_\infty \\ &\leq \epsilon \left(\|T^h \tilde{V}_{k-1}^h - V^{h*}\|_\infty + \|V^{h*}\|_\infty \right) \\ &\quad + \|T^h \tilde{V}_{k-1}^h - V^{h*}\|_\infty \\ &= (1 + \epsilon) \|T^h \tilde{V}_{k-1}^h - V^{h*}\|_\infty + \epsilon \|V^{h*}\|_\infty \\ &\leq (1 + \epsilon) \alpha \|\tilde{V}_{k-1}^h - V^{h*}\|_\infty + \epsilon \|V^{h*}\|_\infty, \end{aligned} \tag{7}$$

where the first and third inequalities follow from the triangle inequality, the second inequality follows from Lemma 17, and the fourth inequality follows from the contraction property of the exact Bellman operator T^h , i.e., Theorem 15. Similarly, for $k = 1$ we obtain

$$\|\tilde{V}_1^h - V^{h*}\|_\infty \leq (1 + \epsilon) \alpha \|\tilde{V}_0^h - V^{h*}\|_\infty + \epsilon \|V^{h*}\|_\infty. \tag{8}$$

Combining equations (7) and (8), we obtain the following direct formulation:

$$\begin{aligned} &\|\tilde{V}_k^h - V^{h*}\|_\infty \\ &\leq \epsilon \|V^{h*}\|_\infty \sum_{i=0}^k (\alpha(1 + \epsilon))^i + (\alpha(1 + \epsilon))^k \|\tilde{V}_0^h - V^{h*}\|_\infty \\ &= \epsilon \frac{1 - (\alpha + \alpha\epsilon)^k}{1 - (\alpha + \alpha\epsilon)} \|V^{h*}\|_\infty + (\alpha + \alpha\epsilon)^k \|\tilde{V}_0^h - V^{h*}\|_\infty. \end{aligned}$$

Note that for $\epsilon = 0$ the conventional α^k bound for exact value iteration is recovered. In order to guarantee the algorithm does not diverge the condition $\alpha + \alpha\epsilon < 1$ must hold. In this case the contractive property is able to bound the accumulated error introduced by the FT approximation. ■

B. Complexity

The computational complexity of a single value iteration (line 4 in Algorithm 1) is given by Theorem 19.

Theorem 19 *Consider a n -dimensional FT value function \tilde{V}_i^h , where h represents a regular MG discretization with l nodes per dimension. Suppose evaluation of the objective function of (5) requires p operations if $\tilde{V}_i^h(z')$ are provided, and must be performed m times in order to obtain the saddle point. Then, $\tilde{V}_{i+1}^h = \tilde{T}_\delta^h \tilde{V}_i^h = \Pi_\delta T^h \tilde{V}_i^h$ is obtained in*

$$\mathcal{O}(nlr^2(mp + n^2lr^2) + nlr^3)$$

operations.

Proof: The FT-rankadapt-cross algorithm represented by operator Π_δ requires $\mathcal{O}(nlr^2)$ function evaluations, which each require m times p operations after calculation of $V(z')$. Using the proposed interpolation scheme $V(z')$ must be calculated for $2n$ points at a cost of $\mathcal{O}(nlr^2)$ for each of these FT evaluations. Finally, FT-rankadapt-cross itself requires $\mathcal{O}(nlr^3)$ operations. ■

Note that by Theorem 19 the complexity scales polynomially with the game dimension n , which lifts the curse of dimensionality as long as the rank does not increase exponentially

with the dimension, *i.e.*, the value function admits an accurate low-rank approximation. Apart from specific cases, *e.g.*, linear-quadratic differential games, estimation of the value function rank is very challenging and subject of current research. Here, this is addressed by the rank-adaptive nature of the cross-approximation used. However, computational cost may become prohibitive as rank increases. Setting a rank bound, $r \leq r_{max}$, results in a quasi-optimal truncated value function $\tilde{V}_{i+1}^h = \operatorname{argmin}_{\tilde{V}^h} \|\tilde{V}^h - \Pi_\delta T^h \tilde{V}_i^h\|_F$ subject to $\operatorname{rank}(\tilde{V}^h) \leq r_{max}$. By Theorem 19, the rank bound also bounds the number of computations per iteration.

VI. COMPUTATIONAL RESULTS

In this section, we present computational results for two nonlinear differential game scenarios: the well-known Homicidal Chauffeur game, and an aerial pursuit scenario involving a quadrotor and a 3D Dubins car. We make use of the FT computational algebra in the C3 toolbox and the framework for DP provided by the C3SC toolbox [8], [27]. All computations were performed on a PC with Intel i9-7900X CPU @ 3.30GHz. The algorithm was run in parallel by distributing the computation over 20 threads.

A. Homicidal Chauffeur

The Homicidal Chauffeur game was introduced by Isaacs in 1965. In this game of cost structure (TC), a homicidal chauffeur tries to run over a pedestrian as quickly as possible. While the chauffeur moves at higher speed, he is limited in his motion by the turning radius of his vehicle. The pedestrian on the other hand can turn instantly. The scenario can be expressed in normalized relative coordinates fixed to the chauffeur, leading to the following differential equations:

$$\dot{x}_1 = -x_2 u + v_1 \sin(v_2), \quad \dot{x}_2 = x_1 u + v_1 \cos(v_2) - 1$$

with x the relative position of the pedestrian, $v_1 \in [0, \frac{1}{4}]$ the pedestrian speed, $v_2 \in [-\pi, \pi]$ the pedestrian direction, and $u \in [-1, 1]$ the pursuer steering input. The termination set is $X_f = \{x | \sqrt{x_1^2 + x_2^2} < 0.4\}$, and the corresponding termination cost is set to zero. The running cost is set to unity, so that the pay-off is equal to the termination time.

FT-based DP is applied to a discretization with 41 nodes from -8 to 8 along each dimension. The value function found has rank 8 and is shown in Fig. 1. Comparison to solutions from literature, *e.g.*, [1], [28], shows very good agreement. Although the value function is only two dimensional, the compressed representation reduces the required storage from 13.4 kB for the full tensor to 5.2 kB for the FT. On average, the Bellman equation is evaluated at 35% of nodes per iteration, reducing the computation time to 23 seconds.

Note that Assumption 4 does not hold for the Homicidal Chauffeur game, and in fact the analytical value function is discontinuous. It is noteworthy that even at a relatively low rank, the FT-approximation is able to capture the discontinuity in the value function by a sharp local gradient. Also note that only (ii) of Assumption 14 applies to this game. Therefore the contractive property of exact DP does not hold, and FT-based DP is not guaranteed to converge under Theorem

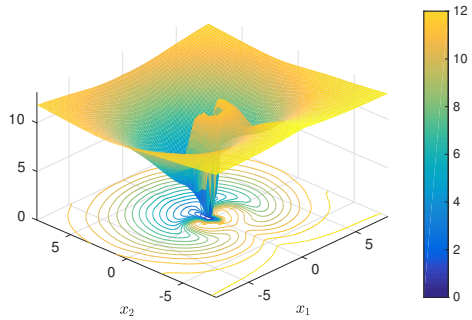


Fig. 1. Value function for Homicidal Chauffeur.

18. Despite the fact that Homicidal Chauffeur evidently does not fit within the framework for which analytical guarantees exist, the computational results indicate the resulting solution is close to the analytical solution.

B. Aerial Pursuit

The second example is a high-dimensional differential game with seven state variables. A pursuer with quadrotor kinematics minimizes the distance to an evader with 3D Dubins car kinematics. The game is formulated in relative states with the origin fixed to the evader, leading to the following equations:

$$\begin{aligned} \dot{x}_1 &= V_{x_1} - V_e \cos \chi \\ \dot{x}_2 &= V_{x_2} + V_e \sin \chi \\ \dot{x}_3 &= V_{x_3} - v \dot{x}_3 \\ \dot{V}_{x_1} &= \frac{u_T - mg}{m} \cos u_\phi \sin u_\theta \\ \dot{V}_{x_2} &= -\frac{u_T - mg}{m} \sin u_\phi \\ \dot{V}_{x_3} &= \frac{u_T - mg}{m} \cos u_\phi \cos u_\theta + g \\ \dot{\chi} &= v \dot{\chi} \end{aligned}$$

where x_1, x_2 , and x_3 are the relative position of the pursuer; V_{x_1}, V_{x_2} , and V_{x_3} are the corresponding pursuer velocity components, and χ is the evader heading. The parameters m, g , and V_e – corresponding to pursuer mass, gravitational acceleration, and evader horizontal speed – are set to 0.8, 9.81, and 1, respectively. The pursuer control inputs $u_T \in [-1.5, 1.5]$, $u_\phi \in [-0.4, 0.4]$, and $u_\theta \in [-0.4, 0.4]$, correspond to thrust increment, roll angle, and pitch angle, respectively. The evader control inputs are its vertical speed $v \dot{x}_3 \in [-0.3, 0.3]$, and heading rate $v \dot{\chi} \in [-5, 5]$. The game has cost structure (IH) with $\beta = 0.2$ and running cost function $g(x, u, v) = 12x_1^2 + 12x_2^2 + 12x_3^2 + 2u_T^2 + u_\phi^2 + 6u_\theta^2 - 4v_{x_3}^2 - 3v_{\dot{\chi}}^2$.

The assumptions underlying Theorem 18 are satisfied, hence FT-based DP is able to converge to within arbitrary bounds of the exact value function. The algorithm is run on the domain $X = [-3.5, 3.5] \times [-3.5, 3.5] \times [-2, 2] \times [-5, 5] \times [-5, 5] \times [-5, 5] \times \mathbb{T}$, where \mathbb{T} denotes the circle group.

The multi-grid algorithm is initialized with 20 gridpoints along each dimension, *i.e.*, $n = 20$, and progresses to $n =$

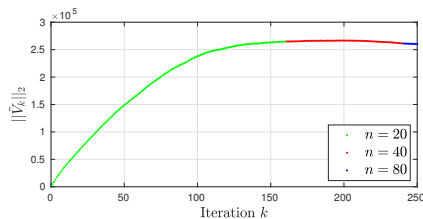


Fig. 2. Convergence of value function for Aerial Pursuit.

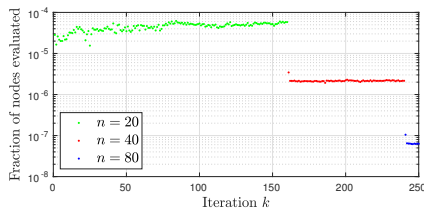


Fig. 3. Fraction of nodes evaluated for Aerial Pursuit.

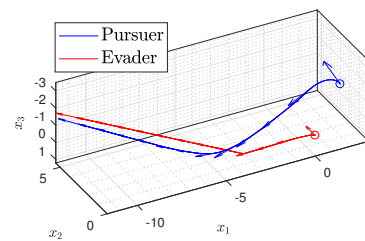


Fig. 4. Optimal trajectory for Aerial Pursuit. Velocity vectors are drawn every time unit.

80. The convergence is shown in Fig. 2 in terms of the norm of the value function.

Figure 3 shows that at $n = 80$, the cross-approximation evaluates the Bellman equation at only about one in 10^7 nodes; leading to seven orders of magnitude decrease in computational cost per iteration. Each iteration at $n = 80$ was performed in about 12 minutes, whereas naive VI would theoretically take over 200 years per iteration. Total computation time for all grids was eight hours. The FT format is a naturally compressed representation of the value function; storing the final value function ($r = 20$) requires 1.31 MB, whereas storing the 80^7 tensor in full format would require 168 TB ($1.68 \cdot 10^8$ MB).

Optimal trajectories, such as the one shown in Fig. 4, show the pursuer initially applying a large acceleration to catch up with the evader. Once the pursuer closes in on the evader the control effort becomes more conservative, while the distance remains small. The evader alternates its steering direction so that the pursuer's overshoot is maximized.

VII. CONCLUSION

Analysis and computational results of a novel differential games dynamic-programming-based algorithm that uses functional tensor-train approximation were presented. The algorithm uses computational resources that grow polynomially with dimensionality and with the rank of the value function and is thereby able to alleviate the curse of dimensionality. The proposed algorithm is shown to converge to optimal solutions with arbitrary bounds for a certain class of differential games. It is demonstrated in several problem instances; in case of a seven-dimensional game the value function representation was obtained with seven orders of magnitude savings in computational and memory cost, when compared to standard value iteration.

VIII. ACKNOWLEDGMENTS

This work was partly supported by Lockheed Martin. Their support is gratefully acknowledged.

REFERENCES

- [1] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. John Wiley and Sons, Inc., New York, 1965.
- [2] T. Başar and G. Olsder, *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, 1999.
- [3] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, 2011.
- [4] M. Falcone, "Numerical methods for differential games based on partial differential equations," *International Game Theory Review*, vol. 8, no. 02, pp. 231–272, 2006.
- [5] M. Chen, S. Herbert, and C. J. Tomlin, "Exact and efficient Hamilton-Jacobi-based guaranteed safety analysis via system decomposition," *arXiv preprint arXiv:1609.05248*, 2016.
- [6] J. Darbon and S. Osher, "Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere," *Research in the Mathematical Sciences*, vol. 3, no. 1, 2016.
- [7] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [8] A. A. Gorodetsky, *C3: Compressed Continuous Computation library*. <https://github.com/goroda/Compressed-Continuous-Computation>, 2016.
- [9] A. A. Gorodetsky, S. Karaman, and Y. M. Marzouk, "High-dimensional stochastic optimal control using continuous tensor decompositions," *arXiv preprint arXiv:1611.04706*, 2016.
- [10] H. J. Kushner, "Numerical approximations for stochastic differential games," *SIAM Journal on Control and Optimization*, vol. 41, no. 2, pp. 457–486, 2002.
- [11] N. V. Krylov, "On the dynamic programming principle for uniformly nondegenerate stochastic differential games in domains and the Isaacs equations," *Probability Theory and Related Fields*, vol. 158, no. 3, pp. 751–783, 2014.
- [12] W. H. Fleming, "The convergence problem for differential games," *Journal of Mathematical Analysis and Applications*, vol. 3, no. 1, pp. 102 – 116, 1961.
- [13] E. Barron, L. Evans, and R. Jensen, "Viscosity solutions of Isaacs' equations and differential games with Lipschitz controls," *Journal of Differential Equations*, vol. 53, no. 2, pp. 213–233, 1984.
- [14] A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*. Springer-Verlag, Berlin, 2008.
- [15] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [16] I. V. Oseledets, "Constructive representation of functions in low-rank tensor formats," *Constructive Approximation*, vol. 37, pp. 1–18, December 2012.
- [17] A. A. Gorodetsky, S. Karaman, and Y. M. Marzouk, "Function-train: a continuous analogue of the tensor-train decomposition," *arXiv preprint arXiv:1510.09088*, 2016.
- [18] D. Bigoni, A. P. Engsig-Karup, and Y. M. Marzouk, "Spectral Tensor-Train decomposition," *SIAM Journal on Scientific Computing*, vol. 38, pp. A2405–A2439, January 2016.
- [19] I. V. Oseledets and E. Tyrtshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra and its Applications*, vol. 432, pp. 70–88, January 2010.
- [20] S. Hart and A. Mas-Colell, "Regret-based continuous-time dynamics," *Games and Economic Behavior*, vol. 45, no. 2, pp. 375–394, 2003.
- [21] J. Hofbauer and S. Sorin, "Best response dynamics for continuous zero-sum games," *Discrete and Continuous Dynamical Systems Series B*, vol. 6, no. 1, pp. 215–224, 2006.
- [22] T. Vincent, B. Goh, and K. Teo, "Trajectory-following algorithms for min-max optimization problems," *Journal of Optimization Theory and Applications*, vol. 75, no. 3, pp. 501–519, 1992.
- [23] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer-Verlag, New York, 2006.
- [24] D. P. Bertsekas, *Dynamic Programming and Optimal Control Part II: Approximate Dynamic Programming*. Athena Scientific, Belmont, MA, 2012.
- [25] D. P. Bertsekas, "Robust shortest path planning and semicontractive dynamic programming," *Naval Research Logistics*, 2016.

- [26] D. P. Bertsekas, *Abstract Dynamic Programming*. Athena Scientific, Belmont, MA, 2013.
- [27] A. A. Gorodetsky, *C3SC: Stochastic Optimal Control with Compressed Continuous Computation*. <https://github.com/goroda/c3sc>, 2016.
- [28] V. S. Patsko and V. L. Turova, “Homicidal chauffeur game: History and modern studies,” in *Advances in Dynamic Games*, pp. 227–251, Birkhäuser, Boston, 2011.