Encoding Defensive Driving as a Dynamic Nash Game†

Chih-Yuan Chiu,* David Fridovich-Keil,* and Claire J. Tomlin

Abstract— **We present a novel formulation of safety and robustness for autonomous systems within a general-sum dynamic game theory framework, based on defensive driving. Specifically, we encode an *adversarial phase* into the ego agent's cost function, i.e., a time interval in which other agents are assumed to be temporarily distracted, to robustify the ego agent's equilibrium trajectory against other agents' potentially dangerous behavior in this time. We illustrate that our new formulation effectively encodes safety in multiple traffic scenarios.**

I. Introduction

In designing autonomous systems, practitioners typically employ one of two prevailing notions of "safety" and "robustness:" *adversarial* and *probabilistic* constraint satisfaction. In this work, we introduce a third, distinct notion of safety. Like adversarial formulations, our work is based upon noncooperative differential game theory; however, unlike such methods, our approach is not equivalent to a single two-player zero-sum differential game, and naturally extends to an arbitrary number of agents.

Our work is based upon the literature in differential game theory and adversarial reachability [1–5]. Adversarial reachability methods [6–9] seek to identify when a system's state can be driven, despite worst-case bounded disturbance, toward one set and away from another. However, in our case, we study safety and robustness in the context of differential games with an arbitrary number of agents. We divide the time horizon into two parts: an *adversarial* part followed by a *cooperative* part. During the adversarial portion, the ego agent presumes that other agents wish to harm it and encodes such behavior in the cost structure of the game, and during the cooperative portion of the time horizon it presumes that other agents will try to help it (e.g., to avoid collision).

II. Related Work

A. Adversarial Reachability

Adversarial reachability methods [6–9] construct a zero-sum differential game between two agents, and appropriately describes many dynamic interactions, e.g., capture-the-flag and reach-avoid games [1, 2]. However, this zero-sum formulation is inadequate for motion-planning tasks, e.g., traffic scenarios with multiple interacting vehicles. Moreover, the adversarial nature of the zero-sum game leads to the construction of extremely conservative ego trajectories, since the ego agent must imagine the worst-case non-ego behaviors that can possibly transpire. Our approach, on the other hand, considers a general-sum game applicable to N-player scenarios, and avoids considering purely adversarial non-ego trajectories. That is, we model antagonistic non-ego behavior using the novel notion of an adversarial-to-cooperative time horizon, rather than as a worst-case bounded disturbance.

B. Probabilistic Constraint Satisfaction

In motion planning, probabilistic constraint satisfaction approaches bound the probability that an ego agent, operating in an environment with stochastic disturbances, becomes unsafe [10]. In particular, risk-sensitive algorithms guard the ego agent from low-probability, yet highly dangerous outcomes, e.g., by using exponential-quadratic cost terms [11] or by associating individual constraint violations with different penalties [12]. However, these methods merely account for the nonzero probability of unsafe outcomes occurring any time within the entire time horizon. By contrast, our work allows the ego to explicitly encode adversarial non-ego behavior inside a specific subset of the time horizon, when such behavior is most expected to occur.

C. Algorithms for Solving Dynamic Games

General-sum differential games can be directly solved by numerically solving a set of coupled Hamilton-Jacobi equations, whose solutions yield Nash equilibrium (NE) strategies [4, 5], via state space discretization. However, the computational cost and memory of these algorithms scale exponentially with the state dimension, and are thus unsuitable for modeling the high-dimensional, multi-player interactions considered in our paper [13]. On the other hand, Iterative Best Response (IBR) algorithms [14, 15] iterate through the players, repeatedly solving the optimal control problem of finding the best-response strategy of each player, assuming all other players' strategies are currently fixed at the previous iterate. Replacing the full dynamic game with a sequence of optimal control problems reduces computation time at each iteration; however, IBR algorithms can still be computationally inefficient overall. Moreover, IBR algorithms often converge only to a local Nash equilibrium (LNE), or fail to converge.

*The first two authors contributed equally to this work.

†The full conference version of this paper was presented at 2021 ICRA (https://arxiv.org/abs/2011.04815).

C-Y Chiu and C Tomlin are with the Dept. of Electrical Engineering & Computer Sciences, UC Berkeley. D Fridovich-Keil is with the Dept. of Aerospace Engineering & Engineering Mechanics, UT Austin. Correspondence to dfk@utexas.edu.

This research is supported by an NSF CAREER award, the Air Force Office of Scientific Research (AFOSR), NSF's CPS FORCES and VeHICaL projects, the UC-Philippine-California Advanced Research Institute, the ONR MURI Embedded Humans, a DARPA Assured Autonomy grant, the SRC CONIX Center, NSF grant DMS 2013985 "THEORINet: Transferable, Hierarchical, Expressive, Optimal, Robust and Interpretable Networks," and U.S. Office of Naval Research MURI grant N00014-16-1-2710..

To iteratively solve linear-quadratic games, our work uses ILQGames [16], a recently developed iterative linear-quadratic algorithm that incurs computational complexity cubic in the number of players and linear in the time horizon, and is guaranteed to rapidly converge to an LNE [17].

III. Preliminaries

Consider the N-player finite horizon general-sum differential game with deterministic nonlinear system dynamics:

$$\dot{x} = f(t, x, u_{1:N}). \tag{1}$$

Here, $x \in \mathbb{R}^n$ is the state of the system, obtained by concatenating the dynamical quantities of interest of each player, $t \in \mathbb{R}$ denotes time, $u_i \in \mathbb{R}^{m_i}$ is the control input of player i, for each $i \in \{1, \cdots, N\} := [N]$, and $u_{1:N} := (u_1, \cdots, u_N) \in \mathbb{R}^m$, where $m := \sum_{i=1}^N m_i$. The dynamics map $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be continuous in t and continuously differentiable in x and u_i, for each $i = 1, \cdots, N$ and each $t \in [0, T]$. Since we wish to ensure the safety of one particular player amidst their interactions with all other players, we refer to Player 1 as the *ego agent*, and the other players as *non-ego agents*. Each player's objective is defined as the integral of a running cost $g_i : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ over the time horizon $[0, T]$:

$$J_i\big(u_{1:N}(\cdot)\big) = \int_0^T g_i\big(t, x(t), u_{1:N}(t)\big)\, dt, \tag{2}$$

for each $i \in \{1, \cdots, N\}$. The running costs g_i depend implicitly on the state trajectory $x(\cdot) : [0, T] \to \mathbb{R}^n$ and explicitly on the control signals $u_i(\cdot) : [0, T] \to \mathbb{R}^m$.

To minimize its cost, each player selects a control strategy to employ over the time horizon $[0, T]$. At each time $t \in [0, T]$, each player i observes the state $x(t)$ (but no other control input $\{u_j(t) \mid j \neq i\}$), and uses this information to design its control, i.e.

$$u(t) := \gamma_i(t, x(t)),$$

where $\gamma_i : [0, T] \times \mathbb{R}^n \to \mathbb{R}^{m_i}$, defined as Player i's *strategy*, is measurable. The *strategy space of Player* i, denoted Γ_i, is defined as the collection of all feasible strategies of Player i's. The overall cost J_i of each Player i is denoted by:

$$J_i(\gamma_1; \cdots; \gamma_N) := J_i\big(\gamma_1(\cdot, x(\cdot)), \cdots, \gamma_N(\cdot, x(\cdot))\big).$$

In practice, we solve for strategies γ_i that are time-varying, affine functions of x.

We now define the *Nash equilibrium* of the above game.

Definition 1: (Nash equilibrium, [3, Ch. 6]) The strategy set $(\gamma_1^\star, \cdots, \gamma_N^\star)$ is called a Nash equilibrium if no player is unilaterally incentivized to deviate from his or her strategy. Precisely, the following inequality, i.e., for each player i:

$$\begin{aligned} J_i^\star &:= J_i\left(\gamma_1^\star, \ldots, \gamma_{i-1}^\star, \gamma_i^\star, \gamma_{i+1}^\star, \ldots, \gamma_N^\star\right) \\ &\leq J_i\left(\gamma_1^\star, \ldots, \gamma_{i-1}^\star, \gamma_i, \gamma_{i+1}^\star, \ldots, \gamma_N^\star\right), \forall \gamma_i \in \Gamma_i. \end{aligned} \tag{3}$$

Computing a global Nash equilibrium is intractable for dynamic games with general dynamics and cost functions. Instead, in this work, we seek a *generalized local* Nash equilibrium, which is defined similarly to (3), but with the inequalities only constrained to hold within a neighborhood of the strategy set $(\gamma_1^\star, \cdots, \gamma_N^\star)$, and with additional constraints imposed on each player. These constraints model appropriate vehicular behavior in traffic scenarios.

IV. Methods

Our main contribution is a novel formulation of safety, best understood through the lens of defensive driving. In Sec. IV-A, we describe how, in the ego agent's mind, the concept of defensive driving can be encoded into the running cost of each non-ego agent, i.e. $g_i(x, u_{1:N})$, for each $i \in \{2, \cdots, N\}$. To demonstrate this defensive driving framework in practice, we simulate realistic traffic scenarios; Sec. IV-B details the dynamics, costs, and constraints imposed on the various agents in these simulations. Finally, in Sec. IV-C, we summarize the ILQGames algorithm as the main feedback game solver used in this work.

A. Encoding Defensive Driving as a Running Cost

In our framework, the ego agent (Player 1) encodes the assumption that all other agents are momentarily distracted, by imagining the overall time horizon $[0, T]$ as divided into two sub-intervals: *adversarial* ($[0, T_{\text{adv}}]$) and *cooperative* ($[T_{\text{adv}}, T]$), with $0 < T_{\text{adv}} < T$. During the adversarial interval, the ego imagines other agents to be "momentarily distracted," and wishes to act *defensively*. This phenomenon is modeled using an adversarial running cost $g_{\text{adv},i} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for each $i \in \{2, \cdots, N\}$. During the cooperative interval, the ego supposes that other agents have reverted to "normal" or "cooperative" behaviors, and thus proceeds in a less conservative manner. This behavior is captured using a cooperative running cost $g_{\text{coop},i} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for each $i \in \{2, \cdots, N\}$. In other words, the running cost of each non-ego agent g_i can be piecewise defined as follows:

$$g_i(t, x, u_{1:N}) = \begin{cases} g_{\text{adv},i}(x, u_{1:N}), & t \in [0, T_{\text{adv}}), \\ g_{\text{coop},i}(x, u_{1:N}), & t \in [T_{\text{adv}}, T]. \end{cases}$$

In this scenario, the net integrated cost J_i, first defined in (2) can be written as follows:

$$J_i = \int_0^{T_{\text{adv}}} g_{\text{adv},i}(x, u_{1:N}) dt + \int_{T_{\text{adv}}}^T g_{\text{coop},i}(x, u_{1:N}) dt\,. \tag{4}$$

With increasing T_{adv}, the ego agent imagines an increasingly adversarial encounter and acts more and more defensively as a result. In practice, the user or system designer would select a suitable T_{adv} before operation, e.g., by choosing the largest T_{adv} such that the solution deviates sufficiently little from a nominal solution (with $T_{\text{adv}} = 0$).

B. Simulation Setup

To test this construction, we simulate two traffic encounters in ILQGames [16] that involve significant interaction (see Sec. V), in which a responsible human driver would likely drive defensibly. Our method captures the spectrum of this "defensive" behavior as T_{adv}, the adversarial time horizon duration, is varied. In each setting, each agent (in this case, each car) has augmented bicycle dynamics, i.e.:

$$\begin{aligned} \dot{p}_{x,i} &= v_i \sin\theta_i, & \dot{v}_i &= a_i, \\ \dot{p}_{y,i} &= v_i \cos\theta_i, & \dot{\phi}_i &= \omega_i, \\ \dot{\theta}_i &= (v_i/L_i)\tan\phi_i, & \dot{a}_i &= j_i, \end{aligned} \tag{5}$$

where $x = (p_{x,i}, p_{y,i}, \theta_i, v_i, \phi_i, a_i)_{i=1}^N$ describes the position, heading, speed, front wheel angle, and acceleration of all vehicles, and for each, $u_i = (\omega_i, j_i)$ describes the front wheel rate and tangent jerk, while L_i is the inter-axle distance.

We define $g_{\text{adv},i}$ and $g_{\text{coop},i}$ as weighted combinations of the following functions, with different behavior encouraged through the use of different weighting coefficients. We denote $p_i = (p_{x,i}, p_{y,i})$ for each agent's position, $d_{\ell_i}(p_i)$, defined below, for the distance between an agent and the corresponding lane centerline ℓ_i in the $(p_{x,i}, p_{y,i})$-plane, and d_{prox} for a constant desired minimum proximity between agents:

$$\text{lane center: } \left[d_{\ell_i}(p_i) := \min_{p_\ell \in \ell_i} \|p_\ell - p_i\|\right]^2 \tag{6}$$

$$\text{ideal speed: } (v_i - v_{\text{ref},i})^2 \tag{7}$$

$$\text{cooperative: } \mathbf{1}\{\|p_i - p_j\| < d_{\text{prox}}\}(d_{\text{prox}} - \|p_i - p_j\|)^2 \tag{8}$$

$$\text{adversarial: } \|p_i - p_j\|^2 \tag{9}$$

$$\text{input: } u_i^T R_{ii} u_i\,. \tag{10}$$

Recall that, for non-ego agents, the "adversarial" cost is only present during the adversarial horizon $[0, T_{\text{adv}})$ and the "cooperative" cost is present thereafter during the cooperative horizon $[T_{\text{adv}}, T]$. We also enforce the following inequality constraints, where d_{lane} denotes the lane half-width, and $\underline{v}_i$ and $\overline{v}_i$ denote speed limits:

$$\text{proximity: } \|p_i - p_j\| > d_{\text{prox}} \tag{11}$$

$$\text{lane: } |d_{\ell_i}(p_i)| < d_{\text{lane}} \tag{12}$$

$$\text{speed range: } \underline{v}_i < v_i < \overline{v}_i\,, \tag{13}$$

Here, the "proximity" constraint is enforced for only the ego agent, to force the ego to bear responsibility for satisfying joint state constraints which encode his or her own safety (e.g., non-collision). In addition, all agents must satisfy individual constraints that encode reasonable conduct in traffic (e.g., staying within a range of speeds). All constraints are enforced over the entire time horizon $[0, T]$. For all tests, we use a time horizon $T = 15\,\text{s}$ and discretize time (following [16] and [3]) at $0.1\,\text{s}$ intervals.

C. Implementation Details

The traffic simulations in this work are solved approximately to local feedback Nash equilibria in real time using ILQGames, an open-source C++-based game-solving algorithm introduced recently in [16]. ILQGames iteratively solves linear-quadratic games, obtained by linearizing dynamics and quadraticizing costs, and incurs computational complexity cubic in the number of players [16]. As discussed above, we must also account for equality and inequality constraints on the game trajectory. While [16] does not address constrained Nash games, here we incorporate constraints via augmented Lagrangian methods [18]. For a more detailed discussion of constraint-handling in feedback Nash games, see [17]. Though other game solvers, e.g., ALGAMES [19] and Iterative Best Response algorithms [15], also handle constraints, they only apply to open-loop games. For a thorough treatment of constraints in games, see [20].

V. Results

We present simulation results for various traffic scenarios in which a responsible traffic participant would likely drive defensively. First, we consider a simple situation involving oncoming vehicles on a straight road, as a proof of concept. Then, we analyze a more complicated intersection example with a crosswalk. In both cases, the ILQGames algorithm solves the defensive driving game quickly, in under $1\,\text{s}$.

A. Oncoming Example

In this example, the ego car is traveling North on a straight road when it encounters another car traveling South. Since the road has a lane in each direction, "ideally" the ego vehicle would not deviate too far from its lane or speed. However, to drive more defensively, the ego vehicle should plan as though the oncoming Southbound car were to act noncooperatively. Our method encodes precisely this type of defensive planning. Fig. 1 shows the planned trajectories that emerge for increasing T_{adv}. As shown, the ego vehicle (bottom) imagines more aggressive maneuvers for itself and the oncoming car (top) as T_{adv} increases. Note, however, that these are merely *imagined* trajectories and that (a) the ego vehicle can always choose to follow this trajectory only for an initial period of time, and recompute its trajectory thereafter with updated state information, and (b) the oncoming vehicle will make its own decisions and will *not* generally follow this "partially adversarial" trajectory. We solve each of these problems (with fixed T_{adv}) in under $0.5\,\text{s}$.

B. Three-Player Intersection Example

We introduce a more complicated scenario designed to model the behavior of two vehicles and a pedestrian at an intersection. As shown in Fig. 2, the ego vehicle is present in the intersection alongside a non-ego vehicle heading in the opposite direction, who wishes to make a left turn, and a pedestrian, who wishes to cross the road. To reach their goal locations, these three agents must cross paths in the intersection. When $T_{\text{adv}} = 0$ s, the ego vehicle swerves left to avoid the non-ego vehicle at the intersection, because it expects the non-ego (turning) vehicle to always behave cooperatively, i.e., continue along its curved path at nominal speeds, resulting in a collision-free trajectory. However, as with the oncoming example, the ego vehicle's trajectory becomes increasingly more conservative as the adversarial time horizon increases in length. In particular, when $T_{\text{adv}} = 0.5\,\text{s}$, the ego slows to avoid the turning vehicle; when $T_{\text{adv}} = 1\,\text{s}$, the ego accelerates aggressively to speed ahead of the non-ego vehicle. This is because in this scenario, the oncoming vehicle is initially slower than the ego vehicle, and

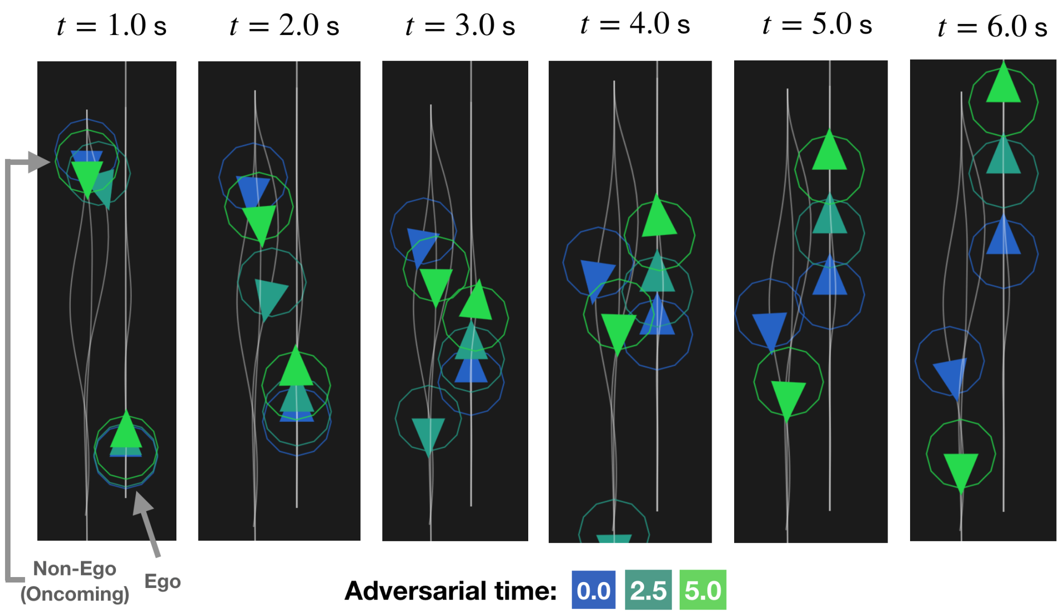

Fig. 1: Oncoming example. The ego (right lane, heading upwards) and oncoming (left lane, heading downwards) vehicles perform increasingly extreme maneuvers as T_{adv} increases. Blue, turquoise, and green represent agents' locations for $T_{\text{adv}} = 0, 2.5, 5$ s, respectively. Panels show agent positions as time elapses. When $T_{\text{adv}} = 0$ s, the ego travels in a straight line because it expects the non-ego vehicle to always be cooperative. However, when $T_{\text{adv}} = 5$ s, the ego swerves outward to dissuade the non-ego vehicle from attempting a collision.

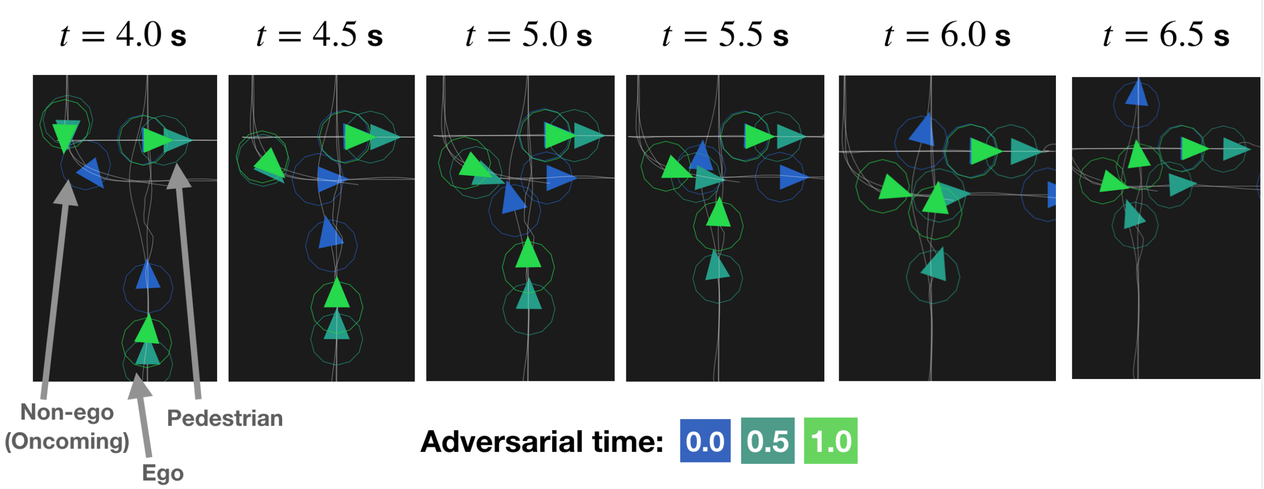

Fig. 2: Three Player Intersection example. The ego agent (right lane, heading upwards) navigates an intersection while avoiding collision with an oncoming vehicle (left lane, heading downwards initially before making a left turn) and a pedestrian (horizontal path at the intersection, left to right). Blue, turquoise, and green represent agents' locations at $T_{\text{adv}} = 0, 0.5, 1$ s, respectively. When $T_{\text{adv}} = 0$ s, the ego expects the non-ego to travel at nominal speeds and approach the intersection first, and thus swerves leftwards, to avoid a collision. However, when $T_{\text{adv}} = 1$ s, the ego vehicle accelerates and swerves rightwards to avoid the non-ego vehicle, to dissuade the oncoming vehicle from attempting a collision. The pedestrian also slows for the same reason.

will thus approach the intersection at the same time as the ego vehicle. Each problem is solved in 0.75 s in single-threaded operation on a standard laptop, via the ILQGames algorithm [16]. This performance indicates real-time capabilities which will be explored in future work on hardware.

VI. Discussion

We present a novel formulation of robustness in motion planning for multi-agent problems. Inspired by defensive driving, our method explicitly models other agents as adversarial in only a limited, initial portion of the overall planning interval. Thus, our approach generates far less conservative behavior than purely adversarial methods. Simulation results illustrate that these "defensive" problems can be solved in real-time. Future work includes modeling human intent and preferences both via players' cost functions and the adversarial horizon T_{adv} itself [21]. This will enable our framework to better respond to human behaviors in various autonomous driving scenarios [22, 23].

References

[1] Rufus Isaacs. *Games of pursuit*. Tech. rep. Rand Corporation, 1951.

[2] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[3] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1999.

[4] Alan Wilbor Starr and Yu-Chi Ho. "Nonzero-sum differential games". In: *Journal of optimization theory and applications* 3.3 (1969), pp. 184–206.

[5] Alan Wilbor Starr and Yu-Chi Ho. "Further properties of nonzero-sum differential games". In: *Journal of Optimization Theory and Applications* 3.4 (1969), pp. 207–219.

[6] L. C. Evans and P. E. Souganidis. "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations". In: *Indiana University mathematics journal* 33.5 (1984), pp. 773–797.

[7] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[8] Jaime F Fisac and S Shankar Sastry. "The pursuit-evasion-defense differential game in dynamic constrained environments". In: *54th Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 4549–4556.

[9] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. "Hamilton-Jacobi reachability: A brief overview and recent advances". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[10] M. Ono, L. Blackmore, and B. C. Williams. "Chance constrained finite horizon optimal control with non-convex constraints". In: *Proceedings of the 2010 American Control Conference*. 2010, pp. 1145–1152.

[11] Farbod Farshidian and J. Buchli. "Risk Sensitive, Nonlinear Optimal Control: Iterative Linear Exponential-Quadratic Optimal Control with Gaussian Noise". In: *ArXiv* abs/1512.07173 (2015).

[12] Farbod Farshidian and J. Buchli. "Chance-Constrained Optimal Covariance Steering with Iterative Risk Allocation". In: *ArXiv* abs/1512.07173 (2015).

[13] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Two Volume Set*. 2nd. Athena Scientific, 2001. ISBN: 1886529086.

[14] Jaime F Fisac, Eli Bronstein, Elis Stefansson, Dorsa Sadigh, S Shankar Sastry, and Anca D Dragan. "Hierarchical Game-Theoretic Planning for Autonomous Vehicles". In: *arXiv preprint arXiv:1810.05766* (2018).

[15] Zijian Wang, Riccardo Spica, and Mac Schwager. "Game Theoretic Motion Planning for Multi-robot Racing". In: *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.

[16] David Fridovich-Keil, Ellis Ratner, Anca D Dragan, and Claire J Tomlin. "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games". In: *arXiv preprint arXiv:1909.04694* (2019).

[17] Forrest Laine, David Fridovich-Keil, Chih-Yuan Chiu, and Claire Tomlin. "The Computation of Approximate Generalized Feedback Nash Equilibria". In: *arXiv preprint arXiv:2101.02900* (2021).

[18] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[19] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. "ALGAMES: A Fast Solver for Constrained Dynamic Games". In: *arXiv preprint arXiv:1910.09713* (2019).

[20] Jong-Shi Pang and Gesualdo Scutari. "Nonconvex Games with Side Constraints". In: *SIAM J. on Optimization* 21.4 (Dec. 2011), pp. 1491–1522. ISSN: 1052-6234.

[21] Andreea Bobu, Dexter R. R. Scobee, Jaime F. Fisac, S. Shankar Sastry, and Anca D. Dragan. "LESS is More: Rethinking Probabilistic Models of Human Behavior". In: *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '20. Cambridge, United Kingdom: Association for Computing Machinery, 2020, pp. 429–437. ISBN: 9781450367462.

[22] Peter Trautman and Karankumar Patel. "Real Time Crowd Navigation from First Principles of Probability Theory". In: *Proceedings of the International Conference on Automated Planning and Scheduling* 30.1 (June 2020), pp. 459–467.

[23] Kathleen Fitzsimons, Aleksandra Kalinowska, Julius P Dewald, and Todd D Murphey. "Task-based hybrid shared control for training through forceful interaction". In: *The International Journal of Robotics Research* 39.9 (2020), pp. 1138–1154.