Infusing Reachability-Based Safety into Planning and Control for Multi-agent Interactions

Xinrui Wang[1⋆], Karen Leung[2⋆], Marco Pavone[2]

***Abstract*— Within a robot autonomy stack, the planner and controller are typically designed separately, and serve different purposes. As such, there is often a diffusion of responsibilities when it comes to ensuring safety for the robot. We propose that a planner and controller should share the same interpretation of safety but apply this knowledge in a different yet complementary way. To achieve this, we use Hamilton-Jacobi (HJ) reachability theory at the planning level to provide the robot planner with the foresight to avoid entering regions with possible inevitable collision. However, this alone does not guarantee safety. In conjunction with this HJ reachability-infused planner, we propose a minimally-interventional multi-agent safety-preserving controller also derived via HJ-reachability theory. The safety controller maintains safety for the robot without unduly impacting planner performance. We demonstrate the benefits of our proposed approach in a multi-agent highway scenario where a robot car is rewarded to navigate through traffic as fast as possible, and we show that our approach provides strong safety assurances yet achieves the highest performance compared to other safety control strategies.**

I. Introduction

Decision-making and control for robots is typically stratified into levels, with each having different purposes. The high-level planner, informed by representative yet simplified dynamics of a robot and its environment, is designed to be far-sighted and selects plans that optimize performance metrics (e.g., minimize time to goal, control effort, and perception uncertainty). While the low-level controller, running at a much higher frequency than the planner, tends to be more short-sighted and respects more accurate models of the robot's dynamics and control constraints in order to implement the controls necessary to follow the desired plan. However, when it comes to ensuring safety for the robot, the divide between these components can lead to a diffusion of responsibilities—the planner and controller may each devise their own safety protocols, or even assume the other bears the full responsibility, but when combined together, they may not necessarily complement each other in achieving the shared goal of ensuring safety for the system.

Safety considerations at the planning level can help discourage a robot from entering potentially dangerous situations. However, ensuring safety typically competes with other planning objectives (e.g., minimizing time and control effort). Additionally, simplifying assumptions that underpin a planner's model of the environment may cause any strong safety assurances claimed by the planner to no longer carry much weight in practice due to model mismatch with the real system. A low-level controller, whose primary purpose is to track the desired plan, is typically too shortsighted to include safety considerations especially with respect to dynamic obstacles. Instead, the controller may switch to a safety controller when near safety violation. The safety controller may be an augmentation of the existing low-level controller, or a different, possibly lower-level, reactive control scheme. However, a switching safety controller that completely overrides the system may severely compromise planner performance and potentially create new unsafe situations (e.g., an autonomous vehicle stopping abruptly on the highway can cause rear-end collisions).

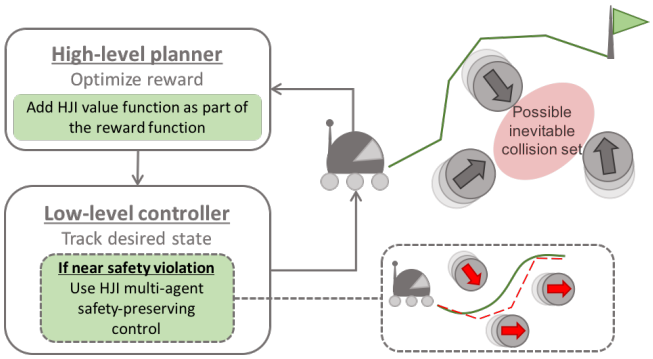

Fig. 1: Both the planner and controller use HJ reachability theory to measure safety of the system, but apply this knowledge in complementary ways. The high-level planner has the foresight to avoid regions of possible inevitable collision. When near safety violation, the safety controller evades multiple agents with minimal intervention (green trajectory), as opposed to a reactive controller (red trajectory).

In this work, we design a safe complementary planning and control stack that provides safety assurance for a robot without unduly impacting planner performance. We leverage Hamilton-Jacobi (HJ) reachability theory to provide the same representation of safety for both the planner and safety controller. In particular, the planner and safety controller maintains their unique purpose, but shares the same sentiment when it comes to understanding safety. Further, in a multi-agent setting, a robot may be in conflict as to which other agents to avoid if there is a danger of colliding with multiple agents. We use HJ reachability theory to optimally select safe controls that reason about collision avoidance with all agents threatening the robot's safety in a minimally interventional way. We demonstrate the efficacy of the proposed framework in a multi-agent highway setting where an autonomous car must move quickly and safely through a densely populated highway. The case study shows that by the planner and

[1]Department of Mechanical Engineering, Stanford University, Stanford, CA 94305. xinrui@stanford.edu

[2]Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305. {karenl7, pavone}@stanford.edu

⋆ Indicates equal contribution

This work was supported by the Office of Naval Research (Grant N00014-17-1-2433) and by the Toyota Research Institute ("TRI"). This article solely reflects the opinions and conclusions of its authors and not ONR, TRI or any other Toyota entity.

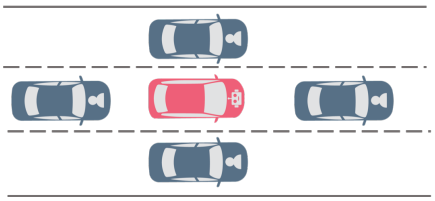

Fig. 2: A robot car (red) boxed-in by other externally controlled (e.g., human-driven) cars (blue). If all cars behave adversarially, collision is inevitable.

controller sharing their interpretation of safety, the overall interaction yields safe interactions that are more efficient and higher performing compared to reactive safety controllers.

Related work: Safety at the planning level can be enforced via hard constraints or incentivized through a planner's objective function. A common approach is to select plans that avoid the inevitable collision set (ICS), though in practice due to computational tractability, the forward reachable set is used instead which is an over-approximation. For example, [1], [2] prevent the robot's planned trajectory from entering the forward reachable set of the other agents in the environment and ensure the existence of a safe stopping maneuver at all times. When no feasible trajectories can be found, the robot switches to the emergency maneuver computed from the last feasible plan. While these approaches guarantee safety with respect to their modeling assumptions, they tend to be overly-conservative for interactive scenarios, and rely on strong modeling assumptions of other agents.

Alternatively, consideration of interaction dynamics between agents can reduce conservatism and enable proactive behaviors. In the context of autonomous driving, a probabilistic prediction model of the environment can be learned from data and then used to inform a robot's decision making algorithm which strives to optimize a multi-objective function with safety being one of the objectives [3], [4]. This can lead to more efficient but potentially unsafe interactions because safety competes with other objectives. [5] designs artificial potential fields (APF) that reflect the reachable sets of the other dynamic agents and use it for path planning. They show that this improves the safety of a robot compared to traditional gaussian APF methods. Regardless of how safety is enforced at the planning level, model mismatch and stochasticity in the environment make it very challenging to provide strict safety assurance.

In turn, low-level controllers typically provide safety reactively; they switch to a safety controller when the system is near safety violation. For instance, [6] selects a maneuver from a set of pre-computed emergency maneuver library when the system is in an unsafe state. A paradigm that can account for general interactive scenarios is HJ reachability analysis; by formulating the interactions between a robot and its environment (e.g., other agents) as an optimal control problem, a collision avoidance control can be computed and applied when near safety violation [7], [8], [9]. However, a switching-based safety controller that completely overrides the system may not be ideal in some contexts, such as in autonomous driving where sudden reactions could disrupt traffic flow. Accordingly, [10] proposes a less invasive approach in passing a HJ reachability-based safety-preserving control constraint into a low-level tracking controller. This results in a minimally interventional safety controller that enables a system to deviate from the desired trajectory to the extent necessary to maintain safety. Through a traffic-weaving case study with a robot car interacting with a human-controlled car, they show that their approach provides a good balance between safety and efficiency.

Rather than treating the planning and control problem separately, [11], [12] combine the two into a single joint optimization problem. They show through an aggressive lane change maneuver and an autonomous car racing example that combining planning and control can be more desirable because the resulting trajectory respects more accurate dynamic capabilities of the car while also cognizant of safety constraints such as avoiding static obstacles and staying within road boundaries.

The key research goal of this paper is to bring the planning and control modules closer together in order to provide stronger safety assurances in the context of multi-agent interactions as illustrated in Figure 1. Solely relying on a safety controller may be insufficient because an optimal collision avoidance control may not always exist when near safety violation, especially in the case of multi-agent scenarios. For instance, consider an autonomous car that is boxed in, as shown in Figure 2; in this situation, a collision is imminent if any of the adjacent cars swerve towards it. This necessitates the need to jointly design a planner and safety controller—the planner should be cognizant of what type of situations the safety controller is able to succeed in, while the safety controller should be aware of the planner's objectives such that it does not unduly impact overall performance.

Statement of Contributions: The key contribution of this paper is in using HJ reachability theory to provide a shared notion of safety between the planner and controller of an autonomy stack. Specifically, the contributions are threefold. First, we provide high-level planners that optimize an objective function with the foresight to avoid regions of possible inevitable collision by incorporating a term into the objective function derived from reachability theory. Second, when the system is near safety violation, we propose a multi-agent safety-preserving controller that is minimally interventional—the controller optimizes for performance while treating safety as a constraint. This extends the safety-preserving controller proposed in [10] to multi-agent settings. Third, we demonstrate our approach on a highway scenario where an autonomous car must safely traverse the highway as fast as possible. We perform an ablation study and compare against using a reactive control strategy, and a baseline safety controller. We show that our proposed approach offers a good balance between safety and performance. The insights gained from the experimental results are general since our approach applies to any planner-control framework as long as HJ reachability analysis is applicable to the system.

Organization: The remainder of the paper is organized as follows. Section II introduces the problem we are approaching and Section III provides a concise overview of HJ reachability analysis. Section IV describes our proposed methodology of aligning the planner and safety controller by providing a shared notion of safety via HJ reachability analysis. In Section V, we illustrate the benefits of our approach with a case study in a multi-agent highway driving scenario. We conclude in Section VI and suggest several

directions for future work.

II. Problem Formulation

We follow a typical decision making and control paradigm of utilizing a high level planner to compute a coarse trajectory and a low-level controller to track it.

A. Robot planner for interactive multi-agent scenarios

We consider a multi-agent setting where a robot is operating in an environment with multiple agents not controlled by the robot (e.g., humans). Let $x_{\mathrm{r}}^{(t)} \in \mathscr{X} \subset \mathbb{R}^{n_{\mathrm{r}}}$ and $u_{\mathrm{r}}^{(t)} \in \mathscr{U} \subset \mathbb{R}^{m_{\mathrm{r}}}$ be the robot planner state and action at time step t respectively. Let $x_{\mathrm{o}}^{(t)} \in \mathscr{X}_{\mathrm{o}} \subset \mathbb{R}^{Jn_{\mathrm{o}}}$ and $u_{\mathrm{o}}^{(t)} \in \mathscr{U}_{\mathrm{o}} \subset \mathbb{R}^{Jm_{\mathrm{o}}}$ be the state and control of J other agents in the environment at time step t. Further, let the time-invariant and discrete-time state space dynamics for a robot and other agents in the environment be given respectively by, $x_{\mathrm{r}}^{(t+1)} = f_{\mathrm{r}}(x_{\mathrm{r}}^{(t)}, u_{\mathrm{r}}^{(t)})$, $x_{\mathrm{o}}^{(t+1)} = f_{\mathrm{o}}(x_{\mathrm{o}}^{(t)}, u_{\mathrm{o}}^{(t)})$. The goal of a robot planner is to find a sequence of robot actions $u_{\mathrm{r}}^{(t:t+N)} = \pi(x_{\mathrm{r}}^{(t)}, x_{\mathrm{o}}^{(t)})$ that maximizes an expected reward $R(x_{\mathrm{r}}^{(t)}, u_{\mathrm{r}}^{(t)}, x_{\mathrm{o}}^{(t)})$ over a fixed horizon of length N. That is, we want to find a solution to the following maximization problem

$$\pi^*(x_{\mathrm{r}}^{(t)}, x_{\mathrm{o}}^{(t)}) = \underset{u_{\mathrm{r}}^{(t:t+N)}}{\arg\max}\ \mathbb{E}\left[\sum_{i=0}^{N} \gamma^i R(x_{\mathrm{r}}^{(t+i)}, u_{\mathrm{r}}^{(t+i)}, x_{\mathrm{o}}^{(t+i)})\right] \quad (1)$$

where $\gamma \in [0,1]$ is a discount factor. Note that for interactive scenarios, (1) is generally difficult to solve due to stochasticity and coupling in the dynamics between the robot and other agents, and that system may not be Markovian (i.e., depend on interaction history) [4], [10]. We assume that such planners typically operate at around $<$10 Hz. As such, they are not always able to account for split-second threats, necessitating the need for a safety controller to intervene.

B. Safe low-level controller

A low-level controller, operating at a higher frequency than the planner, takes the desired trajectory produced by the planner (computed by simulating the system with the desired action sequence) and computes relevant low-level control commands for the system to execute. More concretely, let $z_{\mathrm{r}}^{(t)} \in \mathscr{Z} \subset \mathbb{R}^{p_{\mathrm{r}}}$ and $w_{\mathrm{r}}^{(t)} \in \mathscr{W} \subset \mathbb{R}^{q_{\mathrm{r}}}$ represent the low-level state and control at time step t respectively. The low-level controller may use a higher fidelity model than the planner, $z_{\mathrm{r}}^{(t+1)} = f_{\mathrm{c}}(z_{\mathrm{r}}^{(t)}, w_{\mathrm{r}}^{(t)})$, in order to compute appropriate actuation commands for the system. Then $\mu : \mathscr{Z} \times \mathscr{X} \to \mathscr{W}$: $\mu(z_{\mathrm{r}}^{(t)}, x_{\mathrm{r}}) \mapsto w_{\mathrm{r}}^{(t)}$ is a policy that maps current controller state and desired planner state to actuation commands. When near safety violation, the system may switch to a more reactive policy μ_{safe} that selects control actions that keep the system safe with respect to some metric $J_{\mathrm{safe}}(z_{\mathrm{r}}, w_{\mathrm{r}}, z_{\mathrm{o}})$ where z_{o} is the low-level state of the other agents in the environment. That is, in the most general sense, the optimal safe policy μ^*_{safe} is a solution to the following optimization problem,

$$\underset{w_{\mathrm{r}} \in \mathscr{W}}{\arg\min}\ J_{\mathrm{safe}}(z_{\mathrm{r}}, w_{\mathrm{r}}, z_{\mathrm{o}}), \quad \text{s.t. System is safe.}$$

which selects the safest action. Slack variables or barrier methods can be used to ensure feasibility of the problem.

III. Hamilton-Jacobi Reachability Analysis

In this section, we highlight key HJ backward reachability concepts relevant to our proposed planning and control strategy; see [8] for a more in-depth overview.

A. Overview

Given a dynamics model governing a robotic system incorporating control and disturbance inputs, reachability analysis is the study of the set of states that the system can reach from its initial conditions, i.e., the reachable set. It is often used for formal verification as it can give guarantees on whether or not the evolution of the system will be unsafe, i.e., whether the reachable set includes undesirable outcomes. In this work, we use *backward* reachability analysis because (i) of its non-overly conservative nature stemming from closed-loop computations, and (ii) the set is computed offline and provide near-instant access online via table look-up. See [10] for a more in-depth discussion on the suitability of backward reachability for interactive scenarios.

There are many existing approaches to compute the reachable set of a system [13], [14], [15], [16], [17], but there is a trade-off between modeling assumptions, scalability, and representation fidelity. Compared to alternatives approaches, HJ reachability is the most computationally expensive, but it is able to compute the reachable set exactly[1] for any general nonlinear dynamics with control and disturbance inputs because it uses a brute force computation via dynamic programming. Further, since the sets are used online via a near-instant look-up, we can use them at high operating frequencies.

B. Backward Reachable Tube

Let the system dynamics be given by $\dot{x} = f(x,u,d)$ where $x \in \mathbb{R}^n$ is the state, $u \in \mathscr{U} \subset \mathbb{R}^m$ is the control, and $d \in \mathscr{D} \subset \mathbb{R}^p$ is the disturbance. The system dynamics $f : \mathbb{R}^n \times \mathscr{U} \times \mathscr{D} \to \mathbb{R}^n$ are assumed to be uniformly continuous, bounded, and Lipschitz continuous in x for a fixed u and d. Let $\mathscr{T} \subseteq \mathbb{R}^n$ be the target set that the system wants to avoid. For collision avoidance, $\mathscr{T}$ typically represents the set of states that are in collision with an obstacle. The *backward reachable tube* (BRT) is the set of states that could result in the system entering the target set under worst-case disturbances within some time horizon $|\tau|$. The BRT is denoted by $\mathscr{A}(\tau)$ and is defined as

$$\mathscr{A}(\tau) := \{\bar{x} \in \mathbb{R}^n : \exists d(\cdot), \forall u(\cdot), \exists s \in [\tau, 0], \\ (x(\tau) = \bar{x}) \wedge (\dot{x} = f(x,u,d)) \wedge (x(s) \in \mathscr{T})\}.$$

$\mathscr{A}(\tau)$ represents the set of states from which there does not exist a policy for the system that can prevent the dynamics from being driven into the target set under worst-case disturbances within a time horizon $|\tau|$. As such, to rule out such an eventuality, the BRT is treated as the "avoid set".

C. HJI Value function

Assuming optimal (i.e., adversarial) disturbances, $\mathscr{A}(\tau)$ can be computed by defining a value function $V(\tau, x)$ which obeys the Hamilton-Jacobi-Issacs (HJI) partial differential

[1] With precision dependent on parameters of the numerical solver, e.g., discretization choices in mesh size/time step.

equation (PDE) [18]; the solution $V(\tau,x)$ gives the BRT as its zero sublevel set,

$$\mathscr{A}(\tau) = \{x : V(\tau,x) \leq 0\}.$$

The HJI PDE is solved starting from the boundary condition $V(0,x)$, the sign of which reflects set membership of x in $\mathscr{T}$. We cache the solution $V(\tau,x)$ to be used online as a look-up table. For collision avoidance, $V(0,x)$ typically represents the signed distance between a robot and an obstacle ($x \in \mathscr{T} \iff V(0,x) \leq 0$). Then the solution $V(\tau,x)$ represents the minimum signed distance between the robot and the obstacle within a time horizon $|\tau|$ if the robot follows an optimal policy under worst-case (i.e., adversarial) disturbances. As such, the value function can be interpreted as a quantitative measure of robot safety; the larger the value, the safer.

D. HJI Optimal Control

Given the HJI value function $V(\tau,x)$, the optimal robot policy under worst-case disturbances is,

$$u^* = \arg\max_u \min_d \nabla V(\tau,x)^T f(x,u,d). \tag{2}$$

Many robotic systems employing HJ reachability-based safety switch to using (2) when near safety violation (i.e., when $V(t,x) \leq \varepsilon$, $\varepsilon > 0$) [7], [8], [9]. Recently, [10] considers a less invasive control strategy and computes the set of safety-preserving controls,

$$\mathscr{U}_{\text{safe}}(x) = \{u \in \mathscr{U} \mid \min_d \nabla V(\tau,x)^T f(x,u,d) \geq 0\} \tag{3}$$

which describes the set of controls that keep the value function from non-decreasing under worst-case disturbances. By passing $\mathscr{U}_{\text{safe}}$ as a control constraint when computing appropriate low-level control actions, the system is able to minimally deviate from the desired plan to the extent necessary to maintain safety.

E. Collision avoidance among multiple dynamic agents

We can consider collision avoidance between two dynamic agents by letting x represent the relative state of the system, u correspond to the control inputs of the agent to be controlled (i.e., the robot), and d correspond to the control inputs of the other agent. The target set $\mathscr{T}$ is the set of relative states corresponding to the system being in an undesirable situation (e.g., in collision). In theory it is also possible to formulate the relative state for more than two agents, but is impractical because HJ reachability suffers from the curse of dimensionality. Pairwise extensions are typically used to circumvent the scalability issue, but can lead to issues regarding prioritizing which pairwise interaction to respond to first. We address this precise issue in this work by proposing an optimization problem that prioritizes all violating pairs equally.

IV. Aligning the Planner and Controller

A. Overview

Although the purpose of a safety controller is to keep the robot safe when near safety violation, it is possible that the robot may enter a state of possible inevitable collision, due to shortcomings of the planner. Even when maintaining safety is feasible, the resulting safety control may go against the direction of the planner leading to chattering. We propose using HJ reachability theory to create alignment between a planner and safety controller by ensuring that they both share the same representation of safety. Further, in a multi-agent scenario, a robot safety controller may be conflicted when there are multiple agents to avoid. We use HJ reachability theory to analyse pairwise safety (see Section III-E) and propose a computationally efficient solution that prioritizes collision avoidance with all pairs equally.

B. Assumptions

We assume control over one autonomous agent (i.e., a robot) in an environment with multiple agents uncontrolled by the robot (e.g., humans). In addition to the problem formulation introduced in Section II, we make the following assumptions. Suppose there are J other agents in the environment. Let $x_{\text{rel}}^{(j)}$ denote the relative state between the robot and the jth agent. The relative state is derived from the robot's low-level state z_{r} and the agent's low-level state $z_{\text{o}}^{(j)}$. The robot control is w_{r} and the other agent's control is $w_{\text{o}}^{(j)}$. The HJI value function corresponding to the jth pairwise system is denoted by $V^{(j)}(\tau, x_{\text{rel}}^{(j)})$.

C. HJI-aware Interaction Planner

A high-level planner selects actions that maximize a reward shown in (1) which reflects desired goals, such as reaching a goal state, maintaining speed, or reducing time. We propose adding a term encompassing the HJI value function between the robot and all other agents into a planner's reward function,

$$R_{\text{total}}(x_{\text{r}}, u_{\text{r}}, x_{\text{o}}, x_{\text{rel}}) = \gamma R(x_{\text{r}}, u_{\text{r}}, x_{\text{o}}) + (1-\gamma) R_{\text{HJI}}(x_{\text{rel}}), \tag{4}$$

where x_{rel} is the relative state between the robot and all other agents. The designer can choose how to define $R_{\text{HJI}}(x_{\text{rel}})$ depending on the application. For example, we use $R_{\text{HJI}}(x_{\text{rel}}) = \min_j V^{(j)}(\tau, x_{\text{rel}}^{(j)})$ in the highway driving scenario studied in this paper. The designer can adjust γ to tune the robot's preference for reward-seeking or safety-seeking behavior. When the target set $\mathscr{T}$ is the set of collision states (or a proxy for safety), the value function represents an upper bound on how safe the future will be under worst-case disturbances. With all else being equal, a planner with an additional HJI term in its reward function will select plans that are safer with respect to backward reachability theory, i.e., the planner will avoid states of possible inevitable collision such as being boxed-in by other agents (see Figure 2).

We make little assumptions on the structure of the planner—the only assumption we make is that the planner strives to maximize a reward function, and can accommodate the addition of the proposed HJI term. For instance, this could be applied to the planners used in [3] and [4].

D. Multi-agent Safety-Preserving Control

A system may fall back to a safety controller when near safety violation. In HJ reachability applications, this occurs when $V(\tau,x) \leq \varepsilon$, $\varepsilon > 0$, i.e., when the system is close to the boundary of the BRT. Since we consider pairwise interactions between a robot and every other agent in the environment, there is the problem of prioritizing which pairwise interaction should be tackled first if safety is nearly violated for multiple pairs. To address this issue, we propose selecting feasible controls from the intersection of the safety-preserving control

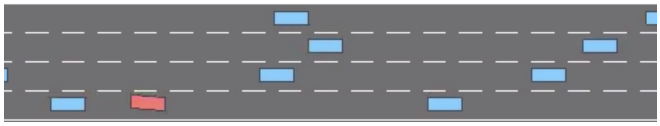

Fig. 3: Snapshot from the highway simulation environment. The robot car (red) is tasked to drive as fast as possible through the traffic but avoid collision with other cars (blue).

sets (3) of all pairwise systems where safety is nearly violated. Let

$$\mathscr{K} = \{j \mid V^{(j)}(\tau, x_{\text{rel}}^{(j)}) \leq \varepsilon \quad \text{for } j = 1, \ldots, J\}$$

be the set of indices k where safety is violated for that robot-agent pair. Further, we strive to select controls that are minimally interventional—controls that maintain safety of the system without abrupt control changes. As such, we propose the optimal safety controller to be the solution of the optimization problem,

$$\min_{w_{\text{r}}} \; g(w_{\text{r}}), \quad \text{s.t. } w_{\text{r}} \in \bigcap_{k \in \mathscr{K}} \mathscr{U}_{\text{safe}}^{(k)}(x_{\text{rel}}^{(k)}) \wedge w_{\text{r}} \in \mathscr{W} \tag{5}$$

where g is a cost function on w_{r} which may strive to achieve high tracking performance or minimize control effort. In the case where a feasible solution to (5) exists, HJ reachability theory guarantees that the system will not enter a less safe state with respect to the chosen dynamics, i.e., the value function will not decrease. For control affine systems, $\mathscr{U}_{\text{safe}}^{(k)}(x_{\text{rel}}^{(k)})$ is a hyperplane. In general, (5) could be nonlinear and intractable to solve, especially at a high operating frequency. Linearizations can be applied to make (5) tractable, as done in [10], but will no longer retain strict safety guarantees afforded by HJ reachability theory.

V. Case Study: Autonomous Highway Driving

A. Problem set-up

Our experiments are conducted in a highway simulation environment (see Figure 3) developed by [19]. The other cars on the road interact with each other using the Intelligent Driver Model (IDM) [20], [21] and minimizing-overall-braking-induced-by-lane-change (MOBIL) model [22] for longitudinal and lateral control, respectively. These are common, and well-studied traffic flow models. Our results, however, do not depend critically on this modeling choice. In our experiments, the high level planner runs at 1 Hz while the low level controller runs at 50 Hz.

1) High-level planner: We formulate the highway environment as a Markov decision process (MDP) and apply the optimistic planning (OP) algorithm proposed in [23]. OP is a tree search algorithm where each branch represents a possible future and the tree is explored optimistically. We omit implementation details about the planner since our proposed approach is agnostic to the type of planner used. Instead we highlight some key features regarding our planner that is representative of the class of planners that our approach is applicable to. First, our planner relies on a model of the environment, but in general, this model is only an approximation. This model mismatch is representative of many model-based control problems, including those for stochastic environments. In this case study, the robot does not have access to the modeling parameters of the other cars which are drawn from a normal distribution, but instead assumes the mean of the distribution. Second, our planner is reward-based and safety is promoted via the objective function. As such, there will be times where the robot will end up in an unsafe state, necessitating the use of a safety controller.

2) Planner dynamics and reward function: The robot planner state is $x_{\text{r}} = (s_{\text{p}}, \ell_{\text{p}}, v_{\text{p}})$ where s_{p} represents the longitudinal distance along a lane, ℓ_{p} is the lane index, and v_{p} is the velocity of the robot. The state representing the J other cars x_{o} is also of this structure, $x_{\text{o}} = (s_{\text{p}}^{(1)}, \ell_{\text{p}}^{(1)}, v_{\text{p}}^{(1)}, \ldots, s_{\text{p}}^{(J)}, \ell_{\text{p}}^{(J)}, v_{\text{p}}^{(J)})$. The action space of the planner is $\mathscr{U} = \{$increase v_{p} by $1ms^{-1}$, decrease v_{p} by $1ms^{-1}$, left lane change, right lane change, idle$\}$. Given this state and action representation, we design the following reward function (without the HJI term),

$$\begin{aligned} R(x_{\text{r}}, u_{\text{r}}, u_{\text{o}}) &= r_{\text{speed}}(x_{\text{r}}) + r_{\text{crash}}(x_{\text{r}}, x_{\text{o}}) + r_{\text{lane}}(x_{\text{r}}) \\ r_{\text{speed}}(x_{\text{r}}) = \gamma_1 \frac{v_{\text{p}} - \underline{v}}{\overline{v} - \underline{v}}, &\quad r_{\text{lane}}(x_{\text{r}}) = \gamma_2 \frac{\ell_{\text{p}} - \underline{\ell}}{\overline{\ell} - \underline{\ell}}, \\ r_{\text{crash}}(x_{\text{r}}, x_{\text{o}}) &= -\gamma_3 \mathbf{1}[x_{\text{r}} \text{ in collision}], \end{aligned} \tag{6}$$

where $\underline{v}$ and $\overline{v}$ corresponds to the speed limits, $\underline{\ell}$ and $\overline{\ell}$ corresponds to the right-most and left-most lanes on the highway. We select $\gamma_1 = 0.4$, $\gamma_2 = 1.0$, $\gamma_3 = 1.0$, $\underline{v} = 15\text{ms}^{-1}$, $\overline{v} = 30\text{ms}^{-1}$ for our experiment. This reward function is designed to encourage the robot to stay in the left most lane, maintain high speed, and avoid collision states. In order to test the effectiveness of our proposed safety controller, we encourage the robot to drive dangerously and weave through dense traffic at a high speed.

3) Low-level controller: We use the dynamically extended simple car model for the low-level dynamics of the robot car and all other cars. Let the low-level controller state be $z = (p_x, p_y, \theta, v)$ and control input be $w = (\delta, a)$ (for ease of notation, we drop the subscript r and o denoting robot and the other cars). The equations of motion for the dynamically extended simple car model is,

$$\dot{p}_{\text{x}} = v\cos\theta, \; \dot{p}_{\text{y}} = v\sin\theta, \; \dot{\theta} = \frac{v\tan\delta}{L}, \; \dot{v} = a, \tag{7}$$

where L is the length of the car, p_x and p_y are the longitudinal and lateral positions of the car in a fixed inertial reference frame respectively, v is the signed speed of the car, and δ and a are the steering and acceleration commands, respectively. For the robot car, a closed-loop feedback controller is deployed to track the desired planner trajectory. In the case of tracking a sequence of waypoints, let $x_{\text{r}} = (s_{\text{p}}, \ell_{\text{p}}, v_{\text{p}})$ be the desired planner state. Let $z_{\text{r}} = (p_{x,\text{r}}, p_{y,\text{r}}, \theta_{\text{r}}, v_{\text{r}})$ be the low-level controller state for the robot and $\Delta\ell$ be the signed lateral distance between the robot and the center line of ℓ_{p} (left of the centerline is positive). The closed-loop feedback controller [19] that computes low-level controls to track the desired planner state $w_{\text{r}} = (\delta, a_{\text{r}})$ is,

$$\delta = \arctan\left(\frac{-LK_\theta}{v_{\text{r}}}\left[\theta_{\text{r}} + \arcsin\frac{K_1\Delta\ell}{v_{\text{r}}}\right]\right), \; a_{\text{r}} = K_2(v_{\text{p}} - v_{\text{r}}) \tag{8}$$

where K_θ , K_1, and K_2 are control gains to be chosen. In our experiments, $K_\theta = 5.0$, $K_1 = 2.0$, and $K_2 = 1.67$.

4) HJI relative dynamics: To compute the HJI value function for a robot-agent system, we need relative dynamics of the pairwise system. We use the dynamically extended

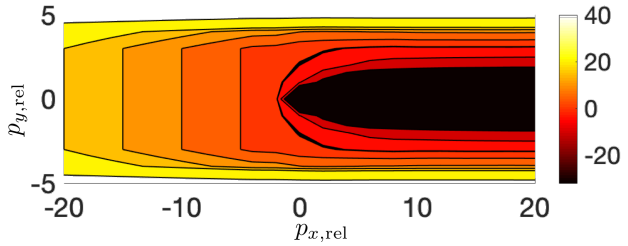

Fig. 4: A slice of the HJI value function (see Section V-A.5) with $v_\text{o} = 20ms^{-1}$, $v_\text{r} = 10ms^{-1}$, and the two cars are parallel. $p_{x,\text{rel}} > 0$ corresponds to the robot car in front of the other car. Since the other car has a higher velocity, it is more unsafe (i.e., negative value) for the robot car to be in front of the other car than behind it.

unicycle model for the robot, and a simplified unicycle model for the other car. For ease of notation, we drop the superscript denoting the jth pair, but the following equations are referring to a particular robot-agent pair. The dynamics for the robot and other car used for HJI value computations are,

$$z_{\text{r,HJI}} = \begin{bmatrix} \dot{p}_{x,\text{r}} \\ \dot{p}_{y,\text{r}} \\ \dot{\theta}_\text{r} \\ \dot{v}_\text{r} \end{bmatrix} = \begin{bmatrix} v_\text{r}\cos\theta_\text{r} \\ v_\text{r}\sin\theta_\text{r} \\ \omega_\text{r} \\ a_\text{r} \end{bmatrix}, \; z_{\text{o,HJI}} = \begin{bmatrix} \dot{p}_{x,\text{o}} \\ \dot{p}_{y,\text{o}} \\ \dot{v}_\text{o} \end{bmatrix} = \begin{bmatrix} v_\text{o}\cos\theta_\text{o} \\ v_\text{o}\sin\theta_\text{o} \\ a_\text{o} \end{bmatrix}, \tag{9}$$

where $w_{\text{r,HJI}} = (\omega_\text{r}, a_\text{r})$ and $w_{\text{o,HJI}} = (\theta_\text{o}, a_\text{o})$ are robot and other car controls, respectively. We assume the control limits: $\underline{\omega}_\text{r} \le \omega_\text{r} \le \overline{\omega}_\text{r}$, $\underline{a} \le a_\text{r}, a_\text{o} \le \overline{a}$, and $\underline{\theta}_\text{o} \le \theta_\text{o} \le \overline{\theta}_\text{o}$. We select dynamics slightly different to (7) to ensure tractability of (5) (see Section V-B), and simpler dynamics when modeling the other cars to (i) provide some conservatism to our model by assuming the other cars are more agile than the robot car, and (ii) prevent the relative state from becoming too large since the value function computation suffers from the curse of dimensionality. We define the relative coordinate frame to be aligned with the inertial frame from (7) and take the difference in position coordinates, $(p_{x,\text{rel}}, p_{y,\text{rel}}) = (p_{x,\text{r}} - p_{x,\text{o}}, p_{y,\text{r}} - p_{y,\text{o}})$. Let the relative state between the robot and the jth other agent be (still dropping the superscript j for ease of notation) $x_\text{rel} = (p_{x,\text{rel}}, p_{y,\text{rel}}, \theta_\text{r}, v_\text{r}, v_\text{o})$. The relative dynamics become,

$$\begin{aligned} \dot{p}_{\text{x,rel}} = v_\text{r}\cos\theta_\text{r} - v_\text{o}\cos\theta_\text{o}, \quad \dot{p}_{\text{y,rel}} = v_\text{r}\sin\theta_\text{r} - v_\text{o}\sin\theta_\text{o} \\ \dot{\theta}_\text{r} = \omega_\text{r}, \quad \dot{v}_\text{r} = a_\text{r}, \quad \dot{v}_\text{o} = a_\text{o} \end{aligned} \tag{10}$$

5) HJI value function: To compute the value function which represents the set of states the robot wants to avoid, rather than using the signed distance function to define $V(0, x_\text{rel})$ which is typically used in HJ reachability literature, we instead use the definition of Responsibility-Sensitive Safety (RSS) introduced in [24]. We consider unsafe states to be a function of both relative position, and velocity. For brevity, we refer the reader to [24] for the mathematical formulation, but provide a brief description here. Safety is decoupled into longitudinal and lateral components. d_long is defined as the longitudinal stopping distance between a front and a rear car if the front car applies maximum braking, and the rear car accelerates maximally over a response time before applying maximum braking. Analogously, d_lat, is the minimum lateral distance. A neighboring car is considered unsafe if both the longitudinal and lateral distances between the robot car and that car are less than d_long and d_lat, respectively. Given the velocities corresponding to a particular x_rel, we define $V(0, x_\text{rel})$ as follows:

$$V(0, x_\text{rel}) = \max(|p_{x,\text{rel}}| - d_\text{long},\, 4(|p_{y,\text{rel}}| - d_\text{lat})^3). \tag{11}$$

Figure 4 is a slice of the value function across the $p_{x,\text{rel}}$ and $p_{y,\text{rel}}$ axes for a time horizon of $|\tau| = 3$ seconds.

B. HJI safety controller

To compute the safety-preserving control set, let $\nabla V(\tau, x_\text{rel}) = (\partial V_{p_{x,\text{rel}}}, \partial V_{p_{y,\text{rel}}}, \partial V_{\theta_\text{r}}, \partial V_{v_\text{r}}, \partial V_{v_\text{o}})$ (for ease of notation, we temporarily drop the superscript j, but this is in reference to a particular robot-agent pair). Then,

$$\min_{w_\text{o} \in \mathscr{W}} \nabla V(\tau, x_\text{rel})^T f_\text{rel}(x_\text{rel}, w_\text{r}, w_\text{o}) = c_1 + c_2 + \partial V_{\theta_\text{r}}\omega_\text{r} + \partial V_{v_\text{r}} a_\text{r},$$

$$c_1 = \min_{w_\text{o} \in \mathscr{W}} \left(\partial V_{v_\text{o}} a_\text{o} - \partial V_{p_{x,\text{rel}}} v_\text{o}\cos\theta_\text{o} - \partial V_{p_{y,\text{rel}}} v_\text{o}\sin\theta_\text{o} \right),$$

$$c_2 = \partial V_{p_{x,\text{rel}}} v_\text{r}\cos\theta_\text{r} + \partial V_{p_{y,\text{rel}}} v_\text{r}\sin\theta_\text{r},$$

where $c_0 = c_1 + c_2$ represents components not dependent on the optimization variables w_r and a_r. From (5), the *minimally interventional* multi-agent safety-preserving control is the solution to the optimization problem (using the superscript notation again for different robot-agent pairs),

$$\begin{aligned} \min_{\omega_\text{r}, a_\text{r}} \; & \lambda_1(\omega_\text{r} - \omega_\text{des})^2 + \lambda_2(a_\text{r} - a_\text{des})^2 + \lambda_3 \max_{k \in \mathscr{K}} \eta_k \\ \text{s.t.} \; & \partial V_{\theta_\text{r}}^{(k)} \omega_\text{r} + \partial V_{v_\text{r}}^{(k)} a_\text{r} \ge -c_0^{(k)} - \eta_k \quad \text{for all } k \in \mathscr{K} \\ & \eta_k \ge 0 \qquad\qquad\qquad\qquad\quad \text{for all } k \in \mathscr{K} \\ & \underline{\omega}_\text{r} \le \omega_\text{r} \le \overline{\omega}_\text{r}, \qquad \underline{a}_\text{r} \le a_\text{r} \le \overline{a}_\text{r}. \end{aligned} \tag{12}$$

where $\lambda_i > 0$ are weights, η_k are slack variables, and ω_des and a_des are the desired controls from (8) (to map the HJI safety control ω to δ, we use $\delta = \tan^{-1}\frac{\omega L}{v}$). (12) is minimally interventional because it minimizes tracking error subjected to safety constraints. Alternatively, the robot can use a switching strategy by letting $\lambda_2 = 0$, $\omega_\text{des} = \omega_\text{prev}$ (to discourage discontinuous steering inputs), and removing the $\eta_k \ge 0$ constraint. The safety controller will choose controls that satisfy, or violate, each safety constraint equally (i.e., it prioritizes safety for all cars equally). By our choice of HJI dynamics (9), the optimization problem (12) is convex and we solve it using CVXPY [25].

C. Experimental results

To evaluate the benefits of our proposed planning and control stack, we perform an ablation study and compare our approach to the RSS policy proposed in [24]. For the high-level planner, we investigate the following configurations,

- **OP** An OP planner only, i.e., $\gamma = 1$ in (4).
- **HJOP** An OP planner with a HJI reward term, i.e., $\gamma = 0.9$ in (4).

With a fixed planner (HJOP or OP), we investigate different low-level safety control strategies used either in a switching (**SW**), or minimally interventional (**MI**) scheme (see Section V-A.5 for the formulation):

- **None** No safety controller is used.
- **RSS** The RSS proper response policy proposed in [24] provides the minimum longitudinal and lateral acceleration necessary to maintain safety. Under this RSS framework, there will be no accidents where the autonomous vehicle is at fault from a planning perspective.

Planner	Controller	Scheme	TTC ≥ 3	TTC 10^{th} percentile	BTN ≤ 1	BTN 90^{th} percentile	STN ≤ 1	STN 90^{th} percentile	Mean v_r (ms^{-1})	Mean $\lvert a_r \rvert$ (ms^{-2})	Interventions %
↑: higher is better, ↓: lower is better			↑	↑	↑	↓	↑	↓	↑	↓	-
HJOP	SPC	SW	1.000	9.927	1.000	0.079	1.000	0.016	21.878	1.273	9.5
HJOP	RSS	SW	0.996	17.125	0.998	0.030	0.994	0.006	20.343	4.066	55.0
HJOP	**SPC**	**MI**	**0.999**	**9.607**	**0.995**	**0.109**	**0.994**	**0.017**	**22.000**	**1.154**	**18.1**
HJOP	RSS	MI	0.996	13.914	0.998	0.028	0.995	0.008	20.301	2.788	57.5
HJOP	None	—	0.956	7.009	0.975	0.213	0.982	0.034	22.554	0.416	0.0
OP	SPC	SW	0.996	7.045	0.977	0.152	0.960	0.084	21.144	5.940	51.7
OP	RSS	SW	0.998	13.148	0.999	0.030	0.998	0.009	19.571	5.398	71.2
OP	SPC	MI	0.999	8.417	0.988	0.091	0.984	0.040	21.039	3.470	63.2
OP	RSS	MI	0.997	13.971	1.000	0.024	0.998	0.008	21.074	2.596	67.2
OP	None	—	0.502	0.785	0.714	5.616	0.782	3.189	28.141	0.386	0.0

TABLE I: Statistics for different planner and safety controller configurations. The values in the TTC ≥ 3, BTN ≤ 1, and STN ≤ 1 column represents the fraction of samples that satisfy the inequality (i.e., higher values are better).

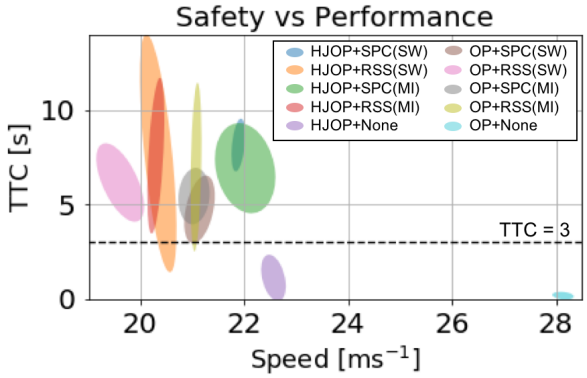

Fig. 5: The trade-off between safety (1^{st} percentile TTC) and performance (mean speed) computed from samples taken over ten-second intervals across each episode. Three standard deviation ellipses are shown.

- **SPC** Our proposed multi-agent HJI safety-preserving controller (SPC) in (5).

To compare each approach, we use the following metrics,

- *Time-to-Collision (TTC):* The estimated time before collision assuming both vehicles continue at constant speed [26]. Lower values indicate a more dangerous situation, while higher values are better with diminishing return.
- *Brake Threat Number (BTN)* and *Steer Threat Number (STN):* The required longitudinal and lateral acceleration for collision avoidance as defined in [27] divided by maximum available longitudinal/lateral acceleration. Lower values indicate safer situations.
- *Mean velocity:* The mean speed of the robot car over all sample points. Higher values indicate better performance (based on the planner's reward function).
- *Mean acceleration magnitude:* The mean magnitude of acceleration of the robot car over all sample points. Lower values indicate more efficient driving.
- *Intervention percentage:* Percentage of samples where the safety controller stepped in. Lower values implies a safer planner.

We performed 20 episodes of 30-second highway simulation with 50 Hz data collection for each planner-controller configuration. Each episode consists of 100 other vehicles that the robot needs to weave through as shown in Figure 3. This set up provides a sufficiently dense and challenging environment to evaluate the effectiveness of each planner-controller configuration. Statistics of the simulations are listed in Table I, and the trade-off between safety, measured by the 1^{st} percentile TTC, and performance, measured by mean speed, is shown in Figure 5. We select TTC $= 3$ as the boundary between unsafe and safe to reflect the popular "three-second rule" while driving, but note that this threshold could be different.

We highlight three key takeaways from these results. First, in the absence of a safety controller, adding an additional HJI term to the planner's reward function significantly improves safety as measured by BTN, STN and TTC, but, as expected, with a decrease in performance (i.e., mean speed). We note that the efficiency (i.e., mean magnitude of acceleration) of these two configurations are similar. Second, SPC and RSS provide very similar levels of safety assurance (considering the diminishing returns of larger TTC), yet SPC provides better performance and higher efficiency than RSS. RSS experiences more interventions, and therefore is likely to execute more braking and swerving maneuvers. We hypothesize that when using the RSS safety controller, the misalignment of the notion of safety between the planner and RSS controller results in chattering. Third, the MI scheme is more efficient (i.e., lower acceleration) than SW, yet performs similarly well in terms of safety metrics. In practice, using MI may be more desirable because the SPC and RSS controller assumes worst-case outcomes by the other agents, but this may not necessarily be what happens over the entire interaction. In other words, MI prevents the robot from *overreacting* every time safety is nearly violated. Instead of prioritizing all agents equally, future work can be directed at assigning priority to which agent to avoid. For example, weighing each safety constraint based each agent's likelihood of following an adversarial policy.

Further, we can visualize the safety-performance trade-off in Figure 5. The OP+None configuration is clearly the fastest but the most unsafe. The top right corner corresponds to the region with highest safety and performance. Although the RSS controllers with HJOP provides higher safety metrics, the diminishing return nature of TTC indicates that our proposed method HJOP+SPC (green) is in the ideal region—it is above the TTC $= 3$ line, and furthest to the right.

D. The mechanism behind HJOP+SPC

We take a closer look at how the HJOP and OP planners behave in a potentially dangerous situation. Given the current situation (left), Figure 6 illustrates the desired plan computed by each planner which, for this example only, assumes the other cars move at constant speed. The next two plots show the robot's roll-out over the next two time steps. The OP planner (in green) chooses to squeeze in between the two cars on its left, while the HJOP planner (in red) changes to the

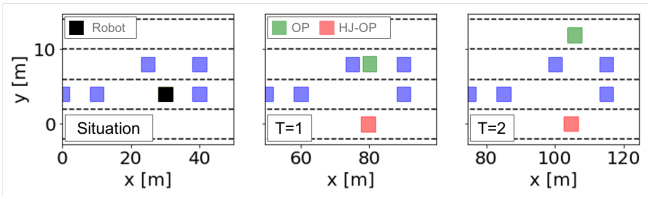

Fig. 6: Snapshots of the robot's desired plan using different planning strategies.

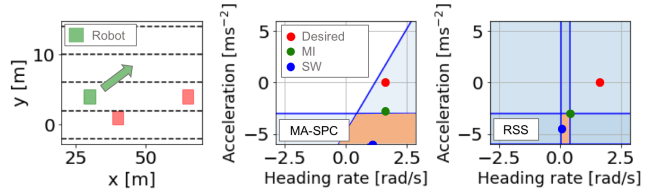

Fig. 7: Optimal controls using the SPC and RSS safety controller. In this example, $\omega_{\text{prev}} = 0$. Left: Front red car is traveling 20ms^{-1}, and other two cars are traveling at 25ms^{-1}.

right lane to avoid being trapped by the other cars. Figure 7 visualizes the set of safety preserving controls considered by the SPC and RSS controllers for a situation where the robot (in green) desires to move to the left lane (see left figure) but safety is near violation by two agents. We see the safety-preserving control constraints imposed by HJ reachability and RSS (orange region indicates the intersection of all safety-preserving control sets), and the solution to (12) for the MI and SW cases. The box-constraint nature of RSS makes it more restrictive than SPC. Due to linear constraints, MI chooses controls on the boundary and as close to the desired control as possible. While SW selects controls close to the ω_{prev} while satisfying the control constraints as much as possible.

VI. Conclusions and Future Work

By sharing the same interpretation of what it means to be safe, the planning and control modules which are typically designed separately are now complementary and can provide safe yet performant behaviors. Our proposed approach leverages HJ reachability theory and, as demonstrated with a multi-agent highway driving scenario, equips a robot with the foresight to avoid regions of possible inevitable collision, and the ability to minimally deviate from the desired trajectory to the extent necessary to avoid collision with multiple agents. We propose three future research directions; the first would be to extend the idea of using a HJI value function as part of the objective function beyond planning algorithms, such as for trajectory optimization, or as a feature in inverse reinforcement learning. The second would entail exploring different ways to define the target set (i.e., $V(0,x)$) when computing the value function, and understanding how it effects the safety-performance trade-off. The third is to design smarter priority assignment, such as through chance constraints, when safety is nearly violated by multiple agents.

References

[1] M. Althoff and J. Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[2] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff. Provably safe motion of mobile robots in human environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2017.

[3] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone. Multimodal probabilistic model-based planning for human-robot interaction. In *Proc. IEEE Conf. on Robotics and Automation*, 2018.

[4] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*, 2016.

[5] N. Malone, H-T. Chiang, K. Lesser, M. Oishi, and L. Tapia. Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field. *IEEE Transactions on Robotics*, 33(5):1124–1138, 2017.

[6] S. Arora, S. Choudhury, D. Althoff, and S. Scherer. Emergency maneuver library – ensuring safe navigation in partially known environments. In *Proc. IEEE Conf. on Robotics and Automation*, 2015.

[7] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *Proc. IEEE Conf. on Decision and Control*, 2019.

[8] M. Chen and C. J. Tomlin. Hamilton–Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):333–358, 2018.

[9] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

[10] K. Leung, E. Schmerling, M. Zhang, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone. On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions. *Int. Journal of Robotics Research*, 2019. Submitted.

[11] J. Wurts, J. L. Stein, and T. Ersal. Collision imminent steering using nonlinear model predictive control. In *American Control Conference*, 2018.

[12] V. Laurense. *Integrated motion planning and control for automated vehicles up to the limits of handling*. PhD thesis, Stanford University, 2019.

[13] M. R. Greenstreet and I. Mitchell. Integrating projections. In *Hybrid Systems: Computation and Control*, 1998.

[14] A. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: Internal approximation. *System and Control Letters*, 41(3):201–211, 2000.

[15] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *Proc. Int. Conf. Computer Aided Verification*, 2011.

[16] M. Althoff and B. H. Krogh. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59(2):371–383, 2014.

[17] A. Majumdar, R. Vasudevan, M. M. Tobenkin, and R. Tedrake. Convex optimization of nonlinear feedback controllers via occupation measures. *Int. Journal of Robotics Research*, 33(9):1209–1230, 2014.

[18] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[19] E. Leurent and contributors. An environment for autonomous driving decision-making, 2018. Available at `https://github.com/eleurent/highway-env`.

[20] M. Treiber, A. Hennecke, and D. Helbing. Microscopic simulation of congested traffic. In *Traffic and Granular Flow '99*. Springer Berlin Heidelberg, 2000.

[21] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000.

[22] A. Kesting, M. Treiber, and D. Helbing. General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board*, 1999(1):86–94, 2007.

[23] J. F. Hren and R. Munos. Optimistic planning of deterministic systems. In *European Workshop on Reinforcement Learning*, 2008.

[24] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars, 2017. Available at `https://arxiv.org/abs/1708.06374`.

[25] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[26] J. Jansson. *Collision Avoidance Theory: With application to automotive collision mitigation*. PhD thesis, Linköping University, 2005.

[27] M. Brännström, J. Sjöberg, and E. Coelingh. A situation and threat assessment algorithm for a rear-end collision avoidance system. In *IEEE Intelligent Vehicles Symposium*, 2008.