2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Human-robot interaction for truck platooning using hierarchical dynamic games

Elis Stefansson[1], Jaime F. Fisac[2], Dorsa Sadigh[3], S. Shankar Sastry[2] and Karl H. Johansson[1]

Abstract— **This paper proposes a controller design framework for autonomous truck platoons to ensure safe interaction with a human-driven car. The interaction is modelled as a hierarchical dynamic game, played between the human driver and the nearest truck in the platoon. The hierarchical decomposition is temporal with a high-fidelity tactical horizon predicting immediate interactions and a low-fidelity strategic horizon estimating long-horizon behaviour. The hierarchical approach enables feasible computations where human uncertainties are represented by the quantal response model, and the truck is supposed to maximise its payoff. The closed-loop control is validated via case studies using a driving simulator, where we compare our approach with a short-horizon alternative using only the tactical horizon. The results indicate that our controller is more situation-aware resulting in natural and safe interactions.**

I. Introduction

A. Motivation

The deployment of autonomous vehicles on public roads has received substantial attention with the DARPA urban challenge [7] and the Waymo car [12] as notable examples. In contrast to earlier proposals often considering the autonomous system in isolation [21], these new approaches aim to integrate the autonomous vehicles into the existing infrastructure. However, new challenges arise since the autonomous vehicles need to interact with human drivers. Accurate interaction models are hence crucial [13], [18], because simple models may result in opaque and potentially unsafe behaviour.

An important vehicle application domain is road freight, where recent work has shown a significant benefit for truck platooning [6] due to reduced aerodynamic drag [1]. Typical platoon controllers have been designed in isolation from other drivers or used simple interaction models [19], which can generate safety and performance degradation once deployed on public roads. In particular, a human-driven car could try to cut in between two trucks in the platoon, as in Fig. 1. Such interference may happen if the human is subject to danger in her own lane (e.g., blocked lane by road work), but should in non-critical situations be discouraged due to performance degradation of the platoon. Consequently, *situation-aware interacting* platoons are needed.

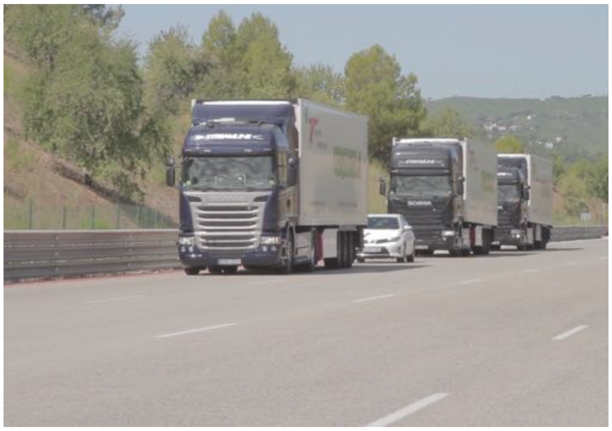

Fig. 1. A human-driven car has cut in between two trucks in the platoon. Such interference may happen if the human is subject to danger in its own lane (e.g., blocked lane by road work), but should in non-critical situations be discouraged due to performance degradation of the platoon.

B. Contribution

In this paper, we propose a platoon controller modelling the human-platoon interaction as a hierarchical dynamic game. The result is a situation-aware interacting platoon which deliberately opens up wider gaps for safety-critical lane changes, but otherwise discourage such cut-ins. More precisely, our contributions are three-fold:

We first propose a hierarchical dynamic game framework for modelling the interaction between an autonomous vehicle and a human driver, building on the formulation proposed in [10]. The game is *dynamic* to capture the sequential decision structure between the vehicle and the human, critical for long-horizon interactions. To obtain tractable solutions, a *hierarchical* approach is used. The hierarchical decomposition is temporal with a high-fidelity tactical horizon predicting immediate interactions and a low-fidelity strategic horizon estimating longer-term interactions. The human driver is represented by the quantal response model [22] for the strategic horizon to capture long-term uncertainties. The solution of the game is constructive as the autonomous vehicle obtains optimal controls by maximising its payoff.

Secondly, we present a situation-aware platoon controller interacting with a human-driven car as in Fig. 1. The platoon controller is decentralised and restricts the human interaction to the nearest truck (the *interacting* truck). The interacting truck uses a hierarchical dynamic game to predict future

[1]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden.

[2]Department of Electrical Engineering and Computer Sciences. University of California, Berkeley, United States.

[3]Computer Science Department, Stanford University, United States. Email: {elisst, kallej}@kth.se, {jfisac, shankar_sastry}@berkeley.edu, dorsa@cs.stanford.edu

This work was partially funded by the Swedish Foundation for Strategic Research, the Swedish Research Council, and the Knut och Alice Wallenberg foundation. The authors would like to thank Ehsan Nekouei for valuable feedback on this paper.

978-3-907144-00-8 ©2019 EUCA

interactions and obtain optimal controls. The non-interacting trucks use instead simple single-agent optimisation schemes to get optimal controls.

Thirdly, we validate the platoon controller using case studies with a human driver. We compare our controller with a short look-ahead alternative using only the tactical horizon. We show that the situation-aware behaviour described above naturally emerges from our controller, whereas the latter alternative is largely situation-agnostic.

C. Related work

Autonomous vehicles typically treat human drivers as immutable and plan thereafter, which may lead to opaque and overly-defence behaviour [18]. A work [13] not requiring this assumption shows how an autonomous car can instead leverage effects on a human-driven car. Here, a high-fidelity short-horizon prediction (0.5 s) is used, assuming that the human can infer the autonomous car's planned trajectory and react accordingly. Unfortunately, for long-horizon predictions, this assumption may generate overconfident autonomous vehicle behaviour. To address long-horizon interaction, dynamic game models have been applied [15], [11], [3]. However, these models use simplified high-level dynamics with pre-defined fixed manoeuvres (e.g., fixed lane changes) not necessarily capturing immediate high-fidelity interactions; the distinction between autonomous vehicles and human drivers is not addressed either. The hierarchical dynamic game framework presented here can be seen as taking the best from these approaches.

Previous work on the human-platoon interaction problem include always maintaining tight truck gaps to discourage car cut-ins [9]; or opening up a wider truck gap when a car enters the platoon identified by sensors [8], [20]. These models are situation-agnostic, as opposed to the platoon controller proposed here.

Finally, the work [10] considers hierarchical dynamic games for an autonomous car interacting with a simulated human-driven car. We here extend their game framework to a more complex problem having an autonomous platoon interacting with a human-driven car.

D. Outline

The remaining paper is structured as follows. Section II presents the hierarchical dynamic game framework. The game considers two agents, an autonomous vehicle and a human driver, interacting on the road. Section III presents the platoon controller, where the game is used to model the interplay between the human driver and the interacting truck. Section IV considers case studies. Finally, Section V concludes the paper with discussion and future research directions.

A more detailed description of the material covered in this paper can be found in the Master's thesis [17].

II. Hierarchical Dynamic Game

Consider an autonomous vehicle $\mathcal{R}$ interacting with a human driver $\mathcal{H}$ on the road. The interaction is formalised by a discrete-time control system with state $x^t \in \mathbb{R}^n$ at time $t \in \mathbb{Z}$. The agents can apply control inputs $u^t_{\mathcal{R}} \in U_{\mathcal{R}}$ and $u^t_{\mathcal{H}} \in U_{\mathcal{H}}$ within permissible compact sets $U_{\mathcal{R}} \subseteq \mathbb{R}^{n_{\mathcal{R}}}$ and $U_{\mathcal{H}} \subseteq \mathbb{R}^{n_{\mathcal{H}}}$ respectively. The control inputs result in a new state x^{t+1} according to a transition map f

$$x^{t+1} = f(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}), \tag{1}$$

updated over time intervals of length $\Delta t \in \mathbb{R}^+$. The objective of the autonomous vehicle is captured by a look-ahead reward. More precisely, setting a finite horizon of $N \in \mathbb{Z}^+$ steps, it seeks to maximise

$$R_{\mathcal{R}}(x^0, u^{0:N}_{\mathcal{R}}, u^{0:N}_{\mathcal{H}}) = \sum_{t=0}^{N} r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}), \tag{2}$$

where x^0 is the current state and $r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$ is the cumulative reward at time t. The reward $r_{\mathcal{R}}$ encodes preferences such as obstacle avoidance, lane-keeping and smooth driving. We assume that the dynamics (1) and x^0 are known to the autonomous vehicle. What remains is thus to predict $u^{0:N}_{\mathcal{H}}$. Since the human is affected by the actions of the robot with objectives that may conflict, a game-theoretic formulation is proposed. One such formulation is to assume that the human can observe x^t at each time, has a similar reward $R_{\mathcal{H}}(x^0, u^{0:N}_{\mathcal{R}}, u^{0:N}_{\mathcal{H}})$ she seeks to maximise, and that we have a predictive model $u^t_{\mathcal{H}} = u^t_{\mathcal{H}}(x^t, u^t_{\mathcal{R}})$. This generates a closed-loop feedback structure where optimal actions can be computed using dynamic programming [4], [16]. However, for long-horizon interactions (large N), computing such schemes can be cumbersome due to the curse of dimensionality; an approximation is needed. Towards this, we note that while immediate interactions (e.g., 0.0–0.5 s) may need a high-fidelity representation of the dynamics, future long-horizon interactions (e.g., 0.5–5.5 s) may instead be accurately estimated using only approximative representations.

With this insight, similar to [10], we introduce a bilevel hierarchical dynamic game decomposed in time by a high-fidelity tactical horizon modelling immediate interactions and a low-fidelity strategic horizon approximating future long-horizon interactions. A strategic planner solves the strategic horizon as a dynamic game, with value functions as outputs. A tactical planner then solves the tactical plus the strategic horizon by augmenting high-fidelity dynamics with long-horizon estimates using the strategic value functions.

A. Strategic Planner

The strategic planner considers a discrete-time hybrid system approximation with simplified dynamics[1]

$$\tilde{s}^{k+1} = \tilde{f}(\tilde{s}^k, \tilde{u}^k_{\mathcal{R}}, \tilde{u}^k_{\mathcal{H}}), \tag{3}$$

representing continuous time steps of $\Delta \tilde{t} \in \mathbb{R}^+$. Here, $\tilde{s}^k = (\tilde{q}^k, \tilde{x}^k)$ is the state with discrete state $\tilde{q}^k \in \tilde{Q}$, continuous state $\tilde{x}^k \in \tilde{X}(\tilde{q}^k) \subseteq \mathbb{R}^{\tilde{n}}$, and continuous control

[1] For brevity, we incorporate both discrete and continuous dynamics in $\tilde{f}$.

inputs $\tilde{u}_i^k \in \tilde{U}_i(\tilde{q}^k) \subseteq \mathbb{R}^{\tilde{n}_i}$.[2] The idea with the hybrid system approach is to partition the original state space into smaller domains $\{\tilde{X}(\tilde{q})\}_{\tilde{q}\in\tilde{Q}}$ where local approximations can be applied to reduce problem size (in Section III, two domains are considered confining the human-driven car to its lane or entering the truck gap). As a result we typically get $\tilde{n} < n$, $\tilde{n}_{\mathcal{R}} < n_{\mathcal{R}}$, $\tilde{n}_{\mathcal{H}} < n_{\mathcal{H}}$, and usually set $\Delta\tilde{t} > \Delta t$ to further reduce computations.

We also specify a simplified reward for the autonomous vehicle of the form

$$\tilde{R}_{\mathcal{R}}(\tilde{s}^0, \tilde{u}_{\mathcal{R}}^{0:K-1}, \tilde{u}_{\mathcal{H}}^{0:K-1}) = \sum_{k=0}^{K-1} \tilde{r}_{\mathcal{R}}(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k, \tilde{u}_{\mathcal{H}}^k) + \phi_{\mathcal{R}}(\tilde{s}^K) \tag{4}$$

which the autonomous vehicle seeks to maximise. The terminal reward $\phi_{\mathcal{R}}$ can be used to represent a critical event (e.g., a collision) without the need of adding extra states. We assume a human reward $\tilde{R}_{\mathcal{H}}$ on an analogous form, which can be based on inverse reinforcement learning [13].

The solution of the game is a modification of the classical closed-loop feedback Stackelberg solution [5] having an uncertain follower acting according to a probability distribution instead of being a (purely) rational maximiser. More precisely, we let the human be the (uncertain) follower with a noisy decision rule $p(\tilde{u}_{\mathcal{H}}^k|\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k)$ at each time step k given by the quantal response model [22]

$$p(\tilde{u}_{\mathcal{H}}^k|\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k) \propto e^{\beta Q_{\mathcal{H}}^k(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k, \tilde{u}_{\mathcal{H}}^k)}. \tag{5}$$

Here, $Q_{\mathcal{H}}^k$ is the state-action value of the human at time step k and $\beta > 0$ is the rationality parameter.[3] The autonomous vehicle, being the leader, maximises instead its state-action value $Q_{\mathcal{R}}^k(\tilde{s}, \tilde{u}_{\mathcal{R}})$ subject to the distribution (5).

The values $Q_{\mathcal{R}}^k$, $Q_{\mathcal{H}}^k$ are obtained in backward time via dynamic programming according to

$$\begin{aligned}
&\tilde{u}_{\mathcal{R}}^{k+1\star}(\tilde{s}) := \arg\max_{\tilde{u}} Q_{\mathcal{R}}^{k+1}(\tilde{s}, \tilde{u})\ ,\ \ \forall \tilde{s} \in \cup_{\tilde{q}\in\tilde{Q}} \{\tilde{q}\} \times \tilde{X}(\tilde{q}) \\
&\tilde{u}_{\mathcal{H}}^i \sim e^{\beta Q_{\mathcal{H}}^k(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k)}\ ,\ \ i \in \{k, k+1\} \\
&Q_{\mathcal{H}}^k(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k, \tilde{u}_{\mathcal{H}}^k) = \tilde{r}_{\mathcal{H}}(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k, \tilde{u}_{\mathcal{H}}^k) + \\
&\qquad\qquad \mathbb{E}_{\tilde{u}_{\mathcal{H}}^{k+1}} Q_{\mathcal{H}}^{k+1}(\tilde{s}^{k+1}, \tilde{u}_{\mathcal{R}}^{k+1\star}(\tilde{s}^{k+1}), \tilde{u}_{\mathcal{H}}^{k+1}) \\
&Q_{\mathcal{R}}^k(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k) = \mathbb{E}_{\tilde{u}_{\mathcal{H}}^k} \tilde{r}_{\mathcal{R}}(\tilde{s}^k, \tilde{u}_{\mathcal{R}}^k, \tilde{u}_{\mathcal{H}}^k) + \\
&\qquad\qquad Q_{\mathcal{R}}^{k+1}(\tilde{s}^{k+1}, \tilde{u}_{\mathcal{R}}^{k+1\star}(\tilde{s}^{k+1}))
\end{aligned} \tag{6}$$

initialised by $Q_{\mathcal{R}}^K(\tilde{s}, \cdot) = \phi_{\mathcal{R}}(\tilde{s})$, $Q_{\mathcal{H}}^K(\tilde{s}, \cdot, \cdot) = \phi_{\mathcal{H}}(\tilde{s})$. The (state) value functions $V_{\mathcal{R}}^k$ and $V_{\mathcal{H}}^k$, determining optimal remaining rewards from step k, are then given by $V_{\mathcal{R}}^k(\tilde{s}^k) = Q_{\mathcal{R}}^k(\tilde{s}^k, u_{\mathcal{R}}^{k\star})$ and $V_{\mathcal{H}}^k(\tilde{s}^k) = \mathbb{E}_{u_{\mathcal{H}}^k} Q_{\mathcal{H}}^k(\tilde{s}^k, u_{\mathcal{R}}^{k\star}, u_{\mathcal{H}}^k)$.

Equation (6) enables a straight-forward numerical implementation by discretising the state-action space, referring to [17] for details.

We stress that the actions in the strategic planner will never be executed. Instead the value functions are used by the tactical planner to estimate long-horizon interactions.

B. Tactical Planner

In the tactical planner, the autonomous vehicle considers its immediate reward over a short horizon $M \ll N$ augmented with an estimate $E_{\mathcal{R}}$ of the future long-horizon reward based on the value functions from the strategic planner. This yields the total reward

$$\bar{R}_{\mathcal{R}}(x^0, u_{\mathcal{R}}^{0:M-1}, u_{\mathcal{H}}^{0:M-1}) = \sum_{t=0}^{M-1} r_{\mathcal{R}}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t) + E_{\mathcal{R}}(x^M) \tag{7}$$

subject to (1). We assume a completely analogous human reward $\bar{R}_{\mathcal{H}}$, where $r_{\mathcal{H}}$ can be obtained via inverse reinforcement learning [13]. The estimates $E_{\mathcal{R}}$ and $E_{\mathcal{H}}$ are constructed by first obtaining value function proxies $\hat{V}_{\mathcal{R}}$ and $\hat{V}_{\mathcal{H}}$ which patch together the strategic value functions over the hybrid domains and the time domain. These proxies represent the strategic value functions but in *continuous* time t and *high-fidelity* state x. More precisely, $\hat{V}_i(t, x)$ represent the strategic value function for agent i starting at time $t \in [0, K\Delta\tilde{t}]$ on the strategic horizon and in state x (in particular, $t = 0$ uses the full strategic horizon), see [17] for details. The estimates are then set to $E_i(x^M) := \hat{V}_i(t, x^M)$. Typically, $t = 0$ yielding a total look-ahead of $M\Delta t + K\Delta\tilde{t}$ seconds (tactical plus strategic horizon), but Section III also considers a varying t adapting the strategic horizon to a critical event.

We finally specify the game's solution. Due to the short horizon (M), the dynamic coupling between the agents is less critical and we therefore use the open-loop Stackelberg solution [5] assuming that the human maximises her reward $\bar{R}_{\mathcal{H}}$. This leads to a nested optimisation

$$\begin{aligned}
&u_{\mathcal{R}}^{0:M-1\star} = \arg\max_{u_{\mathcal{R}}^{0:M-1}} \ \bar{R}_{\mathcal{R}}(x^0, u_{\mathcal{R}}^{0:M-1}, u_{\mathcal{H}}^{0:M-1\star}) \\
&s.t.\ \ u_{\mathcal{H}}^{0:M-1\star} = \arg\max_{u_{\mathcal{H}}^{0:M-1}} \ \bar{R}_{\mathcal{H}}(x^0, u_{\mathcal{R}}^{0:M-1}, u_{\mathcal{H}}^{0:M-1})
\end{aligned} \tag{8}$$

for the autonomous vehicle. We have assumed that the human accurately infers the state x^0, the planned control trajectory $u_{\mathcal{R}}^{0:M-1}$ (justified by small M) and its long-horizon reward estimate $E_{\mathcal{H}}$. Numerically, this nested optimisation can be solved using the Quasi-Newton gradient method L-BFGS [2], see [17] for details.

C. Summary

The presented hierarchical dynamic game predicts future interactions between an autonomous vehicle $\mathcal{R}$ and a human driver $\mathcal{H}$. The solution is constructive and can be used by the autonomous vehicle to obtain optimal controls: At current state x^0, it solves (8) and executes $u_{\mathcal{R}}^{0\star}$, while the human driver executes $u_{\mathcal{H}}^0$ (which may not equal $u_{\mathcal{H}}^{0\star}$); The system then moves to a new current state according to (1) and the process is repeated in a receding horizon fashion.

The next section considers the platoon setup with *multiple* semi-autonomous[4] trucks. The hierarchical dynamic game is

[2] Discrete control inputs could easily be incorporated, but are not used in our application.

[3] In words, (5) models the human as nosily-rational agent (exponentially) more likely to take actions yielding higher payoffs.

[4] The trucks are autonomous longitudinally but fixed to their platoon lane by manual control.

then used to model the interaction between a human driver and *one* of the trucks.

III. Platoon Controller

Consider a platoon of four semi-autonomous trucks driving on a two-lane highway shared with a human-driven car as in Fig. 2 (cf. Fig. 1).

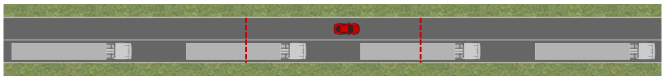

Fig. 2. The four semi-autonomous trucks and the human-driven car on the highway. The interacting truck is currently truck 3 since the car is between truck 2 and 3 (dashed red lines).

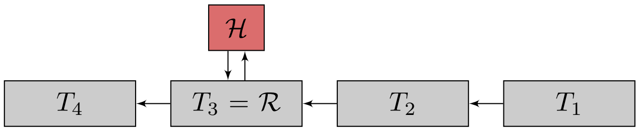

Fig. 3. Control architecture represented as a block diagram. The current interacting truck (truck 3) models the interaction with the human $\mathcal{H}$ as a hierarchical dynamic game.

Towards a platoon controller, we adapt a decentralised control architecture for the platoon controller where the planned trajectory of the truck ahead is passed down to the truck behind, initialised by the leading truck. The truck that is closest behind the human-driven car, the *interacting* truck (see Fig. 2), models the human interaction as a hierarchical dynamic game from Section II; The non-interactive trucks use instead simple single-agent optimisation schemes. The control architecture is summarised in Fig. 3 as a block diagram.

We next specify the interacting truck in Sections III-A to III-D, assuming it to be one of the following trucks for simplicity. We then describe the single-agent optimisation used by the non-interactive trucks in Section III-E and conclude with a summary in Section III-F.

A. Interactive Truck

The interactive truck uses a game as in Section II to predict the interaction with the human-driven car, solving (8) in a receding horizon with time steps $\Delta t = 0.1$ s. The rewards $\bar{R}_i$ in (8) here depend on the current execution via E_i. More precisely, at each time step, the truck monitors any upcoming critical event and acts according to two cases: If a critical event is predicted to happen $M\Delta t < t_c < M\Delta t + K\Delta\tilde{t}$ seconds ahead (i.e., the critical event happens on the strategic horizon), then $E_i(x^M) = \hat{V}_i(t, x^M)$, where t is picked so that the total look-ahead $M\Delta t + (K\Delta\tilde{t} - t)$ coincides with the predicted event time t_c; the critical event can then be incorporated into the strategic terminal rewards ϕ_i. Otherwise, the interactive controller uses the full strategic horizon, $E_i(x^M) = \hat{V}_i(0, x^M)$. The procedure is depicted in Fig. 4. Henceforth, the critical event is restricted to a potential head-on collision between the human-driven car and another car (the *head-on car*) driving in the opposite (wrong) direction, see Fig. 7. This critical event was picked for implementation simplicity and could in principle be exchanged by any other critical obstacle, e.g., road work.

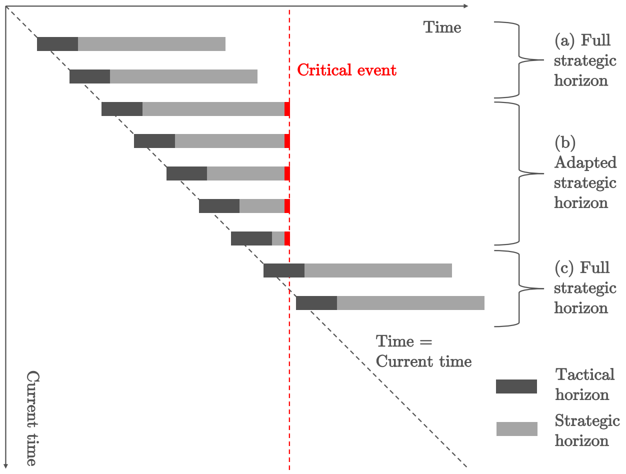

Fig. 4. Execution procedure for the interactive truck: (a) a critical event (e.g., a collision) is too far away on the prediction horizon and hence full strategic horizon is used; (b) the event happens on the strategic horizon and the horizon is adapted to the event and incorporated into the strategic terminal rewards (red bars); (c) the event is not on the strategic horizon and the full strategic horizon is again used.

The next two sections consider the game specification in detail followed by execution details in Section III-D.

B. Interactive Truck: Tactical Horizon

We start by specifying the tactical horizon, i.e., dynamics (1) and rewards r_i as in (7). We set $M = 5$.

1) Dynamics: The human-driven car is assumed to follow unicycle dynamics

$$\begin{bmatrix}\dot{x} & \dot{y} & \dot{\theta} & \dot{v}\end{bmatrix} = \begin{bmatrix}v\cos(\theta) & v\sin(\theta) & vu_1 & u_2 - \alpha_c v|v|\end{bmatrix}, \tag{9}$$

where (x, y), v and θ is the car's position, speed and heading angle, and $\alpha_c = 2.79 \cdot 10^{-4}\ m^{-1}$ is the air friction coefficient for a typical car. The control input $u = [u_1, u_2]^T$ specifies steering input $u_1 \in [-3.7, 3.7] \cdot 10^{-3}$ and acceleration $u_2 \in [-2.5, 2.5]$, bounded by typical car values. The truck is instead fixed in its lane with one-dimensional dynamics

$$\begin{bmatrix}\dot{x} & \dot{v}\end{bmatrix} = \begin{bmatrix}v & u - \alpha_t \cdot v|v|\end{bmatrix} \tag{10}$$

where $\alpha_t = 5.64 \cdot 10^{-5}$ m^{-1} is the air friction coefficient for a typical truck. We confine acceleration input to typical truck values $u \in [-2.9, 0.59]$. Both (9) and (10) are executed discretely with $\Delta t = 0.1$ s.[5]

We also predict future states on the tactical horizon for the truck ahead (the truck in front of the interacting truck) and the head-on car. These predictions are considered fixed in the game: For the head-on car, we use a constant-velocity prediction assuming access to the head-on car's current position x_h^0 and velocity v_h^0; The predicted states of the

[5] Below, labels $\mathcal{R}$ and $\mathcal{H}$ are sometimes used to differentiate between the vehicles' parameters when needed (e.g., $u_{\mathcal{H},2}$).

truck ahead is instead assumed *given* by the truck ahead in accordance with the control architecture (cf. Fig. 3). These predictions together with the truck and car dynamics gives (1).

2) Rewards: The reward $r_{\mathcal{R}}$ consists of two parts:

$$r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) = r^{core}_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) + r^{goal}_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}).$$

The first term $r^{core}_{\mathcal{R}} = \theta^T \Phi_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$ is a linear combination of core features $\Phi_{\mathcal{R}}$ with inverse reinforcement learning weights θ from [13]; Included features are a speed reference term $-(v - v_{ref})^2$ (we set $v_{ref} = 30$ m/s), a control penalty $-u^2$ and a penalty for being near the truck ahead and the human-driven car (using logistic sigmoids with boundaries corresponding to collision). The second term $r^{goal}_{\mathcal{R}}$ encodes more specific objectives: A truck tailing feature $-(x - (x_p - x_{ref}))^2$ where x_p is the position of the truck ahead and x_{ref} is desired truck distance (we set x_{ref} to an optimal truck gap of 9 meters); a penalty for having the human-driven car near the truck ahead (logistic sigmoids)[6]; and an extra high penalty for having a collision between the human-driven car and the head-on car.[7]

The reward $r_{\mathcal{H}}$ similarly equals $r_{\mathcal{H}} = r^{core}_{\mathcal{H}} + r^{goal}_{\mathcal{H}}$. The features $\Phi_{\mathcal{H}}$ of $r^{core}_{\mathcal{H}} = \theta^T \Phi_{\mathcal{H}}$ include staying on lanes, staying inside road boundaries (one-dimensional Gaussians), a control penalty $-u^2$, a speed reference term $-(v - v_{ref})^2$ (again $v_{ref} = 30$ m/s) and a penalty for being near the truck and the truck ahead (logistic sigmoids), with weights θ again from [13]. The reward $r^{goal}_{\mathcal{H}}$ has an extra high penalty for colliding with the head-on car and an extra right lane reward modelling common preference.

Having specified the tactical horizon, we continue with the strategic horizon and how it is solved with the strategic planner as in II-A.

C. Interactive Truck: Strategic Horizon

To reduce problem size, we assume that the truck ahead travels at constant velocity $v_{ref} = 30$ m/s and introduce a hybrid system approximation. The hybrid system consists of two discrete states called the *lane mode* and the *merge mode*. Briefly, the human is fixed to the left lane in the lane mode while it is near the truck gap with small velocity changes in the merge mode, depicted in Fig. 5. Also, the head-on car is not modelled dynamically; instead, its potential collision with the human-driven car is solely incorporated in the terminal rewards. We describe the setup in detail below specifying the dynamics (3) first, then the rewards (4), and conclude with implementation details.

1) Dynamics: Truck dynamics coincide with (10) except that relative coordinates $x_{\mathcal{R},rel} := x_{\mathcal{R}} - x_p$, $v_{\mathcal{R},rel} := v_{\mathcal{R}} - v_{ref}$ with respect to the truck ahead are used

$$\begin{bmatrix} \dot{x}_{\mathcal{R},rel} \\ \dot{v}_{\mathcal{R},rel} \end{bmatrix} = \begin{bmatrix} v_{\mathcal{R},rel} \\ u_{\mathcal{R}} - \alpha_t \cdot (v_{\mathcal{R},rel} + v_{ref})|(v_{\mathcal{R},rel} + v_{ref})| \end{bmatrix}. \tag{11}$$

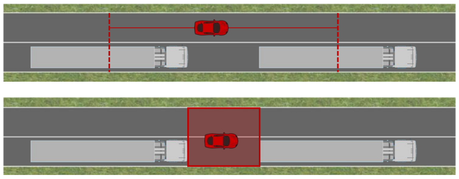

Fig. 5. Domain of lane mode (upper) and merge mode (lower). Note that the domain of the lane mode agrees with the interacting borders in Fig. 2.

The dynamics of the human-driven car depend instead on the discrete state, as detailed below.

Lane mode. In the lane mode, the car is assumed to stay in the left lane with domain as in Fig. 5 and dynamics analogous to (11) but with relative coordinates $x_{\mathcal{H},rel} := x_{\mathcal{H}} - x_p$, $v_{\mathcal{H},rel} := v_{\mathcal{H}} - v_{ref}$ and α_c instead of α_t. Additionally, the car may drive outside the domain (by passing the middle of the truck or the truck ahead). To incorporate this, we set a switch from the lane mode to itself approximately corresponding to periodic conditions: If $x_{\mathcal{H},rel} > 0$ (the car has passed the truck ahead), the state is reset to[8]

$$\begin{aligned} x_{\mathcal{H},rel} &:= x_{\mathcal{H},rel} - x_{ref}, & x_{\mathcal{R},rel} &:= x_{ref} \\ v_{\mathcal{H},rel} &:= v_{\mathcal{H},rel}, & v_{\mathcal{R},rel} &:= v_{ref}. \end{aligned}$$

and if $x_{\mathcal{H},rel} < x_{\mathcal{R},rel}$ (the car has fallen behind the truck)[9]

$$\begin{aligned} x_{\mathcal{H},rel} &:= x_{\mathcal{H},rel} - x_{\mathcal{R},rel}, & x_{\mathcal{R},rel} &:= x_{ref} \\ v_{\mathcal{H},rel} &:= v_{\mathcal{H},rel}, & v_{\mathcal{R},rel} &:= v_{\mathcal{R},rel}. \end{aligned}$$

Merge mode. In the merge mode, the car is near the truck gap having low relative velocity (with respect to the platoon). We then enable the car to change its position both longitudinally and laterally with domain depicted in Fig. 5. This is done at the expense of kinematic approximations

$$\begin{bmatrix} \dot{x}_{\mathcal{H},rel} & \dot{y}_{\mathcal{H}} \end{bmatrix} = \begin{bmatrix} w_{\mathcal{H},x} & w_{\mathcal{H},y} \end{bmatrix},$$

where $y_{\mathcal{H}}$ is the car's lateral position, and we bound the longitudinal velocity input $w_{\mathcal{H},x} \in [-2, 1]$ and the lateral velocity input $w_{\mathcal{H},y} \in [-l/3, l/3]$; the latter corresponds to a 3 s lane change having lane width $l = 3.7$ m.

Discrete transitions. We enable the car to switch between the modes by placing transitions in both directions. If the car slows down and is next to the truck gap, it is natural to switch from lane mode to merge mode. Hence, we place a switching condition

$$x_{\mathcal{R},rel} + L/2 \le x_{\mathcal{H},rel} \le -L/2, \quad -2 \le v_{\mathcal{H},rel} \le 1$$

and reset function $x_{\mathcal{H},rel} := x_{\mathcal{H},rel}$, $y_{\mathcal{H}} := l$.[10] Here, $L = 19.8$ m is the truck length. For the other direction, we

[6] This altruistic penalty is needed in practice since the truck ahead is not actively interacting with the human.

[7] Omitting this penalty may lead to safety issues where the truck tries to block the car by closing the truck gap.

[8] In words, the truck ahead becomes the new truck initially holding the desired truck gap and reference speed. The new truck ahead is also assumed to hold the reference speed v_{ref}.

[9] In words, the truck becomes the new truck ahead and the truck behind it becomes the new truck, initially holding the desired truck gap and the previous following velocity. It is also assumed that the new truck ahead changes its velocity momentarily to v_{ref}.

[10] Here, $y_{\mathcal{H}} = l$ corresponds to the middle of the left lane. The first condition says that the position of the car is next to the truck gap, while the second condition is consistent with the velocity bounds in the merge mode.

place a switching condition

$$x_{\mathcal{H},rel} > -L/2 \text{ or } x_{\mathcal{H},rel} < x_{\mathcal{R},rel} + L/2$$
$$l - \Delta\tilde{t} \cdot l/3 \leq y_{\mathcal{H}} \leq l + \Delta\tilde{t} \cdot l/3$$

and reset function $x_{\mathcal{H},rel} := x_{\mathcal{H},rel}$, $v_{\mathcal{H},rel} := w_{\mathcal{H},x}$.[11] The discrete transitions are summarised in Fig. 6. The dynamics (3) is formed by executing above discretely with $\Delta\tilde{t} = 0.5$ s.

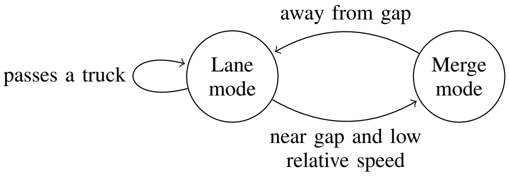

Fig. 6. Informal overview of the discrete transitions with respect to the human-driven car.

2) Rewards: The strategic rewards are based on the tactical rewards. The truck reward equals $\tilde{r}_{\mathcal{R}} = \tilde{r}_{\mathcal{R}}^{core} + \tilde{r}_{\mathcal{R}}^{goal}$ where $r_{\mathcal{R}}^{core}$ is re-used to form $\tilde{r}_{\mathcal{R}}^{core}$ and all terms in $r_{\mathcal{R}}^{goal}$ except the car collision penalty are re-used to form $\tilde{r}_{\mathcal{R}}^{goal}$. The collision is instead represented by a high penalty φ in the terminal reward for having the human-driven car in the left lane (logistic sigmoids) with magnitude consistent with the collision penalty on the tactical horizon. The result is $\phi_{\mathcal{R}}(\tilde{s}^K) = \tilde{r}_{\mathcal{R}}(\tilde{s}^K, 0, 0) + \varphi(\tilde{s}^K)$. This terminal reward is used if a car collision is predicted on the strategic horizon (cf. Fig. 4); otherwise, $\phi_{\mathcal{R}}(\tilde{s}^K) = \tilde{r}_{\mathcal{R}}(\tilde{s}^K, 0, 0)$.

The human reward $\tilde{r}_{\mathcal{H}}$ in both modes re-uses $r_{\mathcal{H}}$ except omitting the collision penalty, and in the merge mode: control and speed features are appropriately modified for having velocities as control inputs. The collision penalty is instead incorporated in the terminal reward if needed, analogous to the truck case.

3) Implementation details: We set $K = 10$ and run the strategic planner with discretised domains as in Table I. We then obtain value function proxies $\hat{V}_i(t, x)$ specified as neural networks, see [17] for details.

D. Interactive Truck - Execution Details

The interacting truck solves (8) with time steps $\Delta t = 0.1$ s and game specified above. More precisely, the predicted collision time is given by $t_c = (x_h^0 - x_{\mathcal{H}}^0)/(|v_h^0| + |v_{\mathcal{H}}^0|)$. If $M\Delta t < t_c < M\Delta t + K\Delta\tilde{t}$, the collision is incorporated into the terminal rewards ϕ_i as in III-C.2 and estimates are set to $E_i(x^M) = \hat{V}_i(t, x^M)$ where t updates according to $M\Delta t + (K\Delta\tilde{t} - t) = t_c$. Otherwise we set $E_i(x^M) = \hat{V}_i(0, x^M)$. This proceeds in a receding horizon fashion.

E. Non-Interactive Trucks

The non-interactive trucks need no game-theoretic setup since the human driver is not close. Hence, we restrict them to simple optimisation schemes $\arg\max_{\mathbf{u}_{\mathcal{R}} \in U_{\mathcal{R}}^M} \sum_{t=0}^{M-1} r_{\mathcal{R}}(x^t, u_{\mathcal{R}}^t) + r_{\mathcal{R}}(x^M, 0)$ with $M = 5$, dynamics as on the tactical horizon and $r_{\mathcal{R}}(x^t, u_{\mathcal{R}}^t)$ including the features speed reference, control penalty and truck tailing similar to the interactive truck.[12] This proceeds in a receding horizon fashion with time steps $\Delta t = 0.1$ s.

TABLE I

Variable	Discretised interval	Number of points
Lane mode		
$x_{\mathcal{R},rel}$	$[-37.8, 19.8]$	41
$v_{\mathcal{R},rel}$	$[-2, 1]$	13
$x_{\mathcal{H},rel}$	$[-27.9, 0]$	21
$v_{\mathcal{H},rel}$	$[-6, 6]$	13
$u_{\mathcal{R}}$	$[-2.9, 0.59]$	7
$u_{\mathcal{H}}$	$[-5, 2.5]$	7
Merge mode		
$x_{\mathcal{R},rel}$	$[-37.8, 19.8]$	41
$v_{\mathcal{R},rel}$	$[-2, 1]$	13
$x_{\mathcal{H},rel}$	$[-27.9, -9.9]$	32
$y_{\mathcal{H}}$	$[-3.7/2, 3 \cdot 3.7/2]$	13
$u_{\mathcal{R}}$	$[-2.9, 0.59]$	7
$w_{\mathcal{H},x}$	[-2,1]	4
$w_{\mathcal{H},y}$	[-3.7/3, 3.7/3]	5

F. Summary

We conclude with a summary. The platoon controller is executed with time steps $\Delta t = 0.1$ s. At each time step, the leading truck initiates the controller by computing its planned state-control trajectory $\{x_{\mathcal{R}}^t, u_{\mathcal{R}}^t\}_{t=0}^{M-1}$. The state trajectory is then passed down to the next truck, which computes its planned trajectory, and the process is repeated until the last truck is done. All non-interactive trucks use the single-agent optimisation specified in Section III-E, while the interacting truck uses the game as in Sections III-A to III-D.

IV. Case Studies

This section presents case studies involving a truck platoon and a car on a two-lane highway, conducted in a driving simulator. The platoon acts according to a specified controller while the car is human-driven. We consider two scenarios, the *head-on car* and the *off-ramp* scenario for short. In both scenarios, the car starts behind the platoon with aim to keep a higher velocity. The human driver therefore starts to overtake the platoon. In the head-on scenario, the driver faces an approaching head-on car when driving next to the middle trucks; the driver is notified by a countdown arrow in the simulator, specifying how many seconds are left till the head-on car appears on the screen (see Fig. 7), starting at 5 seconds. The human driver wants to avoid a head-on collision and typically tries to cut in between the trucks. In the off-ramp scenario, an off-ramp is instead approaching, alerted by a similar 5-second countdown arrow (see Fig. 11). The human driver may or may not want to take the off-ramp; we consider both options.

The dynamics of the car and the trucks are given by (9) and (10). The human driver steers the car via keyboard inputs.

[11] The conditions merely says that the car is outside the truck gap lines and sufficiently close to the middle of the left lane. More precisely, the condition on $y_{\mathcal{H}}$ is set so that the middle of the left lane $y_{\mathcal{H}} = l$ can be reached in one time step from the merge mode.

[12] The truck tailing term is omitted for the leading truck.

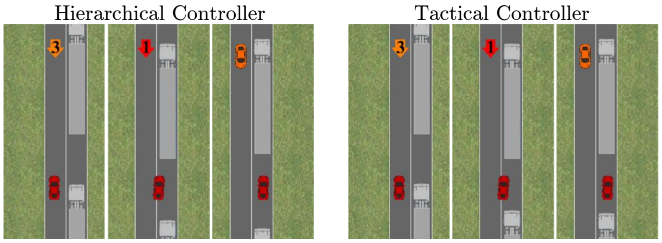

Fig. 7. The controllers' manoeuvres in the head-on scenario run. The orange car is the head-on car.

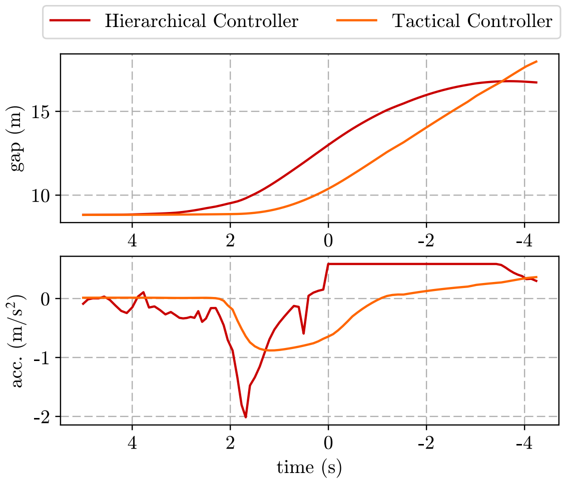

Fig. 8. Truck acceleration and distance to its truck ahead (as a function of time) for the controllers corresponding to Fig. 7.

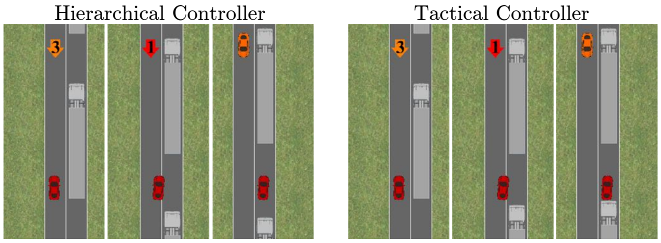

Fig. 9. The controllers' manoeuvres in the head-on car scenario run. Here, the countdown starts when the human driver overtakes the truck ahead.

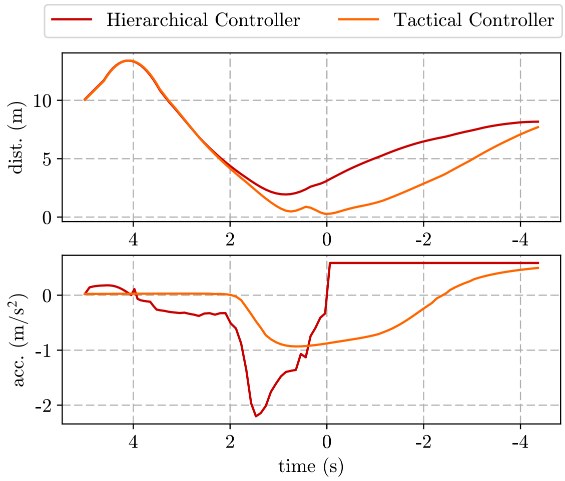

Fig. 10. Truck acceleration and truck-car distance for the controllers corresponding to Fig. 9.

We compare the proposed platoon controller in Section III with a controller using only the tactical horizon for the interactive truck (otherwise identical, see [17] for details). We call them the hierarchical and tactical controller respectively. For consistency, the trajectory of the human driver is pre-recorded with the hierarchical controller and then re-used for both controllers.[13]

Finally, in all plots, 'time' encodes the countdown time until either a head-on car or an off-ramp appears in the driving simulator.

1) Head-on car: The behaviour of the two controllers from a head-on car scenario run can be seen in in Fig. 7 in the form of snapshots, while Fig. 8 plots the acceleration of the (interacting) truck and the distance to its truck ahead. An earlier brake can be noted for the hierarchical controller predicting a merging car, leading to an earlier and wider truck gap; the tactical controller is instead agnostic and starts to brake only when the human actually merges (around 2 s), due to its shorter horizon. This agnostic behaviour can lead to safety-critical issues for the latter controller shown in Fig. 9. Here, the countdown starts when the human overtakes the truck ahead and hence she needs to brake heavily to reach the gap in time. The manoeuvre is not predicted by the tactical controller and the truck therefore almost hits the car as seen in Fig. 10. The hierarchical controller avoids this safety issue with an earlier and heavier brake.

2) Off-ramp: Results from an off-ramp scenario run with a human who does not merge are shown via snapshots in Fig. 11 with truck gap and acceleration plotted in Fig. 12. Since the event is not critical, the hierarchical controller automatically discourages the human from entering its lane by tightening the gap slightly, favouring performance from reduced aerodynamic drag. The tactical controller is again agnostic. If a human merges anyway, both controllers open up a wider gap as seen in Fig. 13.

3) Conclusion: The scenarios indicate that the hierarchical controller generates situation-aware platoon-human interactions. This situation-aware feature emerges from long-horizon predictions and was not seen for the tactical controller using only short-horizon predictions.

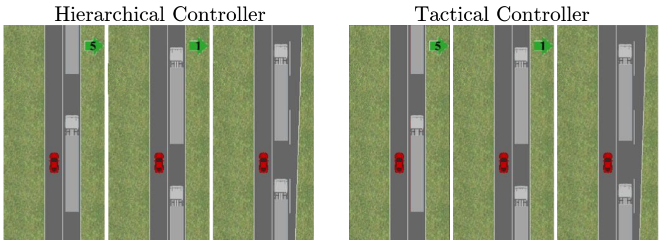

Fig. 11. The controllers' manoeuvres in the off-ramp scenario run with a human who does not merge.

[13] This could give the hierarchical controller an advantage. However, pre-recording with the tactical controller yields similar behaviours indicating that this advantage is minor, see [17] for details.

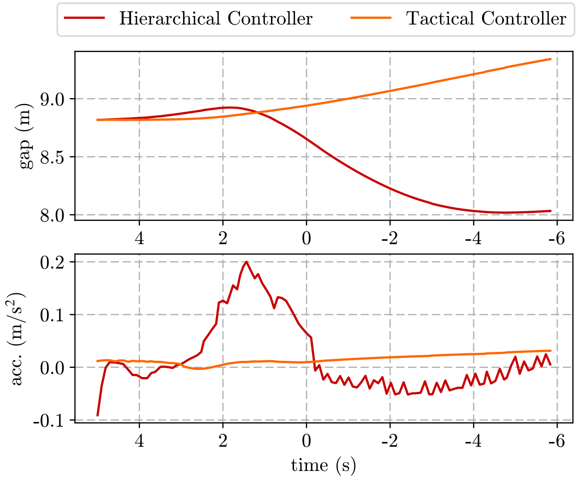

Fig. 12. Truck acceleration and distance to its truck ahead for the controllers corresponding to Fig. 11.

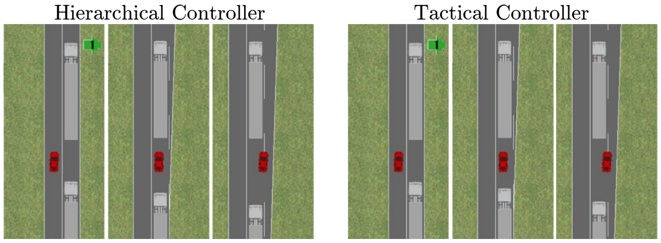

Fig. 13. The controllers' manoeuvres in the off-ramp scenario run with a merging human.

V. Discussion

We have proposed a truck platoon controller interacting with a human-driven car. The platoon controller is decentralised and restricts the human interaction to the nearest truck, where the interacting truck uses a hierarchical dynamic game [10] to predict future interactions. The hierarchical decomposition is temporal with a high-fidelity tactical horizon predicting immediate interactions and a low-fidelity strategic horizon estimating long-horizon interactions. The human is modelled with the quantal response model [22] on the strategic horizon to capture long-term uncertainties. The solution of the game is constructive where the truck obtains optimal controls by maximising its payoff.

We demonstrated the performance of the platoon using case studies with a human-driven car, comparing our controller with a short-horizon alternative using only the tactical horizon. We observed that gap openings emerged naturally to facilitate safety-critical lane changes for the human driver, but were avoided under normal cruising to discourage cut-ins, favouring performance. This situation-aware behaviour was not seen for the short-horizon alternative.

Future work should focus on games with incomplete information structures. In particular, the human may be unsure if the platoon will open up a wider gap, which can be modelled via uncertainties concerning players' intents. Incomplete information structures can also predict different driving styles [14]. Finally, another challenge is to extend the model to a platoon interacting with multiple human-driven vehicles.

References

[1] A. A. Alam, A. Gattami, and K. H. Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 306–311. IEEE, 2010.

[2] G. Andrew and J. Gao. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[3] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr. A Game-Theoretic Approach to Replanning-Aware Interactive Scene Prediction and Planning. *IEEE Transactions on Vehicular Technology*, 65(6):3981–3992, 2016.

[4] T. Başar and A. Haurie. Feedback equilibria in differential games with structural and modal uncertainties. *Advances in Large Scale Systems*, pages 163–301, 1984.

[5] T. Başar and G. J. Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[6] B. Besselink, V. Turri, S. H. van de Hoef, K. Liang, A. Alam, J. Martensson, and K. H. Johansson. Cyber-physical control of road freight transport. *Proceedings of the IEEE*, 104(5):1128–1141, 2016.

[7] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. Springer, 2009.

[8] E. Chan, P. Gilhead, P. Jelinek, P. Krejci, and T. Robinson. Cooperative control of SARTRE automated platoon vehicles. In *19th ITS World Congress*, 2012.

[9] S. Deutschle, G. C. Kessler, C. Lank, G. Hoffmann, M. Hakenberg, and M. Brummer. Use of electronically linked konvoi truck platoons on motorways. *ATZautotechnology*, 10(4):20–25, 2010.

[10] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[11] D. Lenz, T. Kessler, and A. Knoll. Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 447–453. IEEE, 2016.

[12] J. Markoff. Google cars drive themselves, in traffic. *The New York Times*, 10(A1):9, 2010.

[13] 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.

[14] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan. Information gathering actions over human internal state. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 66–73. IEEE, 2016.

[15] W. Schwarting and P. Pascheka. Recursive conflict resolution for cooperative motion planning in dynamic highway traffic. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1039–1044. IEEE, 2014.

[16] M. Simaan and J. B. Cruz Jr. Additional aspects of the Stackelberg strategy in non-zero sum games. *Journal of Optimization Theory and Applications*, 11(6):613–626, 1973.

[17] E. Stefansson. Hierarchical dynamic games for human-robot interaction with applications to autonomous vehicles. Master's thesis, KTH Royal Institute of Technology, 2018.

[18] P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.

[19] S. Tsugawa, S. Jeschke, and S. E. Shladover. A review of truck platooning projects for energy savings. *IEEE Transactions on Intelligent Vehicles*, 1(1):68–77, 2016.

[20] S. Tsugawa, S. Kato, and K. Aoki. An automated truck platoon for energy saving. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4109–4114. IEEE, 2011.

[21] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on automatic control*, 38(2):195–207, 1993.

[22] J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI*, pages 901–907, 2010.