Learning MPC for Interaction-Aware Autonomous Driving: A Game-Theoretic Approach

Brecht Evens   Mathijs Schuurmans   Panagiotis Patrinos

Abstract— **We consider the problem of interaction-aware motion planning for automated vehicles in general traffic situations. We model the interaction between the controlled vehicle and surrounding road users using a generalized potential game, in which each road user is assumed to minimize a common cost function subject to shared (collision avoidance) constraints. We propose a quadratic penalty method to deal with the shared constraints and solve the resulting optimal control problem online using an Augmented Lagrangian method based on PANOC. Secondly, we present a simple methodology for learning preferences and constraints of other road users online, based on observed behavior. Through extensive simulations in a highway merging scenario, we demonstrate the practical efficacy of the overall approach as well as the benefits of the proposed online learning scheme.**

I. Introduction

A. Background and motivation

Despite many efforts in recent years, safe autonomous navigation in the presence of humans remains a highly challenging task. Traditionally, this problem is tackled by separating it into a forecasting problem and a planning problem. That is, predictions of surrounding vehicles' trajectories are generated based on assumptions about their drivers, and these predictions are then fixed and fed into a planning module. However, since the mutual interactions between the autonomous vehicle and the surrounding traffic are neglected, this type of formulation may lead to overly conservative behavior, also referred to as the *frozen robot problem* [1]. In dense traffic, for instance, such a vehicle will typically be unable to merge into lanes cross condensed junctions, even though in practice, feasible solutions may very well exist.

Early work on social navigation such as [1] proposed to solve the frozen robot problem by modelling vehicles as a team of cooperative players, working together to ensure that each player reaches their destination while avoiding collisions. Autonomous navigation can be related to other problems in game theory as well, such as antagonistic and Stackelberg (i.e., leader-follower) games. Antagonistic games are commonly used to account for the worst-case behavior of the surrounding traffic, resulting in safe, robust control strategies [2], which may again suffer from the frozen robot problem. Stackelberg games can be used to model interactions where one driver, i.e., the leader, dominates the decision process [3], [4]. Despite its computational benefits, a drawback of this approach is that it places the burden of collision avoidance unilaterally on the followers. This may lead to undesirably aggressive behavior, as the leader (typically the autonomous vehicle) will assume that the followers can anticipate its planned trajectory accurately.

In this work, we aim to account for the mutual interactions between the controlled vehicle and other road users by modeling it as a cooperative game with coupled constraints, following the previously mentioned work on social navigation. The problem of finding a solution, i.e., a Nash equilibrium, of a game with coupled constraints is called a generalized Nash equilibrium problem (GNEP) [5] and is in general notoriously challenging to solve. Efficiently computing solutions to GNEPs in real-time poses one of the major challenges of this paper.

Moreover, a game-theoretic formulation of road user interaction requires the controlled vehicle to have access to the (implicit) objective function and constraints of the other road users. Among autonomous vehicles, such knowledge can reasonably be assumed to be obtained via vehicle-to-vehicle communication. However, when human drivers are considered, it needs to be inferred from observations, taking into account the possibly time-varying nature of their actions. In the unconstrained setting, this is commonly performed by parametrizing the human objective function and then using Inverse Reinforcement Learning to update these parameters [3], [6], [7]. In this work, we aim to design a practical methodology able to also incorporate and learn constraints.

Our contributions are twofold:

a) A penalty method for interaction-aware motion planning We formalize driving interaction with mutual collision avoidance as generalized potential games (GPGs) [8], a subclass of GNEPs, for which we can find a normalized Nash equilibrium [9] by solving a single nonlinear optimal control problem (OCP) directly, instead of having to use dedicated solvers such as [10], [11] for general GNEPs. This kind of Nash equilibrium is particularly suited for socially aware motion planning as for such an equilibrium the Lagrange multipliers associated with the shared constraints are equal among all players, which introduces some notion of fairness among the players. This idea has been used in various recent papers on traffic control, i.e., without human drivers, such as [12], [13], where ellipsoidal constraints are used to enforce safety. As ellipsoidal overapproximations seem less suitable to model human behavior and may furthermore

This work was supported by the Research Foundation Flanders (FWO) research projects G0A0920N, G086518N, G086318N, and PhD grant 1183822N; Ford KU Leuven Research Alliance Project KUL0075; Research Council KU Leuven C1 project No. C14/18/068; and the Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS project no 30468160 (SeLMA).

KU Leuven, Department of Electrical Engineering ESAT-STADIUS – Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven, Belgium
{brecht.evens,mathijs.schuurmans,panos.patrinos}@kuleuven.be

be conservative in small distances (where interaction is more pronounced), we instead extend the collision avoidance formulation of [14], which involves no approximation of rectangular obstacles.

b) Online learning based on an optimal control model

We propose an online learning methodology to continually update our estimate of other player's costs and constraints by adapting standard inverse optimal control methodologies such as [15]–[17] to our game-theoretical framework. We show empirically that this approach performs well in practice, demonstrating successful closed-loop navigation, where the certainty-equivalent controller exhibits suboptimal or even dangerous behavior.

B. Notation

Given two nonnegative integers $a \leq b$, let $\mathbb{N}_{[a,b]} := \{n \in \mathbb{N} \mid a \leq n \leq b\}$. We define $[\,\cdot\,]_+ := \mathbf{max}\{0, \cdot\}$, where $\mathbf{max}$ is interpreted element-wise. Given a set U, we denote $U^N := U \times \cdots \times U$ as the N-fold Cartesian product with itself, for $N \in \mathbb{N} \setminus \{0\}$. Given a positive definite matrix $Q \succ 0$, let $\|x\|_Q^2 = x^\top Q x$ denote the weighted square norm.

II. Problem statement

We consider the task of controlling an autonomous vehicle interacting with human drivers (and optionally also other autonomous vehicles) in general traffic environments, and formulate this task as a repeated game embedded in a model predictive control (MPC) scheme.

Formally, a game consists of M players, labeled by the index set $\mathcal{A} := \{1, 2, \ldots, M\}$. Following the notation in [3], we denote the index set of autonomous vehicles using symbol $\mathcal{R} \subseteq \mathcal{A}$, and of human drivers using symbol $\mathcal{H} = \mathcal{A} \setminus \mathcal{R}$. Each player $\nu \in \mathcal{A}$ decides on their control variables $u_k^\nu \in \mathbb{R}^{n_\nu}$ at time step k. For notational convenience, we also introduce $u_k^{-\nu} := (u_k^i)_{i \in \mathcal{A} \setminus \{\nu\}}$ as the $(M-1)$-tuple formed by the input variables of all players except for player ν at time step k and $u_k := (u_k^i)_{i \in \mathcal{A}}$ as the M-tuple formed by the input variables of all players at time step k. The state vector x_k^ν represents the physical state of player ν, which evolves over time according to the dynamics $x_{k+1}^\nu = \varphi_k^\nu(x_k^\nu, u_k^\nu)$. Finally, x_k denotes the full physical state vectors, i.e., the M-tuple of state vectors of all players at time step k.

In order to formulate a game-theoretic description of the control task for the autonomous vehicle, we make the following key assumption.

Assumption II.1. *A human driver $\nu \in \mathcal{H}$ behaves optimally according to a (potentially unknown) receding horizon optimal control cost characterized as a sum of stage costs $\ell_k^\nu(x_k, u_k^\nu, u_k^{-\nu})$, over a prediction horizon of N time steps.*

In particular, we assume that at any time step, each player aims to minimize an additive cost $\sum_{k=0}^{N-1} \ell_k^\nu(x_k, u_k^\nu, u_k^{-\nu}) + \ell_N^\nu(x_N)$ over an N-step prediction horizon, where the stage costs ℓ_k^ν may be functions of the state and control variables of all other players. Furthermore, we impose hard constraints to model physical (actuation) limits and to ensure safety. Here, we make the distinction between three types of constraints. First, input constraints $u_k^\nu \in U^\nu$ representing the physical limitations of the vehicle actuation. For convenience, we will assume that U^ν represents box constraints, i.e.,

$$U^\nu = \left\{ u \in \mathbb{R}^{n_\nu} \mid \underline{b}^\nu \leq u \leq \bar{b}^\nu \right\}. \tag{1}$$

Second, continuously differentiable player-specific constraints $h^\nu(x_k^\nu, u_k^\nu) \leq 0$, such as constraints ensuring that the vehicle remains within the boundaries of the road section, and third, collision avoidance constraints $H(x_k, u_k) \leq 0$, shared by all players.

Since the players are unaware of the cost function and the constraints of the other players, the formulation up to this point corresponds to the formulation of a partially observable repeated game. In order to obtain a more tractable formulation, we introduce the following two additional assumptions.

Assumption II.2. *The autonomous vehicles can either communicate with each other or are in possession of the cost function and the constraints of each other.*

Assumption II.3. *The cost function and constraints of each human driver $\nu \in \mathcal{H}$ can be parametrized by θ^ν, and the resulting parametrized cost function and constraints are representative of the real behavior of this human driver.*

Note that Assumption II.2 could be satisfied in practice rather easily. More restrictive is Assumption II.3, first proposed by Kuderer et al. [7], as it assumes that the full range of possible driving styles can be obtained by varying the parameters θ^ν. Therefore, we need to use a sufficiently expressive function class to represent the human's cost and constraints, such that in principle arbitrarily realistic behavior can be obtained. Our approach towards this parametrization is detailed in Section IV. Under previous assumptions, the game reduces to a fully observable repeated game where each player $\nu \in \mathcal{A}$ is assumed to be solving a coupled discrete-time finite-horizon optimal control problem, described by

$$\begin{aligned} \mathbf{P}^\nu : \underset{\substack{x_1^\nu, \ldots, x_N^\nu, \\ u_0^\nu, \ldots, u_{N-1}^\nu \in U^\nu}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell_k^\nu(x_k, u_k^\nu, u_k^{-\nu}) + \ell_N^\nu(x_N) \\ \text{subject to} \quad & x_{k+1}^\nu = \varphi_k^\nu(x_k^\nu, u_k^\nu), \quad \forall k \in \mathbb{N}_{[0,N-1]}, \\ & h^\nu(x_k^\nu, u_k^\nu) \leq 0, \qquad \forall k \in \mathbb{N}_{[0,N-1]}, \\ & h_{\mathrm{f}}^\nu(x_N^\nu) \leq 0, \\ & H(x_k, u_k^\nu, u_k^{-\nu}) \leq 0, \ \ \forall k \in \mathbb{N}_{[0,N-1]}, \\ & H_{\mathrm{f}}(x_N) \leq 0. \end{aligned} \tag{2}$$

Such a system of coupled optimization problems $\{\mathbf{P}^\nu\}_{\nu \in \mathcal{A}}$ is called a GNEP, which as mentioned before is in general very challenging to solve due to the coupling through both the cost function and the constraints. In the next section, we will impose some additional structure onto the given problem, which will facilitate an efficient solution procedure.

III. Solution of the game formulation

A. Reformulation to standard game-theoretic form

In order to simplify notation in the sequel, we reformulate (2) into a more compact form. To this end, we eliminate the state variables by defining Φ_k^ν, $k \in \mathbb{N}$ as the solution maps to the dynamics φ_k^ν starting from x_0, i.e., $\Phi_0^\nu(u^\nu) := x_0^\nu$ and

$$\Phi_{k+1}^\nu(u^\nu) := \varphi_k^\nu(\Phi_k^\nu(u^\nu), u_k^\nu),$$
$$\Phi_k(u) := \left(\Phi_k^1(u^1), \ldots, \Phi_k^M(u^M)\right),$$

where $u^\nu := (u_0^\nu, \ldots, u_{N-1}^\nu)$ and $u := (u_0, \ldots, u_{N-1})$. We may then define the corresponding single shooting cost function J^ν, the player-specific constraints $\mathbf{h}^\nu \equiv (h_k^\nu)_{k=0}^N$ and shared constraints $\mathbf{H} \equiv (H_k)_{k=0}^N$ as

$$J^\nu(u) := \textstyle\sum_{k=0}^{N-1} \ell_k^\nu(\Phi_k(u), u_k) + \ell_N^\nu(\Phi_N(u^\nu)),$$
$$h_k^\nu(u^\nu) := h^\nu(\Phi_k^\nu(u^\nu), u_k^\nu),\ h_N^\nu(u^\nu) := h_\mathrm{f}^\nu(\Phi_k^\nu(u^\nu)),$$
$$H_k(u) := H(\Phi_k(u), u_k),\ H_N(u) := H_\mathrm{f}(\Phi_N(u)),$$

for $k \in \mathbb{N}_{[0,N]}$. We furthermore introduce the constraint set

$$\mathcal{U}^\nu(u^{-\nu}) = \{u^\nu \in \mathbf{U}^\nu |\ \mathbf{H}(u) \leq 0,\ \mathbf{h}^\nu(u^\nu) \leq 0\}, \quad (3)$$

where $u^{-\nu} := (u_0^{-\nu}, \ldots, u_{N-1}^{-\nu})$ and $\mathbf{U}^\nu := (U^\nu)^N$. Thus, (2) may be compactly written in the standard GNEP form

$$\underset{u^\nu \in \mathcal{U}^\nu(u^{-\nu})}{\text{minimize}} \quad J^\nu(u^\nu, u^{-\nu}), \quad \nu \in \mathcal{A}. \quad (4)$$

We now formally define the sought solution.

Definition III.1 (Generalized Nash equilibrium)**.** *We say that an M-tuple $\bar{u}$ is a generalized Nash equilibrium (GNE) of GNEP (4) if no player ν can decrease their cost by unilateral deviation. That is, for all $\nu \in \mathcal{A}$,*

$$J^\nu(\bar{u}^\nu, \bar{u}^{-\nu}) \leq J^\nu(u^\nu, \bar{u}^{-\nu}), \quad \forall u^\nu \in \mathcal{U}^\nu(\bar{u}^{-\nu}). \quad (5)$$

B. Generalized potential games

In the context of traffic interaction, it is natural to impose some additional structure on the cost functions of the different interacting players. This will allow us to specialize problem (4) to the form of a GPG [8] — a subclass of GNEPs with shared constraints.

Definition III.2 (Generalized potential game)**.** *A GNEP corresponds to a GPG if:*

(i) there exists a nonempty, closed set $\mathcal{U} \subseteq \mathbb{R}^n$, with $n := \sum_{\nu=1}^M n_\nu$, such that for every $\nu \in \mathcal{A}$,

$$\mathcal{U}^\nu(u^{-\nu}) = \left\{u^\nu \in \mathbf{U}^\nu \ \middle|\ (u^\nu, u^{-\nu}) \in \mathcal{U}\right\}; \quad (6)$$

(ii) there exists a continuous function $\mathcal{P} : \mathbb{R}^n \to \mathbb{R}$ such that for every $\nu \in \mathcal{A}$ and for every $u^{-\nu} \in \prod_{i \in \mathcal{A}\setminus\{\nu\}} \mathbf{U}^i$,

$$\tfrac{\partial P(u^\nu, u^{-\nu})}{\partial u^\nu} = \tfrac{\partial J^\nu(u^\nu, u^{-\nu})}{\partial u^\nu}.$$

Definition III.2*(i)* readily follows from the problem setup. Using (3), we define

$$\mathcal{U} = \left\{ u \in \mathbb{R}^n \ \middle|\ \begin{array}{l} \mathbf{H}(u) \leq 0, \\ \mathbf{h}(u) \leq 0 \end{array} \right\},$$

with $\mathbf{h}(u) := \left(\mathbf{h}^1(u^1), \ldots, \mathbf{h}^M(u^M)\right)$. Closedness of $\mathcal{U}$ directly follows from continuity of $\mathbf{h}$ and $\mathbf{H}$.

Definition III.2*(ii)* states that in fact, all players are minimizing the same function $\mathcal{P}(u)$, also called the *potential function* of the game. This trivially holds when the cost function of each player ν can be decomposed as $J^\nu(u) = c(u) + d^\nu(u^\nu)$, where $c(u)$ is a common term and $d^\nu(u^\nu)$ is a player-specific term, such that $\mathcal{P}(u) = c(u) + \sum_{\nu=1}^M d^\nu(u^\nu)$. In our particular application, we can thus design the cost function of each vehicle as the sum of a player-specific term (e.g., tracking errors, control effort, slew rate, …) and a common term (e.g., proximity to other vehicles). Such costs are often used in literature, and they are able to capture most of the behavior of human drivers [7].

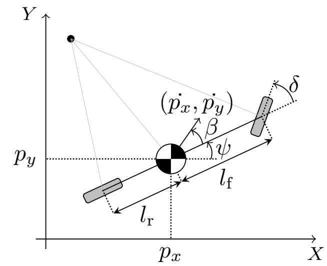

Fig. 1: The kinematic bicycle model of (8).

Under these conditions, finding a GNE of (2) corresponds to solving a single optimal control problem, given by

$$\begin{aligned} \underset{u \in \mathbf{U}}{\text{minimize}} \quad & \mathcal{P}(u) \\ \text{subject to} \quad & \mathbf{H}(u) \leq 0, \\ & \mathbf{h}(u) \leq 0, \end{aligned} \quad (7)$$

where $\mathbf{U} := \prod_{i \in \mathcal{A}} \mathbf{U}^i$ is the Cartesian product of the player-specific actuator bounds.

C. NMPC formulation

1) Dynamics

We will represent the vehicle dynamics using a simple *kinematic bicycle model*(cfr. Fig. 1), i.e.,

$$\begin{aligned} \dot{p_x} &= v\,\mathbf{cos}(\psi + \beta) & \dot{\psi} &= \frac{v}{l_\mathrm{r}}\mathbf{sin}(\beta) \\ \dot{p_y} &= v\,\mathbf{sin}(\psi + \beta) & \dot{v} &= a - \mu v \end{aligned} \quad (8)$$

with $\beta = \mathbf{tan}^{-1}(^{l_r}/_{(l_f+l_r)}\,\mathbf{tan}\,\delta)$, where p_x and p_y denote the position of the center of mass (CM); ψ the heading angle, δ the steering angle, v and a respectively the velocity and the acceleration at the CM, l_r and l_f the distance between the CM and respectively the front and the rear axis, μ the friction coefficient and β the slip angle. The control actions of each driver are the acceleration a and the steering angle δ. We will restrict our attention to the nominal steering region where $\delta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ for which the dynamics is smooth.

2) Collision avoidance constraints

We model each vehicle as a rectangle, represented as

$$O^\nu := \left\{p \in \mathbb{R}^2 \mid p^\top a_i^\nu + b_i^\nu > 0, i \in \mathbb{N}_{[1,4]}\right\}. \quad (9)$$

Thus, for an arbitrary point p,

$$p \notin O^\nu \iff \Psi^\nu(p) = \prod_{i=1}^4 [p^\top a_i^\nu + b_i^\nu]_+ = 0. \quad (10)$$

This is a special case of the obstacle avoidance formulation introduced in [14], [18]. Note that the mapping $\Psi^\nu : \mathbb{R}^2 \to \mathbb{R}$ is such that $\frac{1}{2}\|\Psi^\nu(\cdot)\|^2$ is continuously differentiable with Lipschitz-continuous gradient $\nabla_p \frac{1}{2}\|\Psi^\nu(p)\|^2$, given by

$$\begin{cases} \sum_{i=1}^4 (p^\top a_i^\nu + b_i^\nu) \prod_{j \neq i} (p^\top a_j^\nu + b_j^\nu)^2 a_i^\nu, & \text{if } p \in O^\nu, \\ 0, & \text{otherwise.} \end{cases}$$

We impose collision avoidance between two vehicles by imposing that each corner of every vehicle lies outside the other vehicle using (10). Additionally, we impose that also the nose of each vehicle is also not located within the other vehicle, as this greatly improves performance of our methodology in practice. This yields 10 equality constraints per time step.

3) Solver

We approximately solve (7) by treating the shared constraints using the quadratic penalty method with penalty parameters Σ, such that each subproblem is given by

$$\begin{aligned} \underset{u \in \mathbf{U}}{\text{minimize}} \quad & \mathcal{P}(u) + \frac{1}{2}\|\mathbf{H}(u)\|_{\Sigma}^2 \\ \text{subject to} \quad & \mathbf{h}(u) \leq 0. \end{aligned} \tag{11}$$

The player-specific constraints $\mathbf{h}(u) \leq 0$ are imposed using the augmented Lagrangian method (ALM). If $\mathbf{h}(u)$ is continuously differentiable (as will commonly be the case), then this strategy leads to inner subproblems with a continuously differentiable cost function and simple box constraints, which can be solved approximately using PANOC [19]. If this methodology converges to a feasible pair $(\bar{u}, \bar{y})$, it is an approximate KKT point of (7). Otherwise, it converges to an approximate KKT point of the infeasibility [20].

IV. Online learning of parameters

We have thus far assumed full access to the human cost J^ν and private constraint functions $\mathbf{h}^\nu$, $\nu \in \mathcal{H}$. We now relax this assumption by introducing a practical methodology for learning this information online from observed data. Our approach is similar in spirit to the methodologies proposed in [15]–[17], although there are some key differences.

First of all, we parametrize not only the cost but also the constraints. For each human player $\nu \in \mathcal{H}$, we represent the cost and constraint mappings appearing in (2) in a parametric form, which we denote $\ell_k^\nu(\cdot\,; \theta^\nu), \ell_N^\nu(\cdot\,; \theta^\nu)$, $h^\nu(\cdot\,; \theta^\nu)$, $h_{\mathrm{f}}^\nu(\cdot\,; \theta^\nu)$ and $U^\nu(\theta^\nu)$, with parameter vector $\theta^\nu \in \mathbb{R}^{p_\nu}$. We emphasize that during this design step, care must be taken that III.2*(i)* and III.2*(ii)* hold regardless of the value of θ^ν.

Second, we assume that the observed control actions $\hat{u}_t$ of the other players only correspond to the first entries of their respective optimal control sequences $\bar{u}$. This is motivated by the observation that the remaining $N-1$ controls are open-loop predictions which cannot be observed, as they may significantly differ from the closed-loop policy. Considering again the compact representation (11), we define the Lagrangian

$$\mathcal{L}_\Sigma(u, y) := \mathcal{P}(u) + \langle y, \mathbf{g}(u) \rangle + \frac{1}{2}\|\mathbf{H}(u)\|_{\Sigma}^2, \tag{12}$$

where $\mathbf{g}(u) := \big(\mathbf{h}(u), u - \bar{b}, \underline{b} - u\big)$. Here, we have introduced $\bar{b}$ and $\underline{b}$ as the concatenation of the player-specific actuation bounds $\bar{b}^\nu, \underline{b}^\nu$ in (1), leading to the global box constraints $\mathbf{U} = \{u \in \mathbb{R}^n \mid \underline{b} \leq u \leq \bar{b}\}$. If $\bar{u}$ is a local minimizer of optimization problem (11) for given penalty parameters Σ and a suitable constraint qualification holds at $\bar{u}$ then there exists a $\bar{y}$ such that

$$\nabla_u \mathcal{L}_\Sigma(\bar{u}, \bar{y}; \theta) = 0, \tag{13a}$$

$$\mathbf{g}(\bar{u}; \theta) \leq 0, \tag{13b}$$

$$\bar{y} \geq 0, \tag{13c}$$

$$\bar{y}^\top \mathbf{g}(\bar{u}; \theta) = 0. \tag{13d}$$

The main idea behind the parameter estimation procedure is to select parameter values $\theta := (\theta^i)_{i \in \mathcal{H}}$, such that the observed behavior approximately matches the optimality conditions (13). This naturally leads to the following parameter estimation procedure:

Given the current estimate $\hat{\theta}_t$ for the parameters, set the updated parameters $\hat{\theta}_{t+1}$ as the solution of

$$\begin{aligned} \underset{\theta, u, y}{\text{minimize}} \quad & \left\|\nabla_u \mathcal{L}_{\widetilde{\Sigma}}(u, y; \theta)\right\|_2^2 + r(\theta, \hat{\theta}_t) \\ \text{subject to} \quad & \mathbf{g}(u; \theta) \leq 0, \\ & y \geq 0, \\ & y^\top \mathbf{g}(u; \theta) = 0, \\ & u_0 = \hat{u}_t, \end{aligned} \tag{14}$$

where $r(\theta, \hat{\theta}_t) := \|\theta - \hat{\theta}_t\|_{\Xi}^2$, is a regularization term with parameters $\Xi := \mathbf{diag}(\xi) \in \mathbb{R}_{++}^{p \times p}$ where $p = \sum_{i \in \mathcal{H}} p_i$, which penalizes large deviations between subsequent parameter estimations. The introduction of this term is motivated by the assumption that the human driving style should only change slowly over time. The penalty parameters $\widetilde{\Sigma}$ are selected equal to the values obtained from the solution of (7) by the controlled vehicle at the previous time step.

Optimization problem (14) updates $\hat{\theta}_t$ based on the information gathered during a single time step t of the game. Alternatively, an update can be performed based on the previous L observations $\hat{u}_t^{-\nu}, \ldots, \hat{u}_{t-L+1}^{-\nu}$ by adding the corresponding optimization variables $(u_{(l)}, y_{(l)})$, cost terms $\|\nabla_u \mathcal{L}_{\widetilde{\Sigma}_{(l)}}(u_{(l)}, y_{(l)}; \theta)\|_2^2$ and constraints to the optimization problem, with $l = t - L + 1, \ldots, t$. Using only a single observation, i.e., taking $L = 1$, the parameter estimation problem may admit a large number of (approximate) solutions, as a great deal of freedom remains to choose both the completion of the observed control action over the horizon as well as the parameters θ.

As L is increased and more state-input pairs (with shared θ) are taken into account, the ambiguity in the optimal parameter choices is expected to decrease (at the cost of increased computational complexity).

It is well-known that parameter estimation tasks of this form may inherently suffer from some identifiability issues, as some parameters may be invisible to the learning methodology [21]. Indeed, there may be a large set of parameter values for which the observed sequence optimizes the GPG, locking the learning methodology onto any of these values, regardless of the amount of data [22, Sec. 6.1.2]. Fortunately, our goal is not to arrive at the exact parameters, but rather at any parameter vector that is consistent with the observed driving behavior. The numerical experiments of Section V show that although parameter estimates in general do not converge to the true values, the closed loop behavior of the controlled vehicle is still significantly improved when using online learning.

Remark 1. *Optimization problem (14) can be regarded as a hierarchical optimization problem, where the upper level consist of estimating the parameters θ and the lower level is an optimal control problem, which can be exploited to design dedicated solution methodologies [15]. However, in this work, we opt for the controlled vehicle to solve this optimiza-*

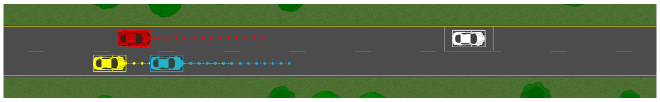

Fig. 2: The merging scenario setup.

tion problem directly using PANOC, *where we address both the primal feasibility constraints and the complementarity constraints using ALM. To warm start learning procedure* (14) *for an autonomous vehicle ν, we use the pair $(\bar{u}, \bar{y})$ which this vehicle has obtained by solving the GPG under its current estimation of the human parameters.*

V. CASE STUDIES

A. Implementation details

All involved optimization problems are implemented using CasADi [23] in Python and solved in real-time using ALPAQA [24] on a 1.7GHz AMD Ryzen 7 PRO 4750U processor with 32 GB RAM (for horizon length $N = 15$ and $T_{\mathrm{s}} = 0.2\,\mathrm{s}$). The source code of the driving simulator and videos of the experiments are available at https://brechtevens.github.io/GPG-drive.

B. Scenario setup

We focus on a merging scenario involving three moving vehicles and one stationary vehicle with dimensions $l_{\mathrm{r}} = 2\,\mathrm{m}$, $l_{\mathrm{f}} = 2\,\mathrm{m}$ and width $2\,\mathrm{m}$. The initial situation is illustrated in Fig. 2, where the controlled vehicle, i.e., the red vehicle r, attempts to merge between the yellow vehicle y and the blue vehicle b. We model this scenario as a two-player game as summarized in Table I: the blue vehicle is considered as an obstacle with constant velocity, as is commonly done in traffic modelling on highways [25]. Cost functions are

TABLE I: Characteristics and initial states of the vehicles.

Vehicle	Type	Dynamics	Initial states			
			$p_{x,0}$	$p_{y,0}$	ψ_0	v_0
Red	GPG	Kinematic bicycle	3	3	0	5
Yellow	GPG	Double integrator	0	0	0	5
Blue	Simple	Constant velocity	7	0	0	5
White	Simple	Stationary obstacle	45	3	0	0

selected for both players which only contain player-specific cost terms and common cost terms, guaranteeing that III.2*(ii)* is satisfied. For the controlled vehicle we propose the following player-specific cost terms:

$$\ell_k^{\mathrm{r}} := \theta_1^{\mathrm{r}}(p_{y,k}^{\mathrm{r}})^2 + \theta_2^{\mathrm{r}}(a_k^{\mathrm{r}})^2 + \theta_3^{\mathrm{r}}(\delta_k^{\mathrm{r}})^2, \tag{15a}$$

$$\ell_N^{\mathrm{r}} := \theta_1^{\mathrm{r}}(p_{y,N}^{\mathrm{r}})^2. \tag{15b}$$

For the yellow vehicle, we assume a quadratic cost on deviations from the desired distance from the blue vehicle:

$$\ell_k^{\mathrm{y}} := \theta_1^{\mathrm{y}}(p_{x,k}^{\mathrm{b}} - p_{x,k}^{\mathrm{y}} - d_{\mathrm{des}}^{\mathrm{y}})^2 + \theta_2^{\mathrm{y}}(a_k^{\mathrm{r}})^2, \tag{16a}$$

$$\ell_N^{\mathrm{y}} := \theta_1^{\mathrm{y}}(p_{x,k}^{\mathrm{b}} - p_{x,k}^{\mathrm{y}} - d_{\mathrm{des}}^{\mathrm{y}})^2, \tag{16b}$$

where $d_{\mathrm{des}}^{\mathrm{y}} = 3$. The ground-truth values for the parameters of the red and yellow vehicle can be found in Table II, where different parameters are introduced to model either courteous or stubborn behavior of the yellow vehicle. To

TABLE II: Cost function parameters.

Vehicle	behavior	Parameters
Red vehicle	-	$\theta^{\mathrm{r}} = \begin{bmatrix}0.0.05 & 0.1 & 0.5\end{bmatrix}^\top$
Yellow vehicle	Stubborn	$\theta^{\mathrm{y}} = \begin{bmatrix}10 & 0.1\end{bmatrix}^\top$
Yellow vehicle	Courteous	$\theta^{\mathrm{y}} = \begin{bmatrix}0.02 & 0.1\end{bmatrix}^\top$

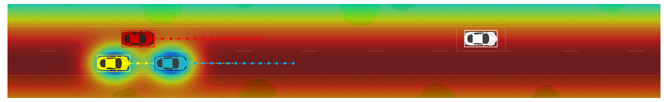

Fig. 3: Stage cost for the controlled vehicle.

model the human tendency to keep a certain safety distance, we add Gaussian cost terms penalizing driving near another vehicle [3], [6], i.e., $c(u) = \kappa e^{-d^\top K d/2}$, where d denotes the difference between the positions p of two different players, with $\kappa = 4$ and $K = \mathbf{diag}(4, 2.25)$. The resulting cost function of the controlled vehicle is visualized in Fig. 3.

Both players attempt to avoid collisions and player-specific boundary constraints have been added to the optimal control problem of the red vehicle to ensure that this vehicle remains within the boundaries of the road section. Finally, the inputs of both players are upper and lower bounded.

C. Numerical results

The introduced GPG formulation is solved for the first 55 time steps. We solve (7) for feasibility 10^{-2} and tolerance 10^{-3}. The yellow vehicle has a perfect estimation of the parameters of the controlled vehicle, but the controlled vehicle might have a wrong belief about the parameters θ^{y}. The yellow vehicle either behaves courteously or stubbornly (cf. Table II). In the first run, we simply set the parameter estimates of the controlled vehicle fixed to either of the corresponding sets of values (we refer to this as its *belief*). The obtained closed loop simulations corresponding to these 4 different combinations are visualized in Fig. 4 at different time steps. Consider the scenario where the yellow vehicle behaves courteously (Fig. 4, top rows). Then, it will tend to leave space for the red vehicle to merge into the right lane. If the red vehicle is aware of this courteous behavior, it merges into the right lane accordingly. However, if it wrongfully anticipates stubborn behavior, it will not attempt to merge into the right lane and keep driving next to the other two vehicles. When approaching the white vehicle, it settles to the strategy of decelerating and merging behind the yellow vehicle. This is suboptimal but not fatal, and accordingly marked in orange.

On the other hand, consider the scenario in which the yellow vehicle is stubborn (Fig. 4, bottom rows). If the red vehicle is aware of this behavior, it does not attempt to merge into the right lane until the yellow vehicle has passed. However, under the *courteous* belief, it plans a future open-loop trajectory which merges into the other lane, (wrongfully) assuming that the yellow vehicle will make space for this manoeuvre. During this manoeuvre, collision

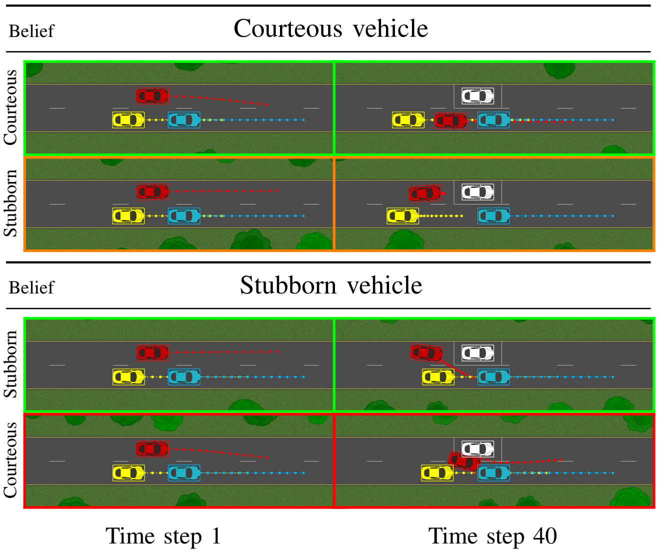

Fig. 4: The planned trajectories at respectively the 1st and 40th time step of the game without online learning.

avoidance constraints are violated, leading to undesired and dangerous behavior (marked in red).

It is apparent that wrong parameter estimates may prohibit safe deployment of a control scheme of this type. We now perform the same simulations using the online learning scheme (14), which continuously updates the estimate of θ^{y}, with $L = 1$ and regularization $\xi = 0.5$. At the first time step, as there are no observations available yet, the open-loop sequence predicted by the red vehicle is equal to the open-loop sequence obtained previously. On the other hand, the behavior at the 40th time step has significantly improved and the previously observed suboptimal/dangerous maneuvers no longer occur.

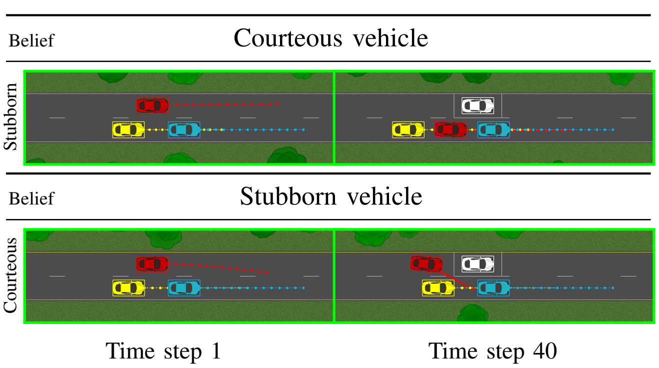

Fig. 5: The planned trajectories at respectively the 1st and 40th time step of the game when using online learning.

Fig. 6 shows the evolution of the parameter estimate θ_1^{y} over time. As discussed in Section IV, we do not observe convergence to the true parameter value (which is to be expected whenever the control signals are not sufficiently exciting). Nevertheless, parameter estimates are obtained which better explain previous observations, resulting in significantly improved behavior. This conclusion is corroborated by Table III, which provides the numerical values for the closed-loop potential and the maximum constraint violation. For

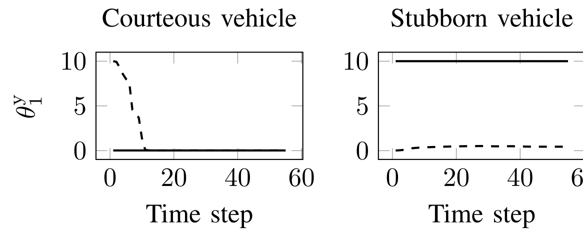

Fig. 6: The estimate of θ_1^{y} over time, where the full lines denote experiments with correct initial estimate and the dashed lines denote experiments with incorrect initial estimate.

TABLE III: The closed-loop potential and maximum constraint violation.

Behavior	Belief	Potential		Violation	
		Simple	Learning	Simple	Learning
Courteous	Courteous	36.18	36.17	0.000	0.000
Courteous	Stubborn	74.55	35.96	0.000	0.000
Stubborn	Courteous	125.50	44.47	3.667	0.009
Stubborn	Stubborn	44.12	44.13	0.010	0.010

both experiments with incorrect initial belief, introducing the parameter estimation scheme results in significantly improved closed-loop potential, approaching the value for the respective experiment with correct initial belief. Furthermore, the maximum constraint violation is smaller than the imposed tolerance 10^{-2}. Note that for both experiments with correct initial belief, the closed-loop potential and maximum constraint violation are (approximately) unaffected by the learning methodology. Finally, Table IV provides the computation times for the controlled vehicle, illustrating the real-time capabilities of the overall scheme, as the maximum computation time for solving the GPG and performing online learning combined are below our sampling period $T_{\mathrm{s}} = 0.2\,\mathrm{s}$.

VI. Conclusion

We presented an interaction-aware MPC strategy for automated driving under shared collision avoidance constraints by formulating the problem as a GPG, which depends on implicit preferences and constraints of surrounding drivers. We present a simple but effective scheme which allows the controller to learn these preferences online using observed behavior. Through numerical simulations, we have illustrated the benefits of the learning scheme, as well as the potential real-time capabilities of the proposed methodology.

TABLE IV: The computation times of the red vehicle.

Behavior	Belief	GPG [s]		Update θ^{y} [s]	
		Max	Avg	Max	Avg
Courteous	Courteous	0.013	0.003	0.006	0.002
Courteous	Stubborn	0.094	0.013	0.103	0.026
Stubborn	Courteous	0.037	0.012	0.102	0.043
Stubborn	Stubborn	0.060	0.013	0.102	0.031

REFERENCES

[1] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, 2010.

[2] R. Lachner, M. H. Breitner, and H. J. Pesch, "Real-Time Collision Avoidance: Differential Game, Numerical Solution, and Synthesis of Strategies," in *Advances in Dynamic Games and Applications* (J. A. Filar, V. Gaitsgory, and K. Mizukami, eds.), (Boston, MA), pp. 115–135, Birkhäuser, 2000.

[3] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, 2018.

[4] A. Liniger and J. Lygeros, "A Noncooperative Game Approach to Autonomous Racing," *IEEE Transactions on Control Systems Technology*, pp. 1–14, 2019.

[5] G. Debreu, "A Social Equilibrium Existence Theorem," *Proceedings of the National Academy of Sciences*, 1952.

[6] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, vol. 1, pp. 41–48, 2012.

[7] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 2641–2646, 2015.

[8] D. Monderer and L. S. Shapley, "Potential Games," *Games and Economic Behavior*, vol. 14, pp. 124–143, May 1996.

[9] J. B. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.

[10] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games," *2020 IEEE International Conference on Robotics and Automation, ICRA 2020*, pp. 1475–1481, 2020.

[11] S. L. Cleac'h, M. Schwager, and Z. Manchester, "ALGAMES: A Fast Solver for Constrained Dynamic Games," *Robotics: Science and Systems XVI*, July 2020.

[12] A. Dreves and M. Gerdts, "A generalized Nash equilibrium approach for optimal control problems of autonomous cars," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 326–342, 2018.

[13] A. Britzelmeier and A. Dreves, "A decomposition algorithm for Nash equilibria in intersection management," *Optimization*, vol. 70, pp. 2441–2478, Nov. 2021.

[14] B. Hermans, P. Patrinos, and G. Pipeleers, "A Penalty Method Based Approach for Autonomous Navigation using Nonlinear Model Predictive Control," *IFAC-PapersOnLine*, vol. 51, pp. 234–240, 2018.

[15] K. Hatz, J. P. Schlöder, and H. G. Bock, "Estimating parameters in optimal control problems," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. 1707–1728, 2012.

[16] M. Menner and M. N. Zeilinger, "Maximum Likelihood Methods for Inverse Learning of Optimal Controllers," *IFAC-PapersOnLine*, vol. 53, pp. 5266–5272, Jan. 2020.

[17] L. Peters, D. Fridovich-Keil, V. Rubies-Royo, C. Tomlin, and C. Stachniss, "Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations," in *Robotics: Science and Systems XVII*, Robotics: Science and Systems Foundation, July 2021.

[18] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," *2018 European Control Conference, ECC 2018*, pp. 1523–1528, 2018.

[19] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1939–1944, Dec. 2017.

[20] P. Sopasakis, E. Fresk, and P. Patrinos, "OpEn: Code Generation for Embedded Nonconvex Optimization," *IFAC-PapersOnLine*, vol. 53, pp. 6548–6554, Jan. 2020.

[21] T. Söderström and L. Ljung, *Theory and Practice of Recursive Identification*. Signal Processing, Optimization, and Control, Cambridge, MA, USA: MIT Press, Oct. 1983.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control. Vol. 1 Vol. 1*. Belmont, Mass.: Athena Scientific, 2005.

[23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, Mar. 2019.

[24] P. Pas, M. Schuurmans, and P. Patrinos, "Alpaqa: A matrix-free solver for nonlinear MPC and large-scale nonconvex optimization," *arXiv:2112.02370 [cs, eess, math]*, Dec. 2021.

[25] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological*, vol. 20, pp. 403–414, Oct. 1986.