CONFIDENTIAL. Limited circulation. For review only.

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

Brecht Evens Mathijs Schuurmans Panagiotis Patrinos

***Abstract*— We present a novel control strategy for controlling autonomous vehicles in general traffic situations which accounts for the mutual interactions between the controlled vehicle and other road users. More specifically, the interaction is modelled as a generalized potential game, where each road user is assumed to minimize a shared cost function subject to shared (collision avoidance) constraints. The shared cost allows the controlled vehicle to cooperate with other road users, while safety guarantees follow from the imposed hard collision avoidance constraints and the introduction of a model predictive control feedback scheme. In the case where the incentives and constraints of other road users, i.e., human drivers, are unknown, we propose a natural and practical methodology for learning this information online from observed data and incorporating it directly into the solution methodology for the game formulation. Extensive numerical simulations in a realistic highway merging scenario have been performed, verifying the practical usability of the developed methodologies.**

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 simple assumptions about their drivers. These predictions are then fixed and fed into a planning module. For instance, one may assume that the vehicles will maintain a constant velocity [1], that they behave as bounded disturbances [2] or that they follow one of a number of predetermined trajectories [3]. This approach thus assumes the trajectories of surrounding vehicles to be independent of the autonomous agent, reducing the interaction to an obstacle avoidance problem.

Since the mutual interactions between the autonomous vehicle and the surrounding traffic are neglected, this type of formulation may lead to overly conservative and defensive behavior, also referred to as the *frozen robot problem* [4]. In dense traffic situations, for instance, such a vehicle will struggle to merge into lanes efficiently or be unable to cross a condensed junction, even though in practice, feasible solutions may very well exist.

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

Early work on social navigation [5] proposed to solve the frozen robot problem by modelling vehicles as a team of cooperative players engaging in joint collision avoidance, working together to make sure that each player reaches their destination while preventing collisions. More generally, the control of autonomous vehicles can be related to other problems in game theory besides cooperative games 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 [6]. However, due to their pessimistic nature, these games again suffer from the frozen robot problem to some extent. Stackelberg games can be used to model interactions where one driver, i.e., the leader, dominates the decision process [7], [8]. The interactions in such a model are only regarded in a one-way sense, i.e., the effect of the leader's actions on the behavior of the followers. 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 in driving scenarios, as the leader (typically the autonomous vehicle) will assume that the followers can anticipate its planned trajectory accurately. This assumption allows for overly confident execution of certain maneuvers which, in reality, the followers may not be able to account for.

In this work, we aim to account for the mutual interactions between the controlled vehicle and other road users by addressing the interaction modelling problem as a cooperative game, following the previously mentioned work on social navigation. To this end, we model each road user as an agent that optimally plans its future trajectory by solving an optimal control problem, while attempting to satisfy constraints ensuring safe control. These constraints are called coupled constraints since they depend on the actions of the other players, and they typically include collision avoidance constraints. 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) [9] and is in general notoriously challenging to solve [10]. 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. Such knowledge can be reasonably assumed when only autonomous vehicles are present on the road, through a vehicle-to-vehicle communication scheme [11]. However,

Manuscript 385 submitted to 2022 European Control Conference (ECC).
Received November 8, 2021.

when human drivers are considered, this information either needs to be manually designed or inferred from observed behavior. One possible solution is to introduce a universal human model, which is assumed to hold for all humans on the road. Of course, this assumption does not hold in general because different drivers behave differently on the road, or the behavior of a single driver might even change over time. This issue can be addressed in a data-driven manner: the human objective function and constraints are parametrized, and the parameters are estimated in an online fashion [7]. In this work, we aim to design a natural and practical methodology for learning this information online from observed data.

Our contributions are twofold:

a) A penalty method for interaction-aware motion planning: We formalize driving interaction with mutual collision avoidance as a GNEP with shared constraints. Even though GNEPs have been studied since 1952, when they were first introduced by Debreu [9], the search for efficient solution methods for the most general form is largely incomplete and many solution algorithms are not globally convergent under reasonable assumptions. As we will show, however, the structure of our model allows us to restrict our attention to a specific subset of GNEPs with shared constraints, namely generalized potential games (GPGs) [12], for which we can find a normalized Nash equilibrium [13] by solving a single nonlinear optimal control problem (OCP) directly. 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. To tractably represent the collision avoidance constraints, we extend the penalty method based approach for obstacle avoidance from [14] to the multi-agent case.

b) Online learning methodology based on an optimal control model: Since the parameters of the human drivers are in general unknown, we propose an online learning methodology to update the current estimation of these parameters by adapting standard inverse optimal control methodologies such as [15] to our game-theoretical framework, allowing the controlled vehicle to learn from examples. We will show that this approach performs well in numerical simulations, as the controlled vehicle is able to successfully estimate the parameters of the surrounding traffic and adjust its behavior accordingly.

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 the operation $[\cdot]_+ := \mathbf{max}\{0, \cdot\}$, where the $\mathbf{max}$ is interpreted elementwise. Given a set U, we denote $U^N := U \times \cdots \times U$ as the N-fold Cartesian product with itself, for $N \in \mathbb{N} \setminus \{0\}$. Let $\|x\|_Q^2 = x^\top Q x$ denote the weighted square norm, with $Q \succ 0$.

II. Problem statement

We consider the task of controlling a single autonomous vehicle, which interacts with human drivers (and optionally also other autonomous vehicles) in general traffic environments. We formalize this task as a repeated game, where each player represents a traffic participant who is optimizing an objective function over a finite prediction horizon in a receding horizon fashion.

Formally, a game consists of M players, labeled by the index set $\mathcal{A} := \{1, 2, \ldots, M\}$. Furthermore, following the notation in [7], we denote the index set of autonomous vehicles using symbol $\mathcal{R} \subseteq \mathcal{A}$, and of human drivers using symbol $\mathcal{H} \subseteq \mathcal{A}$. 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

$$x_{k+1}^\nu = \varphi_k^\nu \left(x_k^\nu, u_k^\nu\right). \tag{1}$$

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 2.1: 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 in general, the stage costs ℓ_k^ν may be functions of the state and control variables of all other players in the game.

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.,

$$U^\nu = \{u \in \mathbb{R}^{n_\nu} \mid \underline{b}^\nu \leq u \leq \bar{b}^\nu\}. \tag{2}$$

Second, continuously differentiable player-specific constraints $h^\nu(x_k^\nu, u_k^\nu) \leq 0$, such as boundary 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.

The controlled vehicle $\nu \in \mathcal{R}$ needs to solve this partially observable repeated game over the control horizon N in order to obtain its corresponding optimal open-loop control

Manuscript 385 submitted to 2022 European Control Conference (ECC).
Received November 8, 2021.

sequence u^ν. In order to obtain a more tractable formulation, we introduce the following two additional assumptions.

Assumption 2.2: The autonomous vehicles can either communicate with each other or are in possession of the cost function and the constraints of each other.

Assumption 2.3: The cost function and constraints of each human driver $\nu \in \mathcal{H}$ can be parametrized by a set of fixed parameters θ^ν, and the resulting parametrized cost function and constraints are representative of the real behavior of this human driver.

Note that Theorem 2.2 could be satisfied in practice rather easily. Much more stringent is Theorem 2.3, first proposed by Kuderer et al. [16], as it assumes that human preferences are fixed over time and that the full range of possible driving styles can be obtained by varying the parameters θ^ν. In order to relax this assumption, we will update the estimation of the human parameters during online operation, allowing their preferences to shift over time. Moreover, by using a sufficiently expressive function class to represent the human's cost and constraints, arbitrarily realistic behavior can in principle be obtained, potentially at the expense of increased computational complexity. We detail our approach towards this issue 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

$$
\begin{aligned}
\mathbf{P}_\nu : \underset{\substack{x_1^\nu,\dots,x_N^\nu,\\ u_0^\nu,\dots,u_{N-1}^\nu}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \ell_k^\nu(x_k, u_k^\nu, u_k^{-\nu}) + \ell_N^\nu(x_N) \\
\text{subject to} \quad & x_{k+1}^\nu = \varphi_k^\nu(x_k^\nu, u_k^\nu) && \forall k \in \mathbb{N}_{[0,N-1]}, \\
& h^\nu(x_k^\nu, u_k^\nu) \le 0, && \forall k \in \mathbb{N}_{[0,N-1]}, \\
& h_{\mathrm{f}}^\nu(x_N^\nu) \le 0, \\
& H(x_k, u_k^\nu, u_k^{-\nu}) \le 0, && \forall k \in \mathbb{N}_{[0,N-1]}, \\
& H_{\mathrm{f}}(x_N) \le 0, \\
& u_k^\nu \in U^\nu, && \forall k \in \mathbb{N}_{[0,N-1]}.
\end{aligned} \tag{3}
$$

Such a system of coupled optimization problems $\{\mathbf{P}_\nu\}_{\nu\in\mathcal{A}}$ is called a generalized Nash equilibrium problem or GNEP. Correspondingly, we refer to a solution as a generalized Nash equilibrium (GNE). In general, a GNEP is very challenging to solve [10], as the optimization problems of the different players are coupled through both their cost function and their 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 (3) into a more compact form. To this end, we eliminate the state variables from the problem by defining Φ_k^ν, $k \in \mathbb{N}$ as the solution maps to the dynamics (1) starting from x_0. That is,

$$
\begin{aligned}
\Phi_0^\nu(u^\nu) &:= x_0^\nu, \\
\Phi_{k+1}^\nu(u^\nu) &:= \varphi_k^\nu\left(\Phi_k^\nu(u^\nu), u_k^\nu\right), \\
\Phi_k(u) &:= \left(\Phi_k^1\left(u^1\right), \dots, \Phi_k^M\left(u^M\right)\right),
\end{aligned}
$$

where $u^\nu := \left(u_0^\nu, \dots, u_{N-1}^\nu\right)$ and $u := \left(u_0, \dots, u_{N-1}\right)$. 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

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

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

$$
\mathcal{U}^\nu(u^{-\nu}) = \left\{ u^\nu \in \mathbf{U}^\nu \;\middle|\; \begin{matrix} \mathbf{H}(u) \le 0 \\ \mathbf{h}^\nu(u^\nu) \le 0 \end{matrix} \right\}, \tag{5}
$$

where $u^{-\nu} := \left(u_0^{-\nu}, \dots, u_{N-1}^{-\nu}\right)$ and $\mathbf{U}^\nu := (U^\nu)^N$. Thus, (3) may be compactly written in the standard form of a GNEP, given by

$$
\underset{u^\nu \in \mathcal{U}^\nu(u^{-\nu})}{\mathbf{minimize}} \; J^\nu(u^\nu, u^{-\nu}), \qquad \nu \in \mathcal{A}. \tag{6}
$$

We now formally define the sought solution.

Definition 3.1 (Generalized Nash equilibrium): We say that an M-tuple $\bar{u}$ is a GNE of GNEP (6) if no player ν can decrease his/her cost by unilateral deviation, that is, for all $\nu \in \mathcal{A}$,

$$
J^\nu(\bar{u}^\nu, \bar{u}^{-\nu}) \le J^\nu(u^\nu, \bar{u}^{-\nu}), \qquad \forall u^\nu \in \mathcal{U}^\nu(\bar{u}^{-\nu}). \tag{7}
$$

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 (6) to the form of a GPG [12] — a subclass of GNEPs with shared constraints.

Definition 3.2 (Generalized potential game[1]): A GNEP corresponds to a GPG if:

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

$$
\mathcal{U}^\nu\left(u^{-\nu}\right) = \left\{ u^\nu \in \mathbf{U}^\nu \mid \left(u^\nu, u^{-\nu}\right) \in \mathcal{U} \right\}; \tag{8}
$$

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

$$
\begin{aligned}
\mathcal{P}\left(\mathbf{v}^\nu, u^{-\nu}\right) - \mathcal{P}\left(\mathbf{w}^\nu, u^{-\nu}\right) = J^\nu\left(\mathbf{v}^\nu, u^{-\nu}\right) \\
- J^\nu\left(\mathbf{w}^\nu, u^{-\nu}\right).
\end{aligned} \tag{9}
$$

Condition (a) readily follows from the problem setup. Using (5), we define

$$
\mathcal{U} = \left\{ u \in \mathbb{R}^n \;\middle|\; \begin{matrix} \mathbf{H}(u) \le 0 \\ \mathbf{h}(u) \le 0 \end{matrix} \right\}, \tag{10}
$$

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

Manuscript 385 submitted to 2022 European Control Conference (ECC).
Received November 8, 2021.

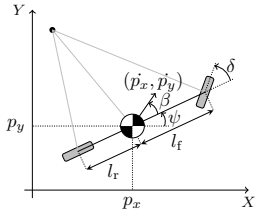

Fig. 1: The kinematic bicycle model of (12).

Condition (b) states that in fact, all players are minimizing the same function $\mathcal{P}(u)$, also called the *potential function* of the game. A common case for which this condition can be verified is 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 cost term and $d^\nu(u^\nu)$ is a player-specific cost term. In this case, a suitable potential function is given by $\mathcal{P}(u) = c(u) + \sum_{\nu=1}^{M} d^\nu(u^\nu)$. This restriction on the cost functions of the autonomous vehicles and human drivers thus needs to be taken into account during the control design. For example, the cost function of each vehicle could be composed of a player-specific cost term (e.g., tracking errors, control effort, slew rate, ...) and a common cost term (e.g., proximity to other vehicles). Such cost function features are often used in literature, and they are able to capture most of the behavior of human drivers [16].

Under these conditions, finding a GNE of (3) corresponds to solving a single optimal control problem, given by

$$\begin{aligned} \underset{u \,\in\, \mathbf{U}}{\text{minimize}} \quad & \mathcal{P}(u) \\ \text{subject to} \quad & \mathbf{H}(u) \leq 0, \\ & \mathbf{h}(u) \leq 0. \end{aligned} \tag{11}$$

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* as visualized by fig. 1, i.e., using the following non-linear ordinary differential equations

$$\begin{aligned} \dot{p_x} &= v\,\mathbf{cos}\,(\psi + \beta) \qquad \dot{\psi} = \frac{v}{l_r}\mathbf{sin}\,(\beta) \\ \dot{p_y} &= v\,\mathbf{sin}\,(\psi + \beta) \qquad \dot{v} = a - \mu v \\ \beta &= \mathbf{tan}^{-1}\left(\frac{l_r}{l_f + l_r}\,\mathbf{tan}\,\delta\right) \end{aligned} \tag{12}$$

where coordinates p_x and p_y denote the position of the centre of mass of the vehicle relative to the inertial reference frame (X, Y); ψ the heading angle of the vehicle, δ the steering angle, v the velocity at the centre of gravity, a the acceleration at the centre of gravity, l_r and l_f the distance between the centre of gravity and respectively the front and the rear axis, μ the friction coefficient and β the slip angle. Each driver can control the motion of their corresponding vehicle using 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 are smooth.

2) Collision avoidance constraints: For the purpose of collision-avoidance, we model each vehicle as a rectangular obstacle, which we represent as the intersection of four open halfspaces

$$O^\nu := \{p \in \mathbb{R}^2 : p^\top a_i^\nu + b_i^\nu > 0, i \in \mathbb{N}_{[1,4]}\}. \tag{13}$$

An arbitrary point p is not located within this rectangle, i.e., $p \notin O^\nu$, if and only if

$$\Psi^\nu(p) = \prod_{i=1}^{4} [p^\top a_i^\nu + b_i^\nu]_+ = 0. \tag{14}$$

This is a special case of the obstacle avoidance formulation introduced in [14], [18], [19]. 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

$$\begin{cases} \sum_{i=1}^{4} (p^\top a_i^\nu + b_i^\nu) \prod_{j \neq i} (p^\top a_j^\nu + b_j^\nu)^2 a_i^\nu, & \text{if } p \in O^\nu, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

An efficient methodology for collision avoidance between two vehicles using (14) is by enforcing that each corner of each vehicle is not located within the other vehicle. 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. In total, this leads to 10 equality constraints at each time step.

3) Solver: To solve optimization problem (11), we address the shared collision avoidance constraints from previous section using the quadratic penalty method with penalty parameters Σ, where each subproblem is given by:

$$\begin{aligned} \underset{u \,\in\, \mathbf{U}}{\text{minimize}} \quad & \mathcal{P}(u) + \frac{1}{2}\|\mathbf{H}(u)\|_\Sigma^2 \\ \text{subject to} \quad & \mathbf{h}(u) \leq 0. \end{aligned} \tag{16}$$

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 [20], an efficient first-order algorithm suited for large-scale embedded nonlinear model predictive control (NMPC) applications. It can be shown that if this methodology converges to a feasible pair $(\bar{u}, \bar{y})$, it is an approximate KKT point of (11) [21]. Otherwise, it converges to an approximate KKT point of the infeasibility [22].

IV. Online learning of parameters

Up to this point, we have assumed that the controlled vehicle has access to the cost functions J^ν and player-specific constraints $\mathbf{h}^\nu$ of all human drivers $\nu \in \mathcal{H}$. We now relax this assumption by proposing a natural and practical methodology for learning this information online from observed data based on the optimality conditions of the GPG. Our approach is similar in spirit to the methodology proposed in Hatz et

Manuscript 385 submitted to 2022 European Control Conference (ECC).
Received November 8, 2021.

al. [15] for parameter estimation in regular optimal control problems.

For each human player $\nu \in \mathcal{H}$, we represent the cost and constraint mappings appearing in (3) 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 $\theta^\nu \in \mathbb{R}^{p_\nu}$. This trivially subsumes the common choice of writing the costs as a linear combination of predetermined *features* [16]

$$\ell_k^\nu(x_k, u_k^\nu, u_k^{-\nu}; \theta^\nu) := \sum_{i=1}^{p_{\text{cost}}} \theta_i^\nu \phi_i(x_k, u_k^\nu, u_k^{-\nu}).$$

We emphasize that while designing such features, care must be taken that Conditions (a) and (b) of Theorem 3.2 hold regardless of the value of θ^ν.

Considering again the compact representation (16), we define the Lagrangian

$$\mathcal{L}_\Sigma(u, y) := \mathcal{P}(u) + \langle y, \mathbf{g}(u)\rangle + \frac{1}{2}\|\mathbf{H}(u)\|_\Sigma^2, \tag{17}$$

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 (2), leading to the global box constraints $\mathbf{U} = \{u \in \mathbb{R}^n \mid \underline{b} \le u \le \bar{b}\}$. If $\bar{u}$ is a local minimizer of optimization problem (16) for given penalty parameters Σ and a suitable constraint qualification holds at $\bar{u}$ then there exists a $\bar{y}$ such that

$$\nabla_u \mathcal{L}_\Sigma(\bar{u}, \bar{y}; \theta) = 0, \tag{18a}$$
$$\mathbf{g}(\bar{u}; \theta) \le 0, \tag{18b}$$
$$\bar{y} \ge 0, \tag{18c}$$
$$\bar{y}^\top \mathbf{g}(\bar{u}; \theta) = 0. \tag{18d}$$

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 (18). To this end, we need to assume that the controlled vehicle ν is able to observe the control actions of all other agents at each time step t. In practice, this information would typically need to be inferred from state observations. We leave the development of such an estimation procedure for future work.

Under previous assumptions, the observed control actions $\hat{u}_t$ of all players correspond to the first entries of their respective optimal control sequences $\bar{u}$. 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

$$\begin{aligned} \underset{\theta, u, y}{\text{minimize}} \quad & \Big\|\nabla_u \mathcal{L}_{\widetilde{\Sigma}}(u, y; \theta)\Big\|_2^2 + r(\theta, \hat{\theta}_t) \\ \text{subject to} \quad & \mathbf{g}(u; \theta) \le 0, \\ & y \ge 0, \\ & y^\top \mathbf{g}(u; \theta) = 0, \\ & u_0 = \hat{u}_t, \end{aligned} \tag{19}$$

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. Here, we take the penalty parameters obtained by the controlled vehicle when solving (11) at the previous time step for $\widetilde{\Sigma}$. Note that we optimize over all control sequences u^ν of all players $\nu \in \mathcal{A}$ starting at the observed control actions $\hat{u}_t^\nu$, as the planned open loop control sequences cannot be observed.

Optimization problem (19) updates the estimated parameters $\hat{\theta}_t$ 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 $\hat{u}_t^{-\nu}, \dots, \hat{u}_{t-L+1}^{-\nu}$ 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 $l = t - L + 1, \dots, t$. 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). We do note, that it is well-known that parameter estimation tasks of this form inherently suffer from some degree of ill-posedness, as some parameters may be invisible to the learning methodology due to identifiability issues [23]. 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 length of the gathered state-input pairs [24, §6.1.2]. Fortunately, however, our goal is not to arrive at the exact parameters, but rather at any parameter vector that is consistent with the observed driving behavior. In the numerical experiments of Section V we will see 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 (19) 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 optimization problem directly using PANOC, where we address both the primal feasibility constraints and the complementarity constraints using ALM. To warm start learning procedure (19) 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

The optimal control problems and the learning problem of the players are implemented using the CasADi library for

automatic differentiation [25] in Python and solved using PANOC, interfaced through the ALPAQA library[2]. The involved optimization problems are solved in real-time on a 1.7GHz AMD Ryzen 7 PRO 4750U processor with 32 GB RAM for horizon length $N = 15$ and $T_s = 0.2\,\mathrm{s}$, where the dynamics are discretized using an explicit fourth-order Runge-Kutta method. The source code of the driving simulator and videos of the experiments performed in this section are available at https://brechtevens.github.io/GPG-drive.



Fig. 2: The merging scenario setup.

B. Scenario setup

We will focus on a merging scenario consisting of three moving vehicles and one stationary parked vehicle with dimensions $l_r = 2\,\mathrm{m}$, $l_f = 2\,\mathrm{m}$ 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 by modelling the blue vehicle as an obstacle with constant velocity, which is commonly performed in traffic modelling on highways [26]. Note that a double integrator model has been used for the dynamics of the yellow vehicle to make explicit that this human driver has no intention to change lanes.

TABLE I: The characteristics and initial states of the vehicles in the merging scenario.

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

Cost function features are selected for both players which only contain player-specific cost terms and common cost terms, guaranteeing that condition (b) of Theorem 3.2 is satisfied. For the controlled vehicle we propose the following player-specific cost terms:

$$\ell^{\mathrm{r}}_k := \theta^{\mathrm{r}}_1 (p^{\mathrm{r}}_{y,k})^2 + \theta^{\mathrm{r}}_2 (a^{\mathrm{r}}_k)^2 + \theta^{\mathrm{r}}_3 (\delta^{\mathrm{r}}_k)^2, \tag{20a}$$

$$\ell^{\mathrm{r}}_N := \theta^{\mathrm{r}}_1 (p^{\mathrm{r}}_{y,N})^2. \tag{20b}$$

For the yellow vehicle, we propose a quadratic cost on the control actions and deviations from the desired following distance from the blue vehicle, i.e.,

$$\ell^{\mathrm{y}}_k := \theta^{\mathrm{y}}_1 (p^{\mathrm{b}}_{x,k} - p^{\mathrm{y}}_{x,k} - d^{\mathrm{y}}_{\mathrm{des}})^2 + \theta^{\mathrm{y}}_2 (a^{\mathrm{r}}_k)^2, \tag{21a}$$

$$\ell^{\mathrm{y}}_N := \theta^{\mathrm{y}}_1 (p^{\mathrm{b}}_{x,k} - p^{\mathrm{y}}_{x,k} - d^{\mathrm{y}}_{\mathrm{des}})^2, \tag{21b}$$

[2]ALPAQA is available at https://github.com/kul-optec/alpaqa

where $d^{\mathrm{y}}_{\mathrm{des}} = 3$. 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.

TABLE II: The 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	$\theta^{\mathrm{y}} = \begin{bmatrix} 0.02 & 0.1 \end{bmatrix}^\top$

To model that humans in general do not like to drive close to other vehicles, we need to add common cost terms. We have opted to add Gaussian cost terms penalizing driving near another vehicle, 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.

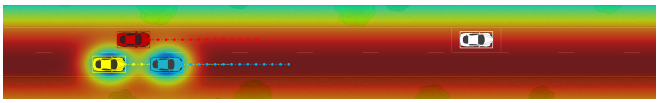

Fig. 3: Heatmap for the controlled vehicle.

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. For these constraint sets, Condition (a) of Theorem 3.2 is satisfied.

C. Numerical results

The introduced GPG formulation will be solved for the first 55 time steps. We solve (11) for feasibility 10^{-2} and tolerance 10^{-3}. The yellow vehicle in these experiments 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 a *belief* in either of those possibilities). 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). For these parameter values, it will tend to leave space for the red vehicle to merge into the right lane. When the red vehicle is aware of the courteous behavior of the yellow vehicle, the red vehicle will merge into the right lane accordingly. However, when the red vehicle wrongfully anticipates that the yellow vehicle behaves in a stubborn manner, it will not attempt to merge into the right lane and keep driving next to the other two vehicles. As the red vehicle approaches the white vehicle, it will think that the optimal strategy is to decelerate and merge

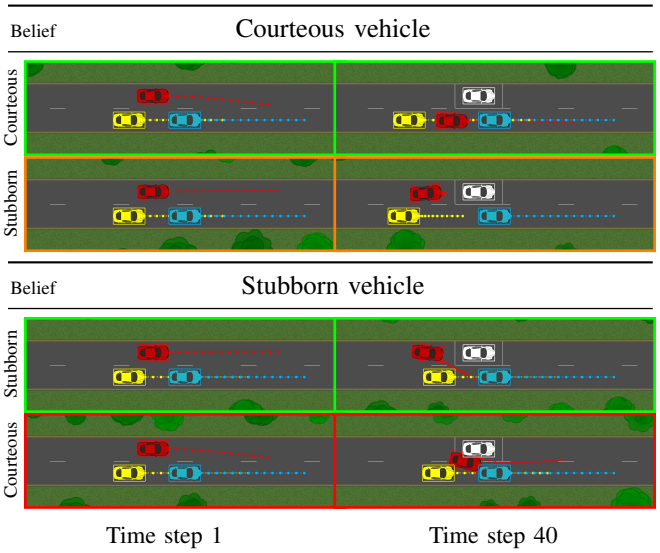

Fig. 4: The planned trajectories in the merging scenario at respectively the 1st and 40th time step of the game.

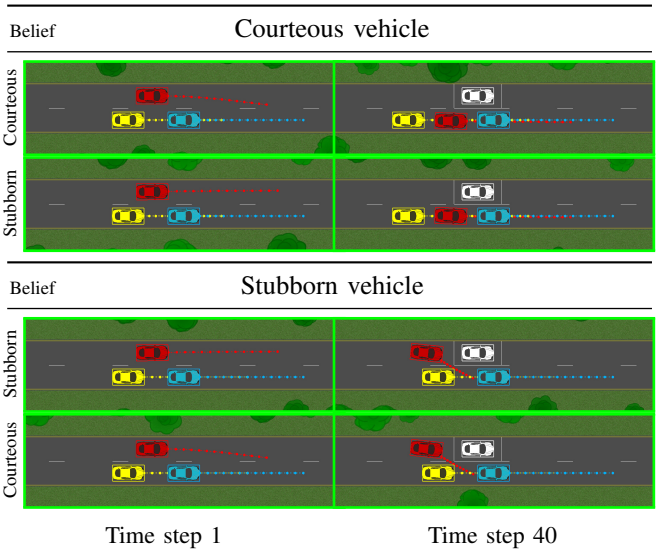

Fig. 5: The planned trajectories in the merging scenario at respectively the 1st and 40th time step of the game when using online learning.

behind the yellow vehicle, which 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). When the red vehicle is aware of this behavior, it will not attempt to merge into the right lane and keep driving next to the other two vehicles. As the red vehicle approaches the white vehicle, it will successfully decelerate to avoid any collisions. However, when the red vehicle believes that the yellow vehicle will behave courteously, the red vehicle 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, various collision avoidance constraints are violated and both vehicles perform large control actions, 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 (19), which continuously updates the estimate of the parameter $\hat{\theta}^{\mathrm{y}}$, 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, consider once again the 40th time step: the suboptimal/dangerous behavior observed before (Fig. 4), no longer occurs, as shown in Fig. 5.

Fig. 6 shows the evolution of the parameter estimate $\hat{\theta}_{\mathrm{y}_1}$ over time. Note that – as discussed in Section IV – we do not always observe convergence to the true parameter value (which is to be expected whenever the control signals are not sufficiently exciting), but nevertheless, parameter estimates are obtained which better explain previous observations, resulting in significantly improved behavior. This conclusion is further corroborated by Table III, which provides

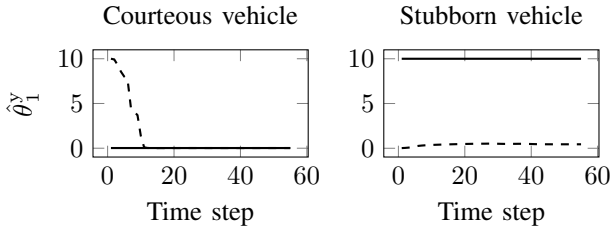

Fig. 6: The estimate of $\hat{\theta}_1^{\mathrm{y}}$ over time in the merging scenario, where the full lines denote the experiments with correct initial estimate and the dashed lines denote the experiments with incorrect initial estimate.

the numerical values for the closed-loop potential and the maximum constraint violation. For 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 a novel control strategy for solving optimal control problems with shared collision avoidance constraints emerging in general traffic situations by formulating the control problem as a generalized potential game, which is

Manuscript 385 submitted to 2022 European Control Conference (ECC).
Received November 8, 2021.

TABLE III: The closed-loop potential and maximum constraint violation in the merging scenario.

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

TABLE IV: The computation time of the red vehicle for solving the GPG and performing online learning during the merging experiment.

Behavior	Belief	GPG [s]		Update $\hat{\theta}^y$ [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

solved repeatedly within a model predictive control scheme. Additionally, in the case where the incentives and constraints of surrounding road users are unknown, we have proposed a practical online methodology for learning this information from observed data, and incorporating it directly into the optimal control problem. Through numerical simulations, we have illustrated the benefits of the learning scheme, as well as the potential real-time capabilities of the proposed methodology.

REFERENCES

[1] M. P. Vitus and C. J. Tomlin, "A probabilistic approach to planning and control in autonomous urban driving," in *52nd IEEE Conference on Decision and Control*, pp. 2459–2464, IEEE, 2013.

[2] A. Gray, Y. Gao, J. K. Hedrick, and F. Borrelli, "Robust Predictive Control for semi-autonomous vehicles with an uncertain driver model," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 208–213, 2013.

[3] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," *IEEE Intelligent Vehicles Symposium*, pp. 652–657, 2009.

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

[5] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: The case for cooperation," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013.

[6] R. Lachner, M. H. Breitner, and H. J. Pesch, "Real-Time Collision Avoidance: Differential Game, Numerical Solution, and Synthesis of Strategies," *Advances in Dynamic Games and Applications*, vol. 5, pp. 115–135, 2000.

[7] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: 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.

[8] A. Liniger and J. Lygeros, "A Noncooperative Game Approach to Autonomous Racing," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2020.

[9] G. Debreu, "A Social Equilibrium Existence Theorem," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 10, pp. 886–893, 1952.

[10] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The Complexity of Computing a Nash Equilibrium," *SIAM Journal on Computing*, vol. 39, pp. 195–259, 2009.

[11] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, "On the solution of the KKT conditions of generalized Nash equilibrium problems," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1082–1108, 2011.

[12] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.

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

[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, no. 20, 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. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2641–2646, 2015.

[17] F. Facchinei, V. Piccialli, and M. Sciandrone, "Decomposition algorithms for generalized potential games," *Computational Optimization and Applications*, vol. 50, no. 2, pp. 237–262, 2011.

[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," in *2018 European Control Conference (ECC)*, pp. 1523–1528, 2018.

[19] B. Hermans, G. Pipeleers, and P. P. Patrinos, "A penalty method for nonlinear programs with set exclusion constraints," *Automatica*, vol. 127, p. 109500, 2021.

[20] 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, 2017.

[21] P. Sopasakis, E. Fresk, and P. Patrinos, "OpEn: Code Generation for Embedded Nonconvex Optimization," in *21st IFAC World Congress*, (Berlin, Germany), 2020.

[22] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*. No. FA10 in Fundamentals of Algorithms, Philadelphia: SIAM, Society for Industrial and Applied Mathematics, 2014.

[23] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. MIT press Cambridge, MA, 1983.

[24] D. P. Bertsekas, *Dynamic programming and optimal control. Vol. 1*. Athena Scientific, 3rd ed., 2005.

[25] 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, no. 1, pp. 1–36, 2019.

[26] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological*, vol. 20, no. 5, pp. 403–414, 1986.