2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Enhancing Game-Theoretic Autonomous Car Racing Using Control Barrier Functions

Gennaro Notomista[1], Mingyu Wang[2], Mac Schwager[3], and Magnus Egerstedt[1]

Abstract— **In this paper, we consider a two-player racing game, where an autonomous ego vehicle has to be controlled to race against an opponent vehicle, which is either autonomous or human-driven. The approach to control the ego vehicle is based on a Sensitivity-ENhanced NAsh equilibrium seeking (SENNA) method, which uses an iterated best response algorithm in order to optimize for a trajectory in a two-car racing game. This method exploits the interactions between the ego and the opponent vehicle that take place through a collision avoidance constraint. This game-theoretic control method hinges on the ego vehicle having an accurate model and correct knowledge of the state of the opponent vehicle. However, when an accurate model for the opponent vehicle is not available, or the estimation of its state is corrupted by noise, the performance of the approach might be compromised. For this reason, we augment the SENNA algorithm by enforcing Permissive RObust SafeTy (PROST) conditions using control barrier functions. The objective is to successfully overtake or to remain in the front of the opponent vehicle, even when the information about the latter is not fully available. The successful synergy between SENNA and PROST—antithetical to the notable rivalry between the two namesake Formula 1 drivers—is demonstrated through extensive simulated experiments.**

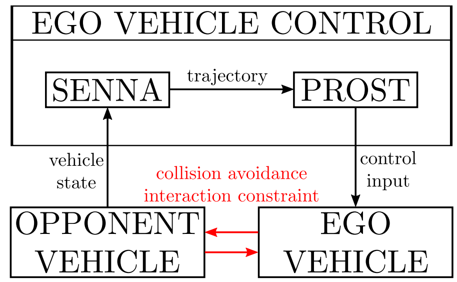

Fig. 1. Block scheme of the approach proposed in this paper for autonomous car racing in presence of uncertainties. The SENNA module acts as a higher-level game-theoretic planning algorithm that optimizes for a trajectory of the ego vehicle in order to overtake or remain in the front of an opponent vehicle. The PROST module is a lower-level robust safety controller that ensures collision avoidance even when there is uncertainty on the model of the opponent, or the measurement of its state is corrupted by noise.

I. INTRODUCTION

Interactions between vehicles have been demonstrated to play a crucial role in the resolution of many traffic situations. From strategically occupying areas of a road junction in order to resolve conflicts at intersections, to inducing a car to slow down during a lane merging maneuver, these interactions can range from being cooperative to competitive [1], [2], [3], [4]. In order to be able to account for human behaviors and optimize motion planning strategies for autonomous vehicles interacting with them, several approaches have been proposed which aim at predicting human intents [5], [6]. More specifically, these models can be employed, for instance, to design maneuvers for autonomous vehicles that are able to influence human drivers in order to mitigate traffic congestions or prevent imminent accidents [7].

The work by Gennaro Notomista and Magnus Egerstedt was sponsored by a Grant DCIST CRA W911NF-17-2-0181 from the Army Research Lab. Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

Gennaro Notomista and Magnus Egerstedt are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30308, USA {g.notomista,magnus}@gatech.edu

Mingyu Wang is with the Department of Mechanical Engineering, Stanford University, Stanford, CA, 94305, USA mingyuw@stanford.edu

Mac Schwager is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305, USA, schwager@stanford.edu

Even more so than in everyday traffic situations, interactions between vehicles are crucial in racing scenarios. In fact, as traffic code does not severely limit operations of vehicles, a wider range of possible behaviors becomes possible—a representative example of this being the leveraging of potential collisions in order to overtake an opponent vehicle during the course of a race [8]. Both in public roads and racing scenarios, the model of the opponent vehicle, whether this is driven by a human or not, plays a central role in the development of motion planning strategies. Following the line of inquiry of [8], in this work we propose a control framework that explicitly accounts for uncertainties in the model used for collision avoidance, which constitutes the main source of interaction with opponent vehicles.

More specifically, leveraging control barrier functions [9], we propose a controller which is able to ensure collision-free motion of the ego vehicle (i.e. the vehicle controlled by our algorithm), while being as least conservative as possible in accounting for the uncertainties in the motion model of the opponent vehicle. Owing to its least conservativeness, this approach—named PROST, Permissive RObust SafeTy—would not be applicable to everyday traffic scenarios. Nevertheless, for the same reason, it is desirable in racing conditions. We then use an optimization-based control framework to minimally modify the controller to track the trajectory generated using the game-theoretic approach in [8]—in this paper referred to as SENNA, Sensitivity-ENhanced NAsh equilibrium seeking—with the objective of winning a two-car racing game. Figure 1 depicts the described approach.

To summarize, our main contributions are the following.

We propose a method to augment the collision avoidance behavior of the trajectory planning approach presented in [8], in order to account for uncertainties in the model as well as the state estimation of the opponent vehicle in a two-car racing scenario. We do that by being as least conservative as possible, a feature that is desirable in racing situations, where it is essential to minimize conservativeness in order to attain the strongest racing performance. And, finally, we give an explicit expression for the most permissive safety certificate that is able to guarantee the collision avoidance objective in presence of bounded modeling and estimation noise.

II. BACKGROUND MATERIAL

A. SENNA: Sensitivity-ENhanced NAsh Equilibrium Seeking Using Iterated Best Response

In this section, we briefly recall the game-theoretic planning approach introduced in [10] for two-drone races, extended in [11] for multi-drone races, and applied to car racing in [8]. In [8], the formulation for single integrator agent dynamics presented in [10] is extended to encompass the nonlinear model of car-like vehicles. The kinematic of a car can be described using the following bicycle model [12]:

$$\dot{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_4 \cos x_3 \\ x_4 \sin x_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{x_4}{L} & 0 \\ 0 & 1 \end{bmatrix} u, \tag{1}$$

where $x = [p_x, p_y, \theta, v]^T$ are the two components of the vehicle position, its speed and its orientation, evaluated with respect to a reference system in the plane where the vehicle moves. The input $u = [u_1, u_2]^T = [\tan\phi, a]^T$ consists of the tangent of the steering angle and the vehicle acceleration. The parameter L represents the distance between front and rear axle of the vehicle. Although in this paper we utilize the kinematic bicycle model, the presented approach can be extended to the dynamic bicycle model by considering the forces acting at the tire level [13].

Owing to the differential flatness of the dynamical system (1) with respect to $\sigma = [\sigma_1, \sigma_2]^T = [p_x, p_y]^T$ (see, e. g., [14]), once desired trajectories are planned for the flat output σ and its first and second derivatives, it is possible to analytically compute the control inputs required to track them, using the following so-called endogenous transformation:

$$u_1 = \frac{\dot{\sigma}_1\ddot{\sigma}_1 + \dot{\sigma}_2\ddot{\sigma}_2}{\sqrt{\dot{\sigma}_1^2 + \dot{\sigma}_2^2}} \qquad u_2 = \frac{\dot{\sigma}_1\ddot{\sigma}_2 - \dot{\sigma}_2\ddot{\sigma}_1}{(\dot{\sigma}_1^2 + \dot{\sigma}_2^2)^{\frac{3}{2}}} L. \tag{2}$$

The inputs given in (2) guarantee perfect trajectory tracking and, above all, respect the non-holonomic kinematic constraints of the car. For this reason, in [8], second-order piecewise polynomials are employed to represent the planned trajectories, i.e.,

$$p_x^{(n)}(t) = \sum_{k=0}^{2} A_{n,k} t^k \qquad p_y^{(n)}(t) = \sum_{k=0}^{2} B_{n,k} t^k$$

for $t \in [n\Delta t, (n{+}1)\Delta t]$ and $n = 0, 1, \ldots, N{-}1$, where $N = T/\Delta t$ is the number of segments in which the trajectory is subdivided and T is the planning horizon. $p_x^{(n)}(t)$ and $p_y^{(n)}(t)$ denote the coordinates of the vehicle in the n-th trajectory segment at time t.

Trajectory continuity and control input constraints can be translated into constraints on the coefficients of the polynomials (see [8] for details). In particular, trajectory continuity constraints result in equality constraints, denoted by $g_{\text{eq}}(\cdot) = 0$, whereas speed, acceleration and curvature bounds can be expressed as quadratic inequality constraints, and are denoted, together with constraints to remain within the track, by $g_{\text{ineq}}(\cdot) \leq 0$. This way, the sensitivity-enhanced iterated best response (IBR) approach [10], [8] can be formulated as:

$$\begin{aligned} \underset{\Theta, A, B}{\text{maximize}} \; & s\left(\left[p_x^{(N)}, p_y^{(N)}\right]\right) + \rho\mu_{\text{o}} \left.\frac{\partial \zeta_{\text{o}}}{\partial \Theta_{\text{o}}}\right|_{(\Theta,\Theta_{\text{o}})} \Theta \\ \text{subject to} \; & g_{\text{eq}}(\Theta, A, B) = 0 \\ & g_{\text{ineq}}(\Theta, A, B) \leq 0 \\ & \zeta(\Theta, \Theta_{\text{o}}) \leq 0, \end{aligned} \tag{3}$$

where $s([p_x, p_y]) = \arg\min_t \left\|\tau(t) - [p_x, p_y]^T\right\|$ is the travel distance of the vehicle along the track, parameterized by the curve $\tau : \mathbb{R} \to \mathbb{R}^2$. ρ is a tunable parameter which encodes how much the optimization cares about influencing the opponent vehicle, μ_{o} is the vector of Lagrange multipliers corresponding to the collision avoidance constraint of the opponent vehicle, ζ and ζ_{o} encode the collision avoidance constraints—(2c) in [8]—, and Θ_{o} is the vector of positions and velocities of the opponent vehicle over the planned time horizon. The expression of the optimization variables are given by

$$\begin{aligned} &\Theta = \{(p_x^{(n)}, p_y^{(n)}, \dot{p}_x^{(n)}, \dot{p}_y^{(n)})\}_{n=0,\ldots,N-1} \\ &A = \{A_{n,k}\}_{\substack{k=0,1,2 \\ n=0,\ldots,N-1}} \qquad B = \{B_{n,k}\}_{\substack{k=0,1,2 \\ n=0,\ldots,N-1}}. \end{aligned}$$

IBR solves the optimization problem (3) iteratively for both ego and opponent vehicles. In [10], it is shown that IBR converges to a Nash equilibrium of the game. Once the optimal trajectory has been generated in terms of the polynomial coefficients A and B using (3), (2) can be used to generate the required control inputs to track it.

As collision avoidance has been demonstrated to be successful in encoding the interactions between two racing vehicles, we would still like to leverage it in racing scenarios, even when the dynamical model of the opponent vehicle is not precise, or its state is corrupted by measurement or estimation noise. Therefore, the trajectory tracking inputs provided by SENNA are used as *nominal* inputs in an optimization-based framework whose objective is that of minimizing the difference from the nominal controller, provided that safety conditions are met. In the following, we recall the concept of control barrier functions [9], which will be used to encode these safety conditions.

B. Control Barrier Functions

Control Barrier Functions (CBFs) [9] are used to ensure *safety* of dynamical systems, intended as the property of rendering a set $\mathcal{C}$ of the system state space forward invariant, i.e. if the state of the system starts inside the set $\mathcal{C}$, it never leaves $\mathcal{C}$. To this end, consider the following nonlinear system in control affine form:

$$\dot{x} = f(x) + g(x)u, \tag{4}$$

with f and g locally Lipschitz, $x \in \mathcal{D} \subset \mathbb{R}^n$ is the state and $u \in \mathcal{U} \subset \mathbb{R}^m$ is the input variable. A control barrier function for this system is defined as follows.

Definition 1 (Control Barrier Function [9])**.** *Let $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$, then h is a control barrier function (CBF) if there exists an extended class $\mathcal{K}_\infty$ function γ such that for the control system* (4)*:*

$$\sup_{u \in \mathcal{U}} \{L_f h(x) + L_g h(x) u\} \geq -\gamma(h(x)). \tag{5}$$

for all $x \in \mathcal{D}$.

The notation $L_f h$ and $L_g h$ is used to denote the Lie derivatives of h along the vector fields f and g, respectively. The following theorem shows the main results associated with CBFs.

Theorem 1 ([9])**.** *Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$. If h is a CBF on $\mathcal{D}$ and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$, then any Lipschitz continuous controller $u(x) \in \{u \in \mathcal{U} \colon L_f h(x) + L_g h(x) u + \gamma(h(x)) \geq 0\}$ for the system* (4) *renders the set $\mathcal{C}$ forward invariant. Additionally, the set $\mathcal{C}$ is asymptotically stable in $\mathcal{D}$.*

Several extensions have been proposed in order to take into account systems with higher relative degree [15], input constraints [16], or input uncertainties [17]. In the following section, CBFs will be leveraged to build collision avoidance constraints between two bicycle models (1), which are as least conservative as possible with respect to modeling errors or noise in the estimation of the state of the opponent vehicle.

III. COLLISION AVOIDANCE CONSTRAINTS FOR CAR RACING SCENARIOS

In [18], CBFs are applied to ensure collision avoidance between robots modeled using double integrator dynamics. The approach takes into account input constraints (corresponding to minimum and maximum acceleration) and proposes a way to ensure the existence of the resulting optimization-based controller for collision avoidance even under these constraints. In this section, we extend this approach to the case of the bicycle model with input constraints (i.e., minimum and maximum values of acceleration and steering angle). In the following subsection, the formulation proposed in [18] for double integrator dynamics will be briefly recalled.

A. Collision Avoidance for Double Integrator Dynamics

Acceleration-controlled robots can be modeled as double integrators whose state $x \in \mathbb{R}^{2n}$ consists of position and velocity of the robot in an n-dimensional Euclidean space, and whose input $u \in \mathbb{R}^n$ corresponds to the acceleration. Mathematically, this can be written as:

$$\dot{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u, \tag{6}$$

where 0 and I are $n \times n$ zero and identity matrices, respectively. For example, an acceleration-controlled ground mobile robot can be modeled using double integrator dynamics with $n = 2$, $x \in \mathbb{R}^4$. Position and velocity of the robot are two-dimensional, $x_1, x_2 \in \mathbb{R}^2$, as is the acceleration control input, $u \in \mathbb{R}^2$.

If there are constraints on the minimum and maximum value of the input, i.e.,

$$\|u\|_\infty \leq \alpha,$$

taking them into account, one can define the following CBF [18] to prevent collisions with an opponent robot, whose variables will be denoted by the subscript $(\cdot)_\mathrm{o}$:

$$\begin{aligned} h(x) = &\sqrt{2(\alpha + \alpha_\mathrm{o})\left(\|x_1 - x_{1,\mathrm{o}}\| - d_\text{min}\right)} \\ &+ \frac{(x_1 - x_{1,\mathrm{o}})^T}{\|x_1 - x_{1,\mathrm{o}}\|}\left(x_2 - x_{2,\mathrm{o}}\right), \end{aligned} \tag{7}$$

where d_min is the minimum distance that has to be kept between the two robots. Keeping the value of $h(x)$ positive corresponds to ensuring collision-free motion. Then, given a nominal controller $\hat{u}$ for the robot to execute, the controller

$$\begin{aligned} u^\star = &\arg\min_u \ \|u - \hat{u}\|^2 \\ &\text{subject to } L_f h(x) + L_g h(x) u \geq -\gamma(h(x)), \end{aligned} \tag{8}$$

where $f(x)$ and $g(x)$ can be read in (6), and $h(x)$ is given in (7), is the controller closest to the nominal one, $\hat{u}$, which is able to guarantee collision free motion of the robots modeled using double integrator dynamics. In the following section, we extend this formulation to the case of robots modeled using bicycle models.

B. Collision Avoidance for the Bicycle Model

The advantages of constructing collision avoidance algorithms using CBFs is that nonholomic constraints of the bicycle model, as well as input constraints, can be seamlessly taken into account without the need of enforcing them at a later stage.

Considering the notation introduced in (1), and letting $p = [x_1, x_2]^T$, we can define the following CBF to be used to prevent collisions between two vehicles modeled using (1):

$$\begin{aligned} h(x) = &\sqrt{4\alpha_\text{max}\left(\|p - p_\mathrm{o}\| - d_\text{min}\right)} \\ &+ \frac{(p - p_\mathrm{o})^T}{\|p - p_\mathrm{o}\|}\left(x_4 \begin{bmatrix} \cos x_3 \\ \sin x_3 \end{bmatrix} - x_{4,\mathrm{o}} \begin{bmatrix} \cos x_{3,\mathrm{o}} \\ \sin x_{3,\mathrm{o}} \end{bmatrix}\right), \end{aligned} \tag{9}$$

where the quantities with subscript $(\cdot)_\mathrm{o}$ are again used to denote the variables related to the opponent vehicle. The main difference between the CBFs in (7) and (9) lies in the evaluation of the value of the maximum acceleration α_max between two bicycle models, which will be discussed in the following.

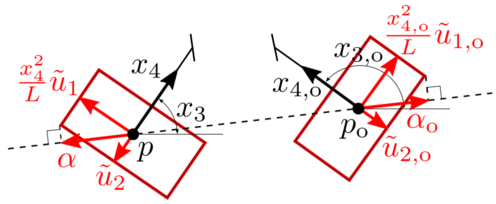

Fig. 2. Quantities used in (10) and Table I for the evaluation of the maximum relative acceleration $\alpha_{\max}$ between two vehicles. The red rectangles depict the sets of achievable accelerations of the two vehicles.

TABLE I

VALUES OF $\tilde{u}_{1,i}$ AND $\tilde{u}_{2,i}$ USED IN (10)

Figure	Condition	Expression
3a	$\left(R(\frac{\pi}{2})(p_j - p_i)\right)^T \begin{bmatrix}\cos x_{3,i}\\ \sin x_{3,i}\end{bmatrix} \geq 0$	$\tilde{u}_{1,i} = u_{1,\max}$
3b	$\left(R(\frac{\pi}{2})(p_j - p_i)\right)^T \begin{bmatrix}\cos x_{3,i}\\ \sin x_{3,i}\end{bmatrix} < 0$	$\tilde{u}_{1,i} = u_{1,\min}$
3c	$(p_j - p_i)^T \begin{bmatrix}\cos x_{3,i}\\ \sin x_{3,i}\end{bmatrix} \geq 0$	$\tilde{u}_{2,i} = u_{2,\min}$
3d	$(p_j - p_i)^T \begin{bmatrix}\cos x_{3,i}\\ \sin x_{3,i}\end{bmatrix} < 0$	$\tilde{u}_{2,i} = u_{2,\max}$

$R(\frac{\pi}{2})$ denotes the matrix that rotates 2-vectors counterclockwise by an angle of $\pi/2$. Also, if i denotes the ego vehicle, j represents the opponent, and vice versa.

In the case of robots modeled using double integrator dynamics with acceleration constraints $\alpha > 0$ and $\alpha_\mathrm{o} > 0$, the value of the maximum relative acceleration is given by $\alpha + \alpha_\mathrm{o}$, for any state x of the robots (see (6)). On the other hand, if the bicycle model is considered to describe the motion of two vehicles, then the maximum relative acceleration between the vehicles depends on the state of each vehicle, and in particular on their relative orientation. A steering input can, in fact, be used to add a lateral component to the acceleration provided by the longitudinal dynamics. Moreover, the former is in general, significantly higher than the latter, both for street and race vehicles [19], [20].

Figure 2 depicts the models of two vehicles, as well as the quantities which will be used in the derivation of the expression of $\alpha_{\max}$ required in the CBF (9). The general expression of the maximum relative acceleration between two bicycle models is given by:

$$\begin{aligned}\alpha_{\max} = \frac{p - p_\mathrm{o}}{\|p - p_\mathrm{o}\|}^T \Bigg(& \frac{x_4^2}{L}\tilde{u}_1 \begin{bmatrix}-\sin x_3\\ \cos x_3\end{bmatrix} + \tilde{u}_2 \begin{bmatrix}\cos x_3\\ \sin x_3\end{bmatrix} \\ & + \frac{x_{4,\mathrm{o}}^2}{L}\tilde{u}_{1,\mathrm{o}} \begin{bmatrix}-\sin x_{3,\mathrm{o}}\\ \cos x_{3,\mathrm{o}}\end{bmatrix} + \tilde{u}_{2,\mathrm{o}} \begin{bmatrix}\cos x_{3,\mathrm{o}}\\ \sin x_{3,\mathrm{o}}\end{bmatrix} \Bigg)\end{aligned} \tag{10}$$

Introducing the following nomenclature for the input bounds of a bicycle model in (1)

$$\begin{aligned} &u_{1,\min} \leq u_1 \leq u_{1,\max} \quad \text{(steering angle inputs)}\\ &u_{2,\min} \leq u_2 \leq u_{2,\max} \quad \text{(acceleration inputs)},\end{aligned}$$

the calculation of the values of $\tilde{u}_1$, $\tilde{u}_2$, $\tilde{u}_{1,\mathrm{o}}$, and $\tilde{u}_{2,\mathrm{o}}$ in (10) can be broken down into four cases, depicted in Figures 3a to 3d and reported in Table I.

Remark 1. *The resulting α_{max}, obtained by substituting the expressions of $\tilde{u}_i$ given in Table I in* (10)*, is a continuous function of the state of the two vehicles.*

With the CBF h given in (9), the optimization-based formulation given in (8) can be used to compute the controller u^* which minimizes the difference from the nominal trajectory tracking controller $\hat{u}$, provided that the safety constraints are not violated, namely that $h(x) \geq 0$.

Remark 2 (Computational complexity)**.** *First of all, it has to be noticed that the collision avoidance constraint has to be enforced only if the relative speed between the two vehicles is negative, i. e., if*

$$(p_j - p_i)^T \left(x_{4,j} \begin{bmatrix}\cos x_{3,j}\\ \sin x_{3,j}\end{bmatrix} - x_{4,i} \begin{bmatrix}\cos x_{3,i}\\ \sin x_{3,i}\end{bmatrix} \right) < 0.$$

Moreover, despite the apparent complexity of the expressions in (9)*,* (10) *and Table I, the collision avoidance constraint is still an affine constraint in the optimization variable $u = [\tan\phi, a]^T$. Therefore, efficient algorithms (see, e. g., [21]) can be utilized in order to solve the optimization problem* (8) *in an online fashion.*

The expression of h introduced in (9) relies on the perfect knowledge of the opponent state x_o. However, if this is corrupted by measurement noise, or there are uncertainties in the model of the opponent, then safety cannot be guaranteed by enforcing a constraint similar to (5). For this reason, in the next section, we propose a method to ensure the desired safety condition, which is also as least conservative as possible.

C. PROST: Permissive RObust SafeTy

A way to design algorithms to race against *unknown* opponents consists in considering modeling, measurements and estimation errors and including them in the formulation. In this section, starting from the CBFs constructed in the previous section, we propose a method to ensure safety under modeling, measurements and estimation uncertainties.

In [22], the robustness properties of CBFs are analyzed. In their main result, the authors show how CBFs are robust to vanishing and even non-vanishing perturbations on the system dynamics. In [17], further work has been done in order to account for uncertainties in the system inputs by propagating the errors forward through the model and ensuring that in the worst cases scenario safety is still guaranteed. If this approach would be applicable in low-risk situations (e. g., public road traffic), in this paper we want to consider racing scenarios where robustness and conservativeness can be sacrified for performance.

To this end, let us consider the expression in (5), which is the inequality constraint that has to be satisfied in order to guarantee collision avoidance between two bicycle models (1) using CBF (9). The inequality in (5) can be written, without the sup operator, as

$$\dot{h}(x, u) \geq -\gamma(h(x)), \tag{11}$$

for some class $\mathcal{K}_\infty$ function γ. In case $\gamma(h(x))$ is measured or estimated, we model its value as the sum of a nominal

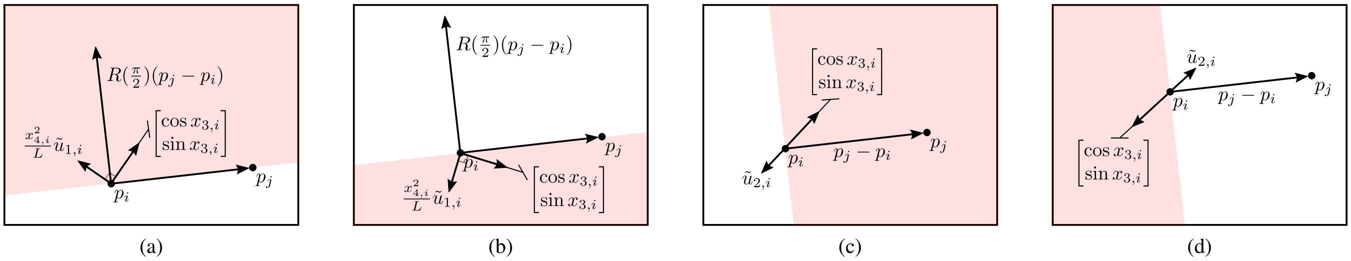

Fig. 3. Geometric configurations of a bicycle model with respect to the line segment joining its position p_i to the one of a second bicycle p_j. The four cases correspond to the four cases reported in Table I, and lead to different expressions of the maximum relative acceleration between two bicycle models. The half planes shaded in red represent the regions in which the vector $[\cos x_{3,i} \;\; \sin x_{3,i}]^T$ lies, under the four conditions reported in Table I.

term and a noise term w, where we assume that the latter is bounded, i.e. $|w| \leq w_{\max}$. The inequality in (11) is then substituted by the following one:

$$\dot{h}(x,u) \geq -\gamma(h(x)) + w.$$

In this case, one could proceed as described in [22], and ensure that only a superlevel set of the set $\mathcal{C} = \{x \in \mathcal{D} \colon h(x) \geq 0\}$ is rendered forward invariant. This is equivalent to the fact that $h(x)$ is not guaranteed to be positive, but only $h(x) \geq -\delta$, for some $\delta > 0$, whose magnitude is itself a class $\mathcal{K}$ function of the magnitude of the disturbance w [22]. Then, a naive approach to robust safety would consist in defining the new CBF $\bar{h} = h - \delta$, so that, enforcing the constraint $\dot{\bar{h}} \geq -\gamma\left(\bar{h}(x)\right)$ will result in $\bar{h} \geq 0$, which is equivalent to $h \geq \delta$. Consequently, in presence of the noise term w, $\bar{h} \geq -\delta$ and, consequently, $h \geq 0$. Nevertheless, this approach is the most conservative one, being designed around the worst case scenario, i.e. to ensure safety when w achieves its maximum value $w_{\max}$.

In the following, based on properties of the class $\mathcal{K}_\infty$ function γ, we give the least permissive, yet robust, CBF that can be employed to guarantee safety. In order to formulate this CBF, we need the following Lemma.

Lemma 1. *Given the CBF $h \colon \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$, any Lipschitz continuous controller u such that*

$$L_f h(x) + L_g h(x) u + \gamma(h(x)) \geq w$$

holds for all $x \in \mathcal{D}$, where w is a bounded additive noise term, with $|w| \leq w_{max}$, renders the set $\bar{\mathcal{C}} = \{x \in \mathcal{D} \colon h(x) \geq -\delta\}$ forward invariant, where

$$\delta = -\gamma^{-1}(-w_{max}) \geq 0. \tag{12}$$

Proof. If $h(x) = -\delta$, one has:

$$\begin{aligned}\dot{h}(x,u) &\geq -\gamma(-\delta) + w \geq -\gamma(-\delta) - w_{\max} \\ &= -\gamma(-\gamma^{-1}(w_{\max})) - w_{\max} = w_{\max} - w_{\max} = 0.\end{aligned}$$

Then, by Nagumo's theorem [23], $\bar{\mathcal{C}}$ is forward invariant. □

Using this Lemma, the following proposition gives the expression of the optimization-based controller that is able to guarantee safety while being as least conservative as possible in accounting for uncertainties in the safety certificate (5).

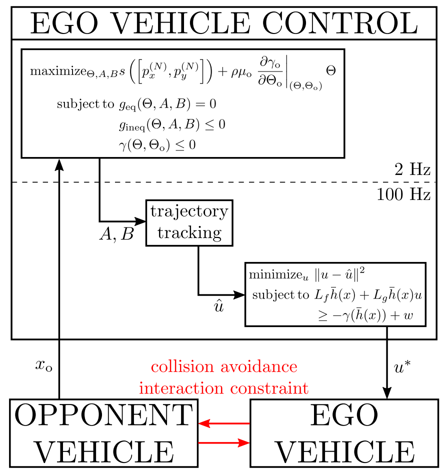

Fig. 4. Block scheme of the approach presented in this paper for autonomous racing in presence of uncertainties. The blocks of Fig. 1 are blown up and detailed with the optimization problems that are solved within them. The trajectory tracking block could theoretically consist of the (open-loop) endogenous transformation (2). Nevertheless, for robustness purposes, in the experiments reported in Section IV, it has been replaced with a closed-loop pure-pursuit controller.

Proposition 1. *The controller*

$$\begin{aligned} u^\star = \;& \arg\min_u \; \|u - \hat{u}\|^2 \\ & \text{subject to } L_f \bar{h}(x) + L_g \bar{h}(x) u \geq -\gamma(\bar{h}(x)) + w, \end{aligned} \tag{13}$$

where $\bar{h} = h - \delta$, h is defined in (9) *and δ is given by* (12), *is the controller closest to $\hat{u}$ that is able to prevent collisions in presence of bounded noise w, $|w| < w_{max}$.*

Proof. By Lemma 1, it follows that the constraint of the optimization problem (13) ensures that $h(x) \geq 0$ for all x. With $h(x)$ given by (9) and using Theorem 1, keeping $h(x) \geq 0$ is equivalent to the condition in which collisions are prevented even in presence of the bounded noise term w in the safety constraint (13). □

If an estimate of the bound $w_{\max}$ is available, using Lemma 1 and Proposition 1, one can choose the class K_∞ function γ that is able to guarantee safety by sacrificing as least as possible the performance of the controller.

Figure 4 summarizes the approach derived so far. The game-theoretic approach SENNA evaluates, every 0.5 seconds, the best trajectory to be followed by the ego vehicle in

TABLE II

NUMBER OF SUCCESSFUL OVERTAKING AND BLOCKING MANEUVERS OUT OF 15 SIMULATIONS OF 8 RACING SCENARIOS.

	two-lap overtaking	two-lap blocking
SENNA VS MPC	2/15	6/15
SENNA VS human	3/15	2/15
SENNA-PROST VS MPC	13/15	12/15
SENNA-PROST VS human	12/15	13/15

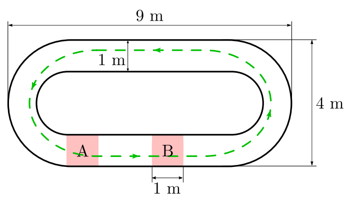

Fig. 5. Parameters of the race track. The red-shaded squares marked with A and B are the regions where the vehicle that attempts the overtaking maneuver and the one that tries to block it start the race, respectively. The dashed green line is the race line given as trajectory input to be tracked to the MPC controller.

order to either overtake or remain in the front of an opponent vehicle. The optimization-based controller PROST runs in a low-level control loop at 100 Hz, and it is designed to mitigate the effects of possible uncertainties in the model of the opponent, or the estimation noise of its state. In the next section, simulation results confirm that the combination of SENNA and PROST allows the ego vehicle to successfully overtake or block an opponent vehicle in a two-lap race.

IV. SIMULATIONS

To validate the proposed control framework, we performed 120 simulated experiments subdivided among 8 different racing scenarios, which are summarized in Table II. As in [11], both overtaking and blocking scenarios are considered. In Fig. 5, the parameters of the simulations related to the race track are reported. The two vehicles are modeled using (1), where $L = 0.4\text{m}$, and the input constraints are $|u_1| \leq \tan(\pi/4)$ and $|u_2| \leq 1\text{m/s}^2$. In the overtaking scenarios, the ego car—either controlled by SENNA only, or using the control architecture depicted in Fig. 4—starts in the square A (see Fig. 5), whereas in blocking scenarios it starts in B. The vehicle that starts behind has the advantage of having higher maximum speed, 2.5 instead of 1.8 m/s.

In order to show the performance of the algorithm when uncertainties in the opponent state or strategy are present, artificial noise is added to the measured state of the opponent vehicle. In particular, the maximum magnitude of the measurement noise is set to 0.04 m for the position of the vehicles $[p_x, p_y]^T$, $\pi/18$ for its orientation θ, and 0.1m/s for its speed v. The opponent vehicle is driven either by a MPC trajectory tracking controller or by a human teleoperator.

First, as a baseline, we let SENNA race against the MPC trajectory tracking controller. Both SENNA and the MPC algorithm are equipped with collision avoidance features, through which they interact during overtaking and blocking maneuvers. However, the collision avoidance algorithm does not account for measurement noise. As a result, the ego vehicle fails most of the times both in the overtaking and the blocking attempts in a two-lap race (first row of Table II). In all unsuccessful experiments, the failure is caused by a collision, either with the opponent vehicle or with the track boundary. Similar outcomes are achieved when SENNA competes with a human driver—who offers a challenge to the game-theoretic approach. Both the lack of a model for the human, and the fact that state measurements are corrupted by noise, prevent the ego vehicle to successfully overtake or block the opponent. Similarly to the previous case, most failing cases are due to collisions between the ego and the opponent vehicle.

The bottom half of Table II shows the results of the application of the control framework presented in this paper in the same scenarios discussed above. Employing the combination of SENNA and PROST allows the ego vehicle to significantly improve its performances. The PROST approach is also used to avoid colliding with the track boundaries. This is implemented using the CBF (9), where the position of the opponent is replaced by the closest point of the track boundary, whose velocity is set to 0.

Note that, owing to the continuity of the CBF h and the function γ in (13), the value of w_{max}—required to evaluate δ used in the definition of $\bar{h}$ in Lemma 1—can be calculated as $w_{\text{max}} = \max_{w_x \in \mathcal{W}_x} |\gamma(h(x + w_x))|$, where, for the considered experiments in which the magnitude of the artificial noise are introduced above, $\mathcal{W}_x$ is the following compact set: $\mathcal{W}_x = [-0.1, 0.1] \times [-0.1, 0.1] \times [-\pi/18, \pi/18] \times [-1, 1]$.

As expected, using the SENNA-PROST approach allows the ego vehicle to succeed in the two-car racing scenario most of the times. By adding the PROST component, in fact, all collisions which could not be prevented before are now mitigated, since the noise present in the state measurements of the opponent is explicitly taken into account. A quantitative measure of the improvement of the performance of the ego vehicle that uses the SENNA-PROST approach is given in Table II in terms of successful overtaking and blocking scenarios in a two-lap race.

V. CONCLUSIONS

In this paper, we presented an optimization-based controller for a two-car racing game. The proposed strategy extends the game-theoretic approach presented in [11]—referred to as SENNA, Sensitivity-ENhanced NAsh equilibrium seeking—in order to take into account measurement, estimation and modeling uncertainties. Using control barrier functions, we formulate conditions for Permissive RObust SafeTy (PROST), which allow an ego vehicle to successfully overtake or remain in the front of an opponent vehicle even when the former only has corrupted information about the latter. This situation encompasses the presence of measurement or estimation noise, as well as unmodeled dynamics of the opponent used in the game-theoretic approach. The successful combination of SENNA and PROST algorithms is demonstrated through extensive simulated experiments.

REFERENCES

[1] G. Ding, S. Aghli, C. Heckman, and L. Chen, "Game-theoretic cooperative lane changing using data-driven models," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3640–3647.

[2] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[3] E. Bıyık, D. A. Lazar, D. Sadigh, and R. Pedarsani, "The green choice: Learning and influencing human decisions on shared roads," *arXiv preprint arXiv:1904.02209*, 2019.

[4] E. Stefansson, J. F. Fisac, D. Sadigh, S. S. Sastry, and K. H. Johansson, "Human-robot interaction for truck platooning using hierarchical dynamic games," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3165–3172.

[5] 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*, vol. 2. Ann Arbor, MI, USA, 2016.

[6] K. H. Yang, "Review of mathematical human models for incorporation into vehicle safety design," *International journal of vehicle design*, vol. 26, no. 4, pp. 430–441, 2001.

[7] D. A. Lazar, R. Pedarsani, K. Chandrasekher, and D. Sadigh, "Maximizing road capacity using cars that influence people," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1801–1808.

[8] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," *Robotics: Science and Systems XV*, 2019.

[9] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[10] R. Spica, D. Falanga, E. Cristofalo, E. Montijano, D. Scaramuzza, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *arXiv preprint arXiv:1801.02302*, 2018.

[11] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.

[12] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[13] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[14] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME international mechanical engineering congress and exposition*. Citeseer, 1995.

[15] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.

[16] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 1656–1661.

[17] Y. Emam, P. Glotfelter, and M. Egerstedt, "Robust Barrier Functions for a Fully Autonomous, Remotely Accessible Swarm-Robotics Testbed," *arXiv e-prints*, p. arXiv:1909.02966, Sep 2019.

[18] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[19] T. D. Gillespie, *Fundamentals of vehicle dynamics*. Society of automotive engineers Warrendale, PA, 1992, vol. 400.

[20] W. F. Milliken, D. L. Milliken *et al.*, *Race car vehicle dynamics*. Society of Automotive Engineers Warrendale, 1995, vol. 400.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[22] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[23] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.