IEEE Robotics and Automation Letters (RAL) paper presented at the
2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

A Predictive Safety Filter for Learning-Based Racing Control

Ben Tearle, Kim P. Wabersich, Andrea Carron, and Melanie N. Zeilinger

***Abstract*—The growing need for high-performance controllers in safety-critical applications like autonomous driving motivated the development of formal safety verification techniques. In this paper, we design and implement a predictive safety filter that is able to maintain vehicle safety with respect to track boundaries when paired alongside any potentially unsafe control signal, such as those found in learning-based methods. A model predictive control (MPC) framework is used to create a minimally invasive algorithm that certifies whether a desired control input is safe and can be applied to the vehicle, or that provides an alternate input to keep the vehicle in bounds. To this end, we provide a principled procedure to compute a safe and invariant set for nonlinear dynamic bicycle models using efficient convex approximation techniques. To fully support an aggressive racing performance without conservative safety interventions, the safe set is extended in real-time through predictive control backup trajectories. Applications for assisted manual driving and deep imitation learning on a miniature remote-controlled vehicle demonstrate the safety filter's ability to ensure vehicle safety during aggressive maneuvers.**

***Index Terms*—Robot Safety, Optimization and Optimal Control, Machine Learning for Control**

I. INTRODUCTION

THE development of robotic systems has led to an ever increasing number of applications that go beyond the isolated task spaces found in legacy industries such as automotive or electronics production. More recent applications encompass dynamic and learning-based interactions with humans in complex task spaces, as is the case with autonomous driving, and therefore require advanced safety mechanisms [1], [2], to prevent potentially dangerous situations. Maintaining safety at the physical limits for highly dynamic systems often requires a task-specific trade-off between performance and conservativeness to ensure safe system operation. As a result, there is an increasing interest in developing theoretically sound safety frameworks with a reduced degree of conservativeness that enable safety in a modular fashion, independent of a task-specific objective.

Manuscript received: February, 23, 2021; Revised May, 21, 2021; Accepted June, 25, 2021.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments. The authors are members of the Institute for Dynamical Systems and Control (IDSC), ETH Zurich, ZH-8092, Switzerland: bentearle@gmail.com, [wkim|carrona|mzeilinger]@ethz.ch. This work was supported by the Swiss National Science Foundation under grant no. PP00P2 157601 / 1. Andrea Carron's research was supported by the Swiss National Centre of Competence in Research NCCR Digital Fabrication and the ETH Career Seed Grant 19-18-2. Ben Tearle's work was also supported by the ETH Career Seed Grant 19-18-2.

Digital Object Identifier (DOI): see top of this page.

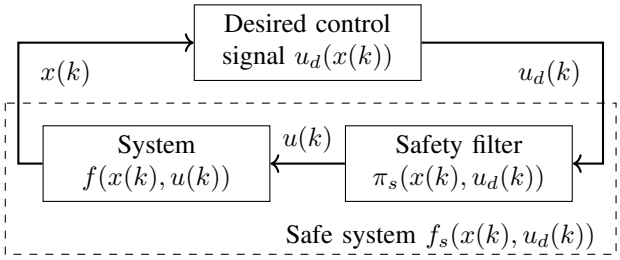

Fig. 1. Concept of a predictive safety filter: Based on the current state $x(k)$, an arbitrary control algorithm provides a desired control input $u_d(k) \in \mathbb{R}^m$, which is processed by the safety filter $u(k) = \pi_s(x(k), u_d(k))$ and applied to the real system.

While some of these methods have been demonstrated in practice, the considered applications are often small-scale or nearly linear control systems that are only operated within conservative regions of their state space [3]. Motivated by the strict safety requirements in autonomous driving, we consider the problem of safe autonomous and assisted racing as a benchmark application for deriving a practically relevant safety mechanism. Racing requires the utilization of a vehicle's full nonlinear dynamics, providing a challenging domain in which safety must be guaranteed.

To provide safety for arbitrary control policies, we rely on a modular safety framework as shown in Figure 1. This approach allows the framework to be used in conjunction with any potentially unsafe control signal, such as those from learning-based controllers. The basic idea is to design a *safety filter*, which analyzes the desired control signal and decides in real-time whether it can be applied to the system, or if it has to be modified to ensure safety. For the racing application considered in this work, this consists of verifying if the vehicle is able to stay within track boundaries in the future given the current steering and drivetrain commands.

This is achieved by finding safe backup control sequences that lead the vehicle towards a set of known safe states, where the first input of the sequence is as close as possible to the desired control signal. This approach allows for verification of the safety of the desired input, while simultaneously providing an alternative safe input.

A. Related Work

The concept of using a safety controller in a closed-loop system was first introduced in [4], where the system can switch between an experimental controller and a reliable safety controller in the case of software faults. Developments in the theoretic use of barrier certificates for verifying system

safety were later proposed in [5], which was further extended to the idea of control barrier functions (CBFs) [6]. More recent work has revisited the notion of using CBFs for safety-critical control of robotic systems, see [1] for an overview. This approach has been combined with a machine-learning framework in [7] to safely learn model discrepancies of a Segway robot, while limiting the operational space during training. Although these methods build off strong theoretical results from control Lyapunov function theory, they rely on the ability to explicitly model a system's safety requirements as a CBF, which is in general difficult to design.

Given the inherent lack of safety guarantees in traditional machine learning methods, the reinforcement learning (RL) field has become increasingly interested in enforcing constraints for training black-box control policies. A general-purpose policy search algorithm for constrained reinforcement learning is introduced in [8], which approximately enforces safety constraints at every policy update. Using a learning-based system model, [9] proposes a method for determination of a safe set of system states under a specific learning-based policy. Although these methods allow for approximately safe policy training, they are limited in that they remain tied to task-specific reinforcement learning algorithms, whereas the safety filter presented in this work is able to function independently of a specific task and thereby enables modular safety.

An approach for providing system safety based on confining a system to a pre-computed set of safe states is introduced in [10]. This uses reachability-based techniques to find a safe set for a given system together with a corresponding control policy that provides invariance within the safe set. The idea is expanded in [3] to perform online updates of the safe set using a non-parametric system dynamics estimate. These approaches suffer from limited scalability in the offline safe set computation required. Recent work attempts to address this by a subsystem decomposition [11], which is not, however, applicable in case of strong lateral-longitudinal couplings as are considered in this work. Approximation techniques include data-based methods [12], sum-of-squares programming [13], and active learning [14].

Closely related to these ideas, a method for establishing safety using an MPC-based control law is derived in [15]. A continuously updating control policy is computed online to find backup trajectories towards safe states, resulting in the implicit representation of the safe set and corresponding safe control law via the MPC optimization problem. This method is extended to consider nonlinear stochastic systems formulated with chance-constraints or parametric uncertainties in [16], [17], and provides the foundation for the task of autonomous racing considered in this work.

B. Contributions

The main contribution of this paper is the design and implementation of a permissive safety filter for autonomous racing that can be combined with any desired control signal, ensuring closed-loop vehicle safety with respect to a track for a diverse range of applications. To this end, we use the concept of predictive safety filters as presented in [15], [17]. To achieve a minimally invasive safety filter supporting aggressive maneuvers, we use a nonlinear dynamic bicycle model with a Pacejka model of the tire forces [18] to simultaneously predict and optimize accurate backup control trajectories. In addition to a high-fidelity system model, the safety filter performance can be improved by using either a longer planning horizon or a larger terminal set. As the planning horizon is typically limited by memory and processing requirements, we derive an iterative optimization-based invariant set computation using convex approximations to obtain an enlarged terminal safe set for the nonlinear dynamic bicycle model, which is valid over a range of constant road curvatures.

The physical miniature racing application demonstrates the proposed safety filter's performance with both human-in-the-loop racing and deep imitation learning. This work presents, to the best of our knowledge, the first application of a predictive safety filter to a complex and highly dynamical nonlinear system demonstrated by experimental results.

II. PROBLEM FORMULATION

Notation: The set of integers in the interval $[a, b] \subset \mathcal{R}$ is denoted by $\mathcal{I}_{[a,b]}$, and the set of integers in the interval $[a, \infty) \subset \mathcal{R}$ is $\mathcal{I}_{\geq a}$. The i-th row of a matrix $M \in \mathcal{R}^{n\times m}$ is denoted by $[M]_i$.

The goal of this work is to design a safety filter that certifies whether or not a desired control input, $u_d(k)$, is safe for a vehicle system, and provides an alternative safe control input at any time. We consider a discrete-time nonlinear system of the form

$$x(k+1) = f(x(k), u(k)),\ \forall k \in \mathcal{I}_{\geq 0}, \tag{1}$$

with dynamics $f : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$. We assume an accurate model and state estimate of the system is available in this work; however, extensions of predictive safety filters considering system uncertainty can be found in [17]. The system is subject to input and state constraints of the form

$$u(k) \in \mathcal{U} \text{ and } x(k) \in \mathcal{X}\ \forall k \in \mathcal{I}_{\geq 0}, \tag{2}$$

which must be satisfied at each time step k to ensure safety.

In order to guarantee this notion of safety for a given $u_d(k)$, a safety control policy, $\pi_{\mathcal{S}}(x(k), u_d(k))$, is provided that guarantees constraint satisfaction for all future time steps if applied to the vehicle. If a safety policy exists with $u_d(k)$ as the current input of the policy, then $u_d(k)$ can be certified as safe and applied to the system. More formally:

Definition 1. *A desired input $u_d(\bar{k})$ is certified as safe for system* (1)*, at a given time step $\bar{k}$, if the safety control policy yields $\pi_{\mathcal{S}}(x(\bar{k}), u_d(\bar{k})) = u_d(\bar{k})$, and application of $u(k) = \pi_{\mathcal{S}}(x(k), u_d(k))$ to the system results in satisfaction of the constraints in* (2) *for all $k \geq \bar{k}$.*

Using a safety policy in accordance with Definition 1 provides a safety filter that can be brought into a closed-loop system as shown in Figure 1. Since the safe control input is recomputed at each time step to verify the incoming desired input, this allows the desired control signal to have control authority over the system whenever possible, i.e.

$\pi_{\mathcal{S}}(x(k), u_d(k)) = u_d(k)$. However, if the desired control signal would put the system at risk of violating its constraints in the future, then alternate inputs, $\pi_{\mathcal{S}}(x(k), u_d(k)) \neq u_d(k)$, must be available that ensure safety for the system.

The next section discusses an approach to compute $\pi_{\mathcal{S}}$ online using an MPC framework that ensures safety for the system while reducing impact on the desired control signal.

III. PREDICTIVE SAFETY FILTER

We define an implicit safety policy through a receding-horizon optimal control problem, referred to as *predictive safety filter problem* [15], which allows for an efficient online computation of the desired safety filter $\pi_{\mathcal{S}}$:

$$\begin{aligned}
\min_{x_{i|k}, u_{i|k}} \quad & J(u_{i|k}, u_d(k)) && \text{(3a)}\\
\text{s.t.} \quad & \forall i \in \mathcal{I}_{[0,N-1]}: \\
& x_{0|k} = x(k), && \text{(3b)}\\
& x_{i+1|k} = f(x_{i|k}, u_{i|k}), && \text{(3c)}\\
& x_{i|k} \in \mathcal{X}, && \text{(3d)}\\
& u_{i|k} \in \mathcal{U}, && \text{(3e)}\\
& x_{N|k} \in \mathcal{S}_f. && \text{(3f)}
\end{aligned}$$

Problem (3) computes a discrete-time state and input backup trajectory, $\{x^*_{i|k}, u^*_{i|k}\}$, of length N, where $x_{i|k}$ is the state predicted i time steps ahead, computed at time k, initialized at $x_{0|k} = x(k)$, and similarly for $u_{i|k}$. The system is predicted along the horizon according to dynamics (3c), subject to an initial condition (3b), state and input constraints (3d) and (3e), and terminal constraint (3f). Different from classical MPC, the objective function in (3a) is chosen to minimize the difference between the desired control input and the first input of the solution trajectory, as

$$J(u_{i|k}, u_d(k)) = \|u_d(k) - u_{0|k}\|^2. \tag{4}$$

The safety policy is then defined by $\pi_{\mathcal{S}}(x(k), u_d(k)) = u^*_{0|k}$.

The cost function in (4) can be modified to include secondary objectives beyond tracking the desired control input. For the racing application, we include a regularization term that penalizes the rate of change of the inputs in order to encourage a smoother control trajectory:

$$J(u_{i|k}, u_d(k)) = \|u_d(k) - u_{0|k}\|^2_W + \sum_{i=0}^{N-1} \|\Delta u_{i|k}\|^2_{R_{\mathcal{S}}}, \tag{5}$$

where $\Delta u_{0|k} := u_{0|k} - u_{0|k-1}$, $\Delta u_{i|k} := u_{i|k} - u_{i-1|k}$ for $i = 1, .., N-1$, and $W, R_{\mathcal{S}} \in \mathbb{R}^{m \times m}$ are cost matrices for the input deviation and input rate respectively. This helps to reduce rapid fluctuations between the desired input and safety filter's input, which can occur with the system at the boundary of the state constraints in practice. To avoid unnecessary input deviations from a desired input that can be certified as safe, the weights are chosen with W much larger than $R_{\mathcal{S}}$ to ensure priority remains on tracking the desired input.

Assumption 1 (Invariant terminal set)**.** *There exists a control law $\kappa_f : \mathcal{S}_f \to \mathcal{U}$, and a corresponding positively invariant set $\mathcal{S}_f \subseteq \mathcal{X}$, such that for all $x \in \mathcal{S}_f$, it holds that $\kappa_f(x) \in \mathcal{U}$ and $f(x, \kappa_f(x)) \in \mathcal{S}_f$.*

As in standard MPC theory, Assumption 1 provides recursive feasibility for the safety control policy obtained from Problem (3), i.e. if the problem has a feasible solution at time step $\bar{k}$, then a feasible solution also exists for all future times $k > \bar{k}$. This results in constraint satisfaction at all times as required in (2). More precisely, a control input $u_d(k)$ can be certified as safe for the system if Problem (3) finds a state and input trajectory that is feasible along the horizon and ends in S_f. Figure 2(a) shows a vehicle at state $x(k)$ where application of $u_d(k)$ would bring the vehicle to state $x_{1|k}$. From $x_{1|k}$ a trajectory exists that keeps the vehicle inside track limits until it reaches $\mathcal{S}_f$. The optimal solution to (3) would therefore be $u^*_{0|k} = u_d(k)$, which achieves a minimal objective cost of zero and satisfies the desired behavior of no intervention for a safe $u_d(k)$.

If the desired input is unsafe, then its application will result in a state from which no trajectory satisfying all constraints exists. In Figure 2(b), the trajectory following $x_{1|k}$ after applying $u_d(k)$ can be seen to leave the track. In this case, Problem (3) will provide an input, $u^*_{0|k} \neq u_d(k)$, that is able to maintain system safety while remaining as close as possible to $u_d(k)$.

As a final remark, note that the combination of the presented predictive safety filter with a learning algorithm can slow down learning convergence induced by modifications of the input $u_d(k)$. As similarly proposed in [19], one can mitigate this effect by adding a penalty on violating the safety condition to the cost/reward of the learning-based controller.

IV. VEHICLE DYNAMICS AND CONSTRAINTS

In this section, the model used to describe the vehicle dynamics is presented, followed by the system constraints.

A. System Model

In this work we consider a miniature RC car modeled using a standard dynamic bicycle model formulation [18], [20]. Using a dynamic model as opposed to a simpler kinematic model as considered in previous related work, [1], allows us to consider the nonlinear tire forces which have a significant impact on motion during aggressive racing. The state of the model is $x = [p_x, p_y, \psi, v_x, v_y, r]$, with inputs $u = [\delta, \tau]$, where p_x, p_y are the x-y coordinates of the car and ψ is the heading angle in the global coordinate frame; v_x, v_y, and r are the velocities and yaw rate of change in the vehicle's body frame. Finally, δ is the steering angle and τ is the drivetrain command. An illustration is shown in Figure 3.

The system model can be described by the differential equations

$$\dot{x} = \begin{bmatrix} v_x \cos(\psi) - v_y \sin(\psi) \\ v_x \sin(\psi) + v_y \cos(\psi) \\ r \\ \frac{1}{m}\left(F_x - F_{yf}\sin(\delta) + m v_y r\right) \\ \frac{1}{m}\left(F_{yr} + F_{yf}\cos(\delta) - m v_x r\right) \\ \frac{1}{I_z}\left(F_{yf} l_f \cos(\delta) - F_{yr} l_r\right) \end{bmatrix}, \tag{6}$$

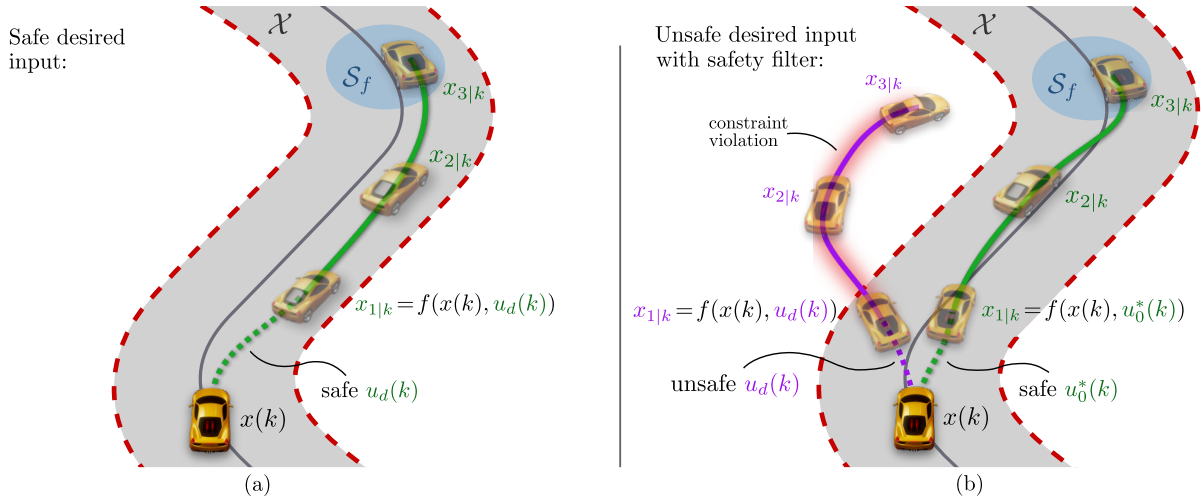

Fig. 2. Diagram (a) shows a possible vehicle trajectory from a safe desired input $u_d(k)$. Diagram (b) shows the resulting vehicle trajectory from an unsafe desired input, where the vehicle ends up leaving the track. An alternate safe input $u_0^*(k)$ applied by the safety filter is shown along with its trajectory.

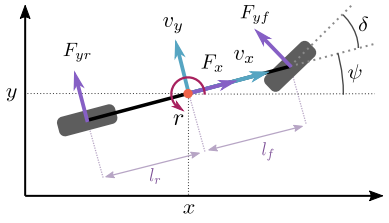

Fig. 3. Dynamic vehicle model diagram.

where m is the car mass, I_z is the yaw moment of inertia, and $l_{f/r}$ is the distance between the center of gravity and the front and rear axles, respectively. The lateral tire forces F_{yf} and F_{yr} are modeled with a simplified Pacejka tire model,

$$\alpha_f = \arctan\left(\frac{v_y + l_f r}{v_x}\right) - \delta, \quad \alpha_r = \arctan\left(\frac{v_y - l_r r}{v_x}\right)$$
$$F_{yf/yr} = D_{f/r}\sin(C_{f/r}\arctan(B_{f/r}\alpha_{f/r})), \tag{7}$$

where α_f and α_r are the tire slip angles [18]. The longitudinal force is modeled as a single force applied to the center of gravity of the vehicle, and is computed as a linear combination of the drivetrain command and velocity as $F_x = C_1\tau + C_2\tau^2 + C_3 v_x + C_4 v_x^2 + C_5 \tau v_x$. The drivetrain command τ can be positive, resulting in forward motion, or negative, resulting in braking.

The continuous-time system in (6) is discretized using Euler forward, obtaining a discrete-time nonlinear system of the form (1).

B. System Constraints

The system is subject to nonlinear state constraints, and polyhedral input constraints of the form

$$\mathcal{X} := \{x \in \mathbb{R}^n | d(x) \leq b\}, \quad \mathcal{U} := \{u \in \mathbb{R}^m | Gu \leq g\}, \tag{8}$$

where $d : \mathbb{R}^n \to \mathbb{R}^{n_b}$, and $G \in \mathbb{R}^{n_g \times m}$. The input constraints consist of bounding the maximum and minimum commands, while the state constraints enforce the safety-critical task of keeping the car within track limits.

To keep the vehicle within the boundaries of the track, we constrain the front two corners of a bounding box around the vehicle, e_{lf} and e_{rf}, as shown in Figure 4. We do not consider drifting maneuvers in this work, but these could be accounted for by also adding similar constraints to the rear corners. The lateral error of the center of gravity with respect to the track center line is e_{lat}, while the yaw error with respect to the track orientation is μ. Given a reference center line position and orientation, x_t, y_t, ψ_t, these states can be written as

$$\begin{aligned} e_{lat}(k) &= -\sin(\psi_t)(x(k) - x_t) + \cos(\psi_t)(y(k) - y_t), \\ \mu(k) &= \psi(k) - \psi_t, \\ e_{lf}(k) &= e_{lat}(k) + l_f \sin(\mu(k)) + \frac{w}{2}\cos(\mu(k)), \\ e_{rf}(k) &= e_{lat}(k) + l_f \sin(\mu(k)) - \frac{w}{2}\cos(\mu(k)), \end{aligned} \tag{9}$$

where w is the width of the vehicle. These two corner points of the bounding box can be bounded by half the width of the track, denoted t, as

$$|e_{lf}| \leq t, \quad |e_{rf}| \leq t. \tag{10}$$

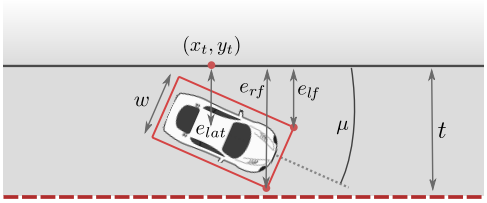

Fig. 4. Track-relative error states used to constrain the vehicle.

V. TERMINAL SET COMPUTATION

The main difficulty in designing a safety filter for the racing application considered is the construction of the positively invariant set, $\mathcal{S}_f$, for the nonlinear vehicle system as described

in Assumption 1. A method for computing polyhedral terminal sets for autonomous driving is presented in [21], but the required simplifying assumptions in the kinematic model used are not suitable for a vehicle performing aggressive maneuvers. Approaches to terminal set design for more general nonlinear systems can be found in [22], [23], [24], where the common idea is to design a set based on a linearized system while using techniques to compensate for linearization errors such that set invariance still holds for the nonlinear system. We take a similar approach that enforces a required Lyapunov dissipation for a range of steady-states to compute the terminal safe set.

We first introduce a transformation in a track-relative coordinate frame that allows for computation of steady-states of the nonlinear vehicle model parameterized by the road curvature. Based on established techniques for terminal set design, we then propose to compute a linear control law capable of stabilizing the nonlinear system in a neighborhood around a specific steady state. We consider a grid of parameter values for the linearized system and compute a positively invariant set for track segments of constant curvature. A-posteriori verification is then performed to ensure invariance holds for the nonlinear system across the full parameter range.

A. Track-Relative Coordinate Transformation and Terminal Steady-States

For the safety certification problem presented in Section III, the terminal set must contain states that are considered safe for the desired system. In a racing context, having the vehicle positioned on the center line and oriented forwards is a safe position, providing the vehicle is able to follow the center line closely under some control law. In order to more easily analyze the system with respect to the center line, the global state is transformed into the track-relative state $x_r = [e_{lat}, \mu, v_x, v_y, r]$, similar to that used in [25]. Here, e_{lat} and μ are the lateral error and orientation error as described in (9), and v_x, v_y, and r remain unchanged from (6). The dynamics of e_{lat} and μ are described by

$$\begin{aligned} \dot{e}_{lat} &= v_x \sin(\mu) + v_y \cos(\mu), \\ \dot{\mu} &= r - c\frac{v_x \cos(\mu) - v_y \sin(\mu)}{1 - c e_{lat}}, \end{aligned} \tag{11}$$

which are parameterized by the curvature of the track, c, at a given point on the center line. We use the same dynamics for v_x, v_y, r as in (6) to describe $\dot{x}_r$, then discretize to obtain

$$x_r(k+1, c) = f_r\left(x_r(k, c), u(k)\right), \ \forall k \in \mathcal{I}_{\geq 0}, \tag{12}$$

with $f_r : \mathbb{R}^{n_r} \times \mathbb{R}^m \to \mathbb{R}^{n_r}$. Constraints keeping the vehicle within track boundaries, $|e_{lat}| \leq t - w/2$, and oriented forwards, $|\mu| \leq \pi/2$, can now be written in polytopic form as $\mathcal{X}_r \coloneqq \{x_r \in \mathbb{R}^{n_r} | H x_r \leq h\}$, where $H \in \mathbb{R}^{n_h \times n_r}$.

The goal is to find a terminal control law for the system (12) that can stabilize the vehicle around the track center line, relating to $e_{lat} = 0$ and a constant velocity $v_x = \mathrm{v}_x$. Since the track-relative dynamics are parameterized by c, different steady-state points $(x_r^e(c), u^e(c))$ exist depending on the current track curvature. The steady-state and corresponding input at a given curvature can be computed by solving (12) for a state and input pairing such that $x_r^e(c) = f(x_r^e(c), u^e(c))$, resulting in

$$x_r^e(c) = [0, \mu^e, \mathrm{v}_x, v_y^e, r^e]^T, \quad u^e(c) = [\delta^e, \tau^e]^T. \tag{13}$$

While direct use of the steady-state (13) as a terminal constraint satisfies the invariance property, the resulting terminal constraints (3f) would become rather restrictive, resulting in conservative behavior of the safety filter. To increase the feasible set of (3) and thereby the safe set of the vehicle states, we propose a design procedure to enlarge the terminal steady state constraint through an invariant set in the following.

B. Terminal Set & Control Law Synthesis

To design a terminal set for the system (6), we use a linearization around the previously introduced equilibrium points (13) to obtain a stabilizing state feedback controller. This allows us to derive a positively invariant set from a Lyapunov function for the corresponding closed-loop system.

We begin by linearizing (12) for a specific steady-state and curvature (13), resulting in

$$\bar{x}_r(k+1, c) = A(c)\bar{x}_r(k, c) + B(c)\bar{u}(k, c) \tag{14}$$

where $A(c)$ and $B(c)$ are the linearization matrices evaluated at a steady state pair $(x_r^e(c), u^e(c))$. The notation $\bar{x}_r(k, c) = x_r(k, c) - x_r^e(c)$ indicates the deviation of the state $x_r(k, c)$ from the steady-state $x_r^e(c)$ for a given curvature, and similarly for $\bar{u}(k, c)$. For the local stabilizing control law, we choose a constant linear controller of the form

$$\kappa_f(k, c) = K\bar{x}_r(k, c), \tag{15}$$

where $K \in \mathbb{R}^{m \times n_r}$.

An ellipsoidal set is chosen for the terminal set as

$$\mathcal{S}_f(c) \coloneqq \left\{\bar{x}_r(k, c) | \bar{x}_r(k, c)^T P \bar{x}_r(k, c) \leq 1\right\} \subseteq \mathcal{X}_r, \tag{16}$$

which is a sublevel set of a quadratic Lyapunov function $V_f(\bar{x}_r(k, c)) = \bar{x}_r(k, c)^T P \bar{x}_r(k, c)$, contained within the state constraints $\mathcal{X}_r$. Although ellipsoidal sets are typically inner-approximations of a system's maximum positive invariant set, they allow for an efficient implementation compared with potentially less conservative, but more complex sets. The matrix $P \in \mathbb{R}^{n_r \times n_r}$ can be obtained by solving the discrete-time Lyapunov equation for the closed-loop system dynamics matrix $A_{cl}(c) = A(c) + B(c)K$, with a pre-specified dissipation rate Q_{dis}:

$$A_{cl}(c)^T P A_{cl}(c) - P \leq -Q_{dis}. \tag{17}$$

The set (16) is then guaranteed to be positively invariant for the system (14) at a given curvature when subject to the control law (15). The dissipation Q_{dis} provides the ability to compensate for linearization errors when stabilizing the original nonlinear system. This dissipation value is chosen using $Q_{dis} = Q + K^T R K$, where Q, R are cost matrices that can be designed to bound the linearization errors by $\bar{x}_r(k, c)^T Q \bar{x}_r(k, c) + \bar{u}(k, c)^T R \bar{u}(k, c)$.

The curvature values of a track with both left and right turns fall into the range $c \in [-c_{max}, c_{max}]$, where c_{max} is the

largest curvature value on the track. We therefore want a single control law that stabilizes the system at any curvature within the given range. This is done by first introducing a set of $n_c \in \mathbb{R}$ equidistant incremental curvature values in $[-c_{max}, c_{max}]$, and computing the corresponding equilibrium states, inputs, and linearization matrices for each: $\left\{x^e_{r,i}, u^e_i, A_i, B_i\right\}$, $\forall i \in \mathcal{I}_{[1,n_c]}$. We then impose the stability condition from (17) with a single common control matrix K for all steady-states. This allows us to compute the control law and resulting invariant set with a semidefinite program (similarly used in [24]):

$$\min_{E,Y} \quad -\log\det E \tag{18a}$$

$$\text{s.t.} \quad \forall i \in \mathcal{I}_{[1,n_c]}:$$

$$E \succeq 0 \tag{18b}$$

$$\begin{bmatrix} \left([h]_j - [H]_j x^e_{r,i}\right)^2 & [H]_j E \\ E[H]_j^T & E \end{bmatrix} \succeq 0, \forall j \in \mathcal{I}_{[1,n_h]} \tag{18c}$$

$$\begin{bmatrix} ([g]_l - [G]_l u^e_i)^2 & [G]_l E \\ E[G]_l^T & E \end{bmatrix} \succeq 0, \forall l \in \mathcal{I}_{[1,n_g]} \tag{18d}$$

$$\begin{bmatrix} E & EA_i^T + Y^T B_i^T & EQ^{\frac{1}{2}} & Y^T R^{\frac{1}{2}} \\ A_i E + B_i Y & E & 0 & 0 \\ Q^{\frac{1}{2}} E & 0 & I & 0 \\ R^{\frac{1}{2}} Y & 0 & 0 & I \end{bmatrix} \succeq 0 \tag{18e}$$

where $E \coloneqq P^{-1}$, and $Y \coloneqq KE$. The solution to (18) allows us to extract a maximal volume ellipsoidal set (16) that is invariant for the closed-loop system $A_{cl}(c)$ at each of the n_c gridded curvature values. The matrix inequalities described by (18c) and (18d) impose each state and input half-space constraint indexed by $j = 1, .., n_h$ and $l = 1, .., n_g$ for each equilibrium point $i = 1, .., n_c$. The constraint in (18e) can be derived from the Lyapunov decrease condition (17) and Schur complements.

Since the resulting set is invariant by design only for the linearized system and only at the chosen curvature values, we must further verify that invariance holds for the nonlinear system as well as for the continuous range of curvatures. This is done via an additional optimization problem that searches the set for any state and curvature pairing that leads to an invariance violation for the nonlinear system under the computed terminal control law:

$$\max_{\bar{x}_r, c} \quad \bar{x}_r(k+1, c)^T P \bar{x}_r(k+1, c) \tag{19a}$$

$$\text{s.t.} \quad \bar{x}_r(k, c)^T P \bar{x}_r(k, c) \leq 1 \tag{19b}$$

$$\bar{x}_r(k+1, c) = f(\bar{x}_r(k), \kappa_f(k), c) \tag{19c}$$

$$c \in [c^{min}, c^{max}]. \tag{19d}$$

If the optimal objective value (19a) is less than 1, then $\mathcal{S}_f(c)$ is verified as invariant for the nonlinear system; otherwise, the problem has found a state for which the set is not invariant under the nonlinear dynamics. In this case, the set can be incrementally scaled down until no violating points are found, with the limit reaching the vehicle steady-state as a feasible solution. While it is generally difficult to find the global optimum of (19), a practical approach is to randomly re-initialize a suitable local optimization technique multiple times, see, e.g. [26] for further details.

Note that the invariance guarantees of the proposed terminal set are valid for constant curvatures. Since we consider a track made up of connecting constant curvature segments, the theoretical invariance property therefore holds on each individual segment. However, the guarantees do not strictly hold for the instantaneous change of curvature between segments due to the resulting shift in the steady-state set point. Since the linearization-based control law and invariant set design inherently introduce some conservativeness, we observe in practice that changing set points can still be efficiently compensated. We therefore do not explicitly account for this change in curvature, and consider invariance for the individual track segments as practically adequate for the terminal safe set.

The curvature value used for the terminal set when solving Problem (3) is taken as the curvature a certain distance ahead of the vehicle along the track. This distance is heuristically chosen as a function of the current desired torque input, $u_{d,\tau}(k)$, and the time horizon of the problem, $t_N = N \cdot T_s$, to generate a reasonable distance ahead for the terminal set.

VI. EXPERIMENTS

To demonstrate the performance of the proposed safety filter, the scheme is implemented to keep a remote control vehicle inside track boundaries. We first present an experiment showing the safety filter in a driver-assistance scenario, where the desired inputs are provided by a human driver. This is followed by a learning-based control application, where a neural network policy is safely learned and deployed on the vehicle using imitation learning.

To ensure feasibility of the MPC problem (3), the track width constraint (10) and terminal set constraint (16), are implemented as soft constraints. The problem is solved online using acados [27] with a real-time iteration SQP scheme, a horizon length of $N = 60$, and a sampling frequency of 80 Hz. The terminal set computations in (18), (19) are solved offline using MOSEK [28], with $n_c = 21$ equilibrium points spanning curvatures $[-2.5, 2.5]$. We additionally verified in simulation that larger instantaneous curvature changes up to 6.66 can be compensated for. The verification problem in (19) is solved $10,000$ times from randomly selected initial conditions, and the resulting objective value never exceeds 1.

A Kyosho Mini-Z 1:28-scale vehicle is used on a 0.80 m constant-width track as the test platform for all experiments. A VICON motion capture system provides position and orientation information, which is used by an Extended Kalman Filter to produce a complete state estimate. The safe control inputs are sent via radio controller to the vehicle. The closed loop system is implemented using ROS running on a Lenovo ThinkPad P1 with Ubuntu 18.04, Intel Core i7-9750H processor, and 32 GB RAM.

A. Manual Driver Assistance

By combining the safety certification with human driver inputs, a driver-assistance system is created that provides necessary intervention should the driver make a mistake that

would endanger the vehicle. Since the safety certification is designed to be minimally invasive, it gives the driver free control of the vehicle as long as their actions remain safe, only intervening when required.

In this experiment, the manual driver inputs are provided by a physical joystick. Figure 5 shows the vehicle trajectory and corresponding inputs from a single lap driven with the safety certification active. In the vehicle trajectory plot, the color-map shows the L2-norm of the difference between the desired and safe control input vectors, indicating the magnitude of modification by the safety filter. The input comparison plots show the safety filter commands initially closely correspond to the driver commands up until the dashed line, indicating that the driver commands are being certified as safe and applied to the vehicle. After this, the safety certification begins to intervene in both steering and throttle inputs as the driver purposefully fails to steer around corners, or swerves the car toward the wall. The plot of the trajectory demonstrates how the safety certification is able to keep the vehicle within track boundaries at all times, while still managing to track the desired inputs whenever possible.

B. Imitation Learning

Imitation learning is a method for learning a policy for a task by replicating actions from expert demonstrations. We implement an iterative imitation learning algorithm, DAgger (Dataset Aggregation) [29], to learn a deterministic racing policy. In DAgger, a policy is first initialized using supervised learning from an expert demonstration, and is then deployed directly on the task. The expert labels all states visited by the learned policy with the optimal action, which is then added to the dataset for the policy to retrain on. This process is repeated iteratively with the intention that the learned policy is able to improve from previous mistakes.

Since DAgger relies on rolling out the learner policy during training, it can be combined with the proposed safety filter to provide a safe training environment to learn a racing controller using the physical vehicle. A feedforward neural network with 3 hidden layers, 64 neurons per layer, and ReLU activation functions is used as the policy architecture, outputting a drivetrain and steering command. The input to the network is the vehicle state in track-relative coordinates, along with 30 curvatures over the next 1.5 meters of track as $x_{NN} = [x_r, c_1 \dots c_{30}]$. Training the network consists of supervised learning to minimize the L2-norm of the difference between expert and network commands. The expert policy used is a Model Predictive Contouring Controller (MPCC), presented in [20], which maximizes track progress while staying inside track boundaries, and has proved successful in other racing applications [30]. Imitating a finite-horizon optimal policy like MPCC can be beneficial, as the states visited by the network controller in each iteration can be labeled offline using an MPCC with a longer horizon that cannot be used in practice due to solve time requirements. The resulting neural network then imitates a high performance policy that otherwise could not be achieved by the expert in real-time. DAgger is set up on the experimental platform alongside the safety filter to allow

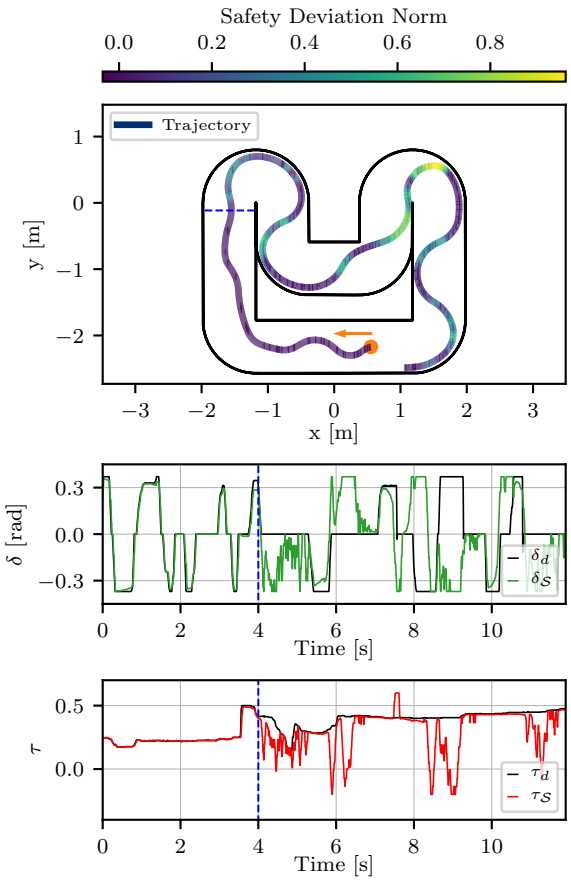

Fig. 5. Vehicle trajectory (top) and control inputs (middle, bottom) for a human providing the desired control signal by joystick. The safety filter intervention is shown via heat map on the trajectory. The orange dot and arrow indicate the starting point and travel direction; the dashed blue line indicates the transition from generally safe driver inputs to unsafe inputs.

for completely automated safe training. Safety is provided both during data collection when the neural network policy is operating, and during transition periods as the vehicle stops to retrain the policy. Figure 6 shows two trajectory plots of DAgger episodes where the neural network policy is active alongside the safety filter. The plot in 6(a) shows the trajectory during the first DAgger episode, where multiple instances of necessary safety filter intervention can be seen, as indicated by the color of the safety deviation norm. In the early stages of training, the neural network policy has only been trained on the initial expert dataset, so it struggles to bring the vehicle onto the optimal racing line without trying to cut corners. The safety filter must then deviate from applying the desired inputs to computing safe inputs that keep the vehicle in the track. The plot in 6(b) shows the trajectory from the 4th episode, which is much more consistent than the initial policy, with almost no major safety filter interventions. The trajectory is aligned more closely with the optimal trajectory from MPCC, demonstrating an improved policy over previous iterations.

VII. CONCLUSIONS

In this work, we have presented a predictive safety filter that is able to render a closed-loop vehicle system safe when

subject to any unsafe control signal. A method for computing and verifying an invariant terminal set for the nonlinear vehicle system on constant curvature track segments is presented, providing a safe operating domain that does not overly restrict the desired policy. The experiments illustrate two applications where the safety filter is able to ensure safety of the vehicle during dynamic high speed maneuvers.

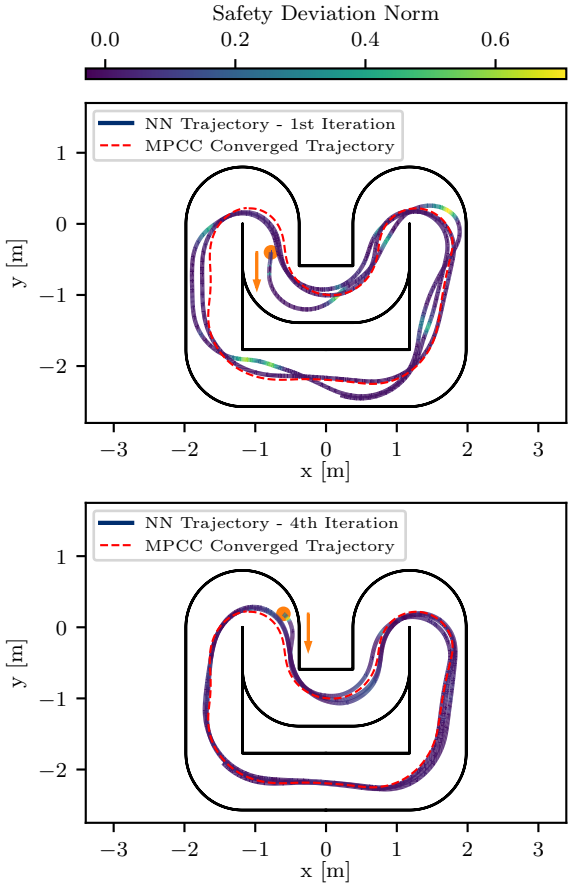

Fig. 6. Vehicle trajectories shown during the first (top) and fourth (bottom) episodes of DAgger; safety filter intervention is shown via heat-map, and the converged expert MPCC trajectory is shown in red. Initial location and direction of travel are shown in orange.

REFERENCES

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, 2019.

[2] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.

[3] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.

[4] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe on-line control system upgrades," in *Proceedings of the American Control Conference*, vol. 6, 1998, pp. 3504–3508.

[5] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. Springer, 2004, pp. 477–492.

[6] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings*, vol. 40, no. 12, pp. 462–467, 2007.

[7] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, vol. 120. PMLR, 2020, pp. 708–717.

[8] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 2017, pp. 22–31.

[9] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, 2016, pp. 4661–4666.

[10] J. H. Gillula and C. J. Tomlin, "Guaranteed safe online learning of a bounded system," in *IEEE Conference on Intelligent Robots and Systems*, 2011, pp. 2979–2984.

[11] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 1517–1522.

[12] K. P. Wabersich and M. N. Zeilinger, "Scalable synthesis of safety certificates from data with application to learning-based control," in *European Control Conference*, 2018, pp. 1691–1697.

[13] L. Wang, D. Han, and M. Egerstedt, "Permissive Barrier Certificates for Safe Stabilization Using Sum-of-squares," *Proceedings of the American Control Conference*, vol. 2018-June, pp. 585–590, 2018.

[14] A. Chakrabarty, C. Danielson, S. Di Cairano, and A. Raghunathan, "Active learning for estimating reachable sets for systems with unknown dynamics," *IEEE Transactions on Cybernetics*, pp. 1–12, 2020.

[15] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *IEEE Conference on Decision and Control*, 2018, pp. 7130–7135.

[16] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Transactions on Automatic Control*, 2021.

[17] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[18] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer US, 2011.

[19] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 1424–1431.

[20] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, Jul 2014.

[21] P. F. Lima, J. Mrtensson, and B. Wahlberg, "Stability conditions for linear time-varying model predictive control in autonomous driving," in *IEEE Conference on Decision and Control*, 2017, pp. 2775–2782.

[22] H. Chen and F. Allgwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205 – 1217, 1998.

[23] A. Carron and M. N. Zeilinger, "Model predictive coverage control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6107–6112, 2020, 21th IFAC World Congress.

[24] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger, "Distributed synthesis and stability of cooperative distributed model predictive control for linear systems," *Automatica*, vol. 69, pp. 117 – 125, 2016.

[25] J. L. Vazquez, M. Bruhlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[26] C. G. E. Boender and H. E. Romeijn, *Stochastic Methods*. Boston, MA: Springer US, 1995, pp. 829–869.

[27] R. Verschueren *et al.*, "acados: a modular open-source framework for fast embedded optimal control," 2019.

[28] M. ApS, *The MOSEK optimization toolbox for MATLAB*, 2019.

[29] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 627–635.

[30] A. Kabzan, Liniger, J. Lygeros, and R. Siegwart et. al, "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.