2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Learning Input Constrained Control Barrier Functions for Guaranteed Safety of Car-Like Robots

Sven Brüggemann[1], Dominic Nightingale[1], Jack Silberman[2], and Maurício de Oliveira[1]

Abstract— **We propose a design method for a robust safety filter based on Input Constrained Control Barrier Functions (ICCBF) for car-like robots moving in complex environments. A robust ICCBF that can be efficiently implemented is obtained by learning a smooth function of the environment using Support Vector Machine regression. The method takes into account steering constraints and is validated in simulation and a real experiment.**

I. Introduction

Safety is often prioritized over performance criteria due to physical limitations of the system leading to unsafe operation (e.g. actuator saturation, power constraints, etc), and/or safety requirements imposed as a result of interacting with the environment. Examples of the latter are obstacle avoidance [1], [2], lane-keeping for autonomous vehicles [3] and navigation tasks [4].

Recently, Control Barrier Functions (CBFs) [5] have become a popular strategy to enforce safety. One common setup for CBFs is that of *supervising* a given nominal action which may not necessarily be safe. The CBF acts as a *safety filter* which computes an overriding safe action that is as "close as possible" to the nominal one if safety is at risk. Yet, whereas CBFs provide a strong theoretical framework upon which to build safety-critical systems, there are still practical/implementation difficulties, and many approaches are validated in simulation only [6]–[13]. One of these difficulties arise for systems of high relative degree which naturally arise for car-like robots as safety is commonly expressed in terms of x-y coordinates only, excluding the steering angle. This work is concerned with the practical deployment of a supervising Input Constrained CBF (ICCBF) based safety filter for car-like robots. The following aspects will be considered in detail:

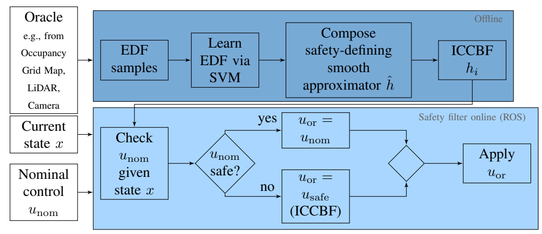
CBF Learning directly from the Euclidean Distance Function (EDF) encoding a global map in the form of a kernelized Support Vector Machine (SVM) representation (loosely speaking, EDFs represent the shortest distance to any unsafe point in space);

Robustification in practice against uncertainties including noisy data, measurement and process noise, discretization and uncertain dynamics (see Section V) to ensure safe operation based on the resulting smooth SVM approximation;

Input Constraints resulting from robot steering limitations are handled by a reformulated model and an ICCBF safety filter;

Validation via simulation;

Implementation on a 1:10 RC car.



(a) Proposed safety filter design process.

Fig. 1: Overview of approach and photos of experiment.

A diagram of the blocks involved in the complete process is given in Fig. 1(a). Notice that though we explicitly learn the EDF via SVM, we implicitly also learn the related ICCBF, as gradients are available in closed form. While we choose to learn the EDF, our method is also applicable to other map representations such as [14].

Contribution: ICCBF-based filter design depends on the explicit knowledge of a valid ICCBF. This is challenging in practice, especially for car-like robots with actuator constraints moving in complex unsafe regions; complex here means areas that cannot be described by simple geometries like ellipsoids. We overcome this difficulty by learning a robust, smooth approximator of the EDF (whose explicit form is unknown) with safety guarantees for arbitrary geometries while considering steering constraints. Unlike most works on CBFs, we consider high-relative degree systems, and towards implementation we develop a closed-form, vectorized algorithm which outperforms a naive formulation by three orders of magnitude (see computing time in Section V). Essential for the implementation of the algorithm is our reformulation of the vehicle model which reduces the number of input constraints to one. We present a real-world experiment to provide evidence that simulations are transferable to real robots. The comprehensive approach from learning from data all the way to the derivation of an implementable, experimentally validated algorithm with much attention to detail is our main contribution.

Related work: Learning and CBF-based design usually assumes that CBFs are known but other ingredients such

[1]Sven Brüggemann, Dominic Nightingale, and Maurício de Oliveira are with the Mechanical & Aerospace Engineering Department, University of California, San Diego, CA 92093-0411, USA {sbruegge,djnighti,mauricio}@eng.ucsd.edu

[2]Jack Silberman is with the Department of Electrical & Computer Engineering, University of California, San Diego, CA 92093-0411, USA jacks@eng.ucsd.edu

as the system dynamics are uncertain (see e.g. [6], [15]). Learning the CBF itself was proposed in [7], [16], where expert demonstrations are used to derive safety guarantees. [17] avoids the assumption of control-affine dynamics by jointly learning the Control Lyapunov Function (CLF), CBF and related policy. Yaghoubi et al. [8] propose imitation learning to learn the CBF-based controller, avoiding solving the related QP. In [9] data is used to learn certifiable safe control laws based on CBFs for hybrid systems. Removing the assumption of known dynamics for multi-agent systems is the focus of [6]. Learning decentralized CBFs for the safety of multi-agent systems is also treated in [18]. A model-based learning approach to synthesize robust feedback control was presented in [19]. The authors of [10] learn the CBFs from sensor data and combine it with the CLFs as part of a QP. A recent overview of the rapidly growing field of learned safety certificates is [20].

Most related to our work are [11]–[13], [21]. The authors of [11] develop a neural network (NN) safety filter for bicycle models facing stationary, fixed-radius obstacles in the plane. Tensor gradients of the NN required for implementation tend to be more expansive than our vectorized formulation. For obstacle avoidance, in [21] NNs are trained online to approximate the distance function, based on which CBFs are directly formulated as a convex second-order cone problem, which is usually more costly than an explicit QP solution. Learning the EDF (and its derivatives) for CBF design was proposed in [12]. Therein, Gaussian Process was used as a supervised learning strategy requiring derivative measurements (or corresponding numerical approximations). Towards safety for obstacle avoidance [13] uses SVM classification to learn the CBF based on sensor data (on- and offline). However, derivatives of the classifier required for the CBF design may vanish or be unbounded with related numerical instabilities or loss of safety (see Section IV). In contrast to the present paper, none of these learning-based methods directly consider systems of high relative degree with input constraints; nor do they discuss implementation details or present real-world validation and related code.

II. Setup

Consider the control-affine system,

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

with $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ as the current state and control input $u \in \mathcal{U}$ for some compact constraint set $\mathcal{U} \subset \mathbb{R}^{n_u}$, and smooth functions $f : \mathcal{X} \to \mathbb{R}^{n_x}$ and $g : \mathcal{X} \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$. Let safety be described by a *safe set* defined as the 0-superlevel set of a sufficiently-often differentiable function $h(x) : \mathbb{R}^{n_x} \to \mathbb{R}$:

$$\mathcal{C} \doteq \{x \in \mathbb{R}^{n_x} : h(x) \geq 0\}, \tag{2}$$

and write $\partial\mathcal{C} \doteq \{x \in \mathbb{R}^{n_x} : h(x) = 0\}$, $\mathrm{int}(\mathcal{C}) \doteq \{x \in \mathbb{R}^{n_x} : h(x) > 0\}$. We assume that $\mathcal{C}$ is non-empty, closed, and simply connected.

Definition 1 (Forward Invariance & Safety)**.** *The set $\mathcal{C}$ is forward invariant if $x_0 \doteq x(t_0) \in \mathcal{C}$ at time t_0 implies that $x(t) \in \mathcal{C}$ for all time $t \geq t_0$. The system is considered safe with respect to $\mathcal{C}$ if $\mathcal{C}$ is forward invariant.*

For common vehicle models, safety is often specified in terms of the vehicle position only, without considering heading information [22]–[24]. This means that CBF-based safety design for car-like robots must overcome two hurdles: it needs to deal with a relative degree larger than one *and* vehicle input constraints, which are not directly expressed in terms of the robot's position, must be enforced (e.g. steering limitations), potentially compromising feasibility. Since CBF safety is often formulated for relative-degree-one systems (e.g. [5], [25]–[27]), here we recall results on ICCBF from [28] which cover constrained systems with higher relative degree. Consider system (1) with constrained input $u \in \mathcal{U}$ and safe set $\mathcal{C}$. Let

$$\begin{aligned} h_0(x) &\doteq h(x), \\ h_1(x) &\doteq \inf_{u \in \mathcal{U}} \{L_f h_0(x) + L_g h_0(x)u + \alpha_0(h_0(x))\} \\ &\vdots \\ h_N(x) &\doteq \inf_{u \in \mathcal{U}} \{L_f h_{N-1}(x) + L_g h_{N-1}(x)u + \alpha_{N-1}(h_{N-1}(x))\}, \end{aligned} \tag{3}$$

where each α_i is a class-$\mathcal{K}$ function (see [29, Section 4, Def. 4.2]) and $N \geq r - 1$, with r as the relative degree of (1) with output function $h(x)$. The case $N = r - 1$ leads to a standard exponential CBF [30]. Define the related safety sets as

$$\mathcal{C}_i = \{x \in \mathcal{X} : h_i(x) \geq 0\}, \tag{4}$$

with $\mathcal{C}_0 = \mathcal{C}$ and assume that $\mathcal{C}^\star = \mathcal{C}_0 \cap \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_N$ is not empty, closed and simply connected.

Definition 2 ([28])**.** *For system* (1) *with safe set $\mathcal{C}$ and continuously differentiable class-$\mathcal{K}$ functions $\alpha_0, \ldots, \alpha_{N-1}$, if there exists a class-$\mathcal{K}$ α_N such that*

$$\sup_{u \in \mathcal{U}} \{L_f h_N(x) + L_g h_N(x)u + \alpha_N(h_N(x))\} \geq 0 \; \forall x \in \mathcal{C}^\star, \tag{5}$$

then $h_N(x)$ is an Input Constrained Control Barrier Function (ICCBF).

Importantly, $\mathcal{C}^\star \subset \mathcal{C}$, so the idea of ICCBFs is to shrink the safe set with each $\mathcal{C}_i$ until safety is guaranteed in the presence of bounded control inputs, as for all $x \in \mathcal{C}^\star$ *any* control for which $h_N(x) \geq 0$ also ensures $h_{N-1}(x) \geq 0$, and eventually $h_0(x) = h(x) \geq 0$. Note that the need to perform the minimization in (3) on each $h_i(x)$ can be very difficult. One of the contributions of this paper is to show how this can be effectively done for car-like robots with steering constraints; see the last paragraph of Section IV.

Provided a valid ICCBF, let the nominal closed-loop be driven by control input $u_{\mathrm{nom}}(x)$, which may not be safe. A safety filter with overriding control, $u_{\mathrm{or}}(x)$, interrupts nominal operation if safety cannot be guaranteed is obtained through:

$$u_{\mathrm{or}}(x) = \underset{u \in \mathcal{U}}{\arg\min} \, \|u_{\mathrm{nom}}(x) - u\|^2, \tag{6a}$$

$$\text{s.t. } L_f h_N(x) + L_g h_N(x)u + \alpha_N(h_N(x)) \geq 0. \tag{6b}$$

Note that vanishing $L_g h_N(x)$ (due to, e.g., a vanishing partial derivatives of $h_N(x)$) risks safety as the inequality constraint would be independent of control u. It follows from smoothness implying locally Lipschitz and [28, Theorem 1] that the above problem admits the closed-form solution.

Lemma 1. *Consider system* (1) *with input constraints* $\mathcal{U}$ *and* $u_{\text{nom}} \in \mathcal{U}$. *Define*

$$u_{\text{safe}}(x) = -L_g h_N(x)^{-1}(L_f h_N(x) + \alpha_N(h_N(x))).$$

If $h_N(x)$ *is an ICCBF and* $L_g h_N(x)$ *is full rank, then*

$$u_{\text{or}}(x) = \begin{cases} u_{\text{nom}}(x), & L_f h_N(x) + L_g h_N(x) u_{\text{nom}} \\ & \quad + \alpha_N(h_N(x)) \geq 0, \\ u_{\text{safe}}(x), & \textit{otherwise}, \end{cases} \tag{7}$$

renders (1) *safe. If* $L_g h_N(x)$ *is not full rank but* $L_f h_N(x) + \alpha_N(h_N(x)) \geq 0$ *then* (1) *is still safe.*

The proof is omitted due to space constraints. We note that if the rank condition on $L_g h_N(x)$ cannot be guaranteed we can numerically solve the QP in (6a). For the car-like robot however, this reduces to $L_g h_N(x) \in \mathbb{R}$ not being zero.

III. Safety design for car-like robots

Assume a car-like robot that moves in a confined planar space described by compact $\mathcal{Z} \subset \mathbb{R}^2$ and suppose certain positions in x- and y-coordinate within $\mathcal{Z}$ are free, denoted by the set $\mathcal{Z}_{\text{free}} \subset \mathcal{Z}$, and others are occupied, written as $\mathcal{Z}_{\text{occupied}}$. We wish to ensure safety by avoiding $\mathcal{Z}_{\text{occupied}}$ using the ICCBF design from Lemma 1. Accordingly, $\mathcal{Z}_{\text{occupied}}$ is associated with $h(x) < 0$.

Characterization of safe regions via Euclidean Distance Function: Safety is characterized by avoidance of occupied map positions, which represent obstacles and off-road areas. These safe/unsafe regions can be related to the EDF $d : \mathcal{Z} \to \mathbb{R}$,

$$d(z) \doteq \inf_{z' \in \mathcal{Z}_{\text{occupied}}} \|z' - z\|, \tag{8}$$

which describes the shortest distance to any occupied point in $\mathcal{Z}$. The EDF cannot be directly used to construct an ICCBF because it is not smooth enough. Instead, a suitable approximation will be learned in Section IV.

Kinematic model with steering state and input constraints: We assume a constant positive forward velocity, v_f, and model the behavior of the front wheels (rather than the rear wheels or center of gravity) using the kinematic bicycle model [31, Section 2, Table 2.1]:

$$\begin{pmatrix} \dot{x}_{\text{f}} \\ \dot{y}_{\text{f}} \\ \dot{\theta} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} v_{\text{f}} \cos(\theta + \delta) \\ v_{\text{f}} \sin(\theta + \delta) \\ v_{\text{f}} \sin(\delta)/L \\ u \end{pmatrix}, \tag{9}$$

where x_{f} and y_{f} are the x- and y-position of the front axle, θ the robot heading, δ the steering angle and $\dot{\delta}$ the front steering rate. We choose the control input to be the control signal $u = \dot{\delta}$ in order to ensure a continuous steering action. Doing so also makes the system control-affine, which is required by the proposed safety filter. We assume that both the steering angle and the angular steering rate are bounded as in:

$$|\delta| \leq \delta_{\max}, \quad |u| \leq u_{\max}. \tag{10}$$

Modified model with only input constraints: Position *and* steering constraints pose difficulties for the framework of ICCBFs[1]. It also complicates the required minimization in (3) (see discussion at the end of Section IV). For this reason, we reduce the number of constraints to one by parameterizing the steering angle through a change in coordinates using the modified sigmoid function:

$$\delta \doteq \phi(\zeta) \doteq \delta_{\max}\left(\frac{2}{1 + e^{-\zeta}} - 1\right). \tag{11}$$

We consider the modified model

$$x = \begin{pmatrix} x_{\text{f}} \\ y_{\text{f}} \\ \theta \\ \zeta \end{pmatrix}, \quad \dot{x} = \begin{pmatrix} \dot{x}_{\text{f}} \\ \dot{y}_{\text{f}} \\ \dot{\theta} \\ \dot{\zeta} \end{pmatrix} = \begin{pmatrix} v_{\text{f}} \cos(\theta + \phi(\zeta)) \\ v_{\text{f}} \sin(\theta + \phi(\zeta)) \\ v_{\text{f}} \sin(\phi(\zeta))/L \\ \left(\frac{\partial \phi(\zeta)}{\partial \zeta}\right)^{-1} u \end{pmatrix}, \tag{12}$$

which ensures that the bounds on the steering angle and steering rate will not be exceeded if the single input constraint,

$$|u| \leq u_{\max}, \tag{13}$$

is satisfied. The next section shows how we can learn an ICCBF for the above model.

IV. Learning a robust safety filter

Safety as defined above relies on the output function, $h(x)$, whose sign characterizes the safe/unsafe regions. Even when $h(x)$ is known, it may be difficult to calculate a suitable ICCBF satisfying Definition 2. Challenges include the need for $h(x)$ to be available in closed form and to be *continuously differentiable* enough to define the output function $h_N(x)$ when $N > 1$. The EDF in (8) accurately defines the safe/unsafe regions but is not continuously differentiable and usually not available in closed form. Yet, samples can be generated, for example, from LiDAR measurements through test runs, occupancy grid representations, and satellite images. Assuming availability of such data, in this section we propose to use an SVM as a smooth surrogate for the EDF. Through an additional *robustness parameter* we also provide theoretical safety guarantees of the learned safety filter. Lastly, we reveal additional computational advantages of the safety filter which are key to practical implementation.

SVM smooth approximation to the EDF: We learn the EDF via ϵ-SVM regression and then use the data to *robustify* the related output function. Reasons for using a regression model instead of SVM classification as used in [13] include the gradients needed for the safety filter. A classification model has an infinite gradient at the barrier and zero gradients in the safe regions, which likely renders the ICCBF invalid due to rank deficiencies in the term $L_g h_N(x)$. In contrast, a regression model has smaller gradients at the boundary and generically non-zero gradients inside the safe region. Further, the simplicity of the SVM model allows us to obtain an efficient expressions for the tensor gradients as part of the safety filter in (7) which enables implementation on resource-limited robot hardware.

[1]It is possible to impose multiple safety-related constraints, at the expense of feasibility guarantees. See the control-sharing property [32], which is often difficult to establish.

The basic idea of ϵ-SVM regression is to find a function that is at least ϵ-close to the all training samples and as flat as possible. The samples themselves serve as function parameters (support vectors), only if they are outside the ϵ-tube. Importantly, the dual problem only involves dot products of samples so that for nonlinear regression, dot products of nonlinear functions $\Phi(z)$ can be implicitly described by kernel functions $k(z, z') = \langle \Phi(z), \Phi(z')\rangle$. Consider training data $(d_i, z_i)_{i=1}^{n_s}$ related (8). The primal is defined as follows:

$$\min_{w,a,\xi^*,\xi} \quad \frac{1}{2}w^\top w + G\sum_{i=1}^{n_s}(\xi_i + \xi_i^*), \tag{14a}$$
$$\text{s.t.} \quad -\epsilon - \xi_i^* \le d_i - w^\top \Phi(z_i) - a \le \epsilon + \xi_i, \tag{14b}$$
$$\xi_i^*, \xi_i \ge 0, \tag{14c}$$

where ξ_i^* and ξ_i are slack variables. The parameter $G > 0$ balances flatness in the feature space against how samples outside the ϵ region are tolerated. We use the standard radial basis kernel,

$$k(z, z') = e^{-\gamma\|z-z'\|^2}, \tag{15}$$

with $\gamma > 0$ describing the impact of each sample on its neighboring points during training. Solving (14) via its dual generates a smooth (hence locally Lipschitz) approximator for the EDF:

$$\hat{d}(z) = \sum_{i=1}^{n_{\text{sv}}} \kappa_i k(z_i, z) + a, \tag{16}$$

with $n_{\text{sv}} \le n_s$ as the number of support vectors and $\kappa \in \mathbb{R}^{n_{\text{sv}}}$ dual coefficients. Typically, the number of support vectors reduces as ϵ increases and C decreases [33].

Robust output function and the ICCBF: Having obtained the smooth approximation $\hat{d}$, we define the output function as

$$h(x) = \hat{h}(x) - \beta, \quad \hat{h}(x) = \hat{d}(Cx), \quad C = \begin{bmatrix} I_2 & 0 \end{bmatrix}, \tag{17}$$

in which I_2 denotes a 2×2 identity matrix and $\beta \ge 0$ is a robustness parameter. Constructing $h_i(x), \hat{h}_i(x), i = 0, \dots, N$, as in (3), it follows that $h_i = \hat{h}_i - \alpha_i \circ \alpha_{i-1} \circ \cdots \circ \alpha_0 \circ \beta$, where is $\circ$ the function composition operator with $\alpha_i \circ \alpha_{i-1} \doteq \alpha_i(\alpha_{i-1})$. In [25] it is shown that a change in the level set, akin to that induced by β, provides robustness against input disturbances; see also Section V for other sources of uncertainties. Appropriate selection of β and ϵ involves data and uncertainty analysis. Note that as common, $h_0(x)$ characterizes safety only in terms of the x-y coordinates, excluding heading and steering information, whereas h_r is the first barrier which explicitly considers steering rate constraints, alleviating the burden of infeasibility of the QP.

The following theorem states that if the approximation of the EDF is sufficiently accurate, then the robust safety filter with control law (7) ensures safety for the robot.

Theorem 1. *Consider system* (1) *with input constraints* $\mathcal{U}$ *and* $u_{\text{nom}} \in \mathcal{U}$. *Let* d *and* $\hat{d}$ *be as in* (8) *and* (16) *and* $\sigma > 0$ *be such that* $\max_{z \in Z_{\text{free}}} |\hat{d}(z) - d(z)| \le \sigma$. *Consider* h *and* $\beta > \sigma$ *in* (17)*. If* $h_N(x)$ *in* (17) *is an ICCBF and* $L_g h_N(x)$ *is full rank then the control* u_{or} *in* (7) *keeps system* (1) *in* $\mathcal{Z}_{\text{free}}$ *and hence safe.*

Proof. By Lemma 1, $h(x) \ge 0$, so that by (17) $h(x) = \hat{d}(Cx) - \beta \ge 0$. Then, by hypothesis and $d(Cx) \ge 0$, $d(Cx) \ge \hat{d}(Cx) - \sigma \implies d(Cx) \ge \beta - \sigma > 0$ for all $Cx \in \mathcal{Z}_{\text{free}}$. □

SVMs can approximate any continuous function and finite number of data to any desired accuracy [34]. Thus, choosing $\epsilon \le \sigma$ in (14) guarantees $|\hat{d}(z_i) - d(z_i)| \le \sigma$ for all data samples, z_i. The assumed bound on $\hat{d}(z)$ is therefore not a limitation of the proposed learning approach, but rather a statement on the quality of the data.

Calculation of the ICCBF: Theorem 1 builds on the modified model (12) with only one input constraint, which is key for implementation. With box constraint $|u| \le u_{\max}$ we can directly compute the closed-form solution of the minimization as part of the ICCBF construction in (3), almost everywhere:

$$\begin{aligned} h_i(x) &= \inf_{u\in\mathcal{U}} \{L_f h_{i-1}(x) + L_g h_{i-1}(x)u + \alpha_{i-1}(h_{i-1}(x))\} \\ &= L_f h_{i-1}(x) - |L_g h_{i-1}(x)|u_{\max} + \alpha_{i-1}(h_{i-1}(x)), \end{aligned}$$

and therefore also its required derivatives. Additionally, the full rank condition of the explicit control law $u_{\text{or}}(x)$ simplifies to $L_g h_N(x) \ne 0$, and no matrix inversion is necessary. Next we apply the proposed learned, robust safety filter in simulation and in a real experiment.

V. Simulation & Experimental Results

The simulation and experiment focus on the application of lane-keeping and obstacle avoidance.

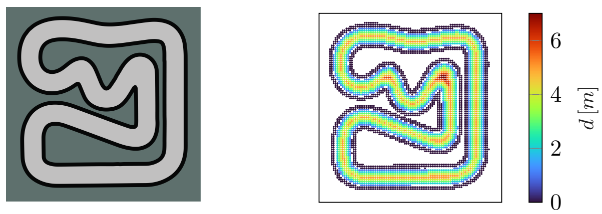
Simulation: Consider the track shown in Figure 2(a) and suppose the car-like robot seeks to move towards some goal z_{goal} without information about the road. To avoid moving off road we implement the safety filter discussed above. From the image in Figure 2(a) we generate EDF data $\mathcal{D}$ (Figure 2(b)) and learn $\hat{d}(z)$ offline. We optimize the learning hyperparameters based on a 10-fold cross-validated grid-search[2] using 50-50 split for training and validation, resulting in $n_{\text{sv}} = 1737$ support vectors and R^2 score of 0.9823 with max. abs. error of 1.01m. The trajectory obtained with the proposed safety filter successfully remains within the road as shown in Figure 3(c).

Implementation: Our simulations and experiment suggest that taking $N = 2$ is sufficient for obtaining an ICCBF[3]. The corresponding safety filter therefore requires the evaluation of the first three partial derivatives of $h(x)$, which involve computing sums of terms over all $n_{\text{sv}} = 1737$ support vectors. By vectorizing all computations we significantly decrease computing time: for the simulation on a laptop[4] the average computing time for $u_{\text{or}}(x)$ decreases from $(1.80 \pm 0.59) \times 10^{-1}$s to $(5.84 \pm 6.42) \times 10^{-4}$s; this makes our approach suitable for implementation. The

[2]The selected hyperparameters are $G = 7, \epsilon = .1, \gamma = 30$.

[3]Note that $N = 1$ does not ensure safety in practice here.

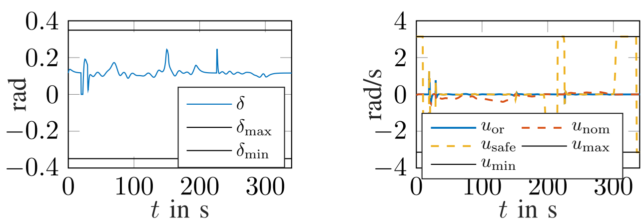
[4]2.5GHz Dual-Core Intel Core i7, 16GB memory, Intel Iris Plus Graphics



(a) Image with safe/unsafe region in black/light gray ($300m \times 300m$).

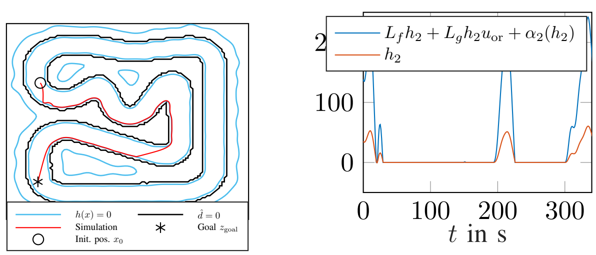
(b) EDF data $\mathcal{D}$ of $n_s = 5444$ training samples with $1m$ resol.

Fig. 2: Preprocessing for offline learning for closed track we wish to follow. From the image we generate data $\mathcal{D}$.



(a) Steering angle.

(b) Control inputs.



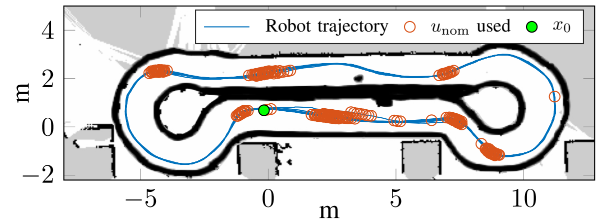
(c) 0-level sets.

(d) ICCBF.

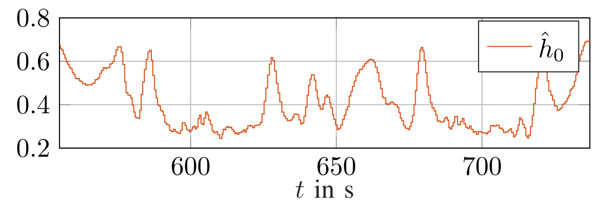
Fig. 3: Simulation results. By observing the simulated travelled path in closed loop we notice that the safety filter keeps the robot safe even though nominal control steers directly off track towards the goal z_{goal}. The resulting steering angle and rate remain within bounds. The safety filter overrides the nominal control whenever the ICCBF $h_2(x)$ approaches zero (the robot approaches the road edges), see Figures 3(b) and 3(d). The safety filter is mostly engaged, except in the three periods in which the nominal steering command drives the robot away from the edges (see Figure 3(c)). Importantly, the track itself represents a simply connected set.

complexity of the calculation of the gradients and higher-order derivatives scales linearly with the number of vectors in the SVM; even in our Python implementation computing time for necessary gradients for $n_{\text{sv}} = 1.4 \times 10^5$ support vectors is at 100Hz. We expect performance to further improve if implemented in compiled languages such as C++ (see `https://github.com/sb-git-cloud/learning_cbf.git` for vectorized code).

Experimental results: The experimental setup is as follows. A modified RC car drives on a closed track with different obstacles (see link above for photos). Its nominal control is set so as to always steer straight. We implement and deploy our learned safety filter with the goal to keep



(a) Travelled vehicle path with highlighted u_{nom} when applied.



(b) ICCBF for one lap.

Fig. 4: Experimental results. In Fig. 4(a): the learned safety filter keeps the car on the road while avoiding obstacles and allowing nominal action, $u_{\text{nom}}(x)$, (zero steering angle) if safe. Fig. 4(b) shows that $\hat{h}_0 \geq 0$ at all time, despite input constraints and uncertainties which include: measurement noise from the LiDAR sensor used for map building and localization; limited map resolution (here 5cm) measurement noise related to the car's velocity; process noise due to discretization of continuous-time model dynamics, and approximated dynamics in general (wheel slip is neglected); input disturbance for the steering angle; limited real-time capabilities of used hardware and ROS.

the car on track. The developed ROS package is run on a Jetson Nano. We generate a map in form of an occupancy grid obtained via Hector Slam [35] and add the race track. The modified occupancy grid map is then used to generate samples of the EDF. For position estimation via LiDAR we use AMCL. The experimental results are shown in Figure 4, with a video in the link above.

VI. Limitations

We acknowledge the limitation of a constant velocity; while a varying velocity could generally be accommodated (e.g., by considering acceleration as another input), it would complicate the implementation. A detailed consideration is part of ongoing research. We note that CBF-based controller implicitly assumes that (6a) always has a feasible solution, i.e., $h_N(x)$ is a valid ICCBF. This is difficult to show in practice, and it is often accommodated with the introduction of slack variables. Nevertheless safety is still improved compared with standard CBF-based methods. If safety is not feasible (due to limited steering angle/rate), the explicit control law (7) automatically selects a solution that minimizes the (safety) constraint violation. Although ϵ-SVM regression for EDFs reduces the risk of rank deficient $L_g h_N(x)$, a theoretical guarantee is generally difficult to obtain. For simulation and experiment we set the class-$\mathcal{K}$ functions $\alpha_i, i = \{0, 1, 2\}$, to be positive constants.

These constants must be tuned in case of a different vehicle model or different constraints. This is generally still an open question and interesting directions towards methodologically selecting linear multiples of class-$\mathcal{K}$ functions or extend learning are shown in [18], [36] for relative-degree one systems.

VII. Conclusion & Future Direction

We derive a robust filter that guarantees safe operation of car-like robots and present a procedure for its calculation and practical implementation. The proposed design considers practical constraints on the steering angle and the steering rate, and uses environmental data to learn the robust safety filter. Practical use has been illustrated via simulation and implementation on a modified RC car. Here the learning was performed offline, but the procedure can be adapted for online or reinforced learning, as well as dynamic environments. Due to space constraints these applications as well as a complete statistical analysis of all the errors involved in the implementation are left for future investigations.

References

[1] E. Aljalbout, J. Chen, K. Ritt, M. Ulmer, and S. Haddadin, "Learning Vision-based Reactive Policies for Obstacle Avoidance," in *Conference on Robot Learning*, pp. 2040–2054, PMLR, 2021.

[2] A. Pandey, S. Pandey, and D. Parhi, "Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.

[3] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "End-to-End Deep Reinforcement Learning for Lane Keeping Assist," *arXiv preprint arXiv:1612.04340*, 2016.

[4] G. Chou, N. Ozay, and D. Berenson, "Uncertainty-Aware Constraint Learning for Adaptive Safe Motion Planning from Demonstrations," *arXiv preprint arXiv:2011.04141*, 2020.

[5] 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)*, pp. 3420–3431, 2019.

[6] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe Multi-Agent Interaction through Robust Control Barrier Functions with Learned Uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020-12-14.

[7] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning Control Barrier Functions from Expert Demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, 2020.

[8] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, "Training Neural Network Controllers Using Control Barrier Functions in the Presence of Disturbances," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020.

[9] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni, "Learning Hybrid Control Barrier Functions from Data," in *Conference on Robot Learning*, pp. 1351–1370, PMLR, 2021.

[10] C. Li, Z. Zhang, A. Nesrin, Q. Liu, F. Liu, and M. Buss, "Instantaneous Local Control Barrier Function: An Online Learning Approach for Collision Avoidance," *arXiv preprint arXiv:2106.05341*, 2021.

[11] J. Ferlez, M. Elnaggar, Y. Shoukry, and C. Fleming, "ShieldNN: A Provably Safe NN Filter for Unsafe NN Controllers," *arXiv preprint arXiv:2006.09564*, 2020.

[12] F. S. Barbosa and J. Tumuva, "Risk-Aware Navigation on Smooth Approximations of Euclidean Distance Fields Among Dynamic Obstacles," *DiVA preprint: diva2:1626326*, 2022.

[13] M. Srinivasan, A. Dabholkar, S. D. Coogan, and P. A. Vela, "Synthesis of Control Barrier Functions Using a Supervised Machine Learning Approach," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7139–7145, 2020.

[14] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. I. Nieto, and R. Y. Siegwart, "Signed Distance Fields: A Natural Representation for Both Mapping and Planning," in *RSS 2016 Workshop: Geometry and Beyond - Representations, Physics, and Scene Understanding for Robotics*, 2016.

[15] A. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "Learning for Safety-Critical Control with Control Barrier Functions," in *Learning for Dynamics and Control*, pp. 708–717, 2020.

[16] M. Saveriano and D. Lee, "Learning Barrier Functions for Constrained Motion Planning with Dynamical Systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 112–119, 2019.

[17] W. Jin, Z. Wang, Z. Yang, and S. Mou, "Neural Certificates for Safe Control Policies," *arXiv preprint: 2006.08465*, 2020.

[18] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning Safe Multi-Agent Control with Decentralized Neural Barrier Certificates," *arXiv preprint: 2101.05436*, 2021.

[19] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe Nonlinear Control Using Robust Neural Lyapunov-Barrier Functions," in *5th Annual Conference on Robot Learning* , 2021.

[20] C. Dawson, S. Gao, and C. Fan, "Safe Control with Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction methods," *arXiv preprint: 2202.11762*, 2022.

[21] K. Long, C. Qian, J. Cortés, and N. Atanasov, "Learning Barrier Functions With Memory for Robust Safe Navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4931–4938, 2021.

[22] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness Guarantees for the Composition of Lane Keeping and Adaptive Cruise Control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018.

[23] M. Cavorsi, M. Khajenejad, R. Niu, Q. Shen, and S. Z. Yong, "Tractable Compositions of Discrete-Time Control Barrier Functions with Application to Lane Keeping and Obstacle Avoidance," *arXiv preprint: 2004.01858*, 2020.

[24] S. Brüggemann, D. Steeves, and M. Krstic, "Simultaneous Lane-Keeping and Obstacle Avoidance by Combining Model Predictive Control and Control Barrier Functions," *arXiv preprint: 2204.06136*, 2022.

[25] S. Kolathaya and A. D. Ames, "Input-to-State Safety With Control Barrier Functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019.

[26] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[27] M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using Control Lyapunov–Barrier Function," *Automatica*, vol. 66, pp. 39–47, 2016.

[28] D. Agrawal and D. Panagou, "Safe Control Synthesis via Input Constrained Control Barrier Functions," *arXiv preprint arXiv:2104.01704*, 2021.

[29] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 2002.

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

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

[32] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.

[33] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[34] B. Hammer and K. Gersmann, "A Note on the Universal Approximation Capability of Support Vector Machines," *Neural Processing Letters*, vol. 17, no. 1, pp. 43–53, 2003.

[35] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A Flexible and Scalable SLAM System with Full 3D Motion Estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, IEEE, November 2011.

[36] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-Critical Control using Optimal-decay Control Barrier Function with Guaranteed Point-wise Feasibility," in *2021 American Control Conference (ACC)*, pp. 3856–3863, 2021.