Data-Driven Safety Filters

HAMILTON-JACOBI REACHABILITY, CONTROL BARRIER FUNCTIONS, AND PREDICTIVE METHODS FOR UNCERTAIN SYSTEMS

JASON J. CHOI

KIM P. WABERSICH, ANDREW J. TAYLOR, JASON J. CHOI, KOUSHIL SREENATH, CLAIRE J. TOMLIN, AARON D. AMES, and MELANIE N. ZEILINGER

Today's control engineering problems exhibit an unprecedented complexity, with examples including the reliable integration of renewable energy sources into power grids [1], safe collaboration between humans and robotic systems [2], and dependable control of medical devices [3] offering personalized treatment [4]. In addition to compliance with safety criteria, the corresponding control objective is often multifaceted. It ranges from relatively simple stabilization tasks to unknown objective functions, which are, for example, accessible only through demonstrations from interactions between robots and humans [5]. Classical control engineering methods are, however, often based on stability criteria with respect to set points and reference trajectories, and they can therefore be challenging to apply in such unstructured tasks with potentially conflicting safety specifications [6, Secs. 3 and 6]. While numerous efforts have started to address these challenges, missing safety certificates often still prohibit the widespread application of innovative designs outside research environments. As described in "Summary," this article presents safety filters

Digital Object Identifier 10.1109/MCS.2023.3291885
Date of current version: 18 September 2023

1066-033X/23©2023IEEE

and advanced data-driven enhancements as a flexible framework for overcoming these limitations by ensuring that safety requirements codified as static state constraints are satisfied under all physical limitations of the system.

To illustrate the fundamental challenges in guaranteeing safety in the form of state constraints for dynamic systems, consider a vehicle driving on a road, as depicted in Figure 1. The vehicle specifies its control action based on its current state, including current position, current velocity, and relative heading to the road. The difficulty arising in safety-critical dynamical systems is that unsafe control actions do not instantly cause a violation of state constraints defining safety requirements but rather can cause a system to evolve into states from which it cannot avoid violating safety in the future. For example, if the steering angle does not correspond to the road's curvature for a fraction of a second, the vehicle does not immediately leave the track, but it may evolve into states from which it is unavoidable that the car goes off track, as depicted by the red trajectory in Figure 1. Safety filters detect such unsafe control inputs that may lead to constraint violations in the future and *minimally* modify them to ensure safety, as illustrated by the green trajectory in Figure 1.

Summary

Some of the most challenging problems in control typically consist of minimizing an objective function under safety constraints and physical limitations. These often conflicting requirements render classical stabilization-based control design tricky, and even modern learning-based alternatives rarely provide strict safety guarantees "out of the box." Safety filters address this limitation through a modular approach to safety. The first part of this article formalizes an ideal safety filter to enhance any controller with safety guarantees and provides a tutorial-style exposition of invariance-based methods using Hamilton-Jacobi reachability, control barrier functions, and predictive control-related techniques. While the first part assumes perfect knowledge of the system dynamics, the second part bridges the gap toward real-world applications through data-driven model corrections. To this end, deterministic, robust, and probabilistic model learning techniques are outlined, and a selection of mini-tutorials for learning-based safety filters is provided. The article concludes with recent applications to demonstrate the capability of various safety filter formulations when combined with stabilizing controllers, learning-based controllers, and even humans.

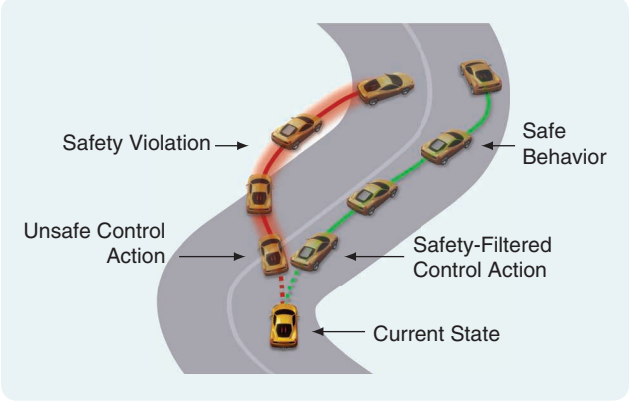


FIGURE 1 An intuitive illustration of safety problems in control using a vehicle. The application of an unsafe control input can result in a safety constraint violation at some point in the future. This is depicted by the red trajectory, where the vehicle ends up leaving the track. The goal of this article is to present safety filters that detect and minimally modify such unsafe inputs to ensure safety at all times.

In this work, we discuss three research directions that have evolved over the past two decades to tackle such safety-critical control problems: reachability-based methods [7], [8], control barrier functions (CBFs) [28], [38], and predictive control techniques [11], [12]. These methods are unified by the common concept of set invariance [13] to ensure that a system must remain within a desired set for the entirety of the system's evolution. Although the three methodologies all address the same fundamental problem of ensuring set invariance, they have developed relatively independently with notable technical differences. Reachability analysis is based on a set-based propagation of all possible system trajectories determined by the system inputs and disturbances. In contrast, CBFs rely on Lyapunov theory to determine inputs to a system that ensure set invariance. Lastly, predictive safety filters (PSFs) are based on a receding-horizon open-loop optimal control problem, which is guaranteed to be solvable and ensures constraint satisfaction at every control sampling time step. In recent years, however, joint research efforts have demonstrated tremendous potential by combining the core competencies of each methodology, enabling high-performance safety-critical applications and promising perspectives for future research [14], [15], [17], [18], [19].

Despite the differences and connections between the methodologies, all the methods rely on a mathematical model that describes the evolution of the dynamic system to ensure safety at all times. The derivation, identification, and verification of these high-fidelity system models are among the most time-consuming tasks in the design phase of safety-critical controllers [20]. To reduce this effort, the increasing availability of low-cost sensing and connectivity capabilities and growing computational resources have triggered research efforts across all three methodologies toward the use of data-driven models [21], [22], [23].

This article provides a comprehensive introduction to the previously described aspects of recent research on safety filters. We present an idealized safety filter problem and demonstrate the capabilities of safety filters based on

Although the three methodologies all address the same fundamental problem of ensuring set invariance, they have developed relatively independently with notable technical differences.

Hamilton-Jacobi (HJ) reachability, CBFs, and predictive control to provide an approximate solution. Once the basic principles are in place, more recently discovered interconnections between the methods are presented to open new perspectives for future research and applications. It is then shown how to enhance the core concepts through data-driven models and how robust and probabilistic uncertainty bounds can be incorporated to ensure safety with high confidence. While we present a selection of successful techniques and state-of-the-art applications, this direction represents a promising dimension worth investigating in the future.

Lastly, we wish to note that while we present a variety of safety filter techniques, it is by no means an exhaustive list of all the methods that have been developed for filtering inputs to a system in an effort to achieve safety. Notable methods that we believe are related to the methods presented in this article include reference and command governors [24] and nonovershooting control [25], [26]. We refer readers seeking to augment their knowledge of safety filters beyond our article to these works.

OUTLINE OF THE ARTICLE

We begin by stating the class of nonlinear dynamical systems considered in this article and specify constraints on the states and inputs that commonly arise in safety-critical applications. Based on this system description, we formalize the desired safety filter module as an optimal control problem. Using this problem formulation, we introduce the fundamental concept of set invariance, followed by techniques for designing and implementing safety filters via HJ reachability, CBFs, and predictive control methods. The similarities and differences between these three methods are highlighted through a simple illustrative example in "Safety Filter Design Example," and a discussion on recent research efforts integrating aspects of these three methods is provided. In the second part of the article, we consider the challenge of safety-critical control in the context of uncertain nonlinear systems. We explore how the preceding methods for safety filter design can be modified to incorporate data-driven components, discuss challenges in working with real-world data in "Learning with Real-World Data," consider a popular data-driven model in "Probabilistic Nonparametric Model: Gaussian Process Regression," and highlight examples of state-of-the-art data-driven safety filter applications. We conclude with a discussion on open research directions in the area of safety filters and their data-driven extensions.

DEFINITIONS AND NOTATION

The natural, real, nonnegative real, and positive real numbers are denoted as $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0} = [0, \infty)$, and $\mathbb{R}_{>0} = (0, \infty)$, respectively. The identity matrix of dimension n is denoted as I_n. Given a set $\mathcal{A} \subseteq \mathbb{R}^n$, we denote its interior as $\text{int}(\mathcal{A})$, its boundary by $\partial\mathcal{A}$, and its complement as $\mathcal{A}^c = \mathbb{R}^n \setminus \mathcal{A}$. The signed-distance function for the set $\mathcal{A}$, $s_{\mathcal{A}} : \mathbb{R}^n \to \mathbb{R}$ is defined as $s_{\mathcal{A}}(x) = \inf_{y \in \mathcal{A}} \| y - x \|$ if $x \in \mathbb{R}^n \setminus \mathcal{A}$ and $s_{\mathcal{A}}(x) = -\inf_{y \in \mathbb{R}^n \setminus \mathcal{A}} \| x - y \|$ for $x \in \mathcal{A}$ and a vector norm $\| \cdot \|$. Given two sets $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathbb{R}^n$, the Minkowski sum of $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as $\mathcal{A}_1 \oplus \mathcal{A}_2 = \{a_1 + a_2 | a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2\}$. Given two sets $\mathcal{A}$ and $\mathcal{B}$, we denote the space of continuous functions, piecewise-continuous functions, and continuously differentiable functions mapping $\mathcal{A}$ to $\mathcal{B}$ by $\mathcal{C}(\mathcal{A}, \mathcal{B})$, $\mathcal{PC}(\mathcal{A}, \mathcal{B})$, and $\mathcal{C}^1(\mathcal{A}, \mathcal{B})$, respectively. A continuous function $\alpha \in \mathcal{C}([0, a), \mathbb{R})$ for some $a > 0$ is said to be class $\mathcal{K}$ ($\alpha \in \mathcal{K}$) if it is strictly increasing and $\alpha(0) = 0$ and is said to be extended class $\mathcal{K}$ ($\alpha \in \mathcal{K}^e$) if it is a class $\mathcal{K}$ function defined on $(-a, b)$ with $a, b > 0$. More details on class $\mathcal{K}$ functions and extended class $\mathcal{K}$ functions can be found in [27] and [28], respectively.

THE SAFETY FILTER PROBLEM WITH KNOWN SYSTEM DYNAMICS

This article considers the construction of safety-filtering mechanisms for nonlinear control systems, which can be described by the differential equation

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0} \tag{1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the system state, and $u(t) \in \mathbb{R}^{n_u}$ is the control input at time $t \in \mathbb{R}_{\geq 0}$. For simplicity, we assume that the function f is continuously differentiable, that is, $f \in \mathcal{C}^1(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_x})$, and that for any initial condition $x_0 \triangleq x(0) \in \mathbb{R}^{n_x}$ and piecewise-continuous control input signal $u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_u})$, there exists a unique solution $x(\cdot) \in \mathcal{C}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_x})$, to (1) for all $t \in \mathbb{R}_{\geq 0}$. While we mainly focus on continuous-time systems of the form (1), the majority of concepts introduced in the following possess analogs for discrete-time systems, which will be pointed out through references to corresponding literature.

Safety for the system (1) is encoded via a state constraint set $\mathcal{X} \subset \mathbb{R}^{n_x}$ and an input constraint set $\mathcal{U} \subset \mathbb{R}^{n_u}$ that must be respected during the evolution of the system, that is,

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U} \text{ for all } t \in \mathbb{R}_{\geq 0}. \tag{2}$$

This article is specifically concerned with ensuring that this safety requirement is met when the system is presented with a piecewise-continuous desired control input signal, $u_{\text{des}}(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_u})$, which does not necessarily enforce the safety requirement (2). Such desired input signals often come from stabilizing controllers hand-designed by domain specialists, learning-based controllers that maximize a particular reward signal, or human input to the system.

A safety filter $\kappa_F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{U}$ (see Figure 2) modifies this desired control input signal to produce an input signal $u(t) = \kappa_F(x(t), u_{\text{des}}(t))$ that ensures the system respects the safety constraint (2) while minimally modifying the desired input signal, that is, minimizing

$$\int_{t=0}^{\infty} \| \kappa_F(x(t), u_{\text{des}}(t)) - u_{\text{des}}(t) \| \, \mathrm{d}t \tag{3}$$

with the goal of preserving as much of the desirable behavior achieved by $u_{\text{des}}(\cdot)$ as possible. Throughout this article, we assume continuous access to direct measurements of the system state and neglect the problem of discrete-time state estimation. Thus, for any initial condition $x_0 \in \mathcal{X}$, an ideal safety filter would return a piecewise-continuous input signal $u(\cdot)$ that solves the optimization problem

$$u(\cdot) = \underset{v(\cdot)}{\operatorname{argmin}} \int_{t=0}^{\infty} \| v(t) - u_{\text{des}}(t) \| \, \mathrm{d}t \tag{4a}$$

$$\text{Subject to } v(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \tag{4b}$$

$$x(0) = x_0, \tag{4c}$$

$$\text{for all } \quad t \in \mathbb{R}_{\geq 0}:$$

$$\dot{x}(t) = f(x(t), v(t)) \tag{4d}$$

$$x(t) \in \mathcal{X}. \tag{4e}$$

While (4) characterizes an ideal safety filter, it is rarely possible to tractably implement for the following reasons:

- The desired input signal $u_{\text{des}}(\cdot)$ is typically not known a priori and can be accessed only at the current time t during closed-loop operation. Examples include when $u_{\text{des}}(\cdot)$ is specified by a feedback controller, when learning-based control applications with random inputs are applied during exploration, and when applications with humans in the loop are providing the desired control inputs, including the teleoperation of robots, driver assist systems, and piloted flight control.
- The optimization problem (4) is not necessarily feasible for each initial condition $x_0 \in \mathcal{X}$. Thus, initial conditions will need to be restricted to a subset $\mathcal{S} \subseteq \mathcal{X}$ of the state constraint set for which (4) is known to be feasible, and the evolution of the system must be constrained to remain in the set $\mathcal{S}$.
- Even if there exists a signal $v(\cdot)$ satisfying the constraints (4b)–(4e), for a given $u_{\text{des}}(\cdot)$, such a signal $v(\cdot)$ may not return a finite value for the cost (4a), rendering the optimization problem "ill defined". Resolving this issue for arbitrary desired input signals will often require considering the cost (4a) over a finite horizon.

We will next tackle these challenges through permissive approximations of the ideal safety filter formulation (4).

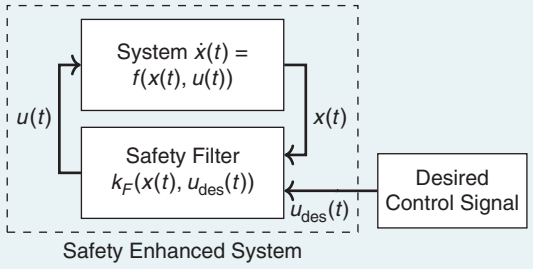


FIGURE 2 An illustration of the safety filter concept. A desired control input $u_{\text{des}}(t)$ is processed by the safety filter to produce a control input signal $u(t) = \kappa_F(x(t), u_{\text{des}}(t))$ that is applied to the system to ensure that safety is maintained at all times.

TABLE 1 An overview of safety filter literature. This table presents references regarding the historical development, core results, and recent data-driven research for each of the three safety filter methodologies presented in this work. While it is not a complete description of all related work on these methodologies, this collection of works serves to highlight the strengths of each approach and is a natural starting point for forming a deeper technical understanding of the results presented in this work.

	HJ Reachability	CBFs	Predictive Filters
Historical development	[29], [30], [31], [32], [33], [34], [35], [36]	[9], [28], [37], [38]	[11], [39], [40]
Core results	[7], [8], [41], [42]	[10], [28], [43], [44], [45], [91]	[12], [46], [47], [48]
Data-driven safety filters	[21], [49], [50]	[23], [51], [52], [53], [54], [55], [56], [57], [58], [59]	[60], [61], [62], [63], [64], [65]

SAFETY FILTER METHODOLOGIES

In this section, we review four approaches for approximating the idealized safety filter defined in (4), see Table 1 for an overview of corresponding literature. We begin by reviewing the fundamental notion of set invariance,

Set invariance is a well-established notion for studying whether the state of a dynamic system is contained in a prescribed set for all time and is thereby instrumental in synthesizing safety filters.

which underlies all of the presented approaches. The first approach we present builds upon the foundational result of Nagumo's theorem to build a switching safety filter. The conservative nature of this approach is then improved by constructing invariant sets using HJ reachability, which is the method of solving reachability problems with optimal control theory based on HJ equations [33]. We next review CBFs that rely on a Lyapunov-like derivative condition to smoothly enforce the safety of a system. Lastly, we review recent advances in PSFs, which utilize a receding-horizon optimal control problem to effectively balance safety with using the desired control input. We outline the strengths and weaknesses of each method and apply them to a simple example problem for comparison in "Safety Filter Design Example." We conclude this section by relating each method back to the ideal safety filter in (4) and highlighting recent research focused on combining the aforementioned techniques in an effort to overcome the limitations facing each method.

Set Invariance

Set invariance [13] is a well-established notion for studying whether the state of a dynamic system is contained in a prescribed set for all time and is thereby instrumental in synthesizing safety filters. While the following concepts of set invariance and controlled set invariance are defined for continuous-time systems of the form (1), they similarly exist for discrete-time systems [66]. Given a feedback controller $\kappa : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$, we may construct a closed-loop system

$$\dot{x}(t) = f(x(t), \kappa(x(t), u_{\text{des}}(t))) \quad t \in \mathbb{R}_{\geq 0} \tag{5}$$

allowing the following definition.

Definition 1 (Set Invariance)

A set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is said to be *(forward) invariant* for the system (5) if for any initial condition $x_0 \in \mathcal{S}$, we have that $x(t) \in \mathcal{S}$ for all $t \in \mathbb{R}_{\geq 0}$.

If a set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is forward invariant for the system (5) and satisfies $\mathcal{S} \subseteq \mathcal{X}$, then we may conclude that for any initial condition $x_0 \in \mathcal{S}$, we have $x(t) \in \mathcal{X}$ for all $t \in \mathbb{R}_{\geq 0}$. Thus, satisfying the state-related part of the safety constraint (2) can be achieved by constructing a controller κ and a corresponding forward invariant set $\mathcal{S}$ contained in the state constraint set $\mathcal{X}$. We note that this construction via invariance requires not only a stronger condition on the initial condition x_0, in that it must lie in $\mathcal{S}$ rather than just $\mathcal{X}$, but it also yields a stronger statement since $x(t) \in \mathcal{S}$ for all $t \in \mathbb{R}_{\geq 0}$ rather than just $x(t) \in \mathcal{X}$. Thus, the particular construction of the feedback controller κ and the forward invariant set $\mathcal{S}$ impacts the resulting performance of the system because the use of a conservative set $\mathcal{S}$ may unnecessarily limit the behavior of the system.

The notion of a control invariant set captures the possibility of controlling the open-loop system (1) in a safe manner, without being confined to using a predefined feedback controller κ and then determining a forward invariant set $\mathcal{S}$ for the closed-loop system under κ.

Definition 2 (Controlled Set Invariance)

A set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is said to be *control invariant* for the system (1) if for any initial condition $x_0 \in \mathcal{S}$, there exists a piecewise-continuous input signal $u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_u})$ such that we have $x(t) \in \mathcal{S}$ for all $t \in \mathbb{R}_{\geq 0}$. If $\mathcal{S} \subseteq \mathcal{X}$ contains all initial conditions $x_0 \in \mathcal{X}$ such that there exists a piecewise continuous input signal $u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_u})$ yielding $x(t) \in \mathcal{X}$ for all $t \in \mathbb{R}_{\geq 0}$, we say that $\mathcal{S}$ is the *maximal control invariant set* in $\mathcal{X}$ for the system (1).

This definition enables various safety filter design techniques given a control invariant set $\mathcal{S}$. Conversely, since finding a control invariant set $\mathcal{S}$ is not restricted to any specific controller, choosing $\mathcal{S}$ can also be done in a more constructive manner. The performance achieved using a safety filter will directly depend on the size of the control invariant set. Ideally, the set $\mathcal{X}$ would be used; however, this is typically not a control invariant set for (1) and merely represents the design goal. We will see in the following sections that the available safety filter techniques produce different feedback controllers and (control) invariant sets that permit varying degrees of performance.

Nagumo's Theorem and Switching Safety Filters

The first safety filter design that we consider is that of a switching safety filter. Although this design is relatively simple and often overly conservative, it highlights key elements that arise in the three advanced safety filter approaches presented next.

Consider a feedback controller

$$\kappa_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u} \tag{6}$$

Safety Filter Design Example

This sidebar illustrates and compares the basic safety filter methodologies by applying each of them to the inverted pendulum system (Figure S1)

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\theta\\ \dot\theta\end{bmatrix}}_{\dot x}=\underbrace{\begin{bmatrix}\dot\theta\\ \frac{g}{\ell}\sin\theta\end{bmatrix}}_{f(x)}+\underbrace{\begin{bmatrix}0\\ \frac{1}{m\ell^2}\end{bmatrix}}_{g(x)}u \tag{S1}$$

where the pendulum angle and angular velocity $[\theta,\dot\theta]=[x_1,x_2]=x$ define the system state, and u is the input torque applied at the base of the pendulum. The system parameters consist of the mass m = 2 kg, length $\ell=1$ m, and gravitational acceleration g = 10 m/s^2. The physical input limitation is a maximum applicable torque of 3 N · m, that is, $\mathcal{U}=\{u\in\mathbb{R}\,|\,|u|\le 3\}$. The safety constraints are defined as pendulum angle and angular velocity constraints of the form $\mathcal{X}=\{x\in\mathbb{R}^2\,|\,|x_1|\le 0.3,|x_2|\le 0.6\}$.

DESIRED CONTROL INPUT SIGNAL

To compare the different safety filter designs with respect to the "ideal" safety filter objective (4), we use the desired control signal

$$u_{\mathrm{des}}(t)=\begin{cases}3, & t\in[0,2),\\ -3, & t\in[2,4),\\ 3, & t\in[4,6),\\ m\ell^2\left(-\frac{g}{\ell}\sin x_1-[1.5,\ 1.5]x\right), & \text{else.}\end{cases} \tag{S2}$$

By alternating between maximum and minimum torque, the desired input signal (S2) tries to violate the system constraints, requiring safety filter intervention. The adversarial input section is followed by a stabilizing feedback control law of the form $u_{\mathrm{des}}(t)=K_{\mathrm{des}}x(t)$, which does not consider constraint satisfaction explicitly.

SWITCHING SAFETY FILTER

This section demonstrates how to construct the switching safety filter (11) using a linear-quadratic regulator (LQR) of the form $\kappa_{\mathcal{S}}(x)=-Kx$. The design of $\kappa_{\mathcal{S}}$ is based on the linearization of the system dynamics (S1) around the upward equilibrium point

$$\Delta\dot x=\underbrace{\begin{bmatrix}0 & 1\\ \frac{g}{\ell} & 0\end{bmatrix}}_{A}\Delta x+\underbrace{\begin{bmatrix}0\\ \frac{1}{m\ell^2}\end{bmatrix}}_{B}\Delta u. \tag{S3}$$

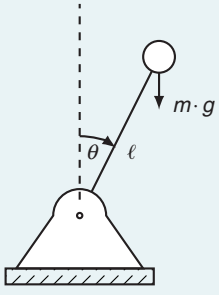


FIGURE S1 An inverted pendulum control system.

Using the state cost $Q=25I_2$ and input cost $R=1$, we obtain the gain $K=[40.62,\ 13.69]$. An invariant set for (S3) is selected as a sublevel set of the LQR Lyapunov function [88, Ch. 4]

$$\mathcal{S}_\gamma=\{x\in\mathbb{R}^2\,|\,\gamma-x^\top Px\ge 0\} \tag{S4}$$

for some $\gamma>0$ and the positive-definite matrix

$$P=\begin{bmatrix}282.26 & 81.23\\ 81.23 & 27.38\end{bmatrix}. \tag{S5}$$

To ensure that $\mathcal{S}_\gamma$ is forward invariant under $\kappa_{\mathcal{S}}$ in the presence of the input constraints $\mathcal{U}$, the level set γ must be selected such that $\kappa_{\mathcal{S}}(x)\in\mathcal{U}\Leftrightarrow|-Kx|\le 3$ for all $x\in\mathcal{S}_\gamma$ and $\mathcal{S}_\gamma\subseteq\mathcal{X}$. Using the support function of $\mathcal{S}_\gamma$ [13], we obtain a maximal value of $\gamma=1.31$, for which we denote the safe set $\mathcal{S}\triangleq\mathcal{S}_{1.31}$. To certify the forward invariance of $\mathcal{S}$ with respect to the nonlinear system (S1), we verify that

$$\max_{x\in\mathcal{S}}\ -2x^\top P(f(x)-g(x)Kx)\ge 0 \tag{S6}$$

through nonlinear programming [39]. The resulting safe set is depicted in Figure S2(a). This construction allows us to implement the switching safety filter in (11) as

$$\kappa_F(x,u_{\mathrm{des}}(t))=\begin{cases}-Kx, & x\in\partial\mathcal{S}\ \text{or}\ |u_{\mathrm{des}}(t)|>3\\ u_{\mathrm{des}}(t), & \text{else.}\end{cases} \tag{S7}$$

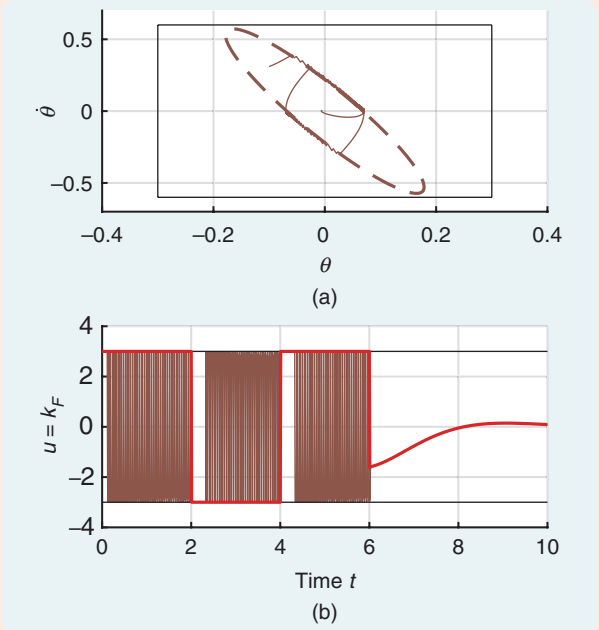


FIGURE S2 The application of the switching safety filter (S7) to the inverted pendulum example (S1). (a) The desired safety constraints $\mathcal{X}$ (solid black line), the switching safety filter safe set (dashed brown line), and the closed-loop trajectory (solid brown line). (b) The applied input trajectory (brown) and desired input signal (red). The constraints are indicated with solid black lines in each plot. While safety is maintained, the switching safety filter (S7) causes undesirable input chattering, and the safe set covers only a small portion of $\mathcal{X}$.

The safety controller is used for 0.01 s when it is activated. The closed-loop simulation of the resulting control structure, as depicted in Figure 2, together with the desired input signal (S2), is shown in Figure S2.

After significant intervention during the first 6 s, the desired control input signal meets safety requirements and input bounds (for $t \in [6, 10]$) and is used. Even though safety is achieved during the entire evolution of the system, limitations that motivate the advanced techniques presented in this article may be observed. The derived safe set $\mathcal{S}$ and safe controller $\kappa_{\mathcal{S}}$ yield a conservative safety filter, which can be seen by the overly large safety margin between the safe set and the angular constraints in Figure S2(a). To reduce such conservativeness, Hamilton-Jacobi (HJ) reachability and predictive safety filters (PSFs) integrate optimal control-based approaches as demonstrated in the upcoming sections. Furthermore, the switching-based safety control law (S7), derived from (11), can result in significant input chattering behavior near the boundary of the safe set, as seen in Figure S2(b). Such behavior is not desirable in practice. To this end, control barrier functions (CBFs) enable a safety filter formulation that yields a smooth control input signal.

HJ REACHABILITY SAFETY FILTER

This section demonstrates how HJ reachability allows one to reduce the conservativeness of the switching safety filter (S7). The value function V defined in (14), which describes the maximal control invariant set in $\mathcal{X}$, is computed by solving the HJ partial differentiable equation numerically using a sufficiently large finite-time horizon T with the HJ optimal control toolbox (helperOC) [8] and the level set toolbox [74]. A 101 × 201 grid is constructed on the set $\mathcal{X}$, and a finite horizon T = 2.5 s is used for this computation, which takes roughly a minute on a standard laptop. The safe set (15) resulting from Theorem 2 is

$$\mathcal{S} = \{x \in \mathbb{R}^2 \mid V(x) \geq \varepsilon\} \tag{S8}$$

with $\varepsilon = 0.02$ to account for numerical approximation errors. See Figure S3(a) for an illustration of $\mathcal{S}$, which represents an approximation of the maximal control invariant set in $\mathcal{X}$ based on Theorem 2. The HJ safety filter is implemented as in (19) and shows a larger safe set than the switching safety filter, leading to fewer interventions (and correspondingly less chatter in the input signal) when $t \in [0, 6]$, as seen in Figure S3.

CONTROL BARRIER FUNCTION SAFETY FILTER

With the goal of reducing the undesirable input chattering of the previous techniques seen in Figures S2 and S3, we next construct a safety filter using control barrier functions (CBFs). To this end, we follow the example presented in [98] and select

$$h_{\mathcal{S}}(x) = 1 - x^{\top} \begin{pmatrix} 1/a^2 & 0.5/ab \\ 0.5/ab & 1/b^2 \end{pmatrix} x \tag{S9}$$

with parameters $a, b > 0$ as a candidate CBF, yielding a zero-superlevel set

$$\mathcal{S} = \{x \in \mathbb{R}^{n_x} \mid h_{\mathcal{S}}(x) \geq 0\} \tag{S10}$$

describing the safe set, similarly to (S4).

The quantities a and b must be selected to ensure that $h_{\mathcal{S}}$ satisfies the condition (24) for some $\alpha \in \mathcal{K}^e$. We consider a function $\alpha \in \mathcal{K}^e$ of the form $\alpha(r) = c_\alpha r$, with $c_\alpha > 0$ to be determined. The CBF supremum condition (24) can be equivalently (modulo input constraints) expressed as the implication [98]

$$\nabla h_{\mathcal{S}}(x) g(x) = 0 \Rightarrow \nabla h_{\mathcal{S}}(x) f(x) + \alpha(h_{\mathcal{S}}(x)) > 0 \tag{S11}$$

which, in the inverted pendulum setting, appears as

$$\nabla h_{\mathcal{S}}(\bar{x}) g(\bar{x}) = 0 \Rightarrow \dot{\bar{x}}_2 = -\frac{b}{2a} \bar{x}_1 \tag{S12}$$

for $\bar{x} \in \mathbb{R}^2$. For an $\bar{x}$ such that $\nabla h_{\mathcal{S}}(\bar{x}) g(\bar{x}) = 0$,

$$\nabla h_{\mathcal{S}}(\bar{x}) f(\bar{x}) + \alpha(h_{\mathcal{S}}(\bar{x})) = c_\alpha + \frac{3}{4a^2} \left(\frac{b}{a} - c_\alpha \right) \bar{x}_1^2. \tag{S13}$$

We see that the required implication is satisfied by choosing $c_\alpha \leq \frac{b}{a}$. We select the values a = 0.137 and b = 0.274 considering the state and input constraints $\mathcal{X}$ and $\mathcal{U}$, respectively, and select $c_\alpha = 0.2$. The resulting safe set is visualized in Figure S4(a). The safety filter can be implemented in simulation

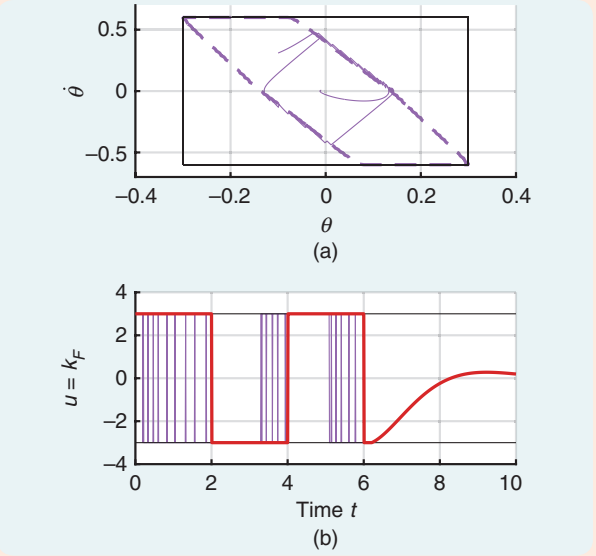


FIGURE S3 The application of the HJ-based safety filter to the inverted pendulum example (S1). (a) The HJ-based safe set (dashed purple line) and closed-loop trajectory (solid purple line). (b) The applied input trajectory (purple) and desired input signal (red). The safety filter is significantly less intrusive compared to the switching safety filter due to the larger safe set $\mathcal{S}$, and displays notably less chattering in the control input signal. The chattering can be resolved further by replacing the simple switching mechanism in (19) with formulations that induce smooth transitions [15].

(Continued)

Safety Filter Design Example *(Continued)*

by solving (26) numerically using the standard YALMIP solver [S1]. We note that the system is kept safe, and the chattering in the control input signal is eliminated (with jumps occurring only at discontinuities in the desired control input signal), as seen in Figure S4. We note that the safe set obtained using this approach is notably smaller than the one used with the HJ reachability safety filter. Developing constructive approaches for synthesizing less conservative CBFs is a topic of ongoing research.

PREDICTIVE SAFETY FILTER

We next implement a PSF that uses a receding-horizon approach to enable smooth filtering of control inputs while maintaining a large control invariant set. The first step to construct a PSF as in (28) is taking the Euler time discretization of the dynamics (27). Using a discretization time of $\Delta T = 0.05$ yields the discrete dynamics

$$x(k+1) = x(k) + 0.5\,(f(x(k)) + g(x(k)))\,u(k). \tag{S14}$$

We next construct a terminal invariant set $\mathcal{S}^{\text{trm}} \subset \mathcal{X}$. The application of the linearization-based approach described in [113, Sec. 4] at the origin yields a terminal invariant set

$$\mathcal{S}^{\text{trm}} = \{x \in \mathbb{R}^{n_x} \mid 1 - x^\top P^{\text{trm}} x \geq 0\} \tag{S15}$$

with

$$P^{\text{trm}} = \begin{bmatrix} 128.10 & 41.13 \\ 41.13 & 15.98 \end{bmatrix}. \tag{S16}$$

We implement (28) with a planning horizon of $N = 20$ using IPOPT [S2] and Casadi [S3]. While solve time is not critical in simulation, real-world applications may require tailored algorithms and software packages (see, for example, [40, Sec. 12] and references therein). Figure S5 illustrates the resulting safe set and closed-loop trajectories. While the PSF both is permissive and smoothly filters the desired control input signal, the required online computations increase by multiple orders of magnitude. The computational load can be balanced by reducing the planning horizon, which, however, also reduces the corresponding implicit safe set, as seen in Figure S5(b).

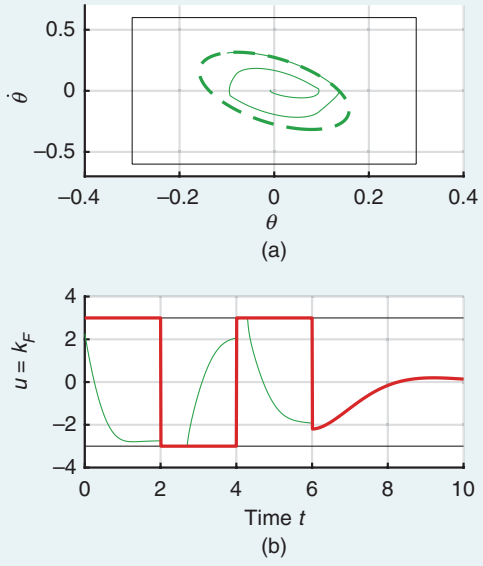


FIGURE S4 The application of the CBF-based safety filter to the inverted pendulum example (S1). (a) The CBF-based safe set (dashed green line) and closed-loop trajectory (solid green line). (b) The applied input trajectory (green) and desired input signal (red). The safety filter smoothly modifies the desired control input signal while ensuring that the system remains safe, though the safe set is smaller than the HJ approach.

CONCEPTUAL IDEAL SAFETY FILTER

To compare the quality of the previous safety filter formulations to the ideal safety filter in (4), we solve (4) approximately using a modified version of the PSF described previously. The

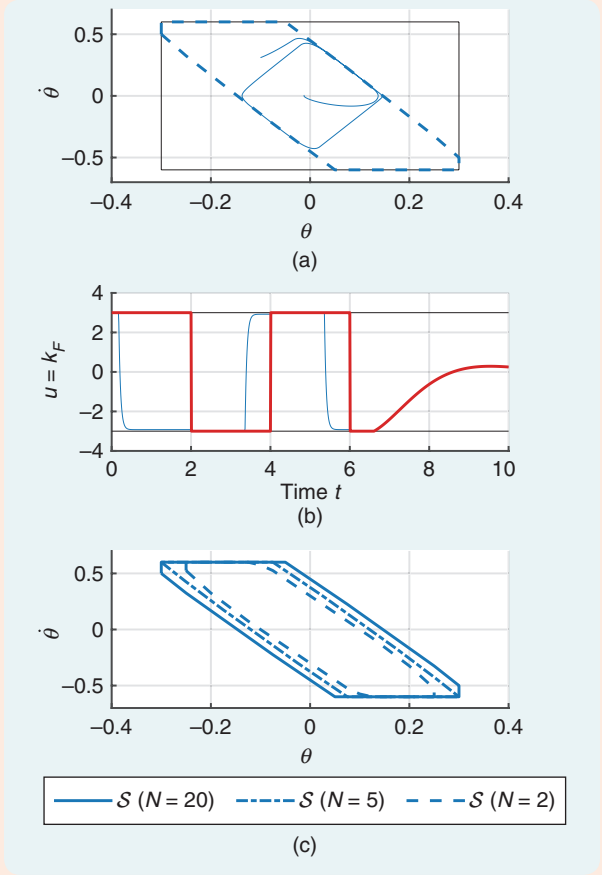


FIGURE S5 The application of the PSF to the inverted pendulum example (S1). (a) The implicit predictive safe set with a horizon length of $N = 20$ (dashed blue line) and closed-loop trajectory (solid blue line). (b) The applied input trajectory (blue) and desired input signal (red). The safety filter anticipates jumps in the desired control input signal and changes the input preemptively, yielding smooth behavior. (c) The implicit safe set $\mathcal{S}_N^{\text{PSF}}$ (29) of the PSF (28) for different planning horizons N. Longer horizon lengths increase the size of $\mathcal{S}_N^{\text{PSF}}$ until it converges to the maximal control invariant set in $\mathcal{X}$.

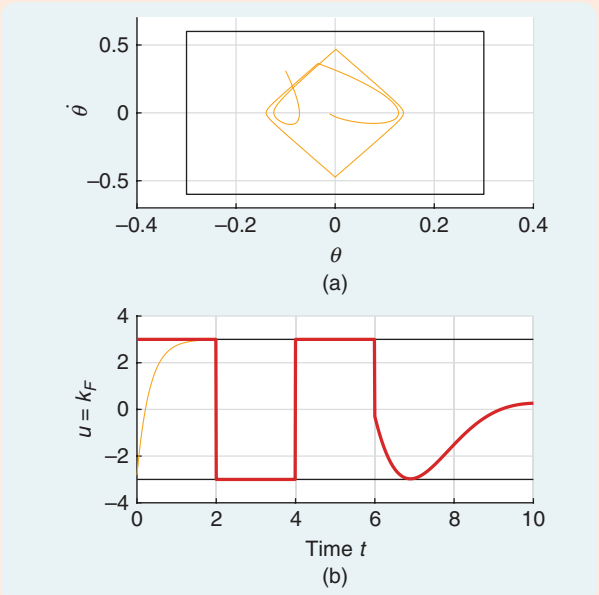


FIGURE S6 The application of the ideal safety filter (4) to the inverted pendulum example (S1). (a) The closed-loop trajectory using the ideal safety filter (solid orange line). (b) The applied input trajectory (orange) and desired input signal (red). The ideal safety filter has access to the typically unknown future desired inputs and provides an optimal filtering behavior by overriding desired inputs only during the first 2 s, after which it directly uses the desired input signal.

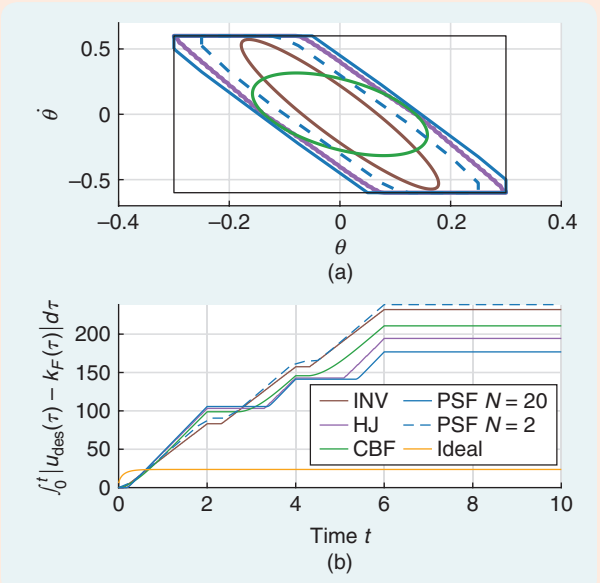


FIGURE S7 A comparison of various safety filter methods. (a) The safe sets associated with the switching, HJ reachability, and CBF- and PSF-based safety filters, using the color codes in Figures S2–S5. (b) The value of the safety filter objective (4a) using the switching, HJ reachability, CBF, PSF, and ideal safety filters. Lower values of quantity indicate that the safety filter permits more use of the desired control input signal. We observe that the PSF-based safety filter with a horizon of $N = 20$ achieves the best performance, while the PSF-based safety filter with a horizon of $N = 2$ achieves the worst performance. We also see that the ideal safety filter vastly outperforms the presented methodologies, which perform relatively similarly compared to the ideal safety filter.

planning horizon is increased to cover the entire task horizon ($N = 200$), and the cost functional (28a) is changed to include the desired control inputs on the whole task horizon, that is, $\Sigma_{i=0}^{120} \| u_{\text{des}}(i\Delta T) - u_{i|0} \|_2^2 + \Sigma_{i=121}^{200} \| K_{\text{des}} x_{i|0} - u_{i|0} \|_2^2$, where we use the squared norm to accelerate the convergence of the underlying optimization algorithm. Note that a requirement for implementing this controller is that all future desired control inputs are known in advance, which is typically not feasible when safety filters are used online. The resulting solution depicted in Figure S6 shows a fundamentally different behavior than the previous approaches. Instead of reactively trying to correct desired control inputs, the ideal safety filter "invests" by overriding the desired control inputs for a short period at the beginning of the evolution, allowing the direct use of the desired input signal from 1.8 s until the end of the horizon. This is possible only because the ideal safety filter can anticipate the effect of aggressive desired input signals in $t = 0 - 6$ s, whereas the other safety filters make instantaneously optimal decisions or consider a much shorter predictive horizon.

COMPARISON OF APPROACHES

We compare the various safety filter implementations in Figure S7. In Figure S7(a), we compare the different safe sets (including two for the PSF using different horizons). We see that the HJ reachability safety filter (purple) contains the safe sets for the switching safety filter, CBF safety filter, and PSF using the shorter horizon length. The PSF using the longer horizon contains the HJ reachability safe set, which is due to using $\varepsilon = 0.02$ to account for numerical error when finding the HJ reachability safe set. Figure S7(b) shows the integral of the deviation of the input from the desired control input signal (4a). The switching safety filter and PSF with a short horizon have the biggest deviation, while the methods resulting in the largest safe sets modify the desired control input signal the least. Compared to the ideal safety filter solution (orange), the relative differences between the individual methods are visible but not significant.

REFERENCES

[S1] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Automat.*, New Orleans, LA, USA, 2004, pp. 284–289, doi: 10.1109/CACSD.2004.1393890.

[S2] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, Mar. 2006, doi: 10.1007/s10107-004-0559-y.

[S3] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, pp. 1–36, Mar. 2019, doi: 10.1007/s12532-018-0139-4.

and a set $\mathcal{S} \subseteq \mathcal{X}$ defined as the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$

$$\mathcal{S} = \{x \in \mathbb{R}^{n_x} \mid h_{\mathcal{S}}(x) \geq 0\} \tag{7a}$$

$$\text{int}(\mathcal{S}) = \{x \in \mathbb{R}^{n_x} \mid h_{\mathcal{S}}(x) > 0\} \tag{7b}$$

$$\partial\mathcal{S} = \{x \in \mathbb{R}^{n_x} \mid h_{\mathcal{S}}(x) = 0\}. \tag{7c}$$

Suppose the set $\mathcal{S}$ is forward invariant for the closed-loop system

$$\dot{x}(t) = f(x(t), \kappa_{\mathcal{S}}(x(t))), \quad \forall t \in \mathbb{R}_{\geq 0} \tag{8}$$

and that $\kappa_{\mathcal{S}}(x) \in \mathcal{U}$ for all $x \in \mathcal{S}$. A classic example of this setting is when $\kappa_{\mathcal{S}}$ is a locally stabilizing controller for some equilibrium point $x_e \in \text{int}(\mathcal{X})$ and $\mathcal{S}$ is the sublevel set of a Lyapunov function. In this setting, $\kappa_{\mathcal{S}}$ is often synthesized based on a linearization of the nonlinear dynamics (1) at the equilibrium point x_e; thus, the sublevel set of the Lyapunov function must be chosen relatively small. A small sublevel set leads to conservative behavior of the safety filter and will motivate later constructions with HJ reachability and PSFs.

Expressing $\mathcal{S}$ as the zero-superlevel set of the continuously differentiable function $h_{\mathcal{S}}$ allows us to consider a fundamental result in studying set invariance established in 1942 and known as *Nagumo's Theorem* [67] (see [13, Sec. 4.2.1] for a modern proof).

Theorem 1

Consider the closed-loop system (8) and a set $\mathcal{S} \subseteq \mathcal{X}$ defined as the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$ with $\text{int}(\mathcal{S}) \neq \varnothing$ and

$$\nabla h_{\mathcal{S}}(x) \triangleq \frac{\partial h_{\mathcal{S}}}{\partial x}(x) \neq 0 \tag{9}$$

for all $x \in \partial\mathcal{S}$. Then, the set $\mathcal{S}$ is forward invariant for (8) if and only if

$$\dot{h}_{\mathcal{S}}(x) \triangleq \nabla h_{\mathcal{S}}(x) f(x, \kappa_{\mathcal{S}}(x)) \geq 0 \tag{10}$$

for all $x \in \partial\mathcal{S}$.

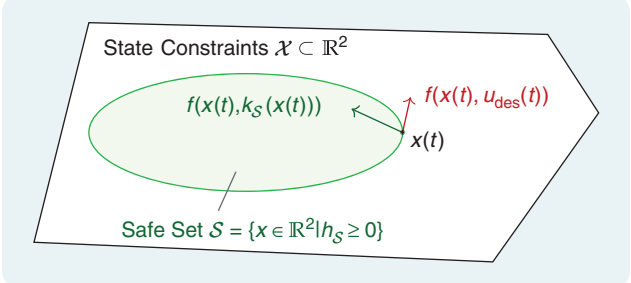


FIGURE 3 A geometric interpretation of Nagumo's theorem. The switching safety filter (11) builds directly off the condition (10) in Nagumo's theorem to enforce safety. At the state $x(t)$, the desired input $u_{\text{des}}(t)$ will cause the system to leave the safe set $\mathcal{S}$ since the vector $f(x(t), u_{\text{des}}(t))$ points outward with respect to the set $\mathcal{S}$. Switching to the safe control law (6) as dictated by (11) leads to the system remaining inside the set since the vector $f(x(t), \kappa_{\mathcal{S}}(x(t)))$ points inward with respect to the set $\mathcal{S}$.

The requirement (10) of Nagumo's theorem has a simple geometric interpretation, as seen in Figure 3. In particular, the vector given by the closed-loop dynamics (8) must point into the set $\mathcal{S}$ at each point on its boundary. Moreover, it is a necessary and sufficient condition for the forward invariance of the set $\mathcal{S}$, implying that the inequality in (10) is satisfied for all $x \in \partial\mathcal{S}$ since $\mathcal{S}$ is forward invariant for the closed-loop dynamics (8). We note that an analog of Nagumo's theorem for discrete-time systems does not exist in general [66, Sec. 3.2].

This property on the boundary of the set $\mathcal{S}$ allows the construction of a simple safety filter that switches between using the desired control input signal $u_{\text{des}}(\cdot)$ and the controller $\kappa_{\mathcal{S}}$. Recalling that $\kappa_{\mathcal{S}}(x) \in \mathcal{U}$ for all $x \in \mathcal{S}$, we construct a safety filter $\kappa_F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{U}$ as

$$\kappa_F(x, u) = \begin{cases} \kappa_{\mathcal{S}}(x), & x \in \partial\mathcal{S} \text{ or } u \notin \mathcal{U} \\ u, & \text{else.} \end{cases} \tag{11}$$

Such a switching safety mechanism was originally proposed in [68]. It is straightforward to see that

$$\nabla h_{\mathcal{S}}(x) f(x, \kappa_F(x, u)) \geq 0 \tag{12}$$

for all $x \in \partial\mathcal{S}$ and $u \in \mathbb{R}^{m_u}$ by virtue of $\kappa_{\mathcal{S}}$ satisfying (10) for all $x \in \partial\mathcal{S}$. Thus, we may conclude by Nagumo's theorem that $\mathcal{S}$ is forward invariant for the closed-loop system (5) using the proposed safety filter with $u_{\text{des}}(\cdot)$.

We note that the form of the safety filter (11) is not rigorous because instantaneous switches at the boundary of the system may not yield a piecewise-continuous input signal if the switches occur infinitely often in a finite period of time (commonly known as *Zeno behavior*). This issue can be resolved both theoretically and practically by requiring the controller $\kappa_{\mathcal{S}}$ to be used for a short time interval when activated. The choice of this time interval has practical consequences since short intervals can yield undesirable chattering behavior, while large intervals can lead to the safety filter rarely using the desired control input signal $u_{\text{des}}(\cdot)$. The main benefit of constructing the switching-based safety filter (11) is its simplicity of implementation whenever a controller $\kappa_{\mathcal{S}}$ and a corresponding forward invariant set $\mathcal{S}$ are available.

HJ Reachability Analysis for Safe Set Synthesis

In this section, we seek to reduce the conservativeness of the preceding switching safety filter design by constructively synthesizing the maximal control invariant set $\mathcal{S}$ in $\mathcal{X}$. We will achieve this through the method of HJ reachability [8]. HJ reachability is a constructive framework for solving reachability problems through optimal control theory based on HJ equations. A reachability problem in control generally seeks to determine the set of states that can be encountered by a trajectory of a dynamical system like (1). This captures a

Available safety filter techniques produce different feedback controllers and (control) invariant sets that permit varying degrees of performance.

broad collection of problems useful for system verification; for instance, reach-avoid problems are concerned with trajectories reaching a goal region while avoiding an unsafe region at the same time [42]. A comprehensive characterization of the various types of reachability problems is beyond the scope of this article. Instead, readers are referred to "Hamilton-Jacobi Reachability Safety Filter Applications" for specific examples of their application to safety verification of autonomous aerial and mobile vehicles and [8] for an in-depth technical description.

We now focus on the category of reachability problems that find the maximal control invariant set contained in the state constraint set $\mathcal{X}$. We study state trajectories *inevitably* reaching the unsafe region, $\mathcal{X}^c$ (the complement of $\mathcal{X}$), regardless of the control effort. The collection of such trajectories constitutes the region where violating state constraints is inevitable. The complement of this set, on the other hand, consists of states from which such failure can be avoided by an appropriate choice of control. Thus, this set becomes the maximal control invariant set contained in $\mathcal{X}$. The formal discussion of this complementary relationship is presented in [30]. Additionally, in [30], [36], this relationship is generalized to finite-horizon safety problems where the safe set does not necessarily have to be control invariant, as analyzed more in-depth in the viability theory literature [32], [36], [69].

Taking the HJ approach to this reachability problem allows the computation of the maximal control invariant set to be posed as an infinite-horizon optimal control problem [21], [36]. To see this, let $s_{\mathcal{X}} : \mathbb{R}^{n_x} \to \mathbb{R}$ denote the signed-distance function for the set $\mathcal{X}$ (see the "Definitions and Notation" section at the beginning of this article). The satisfaction of state constraints requires that $s_{\mathcal{X}}(x(t)) \leq 0$ for all $t \in \mathbb{R}_{\geq 0}$. Equivalently, we may consider a cost functional $J : \mathbb{R}^{n_x} \times \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \to \mathbb{R}$ defined as

$$J(x_0, u(\cdot)) = \inf_{t \in \mathbb{R}_{\geq 0}} -s_{\mathcal{X}}(x(t)) \tag{13}$$

where the state constraints are satisfied if and only if $J(x_0, u(\cdot)) \geq 0$. This cost functional enables us to define a value function $V : \mathbb{R}^{n_x} \to \mathbb{R}$ as

$$V(x_0) = \sup_{u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U})} J(x_0, u(\cdot)). \tag{14}$$

The value function defines an optimal control problem to maximize (13) across all feasible control input signals (piecewise-continuous and satisfying input constraints), ultimately to ensure that it is nonnegative and thereby implying the satisfaction of state constraints. This value function is a core concept in reachability-based safety filter design. It serves as a metric for quantifying safety margins, with negative values indicating violation of safety at some point in the future and with larger positive values reflecting more margin (because it is possible to keep the system's state further from the boundary of $\mathcal{X}$ through control), as captured in the following result [21, Proposition 4].

Theorem 2

For any $\epsilon \in \mathbb{R}_{\geq 0}$, the ϵ-superlevel set of the value function $V : \mathbb{R}^{n_x} \to \mathbb{R}$ defined in (14), denoted as

$$\mathcal{S}_\epsilon = \{x \in \mathcal{X} \mid V(x) \geq \epsilon\} \tag{15}$$

is a control invariant set for (1) and $\mathcal{S}_0$ is the *maximal control invariant set* in $\mathcal{X}$ for (1). Moreover, if $\mathcal{U}$ is compact, for all $x \in \mathcal{S}_0$ where the gradient of V exists,

$$\max_{u \in \mathcal{U}} \nabla V(x) f(x, u) \geq 0. \tag{16}$$

The preceding theorem establishes that we can construct the maximal control invariant set in $\mathcal{X}$ for (1), $\mathcal{S}_0$, through the value function V. The complement of this set captures all the states from which the system will inevitably reach $\mathcal{X}^c$, thereby violating state constraints. Thus, the boundary of the maximal control invariant set $\mathcal{S}_0$, which is characterized by the zero-level set of V, discriminates the region of the state space in which violating safety is inevitable from the region in which satisfying the safety constraints is feasible. In practice, using a control invariant set for (1) that is smaller than $\mathcal{S}_0$, which can be produced by considering ϵ-superlevel sets of V as noted in Theorem 2, provides a tunable buffer for accommodating errors when V is numerically approximated.

The value function (14) can also be used to synthesize a control policy that can be directly incorporated into a safety filter. If the value function V is differentiable, we can construct an optimal safe policy $\kappa_V^* : \mathbb{R}^{n_x} \to \mathcal{U}$ satisfying

$$\nabla V(x) f(x, \kappa_V^*(x)) = \max_{u \in \mathcal{U}} \nabla V(x) f(x, u). \tag{17}$$

Hamilton-Jacobi Reachability Safety Filter Applications

Hamilton-Jacobi (HJ) reachability provides an effective tool for guaranteeing and verifying the performance and safety properties of a system. The notion of a reachable set can be used to describe regions in the state space from which achieving performance goals or satisfying safety constraints is feasible. Such sets are often characterized as level sets of a value function of an optimal control or differential game problem, for instance, as in Theorem 2. Moreover, when a controller is not prespecified, reachability formulations can be used to synthesize controllers that achieve safety in an optimal manner, as in (17). Finally, model uncertainty and exogenous disturbances can be directly incorporated into reachability formulations, permitting the construction of robust control invariant sets.

The availability of tools for computing value functions [74] establishes HJ reachability as a framework for constructive verification and safe control synthesis. This has led to the application of HJ reachability in safety-critical real-world settings, such as aircraft traffic management [7], real-time motion planning [163], and flight envelope verification for new-generation electric vertical take-off and landing aircraft [164]. Figures S8 and S9 show other applications of HJ reachability for the verification of robotic aerial vehicles [S5], [S6], while Figure S10 introduces its use in autonomous vehicle navigation [154].

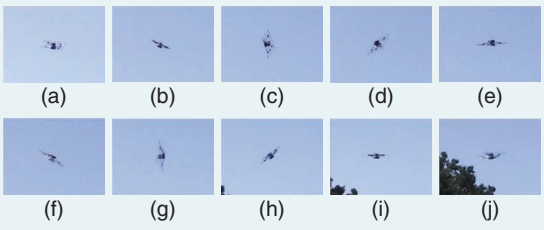


FIGURE S8 A mosaic of an autonomous backflip for the STARMAC quadrotor [S4]. (a)–(j) The controller takes the system through a sequence of mode transitions (a) from initiating the impulse mode, in which the vehicle rotation is induced by strong motor thrust, (b) entering the drift mode, where (b)–(f) it turn offs the motors and continues free-falling in, and (f) entering the recovery mode, (f)–(j) in which the quadrotor returns to hovering. (Source: [S5], ©2010 IEEE.)

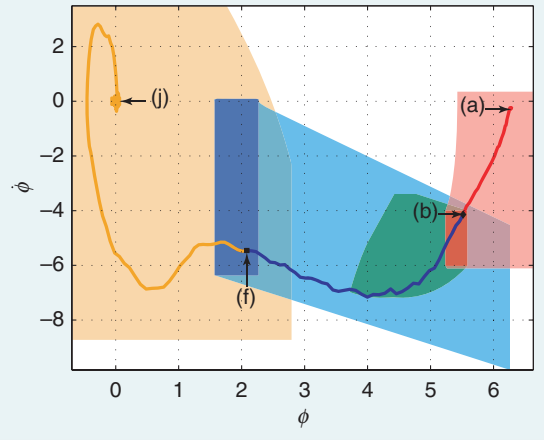


FIGURE S9 Reachable sets in the pitch angle ϕ and pitch rate $\dot{\phi}$ of the drone backflip maneuver seen in Figure S8. A pitching thrust is applied in the light red region, and the drone transitions from a pitch thrust to drifting in the dark red region. The drone transitions from drifting to recovering in the dark blue region, after which it arrives at a hovering equilibrium configuration. The ability to perform the backflip while ensuring a safety constraint on the minimum altitude of the vehicle is verified by analyzing reachable sets for the full system during the impulse, drift, and recovery stages of the vehicle [S6]. (Source: [S5], ©2010 IEEE.)

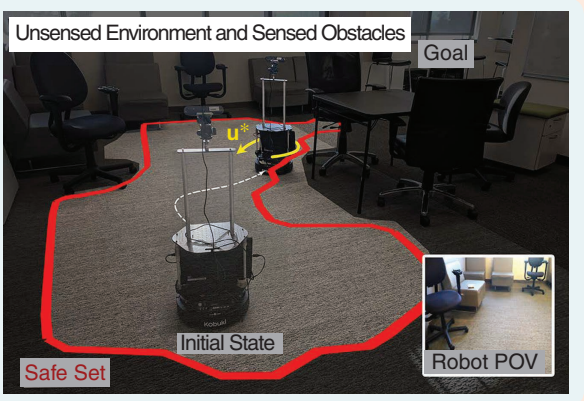


FIGURE S10 A safe autonomous navigation framework for an a priori unknown environment based on HJ reachability. The framework treats unexplored portions of the environment as an obstacle and uses HJ reachability to compute the safe region and the safe controller for the vehicle, which is updated in real time as the vehicle explores the environment. A vision-based and learning-based planner is deployed to reach the navigation goal, while the HJ reachability-based safety filter (19) keeps the robot safe when it is at risk of colliding with obstacles. POV; point of view. (Source: [154], ©2019 IEEE.)

REFERENCES

[S4] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proc. AIAA/IEEE 23rd Digit. Avionics Syst. Conf.*, Salt Lake City, UT, USA, 2004, pp. 12.E.4–121, doi: 10.1109/DASC.2004.1390847.

[S5] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 1649–1654, doi: 10.1109/ROBOT.2010.5509627.

[S6] J. H. Gillula, G. M. Hoffman, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *Int. J. Robot. Res.*, vol. 30, no. 3, pp. 335–354, Feb. 2011, doi: 10.1177/0278364910387173.

For all $x \in \mathcal{S}_0$. By construction,

$$\dot{V}(x) = \nabla V(x) f(x, \kappa_V^*(x)) \geq 0 \tag{18}$$

from (16). We note that if for some $\epsilon \in \mathbb{R}_{\geq 0}$, we have that $\nabla V(x) \neq 0$ for all $x \in \partial S_\epsilon$; this condition coincides with the necessary and sufficient condition of Nagumo's theorem for the set $\mathcal{S}_\epsilon$ to be forward invariant under the control law κ_V^*.

Similar to the switching safety filter in (11), given a desired $\epsilon \in \mathbb{R}_{\geq 0}$, we can design a switching safety filter based on the value of $V(x(t))$,

$$\kappa_F(x,u) = \begin{cases} \kappa_V^*(x), & V(x) \leq \epsilon \text{ or } u \notin \mathcal{U} \\ u, & \text{else.} \end{cases} \tag{19}$$

If $\epsilon = 0$, this safety filter is least restrictive [35], [70] in the sense that the filter intervenes only at the boundary of the (approximate) maximal control invariant set in $\mathcal{X}$. As before, it is necessary to use the controller κ_V^* for a short period of time when it is activated to avoid rapid switching, though this controller often practically displays less chattering than the naive switching safety filter (11).

Computing the value function V is the main task in verifying the maximal control invariant set $\mathcal{S}_0$ and constructing the safety filter (19) since it determines the ϵ-superlevel sets $\mathcal{S}_\epsilon$ and the optimal safe policy κ_V^*. The value function can be characterized as a solution of an HJ partial differential equation (HJ-PDE)

$$0 = \min\left\{-s_{\mathcal{X}}(x) - V(x),\ \max_{u \in \mathcal{U}} \nabla V(x) f(x,u)\right\} \tag{20}$$

that can be derived from the dynamic programming principle [42]. The HJ-PDE (20) does not necessarily admit unique solutions. In practice, the existence of a unique solution can be ensured by using a discounted formulation of the HJ-PDE [71], [72] or using a finite-horizon value function [replacing the time horizon in (13) with $[0, T]$] that approximates V for sufficiently large $T \in \mathbb{R}_{>0}$ [21]. Furthermore, if V defined as in (14) is not differentiable, it is still the viscosity solution of (20), which is a standard type of weak solution for PDEs not necessarily possessing a differentiable solution [73]. In the presence of such nondifferentiability, the optimal safe policy κ_V^* can be constructed using the notion of sub- and superdifferentials [33, Ch. III.3.4]. We note that under the Lipschitz continuity of the dynamics (1) and the signed-distance function $s_{\mathcal{X}}$, the discounted and finite-horizon value functions used to approximate the infinite-time value function are almost everywhere differentiable, implying the applicability of κ_V^* satisfying (17).

Algorithms for numerically computing the value function have been well developed [74], primarily through the notion of viscosity solutions [33], [73] and level-set methods for solving PDEs [75]. These algorithms typically rely upon forming a grid on the set $\mathcal{X}$ and evaluating the value function, its gradient, and the Hamiltonian [the left-hand side of (16)] at each grid point. Consequently, these approaches face challenges with problems possessing high-dimensional state spaces, a traditional challenge in dynamic programming known as the *curse of dimensionality* [29]. Recent research efforts have attempted to alleviate this challenge by using state decompositions [76], warm starting [77], or approximating solutions with neural networks [78]. Other works attempt to compute the maximal control invariant set approximately without relying on solving the HJ-PDE by using sums-of-squares programming [79], [80] or set operations based on polytopes [81], ellipsoids [30], [82], and zonotopes [83], [84]. Other approaches build upon a Bellman equation that captures the dynamic programming principle of reachability problems for discrete-time systems [30], [31], similar to the HJ-PDE (20) for continuous-time systems. This provides a foundation for many discrete-time-based dynamic programming algorithms like value iteration, Q-learning, or deep reinforcement learning (RL) as methods of finding approximate solutions of reachability [85], [86], [87].

Similar to the switching safety filter in (11), the reachability-based safety filter in (19) relies on instantaneously switching the control input from $u_{\text{des}}(t)$ to $\kappa_V^*(x(t))$ when the system encounters the boundary of $\mathcal{S}_\epsilon$. The instantaneous jumps in the control input can produce chattering, which may be infeasible in real-world systems. As a common practice to alleviate the chattering, the transition from $u_{\text{des}}(t)$ to $\kappa_V^*(x(t))$ in (19) as $V(x(t))$ approaches ϵ can be moderated in a smooth manner by blending the two control input values. More sophisticated filtering mechanisms that induce more desirable closed-loop system behavior while preserving safety are an area for future investigation in the practical deployment of reachability-based safety filters. For instance, the design principle of CBFs that will be presented next, which induces a smooth transition, can also be employed in the HJ reachability-based safety filter design [15]. However, we note that the HJ reachability framework was originally developed to build constructive tools for verifying the subset of the state space from which the safety specification (2) can be achieved. The switching safety filter in (19) is the elementary presentation of the usage of the computed maximal control invariant set through the HJ reachability.

Safety Filters Using CBFs

CBFs provide an alternative framework for constructing safety filters. While the relationship between CBFs and set invariance can be understood through the perspective of Nagumo's theorem, it can also be understood through the comparison principle, a fundamental idea in the study of nonlinear systems [88]. Through this

approach, it is possible to construct safety filters that smoothly modify a desired input control signal as the boundary of a set is approached, rather than switching to a safe controller only at the boundary. Moreover, the use of the comparison principle establishes connections between CBFs and Lyapunov functions, allowing a large set of tools developed in the context of stabilization to be adapted for the task of set invariance.

Historically, barrier methods were first developed in the context of constrained optimization [89], wherein constraint satisfaction could be achieved through increasingly large penalties on constraint violations. The idea to use barrier certificates in the context of nonlinear dynamical systems was first proposed in [37] for certifying the forward invariance of a set for a closed-loop system. This result was further developed in [9], yielding the first definition of CBFs as a tool for simultaneously synthesizing a safety-critical controller and a barrier certificate for the corresponding closed-loop system. The controller presented in this work was based on a structured design developed with control Lyapunov functions for stabilization in [90]. A consequence of this structured design was that the controller could not accommodate a desired control input signal that focused on performance instead of safety, making it unamenable for use as a safety filter.

A change to the formulation of CBFs that increased their potential for use as safety filters was proposed in [38]. The first component of this change was incorporating an extended class $\mathcal{K}$ function into the CBF time derivative condition required for safety. This change allowed the system state to approach the boundary of the safe set as long as it displayed a safe degree of "braking," reducing the conservative nature of the original definition of CBFs. The second component of this change was realizing that for control-affine systems, the CBF time derivative was affine in the control input, and thus, could be directly incorporated as a constraint in a convex optimization problem. This resulted in a way to optimally filter a desired control input signal while meeting safety constraints.

We now review CBFs, as first introduced in [28]. We study a broad subset of the class of systems described by (1) in the form of a control-affine nonlinear system

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad t \in \mathbb{R}_{\geq 0} \tag{21}$$

making similar assumptions on differentiability and the existence and uniqueness of solutions as made for (1). Given a feedback controller $\kappa : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$, we may construct a closed-loop system

$$\dot{x}(t) = f(x(t)) + g(x(t))\kappa(x(t), u_{\text{des}}(t)), \quad t \in \mathbb{R}_{\geq 0}. \tag{22}$$

With this definition in mind, we may define the following.

Definition 3 (BF)

Let $\mathcal{S} \subseteq \mathcal{X}$ be the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$. The function $h_{\mathcal{S}}$ is a BF for (22) on $\mathcal{S}$ if there exists $\alpha \in \mathcal{K}^e$ such that for any $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$

$$\dot{h}_{\mathcal{S}}(x,u) \triangleq \nabla h_{\mathcal{S}}(x)(f(x) + g(x)\kappa(x,u)) \geq -\alpha(h_{\mathcal{S}}(x)). \tag{23}$$

The following theorem is proven through comparison principles [43, Theorem 1] (and may also be proven using Nagumo's theorem [28, Proposition 1]) and shows how a BF serves as a certificate of set invariance.

Theorem 3

Let $\mathcal{S} \subset \mathcal{X}$ be defined as the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$. If $h_{\mathcal{S}}$ is a BF for (22) on $\mathcal{S}$, then the set $\mathcal{S}$ is forward invariant for the system (22).

This theorem states that if the closed-loop dynamics (22) satisfy the inequality in (23) at each point in the state space, the set $\mathcal{S}$ is forward invariant for (22). We observe two notable properties of the requirement in (23). The first property is that the time derivative of $h_{\mathcal{S}}$ must be lower bounded by a quantity that increases as $h_{\mathcal{S}}$ gets smaller. This induces a "braking" effect on the system, where it may not approach the boundary of $\mathcal{S}$ too quickly. The second property is that the time derivative of $h_{\mathcal{S}}$ must be positive outside of the set $\mathcal{S}$. This induces a type of asymptotic stability of the set $\mathcal{S}$ and plays a role in CBF safety filters' robustness to disturbances and model uncertainty [44].

As previously discussed, it is often easier to synthesize a safety filter given a control invariant set rather than construct a forward invariant set given a feedback controller. To this end, we define CBFs as in [28].

Definition 4 (CBF)

Let $\mathcal{S} \subseteq \mathcal{X}$ be the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$. The function $h_{\mathcal{S}}$ is a *CBF* for (21) on $\mathcal{S}$ if there exists $\alpha \in \mathcal{K}^e$ such that for any $x \in \mathbb{R}^{n_x}$,

$$\sup_{u \in \mathcal{U}} \nabla h_{\mathcal{S}}(x)(f(x) + g(x)u) > -\alpha(h_{\mathcal{S}}(x)). \tag{24}$$

Given a CBF for (21) on $\mathcal{S}$, we define the pointwise set

$$K_{\text{CBF}}(x) = \{u \in \mathcal{U} \mid \nabla h_{\mathcal{S}}(x)(f(x) + g(x)u) \geq -\alpha(h_{\mathcal{S}}(x))\}. \tag{25}$$

We note that the inequality in (24) is strict, while the inequality in (25) is nonstrict. The strictness of the inequality in (24) is critical for proving regularity properties such as Lipschitz continuity of controllers synthesized using the set K_{CBF} [45], which we will see an example of later. We first state the following result regarding the connection between a CBF and a BF [45, Theorem 1].

Theorem 4

Let $\mathcal{S} \subset \mathcal{X}$ be the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^{n_x} \to \mathbb{R}$. If $h_{\mathcal{S}}$ is a CBF for (22) on $\mathcal{S}$, then the set $K_{\text{CBF}}(x)$ is nonempty for all $x \in \mathbb{R}^{n_x}$, and for any locally Lipschitz continuous controller $\kappa : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ with $\kappa(x,u) \in K_{\text{CBF}}(x)$ for all $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$, the function $h_{\mathcal{S}}$ is a BF for (22) on $\mathcal{S}$.

Control Barrier Function Safety Filter Applications

Control barrier function (CBF)-based safety filters have seen extensive use in real-world applications, including mobile robots [92], robotic swarms [93], aerial vehicles [94], robotic arms [95], robotic manipulators [96], quadrupedal robots [97], bipedal robots [55], and automotive systems [98]. Practical safety tasks can often be encoded using the notion of forward invariance, such as safe foot placement on viable footholds, maintaining a safe following distance, avoiding obstacles in a complex dynamic environment, or respecting positioning constraints, as seen in the various examples in Figures S11–S14.

In each of these dynamic applications, safe control computations must be performed quickly. The formulation of CBF-based safety filters via convex optimization programs, such as in (26), permits a reliable and efficient means to quickly filter the desired control input signals. Several of these examples incorporate horizon-based elements seen in Hamilton-Jacobi (HJ) and predictive filter methods, either in the use of low-rate predictive filters (Figure S11), offline reference trajectories (Figure S13), or backup-set CBFs (Figure S14), to achieve both strong closed-loop performance in addition to safety.

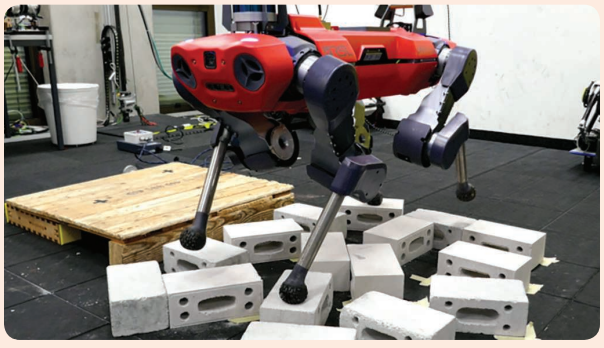


FIGURE S11 A CBF safety filter on a quadruped. A multilayered safety filter design is used that integrates predictive safety filters (PSFs) with CBFs to ensure safe foot placement on viable footholds while maintaining system stability. CBF constraints are integrated into both a mid-level predictive filter and a low-level CBF-based filter given by (26), ensuring a consistent safety specification across planning and control layers. (Source: [97], ©2021 IEEE.)

FIGURE S13 A CBF safety filter on a robotic arm in an industrial kitchen. Maintaining safety in a dynamic work environment shared with human personnel requires online modification of arm reference trajectories, but directly recomputing trajectories online is computationally intractable for real-time operation. CBF-based safety filters are used to efficiently modify trajectories given ongoing changes in the environment. (Source: [95], ©2022 IEEE.)

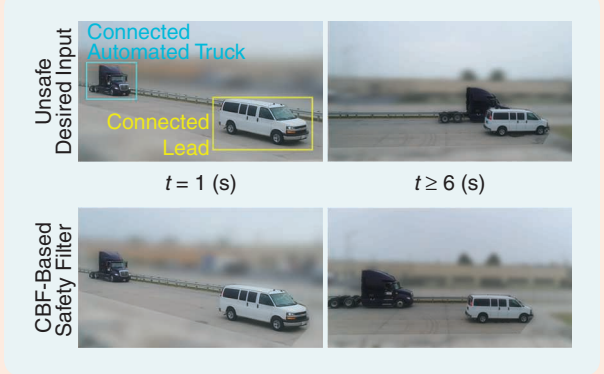


FIGURE S12 A BF safety filter on a connected automated semitrailer truck. The desired control input signal u_{des} is derived from an expert-designed controller that balances speed tracking with passenger comfort but does not keep the system safe. A CBF-based safety filter constructed using the ISSf notion of robustness ensures the safety of the truck in the presence of complex unmodeled braking system dynamics [98].

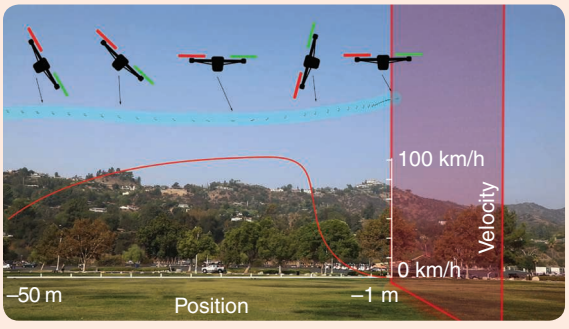


FIGURE S14 A CBF safety filter for geofencing of a high-speed drone. A backup-set CBF safety filter is designed to safely filter pilot inputs for a racing drone flying at high speeds (100 [km/h]), enabling acrobatic maneuvers while maintaining safety. The lightweight nature of CBF-based safety filters permits using only onboard sensing and computation, enabling beyond line-of-sight operation and robustness to ground communication failures. (Source: [165], ©2022 IEEE.)

We use a CBF to synthesize a safety filter as in [91] through the convex optimization problem

$$\kappa_F(x,v) = \underset{u\in\mathcal{U}}{\operatorname{argmin}} \quad \frac{1}{2}\|u-v\|_2^2 \\ \text{subject to} \quad \nabla h_S(x)(f(x)+g(x)u) \geq -\alpha(h_S(x)). \quad (26)$$

This controller is a convex quadratic program that may be efficiently solved. By construction, it satisfies $\kappa_F(x,u_{\text{des}}(t)) \in K_{\text{CBF}}(x)$ for all $x\in\mathbb{R}^{n_x}$ and $t\in\mathbb{R}_{\geq 0}$ such that the conditions of Theorem 4 are met, and thus, by Theorem 3, we can conclude that the set $\mathcal{S}$ is forward invariant for (22). Moreover, it allows the desired input signal $u_{\text{des}}(\cdot)$ to be minimally modified such that $u_{\text{des}}(t)$ is not used only when it is unsafe, and the input $u(t)$ actually used is as close as possible to $u_{\text{des}}(t)$.

The preceding controller has been deployed in several experimental contexts, including mobile robots [92], robotic swarms [93], aerial vehicles [94], robotic arms [95], robotic manipulators [96], quadrupedal robots [97], bipedal robots [55], and automotive systems [98]. A more detailed overview of some of these applications can be found in "Control Barrier Function Safety Filter Applications." This collection of successful practical applications indicates that CBFs are a powerful tool for safety filter design for complex high-dimensional nonlinear systems. Additionally, we note that CBF-based safety filters have been formulated for discrete-time systems [99] and sampled-data systems [96], [100] that fuse continuous-time dynamics with discrete-time controller implementations.

Despite these successes, there remain challenges and limitations facing CBF-based safety filters. A key challenge lies in constructively synthesizing CBFs and verifying that the condition in (24) can be met over the state space (or over some limited part of the state space), especially with bounded inputs. For relatively simple systems, it is often possible to check this condition analytically, but it can be difficult to verify for more complex high-dimensional systems. Recent attempts to solve this challenge have considered numerical optimization-based approaches through sums-of-squares programming [101], [102], [103] using reduced-order models coupled with approaches for handling the full-order system dynamics [104], [105] or learning CBFs from data [106], [107], [108], [109], [110]. Still, well-established and principled methodologies for finding CBFs, especially in the presence of input constraints, remain an open research question.

Predictive Safety Filters

The previously discussed safety filter methods rely on an explicit characterization of the safe set. The underlying computations to produce this characterization are typically limited in scalability, as in the case of HJ reachability, or can be potentially conservative approximations, as with CBF-based methods. Recent concepts such as active set methods [46], Safety Handling Exploration with Risk Perception Algorithm (SHERPA) [47], model predictive safety certification [12], PSFs [60], and predictive shielding [48] aim at addressing this challenge and provide a tradeoff between scalability and performance by a just-in-time computation of predictive backup plans. We specifically focus on PSFs [12], [60] in the following due to their close relation with (data-driven) model predictive control (MPC) literature [22], [65], [111]. This connection provides PSFs with an extensive theoretical background covering a variety of system model classes with uncertainties, data-driven estimates, and efficient computational toolsets for their implementation [11].

Despite these similarities, safety filters solve a fundamentally different task than standard MPC formulations. Instead of minimizing an objective function, the optimum of which commonly describes a desired system behavior, a PSF provides a modular framework that alleviates limiting assumptions on the objective, such as stabilization or reference-tracking conditions. At the same time, the PSF can build on the large body of available theories, ensuring recursive feasibility and constraint satisfaction from the literature.

Once a PSF is implemented for a system, it can be used to enable safe operation in various scenarios, for example, the application of excitation signals, minimizing black-box objectives functions (available only as numerical values) via learning-based approaches, or the minimization of discontinuous reward functions using RL techniques. See [22, Sec. 5] for a detailed discussion.

PSFs are based on the idea of extending a potentially conservative control invariant terminal safe set $\mathcal{S}^{\text{trm}}$ using predictive backup plans. More precisely, for a time $t_0\in\mathbb{R}_{\geq 0}$, consider the system state $x(t_0)$ and the desired input $u_{\text{des}}(t_0)$. Letting $T\in\mathbb{R}_{>0}$ be a prediction horizon, the safety of the desired input $u_{\text{des}}(t_0)$ is certified by searching for a state trajectory $x(\cdot)\in C([t_0,t_0+T],\mathcal{X})$ and an input signal $u(\cdot)\in PC([t_0,t_0+T],\mathcal{U})$ satisfying the system dynamics (1) and the boundary conditions $x(t_0)=x(t_0)$, $x(t_0+T)\in\mathcal{S}^{\text{trm}}$, and $u(t_0)=u_{\text{des}}(t_0)$. If such a state trajectory and input signal exist, then it is possible to use the desired control input and bring the system from the state $x(t_0)$ into the set $\mathcal{S}^{\text{trm}}$ within the finite prediction horizon T while respecting state and input constraints. If the desired input $u_{\text{des}}(t_0)$ is not safe, a safe control input is chosen such that the system can be brought to the terminal safe set $\mathcal{S}^{\text{trm}}$ within the prediction horizon.

We note that because the PSF is implemented with a receding horizon, the actual evolution of the system is not required to follow the backup plan into the terminal safe set $\mathcal{S}^{\text{trm}}$. Instead, the system will evolve using the input at the beginning of the predictive horizon (which must be consistent with a backup plan that returns to $\mathcal{S}^{\text{trm}}$ further in the horizon), after which it will compute a new backup plan. In this way, the system is allowed to freely evolve according to $u_{\text{des}}(\cdot)$ and does not need to return to $\mathcal{S}^{\text{trm}}$ as long as it remains possible to return to $\mathcal{S}^{\text{trm}}$ in the future

(see Figure 4 for an illustration). Note that this mechanism contrasts with standard stabilizing and reference-tracking MPC formulations [11] in which open-loop predictions ideally match the resulting closed-loop behavior to optimize performance rather than serving as an optional backup trajectory for safety. In particular, a PSF does not have to fulfill the task at hand, and in particular, the terminal safe set and backup plan can be based on a more straightforward set of system behaviors than u_{des} tries to achieve.

Implementing a PSF requires solving a predictive control problem online. While efficient solvers are available [11], they require a nonnegligible evaluation time period, compared with CBF or HJ reachability-based safety filters. During an evaluation time period $\Delta T \in \mathbb{R}_{>0}$, the previous input is typically held constant, resulting in zero-order-hold input signals, that is, $u(t) = \kappa_F(x(k\Delta T), u_{\text{des}}(k\Delta T))$ for all $t \in [k\Delta T, (k+1)\Delta T)$, where $k \in \mathbb{N}$ denotes the corresponding sampling time step. A common approximation in predictive control is to integrate the dynamics (1) within $t \in [k\Delta T, (k+1)\Delta T)$ using explicit integration methods. For example, applying a simple standard Euler discretization to (1) yields an approximate discrete-time zero-order hold formulation of the continuous-time system model (1)

$$x(k+1) = x(k) + \Delta T f(x(k), u(k)) = \mathrm{f}(x(k), u(k)). \tag{27}$$

A comprehensive introduction to different numerical integration methods used in predictive control can be found in [11, Sec. 8.2], and a detailed theoretical analysis is provided in [40, Ch. 2], [112].

Let N be the discrete-time prediction horizon. At the sampling time step k, the construction of a safe backup trajectory $\{x_{i|k}\}$ for $i = 1, \ldots, N$ toward the terminal safe set $\mathcal{S}^{\text{trm}}$ is formulated as

$$\begin{aligned}
\min_{u_{i|k}} \quad & \| u_{\text{des}}(k) - u_{0|k} \| && && \text{(28a)}\\
\text{Subject to} \quad & x_{0|k} = x(k) && && \text{(28b)}\\
& x_{N|k} \in \mathcal{S}^{\text{trm}} && && \text{(28c)}\\
& x_{i+1|k} = \mathrm{f}(x_{i|k}, u_{i|k}) && \text{for } i = 0, \ldots, N-1 && \text{(28d)}\\
& x_{i|k} \in \mathcal{X}, && \text{for } i = 0, \ldots, N-1 && \text{(28e)}\\
& u_{i|k} \in \mathcal{U}, && \text{for } i = 0, \ldots, N-1 && \text{(28f)}
\end{aligned}$$

where $i|k$ denotes planned states and inputs computed at time step k predicted i time steps into the future that satisfy the dynamic constraint (28b). An illustration of this planned sequence of states can be seen in Figure 4. The remaining constraints (28c)–(28f) ensure that backup plans lead the system into a safe terminal control invariant set $\mathcal{S}^{\text{trm}}$ [(28e) is referred to as the terminal constraint in MPC [11]] while satisfying the state and input constraints. While the objective (28a) aligns with the primary formulation of PSFs found in literature, additional terms can be included, for example, $\sum_{i=0}^{N-1} \| u_{\text{des}}(k+i) - u_{i|k} \|$ to better approximate (4) in the case that $u_{\text{des}}(\bar{k})$ is available for future time steps $\bar{k} > k$. The following assumption on the terminal safe set $\mathcal{S}^{\text{trm}}$ ensures that this optimization problem yields a safe input.

Assumption 1 (Terminal Control Invariant Set)

Consider the system (27). There exists a terminal set $\mathcal{S}^{\text{trm}} \subseteq \mathcal{X}$ such that for all $x \in \mathcal{S}^{\text{trm}}$, there exists an input $u \in \mathcal{U}$ such that $\mathrm{f}(x, u) \in \mathcal{S}^{\text{trm}}$.

Assumption 1 states that the terminal set $\mathcal{S}^{\text{trm}}$ is a discrete-time control invariant set, similar to the continuous-time version in Definition 2, and thus can be kept safe for all time. A trivial choice for $\mathcal{S}^{\text{trm}}$ is any equilibrium point $x_e = \mathrm{f}(x_e, u_e)$ satisfying $x_e \in \mathcal{X}$ and $u_e \in \mathcal{U}$. Since such terminal equality constraints can be rather conservative, control Lyapunov techniques are commonly used to provide the existence of a locally stabilizing controller for the equilibrium (x_e, u_e), which allows constructing less restrictive invariant terminal set constraints satisfying Assumption 1 (see, for example, [11, Sec. 2.5.3.2], [113]). In addition, various research efforts provide alternative methods for the terminal set design, such as terminal sets based on safe periodic system orbits [114], [115], discrete-time CBFs [17], [18], [99], and adaptive enlargements of the terminal set using previous solutions of (28) [12], [116] or closed-loop system trajectories [117].

The resulting PSF for the discrete-time system (27) is then given by $\kappa_F(x(k), u_{\text{des}}(k)) = u^*_{0|k}$ with $u^*_{0|k}$ being the first element of the optimal backup control sequence obtained from (28). The formal closed-loop safety guarantee under the application of $u(k) = \kappa_F(x(k), u_{\text{des}}(k))$ follows from an induction argument. In particular, assume that (28) was feasible at time $k-1$ with the corresponding optimal input sequence $\{u^*_{i|k-1}\}$. Under the application of $u(k-1) = \kappa_F(x(k-1), u_{\text{des}}(k-1)) = u^*_{0|k-1}$, the system evolves to the state $x(k) = x^*_{1|k-1}$ Because the terminal set is a control invariant set, we can construct a feasible candidate sequence at time step k given by $\{u^*_{1|k-1}, u^*_{2|k-1}, .., u^*_{N-1|k-1}, \bar{u}\}$ with $\bar{u} \in \mathcal{U}$ such that $\mathrm{f}(x^*_{N-1|k-1}, \bar{u}) \in \mathcal{S}^{\text{trm}}$, thereby satisfying all constraints in (28).

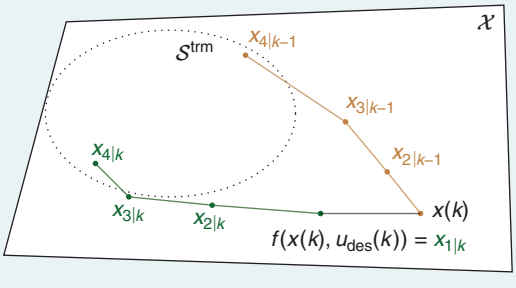


FIGURE 4 The mechanism of PSFs. The current system state $x(k)$ is shown with a safe backup plan (brown) from the solution at time $k-1$. A desired input signal $u_{\text{des}}(k)$ is applied if a feasible backup trajectory (green) can be obtained from the resulting $\mathrm{f}(x(k), u_{\text{des}}(k))$ via the optimization problem (28).

The key approximation of that all methodologies follow is through the use of a control invariant set.

By induction, we may conclude the feasibility of (28), and consequently, the satisfaction of state and input constraints due to (28e) and (28f), if (28) is initially feasible at $k = 0$. This result also implies that the set of feasible initial conditions

$$\mathcal{S}_N^{\text{PSF}} = \{x(k) \in \mathbb{R}^{n_x} \mid (28b) - (28f)\} \tag{29}$$

implicitly defines a control invariant set. This eliminates the need for an explicit safe set representation as, for example, a superlevel set of a function, which is often difficult to compute for high-dimensional systems.

While PSFs provide a flexible framework for approximately optimal safety filtering and approximate optimal control, the central challenge is to solve (28) reliably in real time. This is addressed theoretically and through software tools [11, Sec. 8] and is a central part of ongoing MPC research. Another practical challenge when implementing a PSF arises if disturbances drive the plant into a state for which the problem (28) is infeasible and no safe control input can be computed. A systematic method for dealing with infeasibility is to "soften" the constraints by including slack variables into the problem, as commonly done in MPC [118]. For instance, when the state and terminal constraints can be described by $\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid a^{\mathcal{X}}(x) \le 0\}$ and $\mathcal{S}^{\text{trm}} = \{x \in \mathbb{R}^{n_x} \mid a^{\mathcal{S}^{\text{trm}}}(x) \le 0\}$ for some functions $a^{\mathcal{X}}$, $a^{\mathcal{S}^{\text{trm}}}$, respectively, the soft-constrained PSF problem (28) is

$$\min_{u_{i|k}, \xi_{i|k}} \left\| u_{\text{des}}(k) - u_{0|k} \right\| + \sum_{i=0}^{N} l_\xi(\xi_{i|k})$$

subject to (28*b*), (28*c*), (28*f*),

$$\begin{aligned} &\xi_{i|k} \ge 0, && \text{for } i = 0, \ldots, N, \\ &a^{\mathcal{X}}(x_{i|k}) \le \xi_{i|k}, && \text{for } i = 0, \ldots, N-1, \\ &a^{\mathcal{S}^{\text{trm}}}(x_{N|k}) \le \xi_{N|k}. \end{aligned} \tag{30}$$

The nonnegative slack variables $\{\xi_{i|k}\}$ ensure feasibility for any $x(k)$ and any input sequence $u_{i|k} \in \mathcal{U}$. The corresponding penalty function l_ξ can, for example, be selected as $l_\xi(\xi) = \| \xi \|^2 + \rho_\xi \| \xi \|$, where ρ_ξ is a positive constant. The goal is to select ρ_ξ large enough such that the second term in (30) admits an exact penalty function, implying that the slack variables are nonzero only if the constraint satisfaction of the corresponding constraints is not possible. If the original hard-constrained problem (28) is feasible, the soft-constrained problem should produce the same control input [118]. It should be noted that the slack variables are, however, not guaranteed to vanish in closed-loop operation; that is, the system may not return to the implicit safe set defined by (28). Current research efforts in MPC [119], [120] and PSFs [18] investigate such cases, for example, by connecting PSF and CBF theory [18] (see also the discussion in the "CBFs + PSFs" section).

Discussion on Basic Safety Filters

In this section, we first discuss in what ways HJ reachability, CBF, and PSF safety filters approximate the ideal safety filter defined in (4). Next, we provide a brief overview of the relationship between the three methods, with a focus on recent work at the intersection of the approaches.

Approximation of Ideal Safety Filter

As discussed in the introduction of the ideal safety filter (4), it is often infeasible to directly solve (4), either due to computation limits or because the entirety of the desired input signal $u_{\text{des}}(\cdot)$ is unavailable in advance. All three previously described methods can be seen as approximating various features of the ideal safety filter in (4).

The key approximation of (4) that all methodologies follow is through the use of a control invariant set. While it is desired for the ideal safety filter to work for any initial condition $x_0 \in \mathcal{X}$, it will be feasible only for initial conditions x_0 in the maximal control invariant set in $\mathcal{X}$. Each of the methodologies considers a subset of $\mathcal{X}$ that may be rendered forward invariant, ensuring that the constraint (4e) is satisfied. HJ reachability finds an explicit representation of the maximal control invariant set and can return $\mathcal{X}$ itself if $\mathcal{X}$ itself is control invariant. CBFs often use an inner approximation of the set $\mathcal{X}$ that the CBF condition (24) can be verified over but does not necessarily seek a maximal control invariant set. Lastly, PSFs return an implicit representation of a control invariant set contained in $\mathcal{X}$, with the set being implicitly defined by the feasibility of the optimization problem in (28). In practice, this control invariant set can closely approximate the maximal control invariant set in $\mathcal{X}$ with a sufficiently large time horizon.

The second major way in which the three methodologies approximate (4) is the minimization of the cost (4a). HJ reachability does not explicitly consider the minimization of this cost. However, if from a given initial condition x_0, the desired input signal $u_{\text{des}}(\cdot)$ keeps the system in the maximal control invariant set, then a cost of zero can be obtained. If instead $u_{\text{des}}(\cdot)$ would cause the system to leave the maximal control invariant set, the filter switches to the optimal safe policy for a period of time according to (19). These switches lead to accruing of some cost according to (4a) and are generally not ensured to be minimal.

In contrast, the CBF safety filter specified in (26) seeks to minimize an instantaneous deviation from the desired input signal at each time $t \in \mathbb{R}_{\geq 0}$, given by $u_{\text{des}}(t)$, subject to the CBF inequality in (24). There are two consequences of this with respect to minimizing the cost (4a). First, even if from a given initial condition x_0, the desired input signal $u_{\text{des}}(\cdot)$ would keep the state inside the zero-superlevel set of the CBF (and hence inside $\mathcal{X}$), it may be modified to ensure that the stricter requirement specified by the CBF inequality in (24) is satisfied. Second, the CBF safety filter does not explicitly consider the behavior of the system along a horizon (it does so implicitly through the time derivative requirement on the CBF h). Hence, the CBF safety filter may choose to return an instantaneously optimal input but accrue more cost along the evolution of the system.

With respect to optimality, the finite-horizon formulation of PSFs makes them an intermediate between the infinite horizon used in HJ reachability and the instantaneous optimization used with CBFs. The cost of the PSF in (28a) is an instantaneous optimization in that it considers $u_{\text{des}}(k)$. Despite this instantaneous nature, the PSF will not need to modify $u_{\text{des}}(k)$ if there exists a subsequent input sequence that keeps the system within the set $\mathcal{X}$. Practically, this can lead to a significant improvement in minimizing the cost (4a) over purely instantaneous approaches not considering a horizon. Moreover, suppose the input signal $u_{\text{des}}(\cdot)$ is known in advance. In that case, future values of the desired input signal (such as $u_{\text{des}}(k+1)$) can be incorporated into the cost function of the PSF, yielding more optimal behavior. Examples of such inputs include open-loop excitation signals or if they can be parameterized in terms of the current system state, such as with a feedback controller.

Lastly, there are two additional, but minor, ways in which the ideal safety filter in (4) is being approximated. First, the PSF does not use the exact continuous-time nonlinear dynamics in (4d) but rather uses a discrete-time approximation. The accuracy of this approximation can play a role in both the safety guarantees and optimality of the resulting filter. Secondly, the switching nature of the HJ reachability safety filter and the discrete-time nature of the PSF limit the possible piecewise-continuous input signals the system can take. As noted when discussing the switching safety filter, it is necessary for the switch to the safe controller to be held for a minimum period of time to prevent Zeno behavior. This restricts the piecewise-continuous input signals of the HJ reachability safety filter to correct a minimum amount of time between their discontinuities. The discrete-time formulation used for the PSF also enforces a minimum amount of time between the discontinuities of the piecewise-continuous signals achievable by the filter. In contrast, the continuity properties of the CBF and the system dynamics permit the full class of piecewise-continuous input signals given by the constraint (4b), with the only discontinuities in the input signal arising from discontinuities in $u_{\text{des}}(\cdot)$.

HJ Reachability + CBFs

Both the methods of HJ reachability and CBFs are built on determining an explicit representation of a control invariant set, typically through the superlevel sets of a continuous scalar function (Figure 3). This similarity leads to connections between the two approaches, both theoretically and in practical behavior. Moreover, the combination

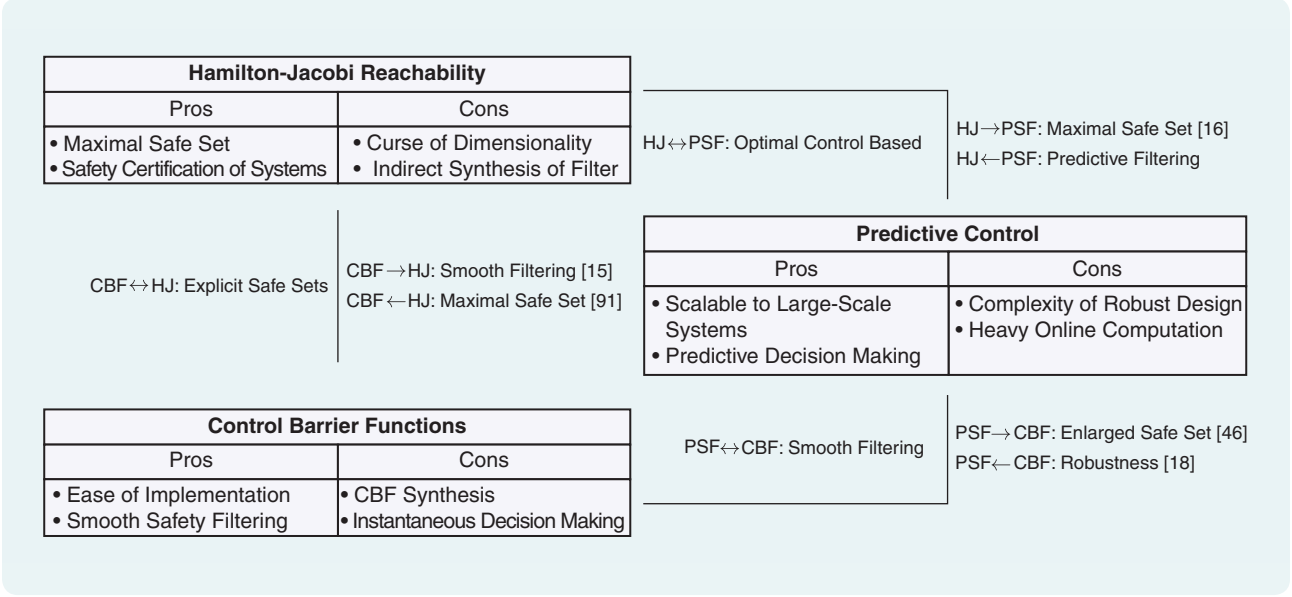


FIGURE 5 The key advantages and drawbacks of HJ reachability, CBFs, and PSFs. The relationship between them is outlined (↔), and how techniques enhance each other; for example, CBF techniques can be used to improve HJ safety filters (CBF → HJ).

of these approaches is complementary since HJ reachability can increase the size of a control invariant set used in a safety filter, while CBFs provide a succinct approach for smoothly filtering a desired input signal.

The use of HJ reachability for the construction of CBFs was first explored in [91], in which a piecewise polynomial function whose zero-superlevel set smoothly approximates the maximal control invariant set computed from HJ reachability was constructed by sums-of-squares programming and used as a CBF. This approach enabled a large control invariant set while preserving the smoothness properties needed by CBFs. The work in [14] conducts a comparative study of the control invariant sets found using HJ reachability and backup CBF methods (discussed later in the "CBFs + PSFs" section). This work found that given an adequately designed backup controller and backup set, the control invariant sets found with backup CBF approaches closely approximate the maximal control invariant sets found through HJ reachability.

Other recent work has looked at how elements from CBF-based safety filters can be directly incorporated into HJ reachability computations. The work in [15] integrates the comparison function seen in CBF-based safety filters into the HJ-PDE (20) that is solved numerically, allowing for the synthesis of CBFs through the toolsets typically used in HJ reachability. In this new reachability formulation, the safety filter (26) based on the resulting value function is verified to be the optimal policy of the value function. This allows the reachability community to expand their choice of the safety filters from the primary switching safety filter (19) to those in the CBF community, which have better practical behaviors. The work in [121] integrates with this previous work by making use of the ability to warm-start the process of numerically solving the HJ-PDE (20) to use dynamic programming to iteratively update a CBF candidate until it converges to a valid CBF. Though these approaches benefit from the strengths of both HJ reachability and CBFs, their numerical approach still faces challenges with high-dimensional systems.

Beyond finding efficient approaches for tackling the curse of dimensionality in this context, an open research direction at this intersection focuses on rigorously studying the regularity properties of CBFs constructed through reachability frameworks. Such an effort would rigorously codify the regularity properties achieved by weak viscosity solutions to the HJ-PDE and develop similarly rigorous results connecting the resulting CBFs and safety in the face of these regularity limitations, similarly to those in [122].

HJ Reachability + PSFs

While both HJ reachability and PSFs aim to ensure safety through an optimal control problem formulation, there are differences in their respective problem structures and corresponding algorithms. First, HJ reachability incorporates safety constraints through an appropriate value function (14), whereas PSFs consider them as part of a constrained optimization problem (28d). As a result, HJ reachability-based safety filters decouple safe set synthesis and filter design, while PSFs implicitly capture a safe set and filter inputs through a single optimization problem.

Second, HJ reachability uses the machinery of dynamic programming [123] to find an optimal solution offline for all states. Typically, the value function (14) is explicitly computed as the solution of an HJ-PDE, which is feasible for a class of optimal control problems [33], including both reachability formulations and state-constrained general-cost problems [124]. The computation of the value function globally for all states faces the curse of dimensionality. In return, it explicitly characterizes the maximal control invariant set in $\mathcal{X}$ before deploying the controller, which makes it attractive for the verification of safety-critical systems [7].

In contrast, PSFs leverage online optimization to approximately solve a state-constrained optimal control problem (28) by using only the current state and a receding-horizon principle [11]. The online computation concept of PSFs avoids an explicit precomputation of an optimal control policy, thereby enabling the scalability of the approach. Instead, PSFs require efficient nonlinear programming solvers working in real time with significant system processing power. If sufficient computation power is available online, PSFs can provide a near-ideal safety filter even for high-dimensional systems. However, evaluating whether the system will be safe given an initial state can be verified only by evaluating the feasibility of the optimization problem (28) as an explicit representation of the safe set is not available.

Despite these differences between the two approaches, there are similarities between the approaches that suggest the potential for integrating them. In particular, the implicit safe set defined by a PSF using a sufficiently long planning horizon coincides with the explicit safe set from HJ reachability. This effect is demonstrated in "Safety Filter Design Example." Finally, recent approaches are exploring various ways of exploiting the benefits of both methods; see, for example, [16], where condition (18) from HJ reachability is incorporated as a constraint in a predictive controller.

CBFs + PSFs

CBFs and PSFs naturally complement each other in a way that reduces the weakness of each individual method. The predictive horizon present in PSFs can help to reduce poor closed-loop behavior induced by the instantaneous optimization of CBF-based safety filters by incorporating future desired control inputs into (28a) and by increasing the planning horizon. This improvement comes with the burden of solving a nonlinear optimization problem in real time,

Empirical information about the unknown system is integrated into various elements of the safety filter synthesis process.

which substantially increases the complexity of the safety filter design and implementation compared to CBF-based filters. Furthermore, PSFs do not provide intrinsic robustness properties, often resulting in complicated design procedures to ensure safety with disturbances.

This complementary relationship has yielded several recent results integrating the two methods. Integrating CBF constraints directly into the optimization problem specifying the predictive filter, either as an instantaneous derivative condition [97] or as a decrement condition [17], [125], [126], [127], leads to the dynamic "braking" typical of CBFs and often yields robust behavior. In addition, the use of CBFs as a terminal constraint can formally render the sum of slack variables in the PSF problem (30) into a "predictive" CBF [18]. Further approaches include multirate architectures, in which a high-level predictive controller provides a desired input signal that is filtered using a CBF-based safety filter [19], [97], [128]. These approaches allow for the complex nonlinear predictive optimization problem to be solved at slower frequencies since the CBF-based filter keeps the system close to the planned trajectory at a high frequency. Other approaches have introduced predictive elements to consider safety along solution trajectories [129] or used predictive elements for trajectory tracking and CBFs for obstacle avoidance [130].

The thread of work in [46], [131], [132] focuses on the notion of backup-set methods using CBF-based safety filters. This approach shares conceptual elements with PSFs by using a backup set that can be kept forward invariant with a backup controller to implicitly define a larger control invariant set. The backup-set methods consider a predictive horizon over which a CBF constraint must be enforced, ensuring that the system can always reach the backup set. Structural differences between these backup-set approaches and predictive filters often lead to different approximations for tractably handling the use of a predictive horizon, suggesting a distinction between the two methods.

DATA-DRIVEN SAFETY FILTERS

The safety filter techniques summarized in the first half of this article were presented assuming perfect knowledge of the system dynamics (1). However, in most practical settings, high-fidelity system models are difficult to construct, and systems are subject to external disturbances, which can lead to the loss of safety guarantees. This challenge has been a topic of significant research interest from the perspective of data-driven control, in which empirical information about the unknown system is integrated into various elements of the safety filter synthesis process. We now present this problem setting and a selection of data-driven results related to HJ reachability, CBFs, and PSFs.

Consider the nonlinear control system

$$\dot{x}(t) = f_{\text{true}}(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0} \tag{31}$$

where $f_{\text{true}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, which, for simplicity, we assume to be continuously differentiable in its arguments. For many systems in engineering, first principles, such as Lagrangian mechanics or the laws of thermodynamics, allow for the derivation of simplified model structures to construct the function f in (1). In many real-world applications, generating a sufficiently accurate first-principles model can, however, require significant engineering effort, leading to a discrepancy between the model f and the actual dynamics of the system, given by f_{true}.

We note that the mathematical formulation in (31) can describe the frequent setting of parametric uncertainty. Even for systems for which the structure of f accurately characterizes f_{true}, there may be errors between the parameters of the model and the parameters of the actual system. For instance, different cars of the same vehicle type may have the same model structure but may differ in the model parameters due to manufacturing tolerances, wear of components, or replacement parts. Manually identifying parameters through laboratory testing is often difficult and costly, and designs that are robust to large parameter uncertainties are often conservative. Data-driven techniques specialized for addressing safety in the face of parametric uncertainty have been proposed, including adaptive control [133], [134] and Bayesian estimation [135], [136].

This section discusses how to leverage data-driven techniques to improve a model obtained from first principles, given by (1), to more accurately reflect (31) and presents selected techniques for using these concepts in the context of safety filters. To this end, we consider a sequence of measured states, inputs, and state time derivatives

$$D = \{(x_k, u_k, \dot{x}_k)\}_{k=1}^{n_D} \triangleq \{(x(kT_s), u(kT_s), \dot{x}(kT_s)\}_{k=1}^{n_D} \tag{32}$$

at sampling time steps kT_s. Collecting data of the form (32) assumes prior physical knowledge about the system dynamics to determine a suitable selection of system state measurements. "Learning With Real-World Data" outlines some

preprocessing steps to obtain (32) from state measurements. If the true system provides only limited access to system states through noisy sensor measurements of the form $y_k = g(x(kT_s))$, $y_k \in \mathbb{R}^{n_y}$ with $n_y < n_x$, more advanced methodologies from the field of system identification [137], [138] may be used. We note that data of the form (32) can equally handle episodic measurements, including multiple resets of the system state, enabling iterative model refinement.

Learning With Real-World Data

Data-driven safety filters provide a promising approach for infusing information collected from experiments on a real-world system into the control design process to improve the safety of a system. Achieving this goal requires overcoming challenges that often arise when data are produced by real-world dynamic systems. In this sidebar, we discuss some of the tradeoffs faced when solving these challenges in the context of an example using real-world data.

Consider a scalar nonlinear dynamical system given by

$$\dot{x}(t) = f_{\text{true}}(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0} \tag{S17}$$

with state $x(t) \in \mathbb{R}$ and input $u(t) \in \mathbb{R}$ at time $t \in \mathbb{R}_{\geq 0}$. Though this is a continuous-time system, data collected during its evolution are typically discrete time in nature, with states x_k and inputs u_k measured at sample times kT_s, as in the dataset (32). An example of such a state sequence is given by the black line in Figures S15(a) and S16(a), produced by the Segway system in [23]. Typically absent from these collected data are direct measurements of the state derivative, $\dot{x}_k$. To build the dataset D in (32) and characterize the function e^n that arises in the model-based decomposition of the system dynamics in (33), information about $\dot{x}_k$ is, however, required.

In this example, we assume that we have true measurements of $\dot{x}_k$, given by the black line in Figures S15(b) and S16(b). A naive approach for approximating $\dot{x}_k$ is to take a finite difference of sequential x_k measurements. The approximation is seen by the red line in Figures S15(b) and S16(b). As expected, when taking numerical derivatives of dynamic system data, noise in the x_k data is amplified, leading to large oscillations and errors in approximating the true value of $\dot{x}_k$.

An alternative is to fit a smooth function to approximate x_k and differentiate this signal. In this example, we will use the smoothing spline approximation captured by the MATLAB function `spaps` and based on [S7]. Approximations of the sequence x_k using various tolerances (a smaller tolerance requires less error in the approximation at the expense of smoothness) can be seen in Figures S15(a) and S16(a), with their derivatives shown in Figures S16(b) and S16(b). We see that using a tolerance of $1e-6$ (blue) leads to a spline that captures the sequence x_k accurately and also captures some of the undesirable oscillations in the derivative. In contrast, using tolerances of $1e-4$ (green) and $5e-4$ (magenta) leads a to worse approximation of the sequence x_k but smooths out the signal such that the derivative does not feature the same oscillations.

To highlight the importance of smoothing a state trajectory in an effort to more accurately capture the state derivative trajectory, consider the following regression problem:

$$\min_{\hat{f} \in \mathcal{H}} \sum_k \left| \dot{x}_k - \hat{f}(x_k, u_k) \right|^2 \tag{S18}$$

where $\hat{f}$ is a model from a class of models denoted by $\mathcal{H}$. We note that we have dropped any prior model knowledge from this problem for the sake of simplicity. As we do not have direct access to the measurements $\dot{x}_k$, we must use an approximate value, which we denote by $\dot{x}_k^f$, yielding the regression problem

$$\min_{\hat{f} \in \mathcal{H}} \sum_k \left| \dot{x}_k^f - \hat{f}(x_k, u_k) \right|^2. \tag{S19}$$

We may rewrite the cost function in (S18) as

$$\left| \dot{x}_k - \dot{x}_k^f + \dot{x}_k^f - \hat{f}(x_k, u_k) \right|^2. \tag{S20}$$

Solving the regression problem in (S19) to a high degree of accuracy such that $\hat{f}(x_k, u_k) \approx \dot{x}_k^f$, implies with (S20) that the cost function in (S18) attains an approximate value of

$$\sum_k \left| \dot{x}_k - \dot{x}_k^f \right|^2. \tag{S21}$$

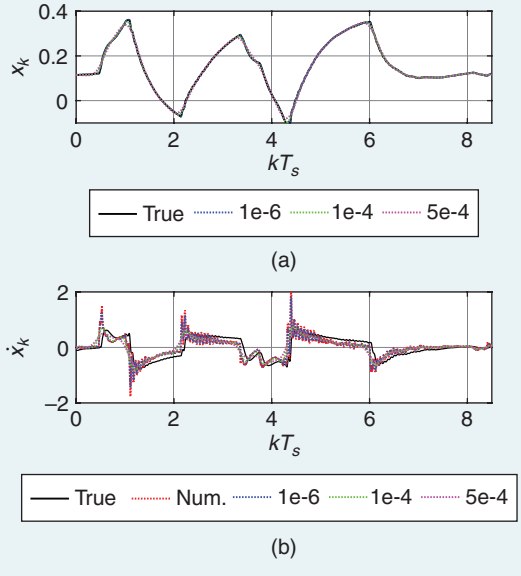


FIGURE S15 Real-world system state and state derivative trajectory data. (a) The state trajectory and smoothing splines using various accuracy tolerances. (b) The state derivative trajectory, numerical derivatives of raw state trajectory, and derivatives of state trajectory smoothing splines. Num.: numerical.

While this article focuses on learning the dynamics model (31), such a dataset could be used in other ways in the safety filter design process, such as learning control invariant sets [106], [107], [108].

Model Uncertainty Decomposition

System modeling by domain experts using physical principles is typically the first step of safety filter design and yields an imperfect nominal model f, as in (1). Using

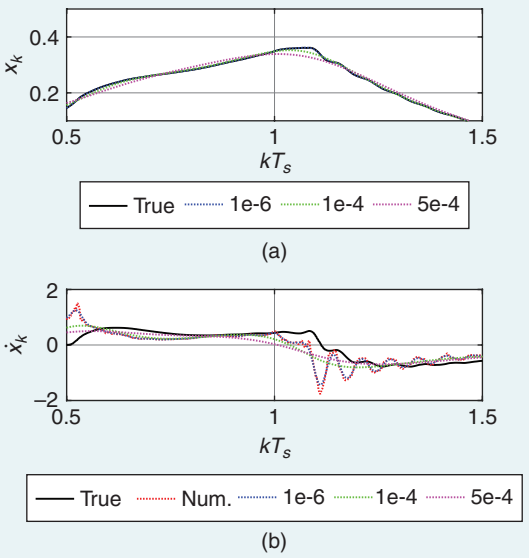


FIGURE S16 An enhanced view of real-world system state and state derivative trajectory data. (a) The spline with a tolerance of 1e-6 (blue) accurately captures fine features in the true state trajectory (black), while the splines with tolerances of 1e-4 (green) and 5e-4 (magenta) capture the general trend of the true state trajectory while ignoring fine features. (b) The numerical derivative of the true state trajectory (red) and the derivative of the spline with a tolerance of 1e-6 (blue) display large oscillations and fail to accurately track the true state derivative trajectory (black). The derivatives of the smoother splines using tolerances of 1e-4 (green) and 5e-4 (magenta) do not display these oscillations and follow the general trend of the true state derivative trajectory.

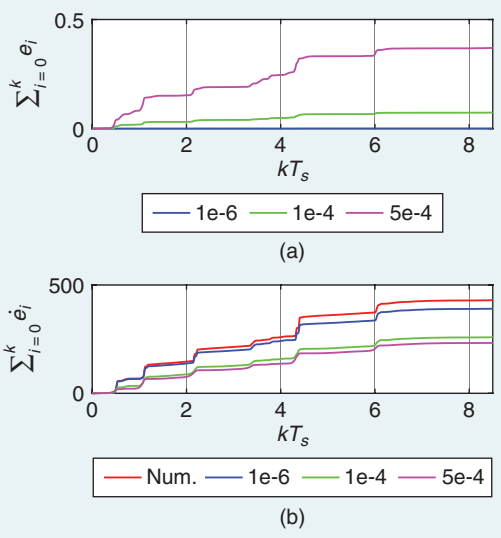


FIGURE S17 The cumulative error of smoothing splines. (a) The cumulative error in the smoothing splines used to approximate the sequence of state measurements x_k. (b) The cumulative error in the approximation of the state derivatives $\dot{x}_k$. The numerical differentiation of the state measurement signal x_k (red) incurs the highest loss, while the lowest tolerance spline (blue) accrues a similar error. The splines that prioritize the smoothness overapproximation accuracy of x_k (green, magenta) have derivatives that accrue less error.

Thus, our ability to minimize the cost of the ideal regression problem in (S18) is limited by how accurately $\dot{x}_k^f$ captures $\dot{x}_k$.

Let us denote x_k^f as the value of the smoothing spline at time kT_s and $\dot{x}_k^f$ as the value of the smoothing spline derivative at time kT_s. Let us also denote the errors $e_k = |x_k - x_k^f|^2$ and $\dot{e}_k = |\dot{x}_k - \dot{x}_k^f|^2$. We can see the cumulative sums of these errors for the various smoothing splines in Figure S17. In Figure S17(a), we observe that the spline using a tolerance of $1e-6$ (blue) approximates the true signal with low error, while the error increases for $1e-4$ (green) and $5e-4$ (magenta). In contrast, in Figure S17(b), we observe that the numerical derivative (red) and derivative of the spline using a tolerance of $1e-6$ (blue) accrue nearly double the error of that achieved by the derivative of the spline using tolerances of $1e-4$ (green) and $5e-4$ (magenta). It is this error that appears in (S21) and thus determines the accuracy of our learning algorithm. Hence, we see that there is a balance that must be met when smoothing real-world data to make them well conditioned for learning and that it may be desirable to accrue more error in our smooth models in an effort to reduce error in the derivative approximation.

This example further highlights an important aspect when designing data-driven safety filters. In particular, in most real-world settings, it will be impossible to completely learn a system or make the function e^l in (34) uniformly equal to zero. This is due not only to filtering of the data removing content from the signal but also due to the fact that data collected by sampling a system cannot accurately capture high-frequency content. Therefore, it is necessary to develop data-driven safety filters that are robust to learning errors, as we explore in this article.

REFERENCE

[S7] C. Reinsch, "Smoothing by spline functions," *Numer. Math.*, vol. 10, pp. 177–183, Oct. 1967, doi: 10.1007/BF02162161.

this model, we can rewrite the actual system dynamics (31) as

$$\dot{x}(t) = f(x(t),u(t)) + \underbrace{f_{\text{true}}(x(t),u(t)) - f(x(t),u(t))}_{=e^n(x(t),u(t))} \tag{33}$$

where the function $e^n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ captures all errors between the system model and the actual system. While assumptions on the uncertainty of the system may be used to construct a state- and input-dependent set $\mathcal{E}^n(x,u)$ such that $e^n(x,u) \in \mathcal{E}^n(x,u)$ for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, this set often significantly overapproximates the model error, yielding robust designs that are excessively conservative. Data-driven techniques tackle this challenge by reducing the model error e^n to a smaller learning error $e^l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ using a learning-based correction term $f^l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, that is,

$$\begin{aligned}\dot{x}(t) = {} & f(x(t),u(t)) + f^l(x(t),u(t)) \\ & + \underbrace{e^n(x(t),u(t)) - f^l(x(t),u(t))}_{=e^l(x(t),u(t))}.\end{aligned} \tag{34}$$

Learning f^l from the data (32) to mitigate e^l can be posed as a classical regression problem, which can be divided into parametric approaches with a fixed number of parameters independent of the number of measurements n_D and nonparametric approaches that have a variable number of parameters that grows with n_D. Parametric approaches are particularly suitable when the structure f_{true} in (31) is well understood and when fast predictions are required. In contrast, nonparametric approaches can compensate for both parametric and structural uncertainty but can be computationally more expensive to use for prediction. In addition to reducing the model error e^n, some learning techniques bound the residual learning error e^l with a state- and input-dependent set such that $e^l(x,u) \in \mathcal{E}^l(x,u)$ for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$. The structure of such a set yields a collection of distinct data-driven safety filter approaches as follows [22], [65], [111, Sec. II].

Deterministic Models

The first class of approaches considers the integration of deterministic data-driven models into safety filter design, without any specific quantification of the residual learning error e^l. Such approaches can provide good predictive performance and are commonly employed in practical applications. Examples include parametric models with simple least-squares regression or (recurrent) artificial neural networks and nonparametric techniques based on k-nearest-neighbors techniques [139]. Since these approaches typically do not provide an explicit bound on the residual learning error, safety according to (2) is practically achieved using tightened constraints of the form $\alpha\mathcal{X}, \alpha\mathcal{U}$ with $\alpha \in (0,1)$ in safety filter design. The resulting safety margin $(1-\alpha)$ is then hand-tuned to achieve constraint satisfaction. Examples include the use of deterministic models with CBFs [23], [52], [55].

Robust Models

The second class of approaches directly incorporates an explicit bound on the residual learning error e^l into the safety filter design, yielding robust safety filters. As previously noted, certain data-driven models bound the residual learning error through a state- and input-dependent set such that $e^l(x,u) \in \mathcal{E}^l(x,u)$ for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$. Safety filter design is done such that the system in (34) is safe for all possible residual learning error values in the set $\mathcal{E}^l(x,u)$. Error quantification for parametric methods often uses regularity properties of a class of parametric learning models, such as the use of spectral normalization and Lipschitz constants with recurrent neural networks [141], [142]. Nonparametric methods often use assumptions on the actual dynamics of the system (such as Lipschitz continuity) in conjunction with data to synthesize robust safety filters [143], [22, Sec. 3.1.2]. Such approaches have been taken using HJ reachability through a differential game formulation [144], using CBFs through robust optimization [56], [145], and using PSFs by determining an appropriate constraint-tightening mechanism [61].

Probabilistic Models

The preceding robust approaches guarantee the safety of a system, but they can often be unnecessarily conservative because they must capture all possible residual learning errors. Moreover, they tend to neglect the fact that the measurements composing D are noisy, and the resulting guarantees on learning accuracy are inherently probabilistic. The third class of approaches uses distributional information about the residual learning error in the safety filter design process, permitting practical designs that can balance the need for safety with strong performance. The corresponding data-driven models typically provide a data-driven description of the residual learning error e^l in the form of a probability distribution, $p(e^l \mid D)$. An overview of parametric and nonparametric probabilistic regression techniques often used in control can be found in [22], [65], [111, Section II], and references therein. A common learning technique to estimate the model error e^n with a function f^l and error set $\mathcal{E}^l(x,u)$ constructed from data is based on Gaussian process (GP) regression [64], [146], [147], [148], explored in "Probabilistic Nonparametric Model: Gaussian Process Regression."

Though it may be possible to construct probabilistic descriptions of structural uncertainties, parametric uncertainties, and external disturbances, it can be challenging to translate these descriptions into a safety filter formulation. A common simplification is to consider overall safety guarantees from a probabilistic perspective by considering *robustness at a certain probability level* [22, Sec. 3.2], [63],

[64], [148], [149], [150], [151]. To this end, we construct a state- and input-dependent uncertainty set $\mathcal{E}^l(x,u)$ based on available data D (32) similar to the robust case, which is, however, valid only in probability such that

$$\Pr\Big(\underbrace{e^l(x,u) \in \mathcal{E}^l(x,u) \text{ for all } x \in \mathcal{X} \text{ and } u \in \mathcal{U}}_{\star}\Big) \geq p_s \quad (35)$$

at some desired probability level p_s. We note that compared to the robust approach, we do not require $e^l(x,u) \in \mathcal{E}^l(x,u)$ with certainty but rather only at the specified probability level p_s. In practice, this can eliminate the need to address extremely unlikely scenarios that lead to conservative behavior of robust approaches. Recalling the safety specification given in (2), any robust design that is safe for all possible residual learning error values in $\mathcal{E}^l(x,u)$ yields that $\Pr((2)\,|\,\star) = 1$, where $\star$ is defined in (35), implying

$$\Pr((2)) \geq \Pr((2),\star) = \Pr((2)\,|\,\star)\Pr(\star) \geq p_s \quad (36)$$

with $\Pr((2),\star)$ denoting the joint probability of the random event that (2) and $\star$ both happen and $\Pr((2)\,|\,\star)$ denoting the probability of (2) happening conditioned on $\star$ happening. From relation (36), it follows

$$\Pr(x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U} \text{ for all } t \in \mathbb{R}_{\geq 0}) \geq p_s. \quad (37)$$

The relation between the probabilistic error bound (35) and constraint satisfaction in probability (37) provides an intuitive way for trading off safety and permissiveness since lower probability levels p_s typically lead to a smaller learning error bound $\mathcal{E}^l(x,u)$ and less conservative robust safety filter designs. The type of probabilistic condition in (36) has been utilized in the design of probabilistic safety filters through HJ reachability [21], CBFs [54], [58], [151], and PSFs [60], [63].

In theory, any of the advanced safety filter techniques presented can be combined with the preceding model classes. In practice, some safety filter techniques naturally lend themselves to being used with a specific type of model class, as we highlight in the following sections. We also note that systems are often subject to unmodeled external disturbances caused by environmental perturbations, such as wind acting on an airplane or changing road friction coefficients for a ground vehicle. In contrast to uncertainty in the model, these disturbances often do not have an underlying structure that can be discovered by data. Rather, data are often used to quantify the magnitude of disturbances, which is then used for a robust design. For simplicity, the following formulation is presented in the absence of such disturbances, but we note that the following methods for developing safety filters that are robust to learning error can be used for (and in fact, originated from) robustness to disturbances.

Data-Driven HJ Reachability

Due to the inherent separation of safety from performance in HJ reachability, reachability-based safety filter designs can be used together with any type of controller emitting the desired control input signal. In particular, reachability-based safety filters are suitable for filtering learning-enabled systems like autonomous vehicles throughout the process of training the learning-based components in the system. We describe such an HJ reachability-based safety framework for uncertain systems as proposed in [21]. Several extensions and variants of this framework have been proposed to demonstrate the applicability of the framework to high-dimensional systems [50], [152]. We highlight simulation and experimental results utilizing this framework in "Reachability-Based Safe Learning Framework: Experimental Results" to demonstrate the effectiveness of reachability-based frameworks in real-world applications.

HJ Reachability With Learning Error

First, we describe the HJ reachability analysis that is extended to account for learning errors by using a differential game-based formulation [153], resulting in a characterization of the maximal control invariant set and an associated optimal safe policy that are robust to bounded learning error. For the sake of simplified exposition, consider a setting where the model error e^n in (33) does not depend on the input u. Consequently, a learning model $f^l : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, a learning error $e^l : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and a pointwise set $\mathcal{E}^l(x) \subset \mathbb{R}^{n_x}$ such that $e^l(x) \in \mathcal{E}^l(x)$ for all $x \in \mathcal{X}$ can be considered. As the value of the learning error is unknown, it is desirable for a safety filter design to be robust to all possible learning errors permitted by the pointwise set $\mathcal{E}^l(x)$. To this end, consider the dynamics

$$\dot{x}(t) = f(x(t),u(t)) + f^l(x(t)) + d(t) \quad t \in \mathbb{R}_{\geq 0} \quad (38)$$

where $d(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_x})$ is a disturbance signal modeling the possible effects of the unknown learning error $e^l(x(t))$ on the dynamics. To construct the maximal control invariant set contained in $\mathcal{X}$ in this setting, we consider a cost functional $J_d : \mathbb{R}^{n_x} \times \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \times \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_x}) \to \mathbb{R}$ similar to (13)

$$J_d(x_0, u(\cdot), d(\cdot)) = \inf_{t \in \mathbb{R}_{\geq 0}} -s_{\mathcal{X}}(x(t)) \quad (39)$$

where $x(\cdot)$ is the solution to (38) with initial condition x_0, input signal $u(\cdot)$, and disturbance signal $d(\cdot)$.

The set of nonanticipative disturbance strategies, denoted by $\mathcal{D}$, is defined as the set of all mappings $\delta : \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \to \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathbb{R}^{n_x})$ that satisfy

$$\delta[u](t) \in \mathcal{E}^l(x(t)) \text{ for all } u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \text{ and } t \in \mathbb{R}_{\geq 0} \quad (40)$$

and

$$\begin{aligned}&\delta[u_1](t) = \delta[u_2](t) \text{ for almost all } t \in \mathbb{R}_{\geq 0}\\ &\text{for all } u_1(\cdot), u_2(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \text{ s.t.}\\ &u_1(t) = u_2(t) \text{ for almost all } t \in \mathbb{R}_{\geq 0}.\end{aligned} \tag{41}$$

Intuitively, the disturbance signal $d(\cdot)$ resulting from the strategy δ reacting to the control signal $u(\cdot)$, that is, $d(t) = \delta[u](t)$, should satisfy the learning error bound $d(t) \in \mathcal{E}^l(x(t))$ for all time, and the nonanticipative restriction prohibits $d(\cdot)$ from depending on the future information of the control signal $u(\cdot)$ [153].

A value function $V_d : \mathbb{R}^{n_x} \to \mathbb{R}$ that accounts for disturbances can be constructed similarly to (14) through a zero-sum differential game

$$V_d(x_0) = \inf_{\delta[u] \in \mathcal{D}} \sup_{u(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U})} J_d(x_0, u(\cdot), \delta[u](\cdot)). \tag{42}$$

The computation of V_d is done by solving the HJ-PDE [42]

$$0 = \min\left\{-s_{\mathcal{X}}(x) - V_d(x), \max_{u \in \mathcal{U}} \min_{d \in \mathcal{E}^l(x)} \nabla V_d(x)(f(x,u) + d)\right\} \tag{43}$$

which has a viscosity solution that characterizes V_d. Similar to Theorem 2, the value function V_d (42) can be used to characterize control invariant sets in $\mathcal{X}$ that are robust to learning errors. More precisely, for any $\epsilon \in \mathbb{R}_{\geq 0}$, the set $\mathcal{S}_\epsilon = \{x \in \mathcal{X} \mid V_d(x) \geq \epsilon\}$ is a control invariant set that is robust to learning errors, and $\mathcal{S}_0$ characterizes the maximal control invariant set contained in $\mathcal{X}$ that is robust to learning errors [21]. Finally, the robust optimal safe policy $\kappa^*_{V_d} : \mathbb{R}^{n_x} \to \mathcal{U}$ can be constructed as

$$\kappa^*_{V_d}(x) = \operatorname*{argmax}_{u \in \mathcal{U}} \min_{d \in \mathcal{E}^l(x)} \nabla V_d(x)(f(x,u) + d) \tag{44}$$

which ensures that the set $\mathcal{S}_\epsilon$ is forward invariant in the presence of learning errors. Compared to the optimal safe policy defined in (17), this controller introduces the term $\min_{d \in \mathcal{E}^l(x)}$, which considers the worst-case effect of the learning error d at the current state when synthesizing the safe control input.

Reachability-Based Safe Learning Framework

The safe set $\mathcal{S}_\epsilon$ and the safe policy $\kappa^*_{V_d}(x)$ in the previous formulation can be overly conservative when the set $\mathcal{E}^l(x)$

Probabilistic Nonparametric Model: Gaussian Process Regression

Gaussian process (GP) regression is a nonparametric learning method that can serve as an effective framework for learning the model correction term f^l in (34). A GP is a random process in which every finite sample has a joint Gaussian distribution. Taking a Bayesian approach with an assumption that the function we want to learn is sampled from a GP, we are able to obtain the regression model of the function and its uncertainty bound at any desired probability level by inferring the posterior distribution of the GP with respect to the data. In this sidebar, we apply this approach for each dimension of the uncertainty e^n_i to obtain regression models f^l_i, and uncertainty bounds, $\mathcal{E}^l_i(x,u)$, where $i = 1, \ldots, n_x$ (see Figure S18).

We first assume prior knowledge of e^n_i in the form of a mean function $\mu_i : \mathbb{R}^{n_x+n_u} \to \mathbb{R}$ and a covariance kernel $k_i(\cdot,\cdot) : \mathbb{R}^{n_x+n_u} \times \mathbb{R}^{n_x+n_u} \to \mathbb{R}_{\geq 0}$ such that the GP prior of $e^n_i(\cdot)$ is given by

$$e^n_i(\cdot) \sim \mathcal{GP}(\mu_i(\cdot), k_i(\cdot,\cdot)). \tag{S22}$$

We assume that we have the measurements $y_{k,i} = e^n_i(x_k, u_k) + \epsilon_{i,k}$, given a state x_k and input u_k pair sample in the dataset D, where the measurement noise $\epsilon_{i,k}$ is independently identically distributed and sampled from a normal distribution, $\mathcal{N}(0, \sigma_i)$. The choice of the class of the prior mean and covariance function is typically based on prior knowledge of the system. When the prior knowledge is already incorporated in the nominal term f in (34), the prior mean is typically set to $\mu_i = 0$. For the covariance function, a frequent choice is the squared exponential kernel, given by

$$k_i(z_p, z_q) = \sigma^2_{f,i} \exp\left(-\frac{1}{2}(z_p - z_q)^\top L_i^{-1}(z_p - z_q)\right) \tag{S23}$$

with L_i a positive-definite diagonal length-scale matrix and $\sigma^2_{f,i}$ the signal variance. z denotes the state and input pair, $[x^\top u^\top]^\top$. The parameters $\sigma_{f,i}$, and L_i are called *hyperparameters*, and their appropriate values can be selected automatically by inferring from the measurements (32) [160, Ch. 5]. See [160, Sec. 4.2] for an overview of the effect of these values on the GP prior distribution. Let

$$D_i = (Z = [z_0, .., z_{n_D}]^\top, \;\; Y_i = [y_{0,i}, .., y_{n_D,i}]^\top) \tag{S24}$$

denote the data for each state dimension $i = 1, .., n_x$. Based on the prior distribution (S22), we can state the joint distribution

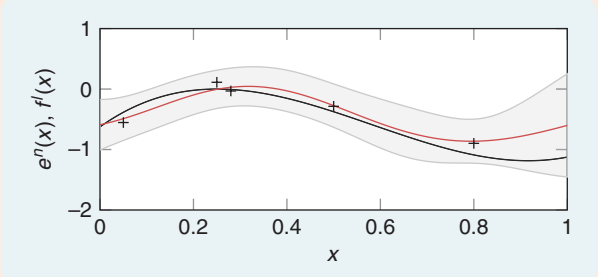


FIGURE S18 Noisy measurements (crosses) of the model error $e^n(x)$ (black), with a Gaussian process regression mean estimate $f^l(x)$ (red) using a squared exponential kernel function, with the 2-σ confidence bound shown in gray.

is overapproximated. Moreover, underapproximating the set $\mathcal{E}^l(x)$ in the construction of the value function V_d can lead to the failure of the system to remain safe in the presence of learning errors that exceed the underestimated error bound. This motivates incorporating data-driven techniques that accurately characterize $\mathcal{E}^l(x)$ into the differential game formulation.

The framework in [21] employs GP regression for this purpose. However, it is worth noting that any robust or probabilistic data-driven models that provide an accurate characterization of model uncertainty can function well in this reachability framework. The constructed GP regression model f^l approximates the model error e^n with its mean prediction and captures the residual learning error e^l with its posterior variance. For more details, see "Probabilistic Nonparametric Model: Gaussian Process Regression."

While a conservative estimate of the possible learning errors $\mathcal{E}^l(x)$ may satisfy (35) with a high probability ps, reducing the conservativeness of an estimate of the possible learning errors can permit better performance. The following result on the differential game form of HJ reachability establishes a property of HJ reachability-based safety filter designs relating two estimates of possible learning errors [21, Proposition 5].

Theorem 5

Consider two learning models, $f_1^l, f_2^l : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and corresponding pointwise error sets $\mathcal{E}_1^l$, $\mathcal{E}_2^l$. Suppose that for all $x \in \mathbb{R}^{n_x}$, we have $f_2^l(x) \oplus \mathcal{E}_2^l(x) \subseteq f_1^l(x) \oplus \mathcal{E}_1^l(x)$. If a set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is control invariant for (38) for all disturbance signals satisfying $d(t) \in \mathcal{E}_1^l(x(t))$ for all $t \in \mathbb{R}_{\geq 0}$, then $\mathcal{S}$ is also a control invariant set for (38) for all disturbance signals satisfying $d(t) \in \mathcal{E}_2^l(x(t))$ for all $t \in \mathbb{R}_{\geq 0}$.

In plain words, Theorem 5 states that a control invariant set for the dynamics (38) that is robust against larger error bounds is also robust against the smaller subset error bounds. Thus, alleviating the conservativeness of the error bound not only can permit better performance but also still preserves safety. This serves as the central principle underlying the safe learning framework.

Based on this principle, the safe learning framework conducts the following steps. When it is initiated, the learning model has little to no data, and the estimate of possible learning errors $\mathcal{E}^l(x)$ is typically quite large. The resulting

of Y_i together with the observation distribution y_i at a desired prediction point $z = (x, u)$ as

$$\begin{bmatrix} Y_i \\ y_i \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mu_i(Z) \\ \mu_i(z) \end{bmatrix}, \begin{bmatrix} K_i(Z,Z) + I\sigma_{s,i} & K_i(Z,z) \\ K_i(z,Z) & K_i(z,z) \end{bmatrix}\right) \tag{S25}$$

with Gram matrix $[K_i(Z,Z)]_{pq} = k_i(z_p, z_q), [K_i(Z,z)]_p = k_i(z_p, z)$, $K_i(z,Z) = K_i(Z,z)^\top$, and $K_i(z,z) = k_i(z,z)$. The conditional distribution of y_i is then obtained using Gaussian distribution identities [160] as

$$p(y_i^z \mid Y_i) = \mathcal{N}(\mu_{i|D_i}(z), \sigma_{i|D_i}(z)) \tag{S26}$$

with

$$\begin{aligned} \mu_{i|D_i}(z) &= \mu_i(z) + K_i(z,Z)(K_i(Z,Z) + \sigma_{f,i}I)^{-1}(Y_i - \mu_i(Z)) \\ \sigma_{i|D_i}(z) &= K_i(z,z) - K_i(z,Z)(K_i(Z,Z) + \sigma_{f,i}I)^{-1}K_i(Z,z). \end{aligned} \tag{S27}$$

The overall GP regression model of e^n is then obtained by stacking (S27), that is, $e^n(x,u) \sim \mathcal{N}(\mu_D(x,u), \Sigma_D(x,u))$ with $\mu_D(x,u) = [\mu_{1|D_1}(z), .., \mu_{n_x|D_{n_x}}(z)]$, $\Sigma_D(x,u) = \text{diag}([\sigma_{1|D_1}(z), .., \sigma_{n_x|D_{n_x}}(z)])$, where $z = (x,u)$. The mean estimate $\mu_D(x,u)$ is used to represent the regression model $f^l(x,u)$, while the variance $\Sigma_D(x,u)$ can be used to construct the desired uncertainty set (35) in the form of a hypercube

$$\mathcal{E}_i^l(x,u) = [-\beta\sigma_{i|D_i}(z), \beta\sigma_{i|D_i}(z)] \tag{S28}$$

for each dimension i. The common choice of the quantile constant β for the ease of implementation is a constant value $\beta = \sqrt{2}\,\text{erf}^{-1}(p_s^{1/n_x})$ [21]. While this choice implies that $\Pr(e^l(x,u) \in \mathcal{E}^l(x,u)) \geq p_s$ for each x and u, it does not necessarily provide the desired bound on the entire function, as required by (35). The computation of β that strictly satisfies the condition (35) is further discussed in [148] and [21]. This usually requires some prior knowledge of the uncertainty terms, for instance, their Lipschitz constants, and the resulting value can sometimes be too overly conservative to be used in practice.

One important property of the set-valued map (S28) is its Lipschitz continuity under the Hausdorff metric for any Lipschitz continuous prior mean and kernel functions $\mu_i(\cdot)$ and $k_i(\cdot,\cdot)$ [21, Proposition 10], including the important special cases of zero prior mean and squared exponential kernels [160]. The Hausdorff metric between any two sets A and B in a metric space (M, d_M) is defined as $d_H(A,B) = \max\{\sup_{a\in A}\inf_{b\in B} d_M(a,b), \sup_{b\in B}\inf_{a\in A} d_M(a,b)\}$.

Finally, the computational complexity of evaluating the mean and variance in (S27) is $O(n_D n_u n_x^2)$ and $O(n_D^2 n_u n_x^2)$, respectively, which scales unfavorably with the number of data points. This can be problematic for the application of GP regression to large datasets for real-time safety filter applications. Various approximation techniques have been proposed to resolve this problem (see [160, Ch. 8] and [S8] for an overview).

REFERENCE

[S8] J. Quinonero-Candela, C. E. Rasmussen, and C. Williams, "Approximation methods for gaussian process regression," in *Large-Scale Kernel Machines*. Cambridge, MA, USA: MIT Press, 2007, pp. 203–223.

control invariant set constructed through HJ reachability that is robust to these learning errors is conservative and limits the performance of the system. As the learning proceeds, new data are incorporated into the learning model, and the control invariant set that requires robustness to smaller error estimates $\mathcal{E}^l(x)$ is updated accordingly by recomputing the value function $V_d(x)$ based on the updated learning model. The value function that was being used previously can be recycled to make the computation more efficient, for instance, by using it as a warm-starting solution [50] or by updating only locally for the region where the learning model is updated [154]. Ideally, learning the smallest set of possible errors results in a control invariant set that is the maximal control invariant set in $\mathcal{X}$ that can be made robust to the presence of minimal learning error. "Reachability-Based Safe Learning Framework: Experimental Results" displays this safe learning framework working in practice.

Data-Driven CBFs

The use of data-driven techniques with CBFs has been an active area of research interest, with a wide range of approaches, including using models that are deterministic [23], [52], [53], [55], [155], robust [56], and probabilistic [51], [54], [57], [58], [59], [93], [150], [151] (see the examples in "Data-Driven Control Barrier Function Safety Filter Applications"). An underlying robustness property of CBF-based safety filter design known as *input-to-state safety* (*ISSf*) [44], [155] manifests in each of these approaches. We now present this property in a general context.

Consider the control-affine model (21) with a learning model f^l and a corresponding learning error e^l given by

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + f^l(x(t), u(t)) + e^l(x(t), u(t)) \quad t \in \mathbb{R}_{\geq 0}. \tag{45}$$

Let $\mathcal{S}$ be defined as the zero-superlevel set of a continuously differentiable function $h_{\mathcal{S}}: \mathbb{R}^{n_x} \to \mathbb{R}$, and suppose that using the learned model f^l, we design a safety filter $\kappa: \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{U}$ such that there exists an $\alpha \in \mathcal{K}^e$ satisfying

$$\nabla h_{\mathcal{S}}(x)\left(f(x) + g(x)\kappa(x, u) + f^l(x, \kappa(x, u))\right) \geq -\alpha(h_{\mathcal{S}}(x)) \tag{46}$$

Reachability-Based Safe Learning Framework: Experimental Results

The Hamilton-Jacobi (HJ) reachability-based safe learning framework proposed in [21] has been demonstrated on a quadrotor subjected to unknown dynamics due to wind effects. The quadrotor attempts to track a reference trajectory using either a linear-quadratic regulator or a tracking policy learned online. The HJ reachability safety filter is utilized to prevent the quadrotor from colliding with its environment. However, a safety filter that is overly conservative may hinder

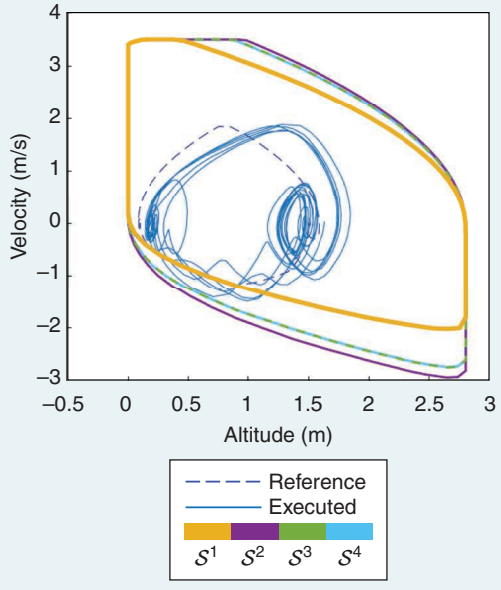


FIGURE S19 The quadrotor altitude HJ reachability safe sets being updated online through learning. The sets progress from $\mathcal{S}^1$ to $\mathcal{S}^4$ as the system gathers data, successively improving the learned dynamics model. (Source: [21], ©2019 IEEE.)

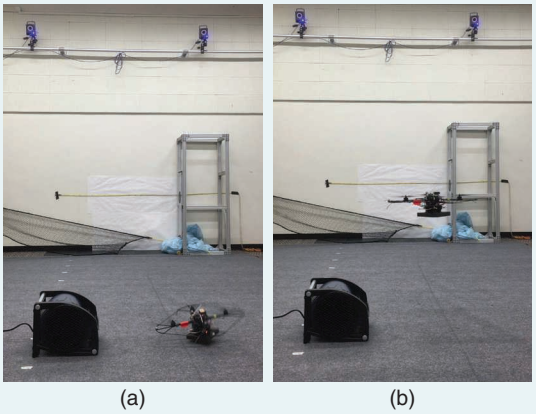


FIGURE S20 A quadrotor learning a vertical flight policy while avoiding collisions with the ground. When the fan is turned on, the system experiences unknown dynamics that have not appeared in previous data, which can lead to a ground collision using the previously learned policy. An online validation method detects that the previously learned model fails to describe the new unknown dynamics and utilizes a safe controller that avoids regions of the state space (close to the fan) where the new unknown dynamics are present. (a) Without online guarantee validation. (b) With online guarantee validation. (Source: [21], ©2019 IEEE.)

for all $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$. This safety filter is designed to meet the original safety specification encoded by the BF $h_{\mathcal{S}}$ and the function α, but does so by incorporating the learned model f^l. Let us further suppose that there exists an $\bar{e} \in \mathbb{R}_{\geq 0}$ such that

$$\left| \nabla h_{\mathcal{S}}(x) e^l(x, \kappa(x, u)) \right| \leq \bar{e} \tag{47}$$

for all $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$. This inequality implies that the effect of the residual learning error on the time derivative of the BF $h_{\mathcal{S}}$ is bounded by a constant $\bar{e}$. Intuitively, this bound can be made smaller through more accurate learning models.

Combining (46) and (47), we have that

$$\dot{h}_{\mathcal{S}}(x, u) \geq -\alpha(h_{\mathcal{S}}(x)) - \bar{e} \tag{48}$$

for all $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$. Noting that $\alpha \in \mathcal{K}^{\mathrm{e}}$ implies it has an inverse $\alpha^{-1} \in \mathcal{K}^{\mathrm{e}}$, we have the implication that

$$h_{\mathcal{S}}(x) \leq \alpha^{-1}(-\bar{e}) \Rightarrow \dot{h}_{\mathcal{S}}(x, u) \geq 0. \tag{49}$$

This preceding implication states that the time derivative of the BF $h_{\mathcal{S}}$ is nonnegative on the boundary of the $\alpha^{-1}(\bar{e})$-superlevel set of $h_{\mathcal{S}}$

$$\mathcal{S}_{\bar{e}} = \{x \in \mathbb{R}^{n_x} \mid h_{\mathcal{S}}(x) \geq \alpha^{-1}(-\bar{e})\} \tag{50}$$

and thus, we can conclude via Nagumo's theorem ($\nabla h_{\mathcal{S}}(x) \neq 0$ when $h_{\mathcal{S}}(x) < 0$ [156]) that $\mathcal{S}_{\bar{e}}$ is forward invariant, see Figure 6. This analysis highlights a fundamental robustness property of CBF-based safety filter designs since the set kept forward invariant does not increase dramatically with small amounts of residual learning error but rather scales proportionally. Moreover, this expansion can be controlled by reducing residual learning error through more data and better learning models that serve to reduce $\bar{e}$.

This notion of a safe set that scales with residual learning error is captured by the idea of ISSf [44]. We note that not only can ISSf describe the impact of model error in (33) (without introducing learning models), but it can enable a simplified design procedure [98], [156]. This property can allow one to utilize a margin built into a system's design to

not only the tracking performance but also the training of the learning-based policy by preventing it from adequate exploration. To reduce conservativeness, the safety filter must address the unknown dynamics by learning from the actual system data, revealing a balance between safety and learning that must be achieved.

Figure S19 shows a phase portrait of the vertical position and velocity coordinates of the quadrotor as it learns a tracking policy. The conservativeness of the safe set is reduced over time as the learning model improves, eventually allowing the learning-based policy to successfully perform the tracking task while avoiding collisions. In Figure S20, a strong wind is introduced near the ground, which the system has not encountered before. Reliance on the previously learned policy that is unaware of this disturbance leads to a deterioration of safety, as seen in Figure S20(a). However, when the accuracy of the learned model is validated online using data that capture the new unknown dynamics, the system is kept away from the region where the model is unreliable until a new model can be trained, thus leading to safety, as seen in Figure S20(b).

Finally, the experiment is extended to simulation with the quadrotor tracking a figure-eight reference trajectory in 3D space (Figure S21). While in the previous scenarios, the HJ safe set computation is done only for the vertical dynamics, to ensure safety constraints in the full 3D environment, the HJ safe set computation is done online for the 10 dimensional (10D) full quadrotor dynamics. The computation is facilitated by incorporating modern reachability computational techniques,

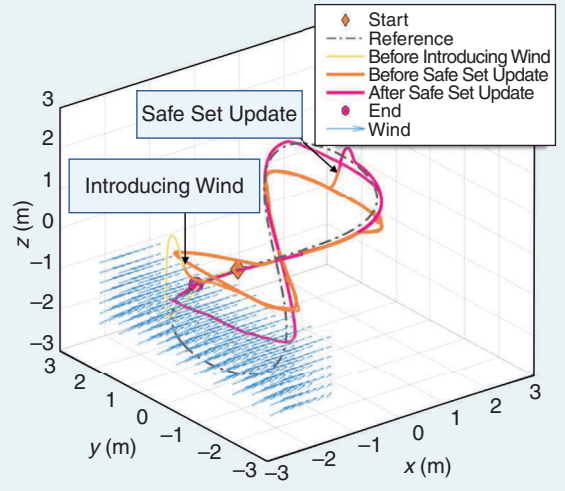


FIGURE S21 The trajectories of the quadrotor tracking a reference trajectory using a linear-quadratic regulator in 3D space. The quadrotor begins in yellow and then experiences a sudden change in wind (blue arrows). While the safe set is updated to account for the new unknown dynamics, online validation of the learned model prevents the trajectory from passing the uncertain wind area (orange trajectory) until the safe set update is complete (pink trajectory). (Source: [50], ©2021 IEEE.)

including state decomposition [76], warm starting [77], and adaptive gridding [50], which took an average of 206.6 s to update the safe set online.

Data-Driven Control Barrier Function Safety Filter Applications

While the application of control barrier function (CBF)-based safety filters does not require data-driven methods to deal with imperfect system models, there have been several applications where the incorporation of data has led to improvements in the safety of a system. In this sidebar, we highlight successful experimental implementations of data-driven

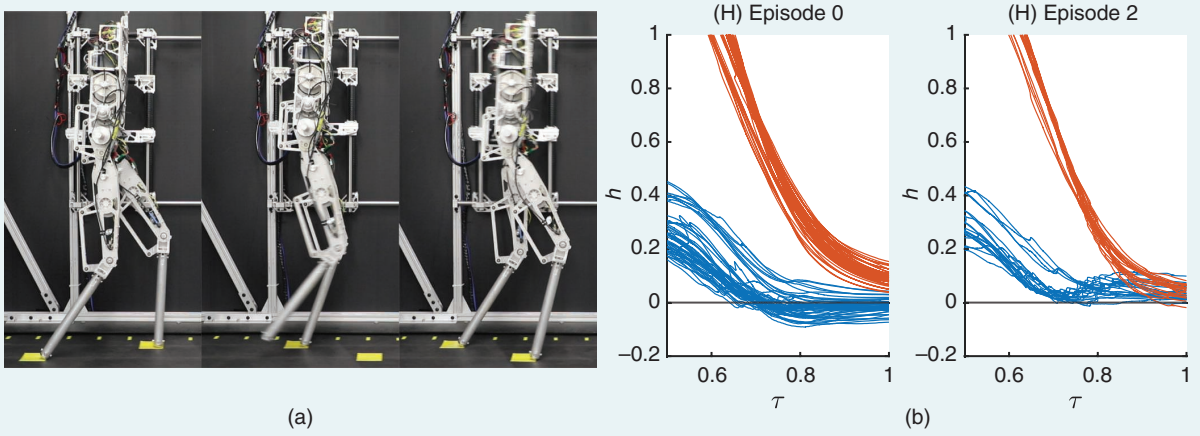
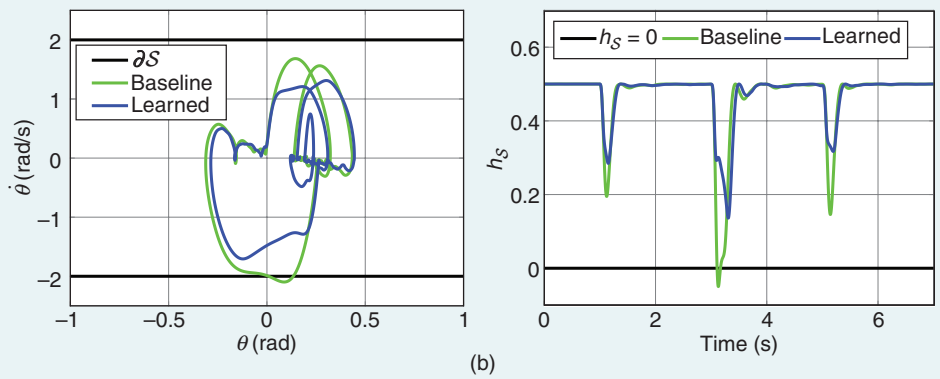


FIGURE S22 (a) and (b) Learning CBF time derivatives on the AMBER-3M bipedal robot. Walking robots often possess model uncertainty, making it difficult to satisfy precise foot placement constraints. By learning the impact of this model uncertainty on the dynamics of the CBF defining foot placement constraints, a CBF-based safety filter using learning models can be synthesized that reduces constraint violation. The two colored curves correspond to CBF values for constraints on each foot across multiple steps, with the constraint corresponding to the blue curves improving (remaining above zero) after incorporating learning models [55].

(a)



(b)

FIGURE S23 (a) and (b) Learning CBF time derivatives on a Segway robot. The baseline CBF-based safety filter (green curves) does not respect safety constraints on the pitch and pitch rate of the Segway due to errors between the system model and the physical system. By integrating data-driven learning models into the CBF-based safety filter, the safety of the system is achieved (blue curves). We note that although the baseline CBF-based safety filter does not respect safety constraints, the system remains close to the safe set, indicating input-to-state safe behavior with respect to model uncertainty inherent in CBF-based safety filters [23].

(Continued)

Data-Driven Control Barrier Function Safety Filter Applications (*Continued*)

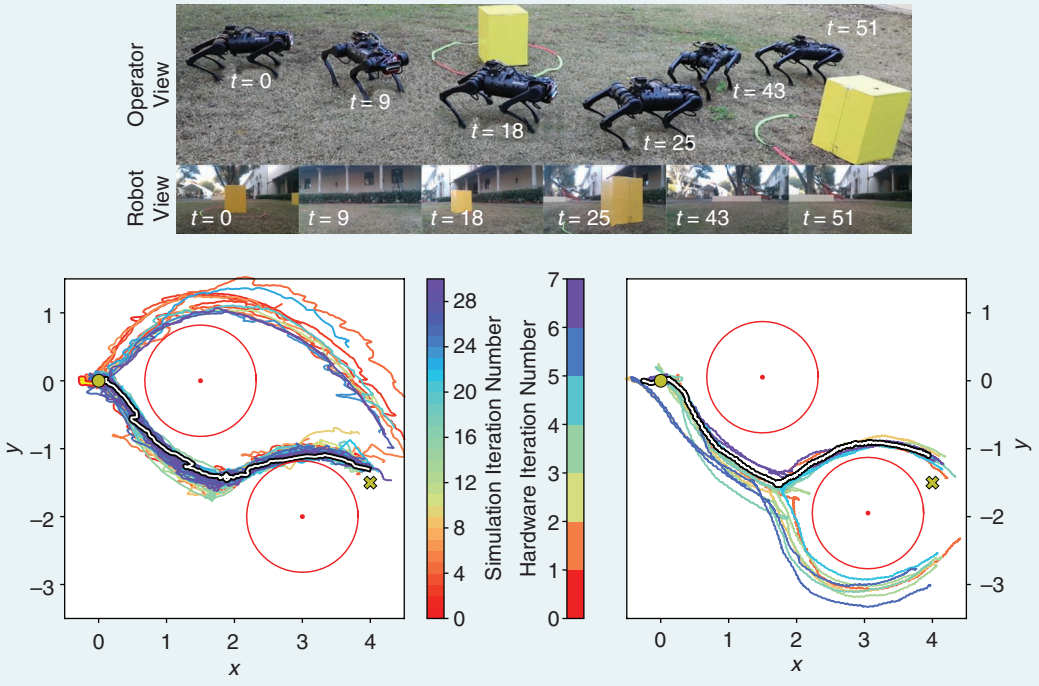


FIGURE S24 Preference-based learning for human-in-the-loop CBF safety filter tuning. Facing uncertainty, the design of CBF-based safety filters must balance robustness and performance. Preference-based learning can translate a designer's evaluation of closed-loop behavior into controller parameter updates that achieve this balance. The initial CBF-based safety filter design overestimates uncertainties and yields conservative behavior with the quadruped remaining stationary. Incorporating user preferences to modify safety filter parameters allows the quadruped to navigate safely navigate obstacles, thus balancing safety and performance [167].

CBF-based safety filters. In Figures S22 and S23, we see applications where learning models are used to mitigate the error between a system model and the physical system. In both examples, a baseline CBF-based safety filter given by (26) is modified with learning models, yielding safe behavior. Figure S24 shows an example of preference-based learning [166] being used to tune the parameters of a robust CBF-based safety filter. By iteratively incorporating designer preferences on closed-loop system behavior, a CBF-based safety filter that balances performance with safety can be synthesized. These results demonstrate the potential of data-driven CBF-based safety-critical control design methodologies.

simplify the design of a data-driven safety filter. In particular, if the safety requirement (that $x(t) \in \mathcal{X}$ for all $t \in \mathbb{R}_{\geq 0}$) is specified with some amount of margin such that it is practically acceptable if $s_{\mathcal{X}}(x(t)) < \epsilon$ for some $\epsilon \in \mathbb{R}_{>0}$ (where $s_{\mathcal{X}}$ is the signed-distance function for the set $\mathcal{X}$; see the "Definitions and Notation" section at the beginning of the article), then the ISSf property can be used to synthesize a controller that meets this practical safety requirement without complicating the controller design to explicitly address learning error.

Data-Driven PSFs

The close relation between the nominal PSF formulation (28) and common model predictive controllers using a terminal set [39] allows one to take advantage of existing advances in the field of robust [11, Sec. 3], [40, Sec. 7] and learning-based

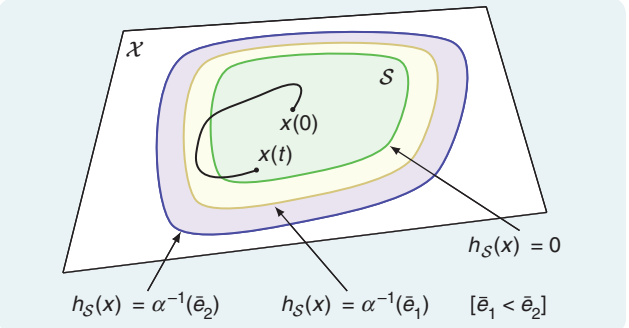


FIGURE 6 A schematic of input-to-state safety (ISSf). In the presence of a residual learning error, a controller that satisfies the CBF constraint using the learning model (46) may not render the set $\mathcal{S}$ forward invariant. Rather, a larger set that scales with the magnitude of the learning error is kept forward invariant, reflected by the two nested sets for the progressively larger learning error bounds $\bar{e}_1$ and $\bar{e}_2$.

MPC [22], [64], [111, Sec. 5] (see the examples in "Predictive Safety Filter Applications: Experimental Race Cars and Simulated Quadrotors"). As the focus in the case of PSFs is to provide formal guarantees regarding constraint satisfaction, most of the underlying mechanisms applied originate from robust MPC literature. Data-driven PSFs have been developed for linear robust (distributed) models [61], [62] and linear (distributed) stochastic models with unbounded process noise [63], [157, Remark 5]. The support of nonlinear system dynamics and exploration beyond available data has been enabled through leveraging probabilistic state- and input-dependent system models [60], [64]. While the precise details of each of these methods vary, they all operate using the idea that instead of directly working with the original safety specifications along predictions (28d), the constraints are enforced with an additional safety margin. This margin is designed to compensate for residual learning errors and disturbances in a closed loop without violating the original safety constraints of the system. The rigorous computation of these margins is at the core of robust-, stochastic-, and learning-based MPC methods. In the following, we focus on the computationally efficient technique for combining PSFs with robust and probabilistic learning models in [60].

Similar to the nominal PSF formulation and consistent with learning-based MPC literature [22], [111], we work with a discrete-time version of the learning-based model (34)

$$x(k+1) = \mathrm{f}(x(k), u(k)) + \mathrm{f}^l(x(k), u(k)) + \mathrm{e}^l(k) \tag{51}$$

where we use $\mathrm{e}^l(k)$ to denote $\mathrm{e}^l(x(k), u(k))$. The learning-based model in (51) and an uncertainty bound of the form (35) can be estimated using measurements of the form $y^k = x_{k+1} - x_k + \epsilon_k$ with ϵ_k independent and identically distributed noise, as exemplified in "Probabilistic Nonparametric Model: Gaussian Process Regression." The central idea of the following approach is to restrict backup trajectories $\{x_{i\,|\,k}\}$, $\{u_{i\,|\,k}\}$ to high-confidence subsets of the state and input space by imposing

$$\mathcal{E}^l(x_{i\,|\,k}, u_{i\,|\,k}) \subseteq \bar{\mathcal{E}}^l, \quad \text{for } i = 0, \ldots, N \tag{52}$$

along predictions, where $\bar{\mathcal{E}}^l \subset \mathbb{R}^{n_x}$ captures a tolerable amount of one-step prediction error. This mechanism causes trajectories to avoid regions with low model confidence due to sparse data coverage, as seen in Figure 7. We note that (52) can be reformulated as a set of inequality constraints in the case of GP regression or Bayesian linear regression and becomes a convex constraint in the case of linear features [149, Sec. 4.1], [60, Sec. 5.1].

While various existing robust predictive control techniques can be used to obtain robustness in probability (36), we focus on a constraint-tightening approach based on [158], [159]. The idea is to introduce increasing safety margins for all constraints along the prediction horizon, ensuring recursive feasibility and constraint satisfaction in a closed loop. In the case of polytopic state, input, terminal, and learning error constraints (52) of the form $\{x \in \mathbb{R}^n \,|\, Ax \leq \mathbb{1}^{n_A}\}$ with $A \in \mathbb{R}^{n_A \times n}$, the tightening of the constraint sets is

$$\bar{\mathcal{X}}_i = \{x \in \mathbb{R}^{n_x} \,|\, A^x x \leq (1-\epsilon_i)\mathbb{1}^{n_{Ax}}\} \tag{53a}$$

$$\bar{\mathcal{U}}_i = \{u \in \mathbb{R}^{n_u} \,|\, A^u u \leq (1-\epsilon_i)\mathbb{1}^{n_{Au}}\} \tag{53b}$$

$$\bar{\mathcal{E}}_i^l = \{x \in \mathbb{R}^{n_x} \,|\, A^{\mathcal{E}} x \leq (1-\epsilon_i)\mathbb{1}^{n_{A\mathcal{E}}}\} \tag{53c}$$

with $\mathbb{1}^n$ denoting the vector of ones of dimension n and with a monotonically increasing tightening sequence ϵ_i satisfying $\epsilon_0 = 0$ and $\epsilon_{i+1} > \epsilon_i$. Integrating the learning-based model (51) and the tightened constraints (53) into the PSF problem (28) yields

$$\min_{u_{i\,|\,k}} \left\| u_{\mathrm{des}}(k) - u_{0\,|\,k} \right\| \tag{54a}$$

$$\text{Subject to } x_{0\,|\,k} = x(k) \tag{54b}$$

$$x_{N\,|\,k} \in \mathcal{S}_N^{\mathrm{trm}} \tag{54c}$$

$$\text{for } i = 0, \ldots, N-1\text{:}$$

$$x_{i+1\,|\,k} = \mathrm{f}(x_{i\,|\,k}, u_{i\,|\,k}) + \mathrm{f}^l(x_{i\,|\,k}, u_{i\,|\,k}) \tag{54d}$$

$$x_{i\,|\,k} \in \bar{\mathcal{X}}_i \tag{54e}$$

$$u_{i\,|\,k} \in \bar{\mathcal{U}}_i \tag{54f}$$

$$\mathcal{E}^l(x_{i\,|\,k}, u_{i\,|\,k}) \subseteq \bar{\mathcal{E}}_i^l. \tag{54g}$$

Similar to the nominal case, constraint satisfaction under application of $u(k) = u_{0\,|\,k}^*$ can be shown through the recursive feasibility of (54) using the tightened constraints together with a robust terminal invariant $\mathcal{S}^{\mathrm{trm}}$ set.

Assumption 2 (Robust Terminal Control Invariant Set)

Consider the system (51). There exists a terminal set $\mathcal{S}^{\mathrm{trm}} \subseteq \bar{\mathcal{X}}_N$ and a Lipschitz continuous control law $\kappa^{\mathrm{trm}} : \mathcal{S}^{\mathrm{trm}} \to \mathbb{R}^{n_u}$ such that for all $x \in \mathcal{S}^{\mathrm{trm}}$ and $\mathrm{e} \in \bar{\mathcal{E}}_N^l$, we have

1) $\mathcal{E}^l(x, \kappa^{\mathrm{trm}}(x)) \subseteq \bar{\mathcal{E}}_N^l$
2) $\kappa^{\mathrm{trm}}(x) \in \bar{\mathcal{U}}_N$
3) $\mathrm{f}(x, \kappa^{\mathrm{trm}}(x)) + \mathrm{f}^l(x, \kappa^{\mathrm{trm}}(x)) + \mathrm{e} \in \mathcal{S}^{\mathrm{trm}}$.

Suppose $0 \in \mathrm{int}(\mathcal{X} \times \mathcal{U})$ and the linearization of (51) at the origin is stabilizable. In that case, a sufficiently small learning error $\mathcal{E}^l(x, u)$ allows the construction of a terminal set $\mathcal{S}^{\mathrm{trm}}$ and controller κ^{trm} satisfying Assumption 2 [11, 3.3.2]. Compared with the nominal PSF terminal set assumption (Assumption 1), Assumption 2 ensures forward invariance of a polytopic terminal set $\mathcal{S}^{\mathrm{trm}}$ for all possible learning errors and requires κ^{trm} to be Lipschitz continuous. Combining Assumption 2 with assuming Lipschitz continuity of the dynamics model (51) and Lipschitz continuity of $\mathcal{E}^l(x, u)$ under the Hausdorff metric (see "Probabilistic Nonparametric Model: Gaussian Process Regression" for such an $\mathcal{E}^l(x, u)$ enables the following data-driven PSF result [60, Theorem 4.6].

Predictive Safety Filter Applications: Experimental Race Cars and Simulated Quadrotors

In the following, we demonstrate two applications of predictive safety filters (PSFs). The first example considers an experimental miniature race car application, as in [140], which implements the soft-constrained PSF (30) to enhance either a human driver or an imitation learning-based policy with safety guarantees. Parameters used in the drive-train dynamics and the Pacejka [S10, Sec. 13.5] tire model are identified from measurements. The second example demonstrates a probabilistic PSF formulation for a quadrotor as in [60]. The constraints in (54) are implemented using a Bayesian regression model to ensure safety in probability during online controller tuning, during which ground crashes would occur without the filter in place.

SAFE MINIATURE RACE CAR OPERATION AND IMITATION LEARNING

We consider a dynamic bicycle model [S9, Sec. 2] with states $x = [p_x, p_y, \psi, v_x, v_y, r]$ and inputs $u = [\delta, \tau]$ as described in Table S1 and dynamics given by

$$\dot{x} = \begin{bmatrix} v_x \cos(\psi) - v_y \sin(\psi) \\ v_x \sin(\psi) + v_y \cos(\psi) \\ r \\ \frac{1}{m}(F_x - F_{yf} \sin(\delta) + m v_y r) \\ \frac{1}{m}(F_{yr} + F_{yf} \cos(\delta) - m v_x r) \\ \frac{1}{I_z}(F_{yf} l_f \cos(\delta) - F_{yr} l_r) \end{bmatrix} \tag{S29}$$

where the lateral forces are modeled according to a Pacejka tire model [S10, Sec. 13.5] as

$$\alpha_f = \arctan\left(\frac{v_y + l_f r}{v_x}\right) - \delta, \quad \alpha_r = \arctan\left(\frac{v_y - l_r r}{v_x}\right) \tag{S30}$$

TABLE S1 States, inputs, and parameters of a vehicle.

State Symbol	Quantity
$p_{x/y}$	x-y coordinates of the car
Ψ	Heading angle
$v_{x/y}$	Velocity in the car frame
r	Yaw rate in the car frame
Input Symbol	**Quantity**
δ	Steering angle
τ	Drive-train command
Parameter Symbol	**Quantity**
m	Mass
I_z	Yaw moment of inertia
$l_{f/r}$	Distance between center of gravity and front/rear axles
$D_{f/r}, C_{f/r}, B_{f/r}$	Pacejka tire model parameters
C_1, C_2, C_3, C_4, C_5	Drive-train model parameters

and

$$F_{yf/yr} = D_{f/r} \sin(C_{f/r} \arctan(B_{f/r} \alpha_{f/r})) \tag{S31}$$

and a drive-train model is used for the longitudinal force

$$F_x = C_1 \tau + C_2 \tau^2 + C_3 v_x + C_4 v_x^2 + C_5 \tau v_x. \tag{S32}$$

All parameters are described in Table S1, and they have been identified using least-squares regression.

The input is limited by the maximum steering angle and maximum drive-train authority, and the safety constraints require the vehicle to stay within the track boundaries, as depicted in Figure S25. The constraint set $\mathcal{X}$ is formulated in track-relative error states, which also simplifies the computation of the terminal invariant set, according to Assumption 1

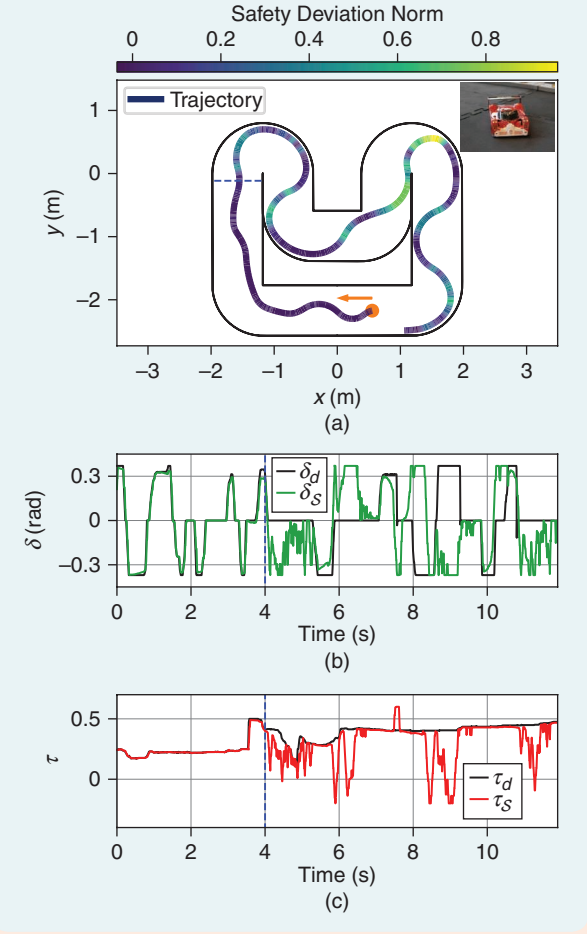


FIGURE S25 A miniature race car example. (a) The vehicle trajectory with the magnitude of safety filter intervention. (b) and (c) Human driver control inputs providing the desired control signal by a joystick as well as control inputs resulting from the PSF. The dashed blue line indicates the transition from safe driver inputs to unsafe inputs. (Source: ©2021 IEEE.)

(Continued)

Predictive Safety Filter Applications: Experimental Race Cars and Simulated Quadrotors (*Continued*)

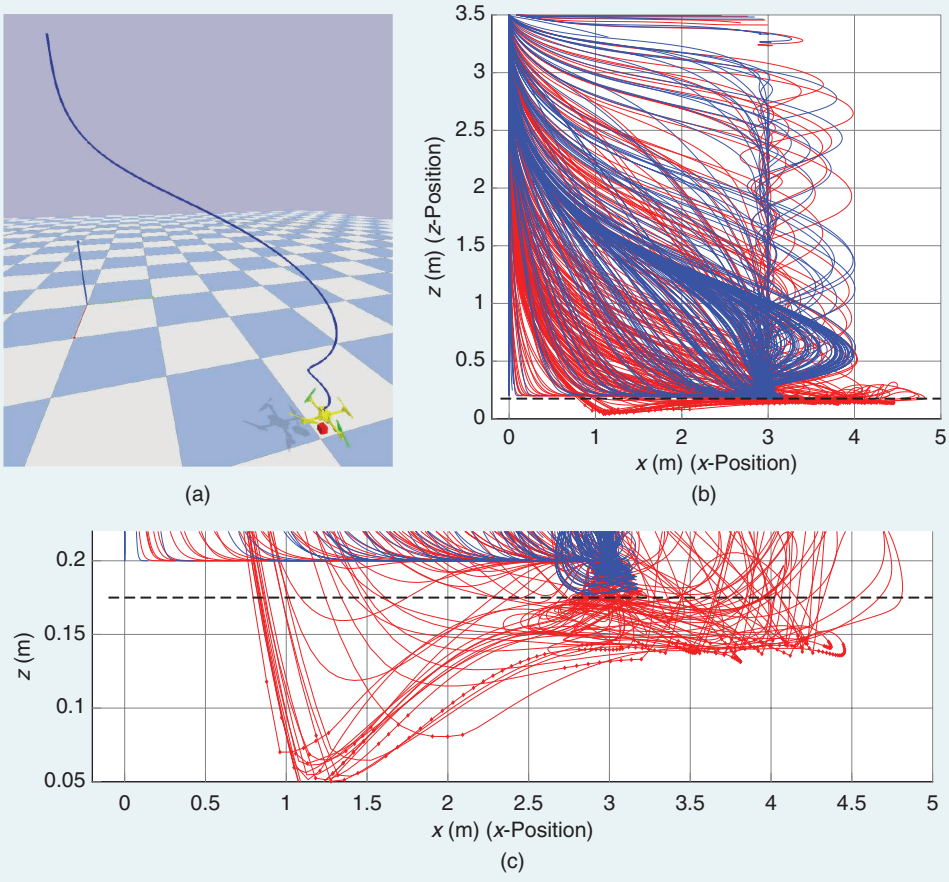


FIGURE S26 Safe quadrotor gain tuning. (a) A PyBullet quadrotor simulation, showing the optimal safe trajectory (blue line). (b) Learning episode trajectories without (red lines) and with (blue lines) the safety filter. (c) Ground collisions are indicated with red squares.

using convex approximations techniques [140]. The PSF is implemented in a nominal fashion using soft constraints (30) to ensure practical feasibility. We consider a driver-assistance scenario as an experiment, with the desired input signal $u_{\text{des}}(k)$ provided by a human driver that is potentially unsafe with respect to the track boundary safety requirements. The PSF provides necessary interventions online to keep the vehicle safe in a minimally invasive fashion, yielding control of the vehicle to the driver as long as the driver's actions remain safe.

Figure S25 illustrates a corresponding experiment with safety intervention magnitudes along a closed-loop trajectory. The input comparison plot shows the proposed desired input signals and the filtered applied input signals. The human performs safe driving during the first 4 s, which can be seen by the unfiltered application of the proposed input signals. In contrast, after this initial time period, the driver purposefully applies unsafe actions, which do not pass the PSF and get modified to ensure safety as desired. As shown in the plot, the PSF keeps the vehicle within the track boundaries at all times.

In addition to the driver-assistance scenario, [140, Sec. VI.B] demonstrates the combination of the same PSF with an imitation learning algorithm that reproduces a carefully selected expert policy using a deep neural network approximation. The PSF successfully keeps the system safe during the so-called DAgger learning episodes [168] and shows minimal intervention after convergence to an approximately optimal control policy.

PREDICTIVE SAFETY FILTERS USING BAYESIAN MODEL ESTIMATES FOR SAFE QUADROTOR TUNING

In the second example [60], we consider the AscTec Hummingbird drone, simulated in the Bullet Physics SDK [S11], as seen in Figure S26(a). A two-layer control structure enables position tracking, where the inner control loop takes the pitch, roll, and vertical

acceleration as input and commands motor torques. The outer controlled system model [S12] consists of states $x \in \mathbb{R}^{10}$, inputs $u \in \mathbb{R}^3$, and dynamics of the form $x(k+1) = \theta_{\text{true}}^{\top}\phi(x,u)$. The safety constraint is to stay above the ground, while the learning task is to efficiently tune an outer saturated PD controller to approach a specific landing position x_d, y_d, z_d. The outer PD controller takes the form

$$\pi_{\text{des}}(x;p,d) = \begin{cases} \text{clip}\,(p_{12}(x_d - x) + d_{12}\dot{x}, -1, 1) \\ \text{clip}\,(p_{12}(y_d - y) + d_{12}\dot{y}, -1, 1) \\ \text{clip}\,(p_3(z_d - z) + d_3\dot{z}, -1, 1) \end{cases}$$

where $\text{clip}\,(x, c_1, c_2) = \max(\min(x, c_2), c_1)$, with PD controller gains $p_{12}, p_3 \in [0, 10]$, and $d_{12}, d_3 \in [-10, 0]$. A Bayesian optimization algorithm [S13] episodically adjusts the PD gains to minimize $|x_d - x| + |y_d - y| + |z_d - z| + 100\,\|\pi_{\text{des}}(x) - u^{\star}_{0|k}\|$, where safety-ensuring actions are largely penalized during the learning process. As depicted in Figure S26(c), the direct application of the learning procedure results in ground crashes.

The learning-based safety filter model (51) is obtained from hovering data at a safe altitude and inferred using Gaussian process (GP) regression in a parametric fashion. The learning-based PSF of the form (54) was designed using $L = 0.999$ (based on an incremental stabilizability argument instead of Lipschitz continuity) with the constraint-tightening fraction $\epsilon = 0.01$. The confident subset constraint was designed to achieve constraint satisfaction with probability $p_s = 0.9$. The terminal set was formulated as a subset of the value function corresponding to a linear-quadratic regulator for the hovering position using a linearization of (51). The Bayesian optimization PD tuning results with the safety filter are shown in Figure S26, where safety is ensured during all 240 learning episodes. The learned controller achieves good performance and does not require safety interventions after the completion of learning.

REFERENCES

[S9] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optim. Contr. Appl. Methods*, vol. 36, no. 5, pp. 628–647, Sep./Oct. 2015, doi: 10.1002/oca.2123.

[S10] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer Science & Business Media, 2011.

[S11] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," Tech. Rep., 2016. [Online]. Available: https://pybullet.org/

[S12] H. Hu, X. Feng, R. Quirynen, M. Villanueva, and B. Houska, "Real-time tube MPC applied to a 10-state quadrotor model," in *Proc. Amer. Contr. Conf.*, 2018, pp. 3135–3140, doi: 10.23919/ACC.2018.8431112.

[S13] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Timpe, "Data-efficient autotuning with Bayesian optimization: An industrial control study," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 730–740, May 2020, doi: 10.1109/TCST.2018.2886159.

Theorem 6

Let Assumption 2 hold and assume that (51) and the corresponding uncertainty bound $\mathcal{E}^l(x,u)$ satisfying (35) are Lipschitz continuous mappings with Lipschitz constants L_f, $L_{\mathcal{E}}$. Consider (53) with constraint tightening sequence

$$\epsilon_i = \epsilon \frac{1 - \sqrt{L_f}^{\,i}}{1 - \sqrt{L_f}} \text{ for some } \epsilon > 0 \tag{55}$$

and allowable disturbance bound $\bar{\mathcal{E}}^l_\gamma = \{x \in \mathbb{R}^n \mid A^{\mathcal{E}} x \leq \gamma \mathbb{1}^{n_{\mathcal{E}}}\} \subset \mathbb{R}^{n_x}$ in (52) with the scaling factor $\gamma > 0$. If $L_{\mathcal{E}} \leq c\epsilon$ for some $c > 0$, then there exists a $\gamma > 0$ small enough that the initial feasibility of (54) ensures safe system operation for all future times according to (2) at probability level p_s.

Theorem 6 states that Lipschitz continuity allows one to design the learning-based PSF problem (54) using the iterative constraint tightening sequence (55) in combination with the admissible disturbance bound $\bar{\mathcal{E}}^l_\gamma$ along backup trajectories. The remaining tuning parameters are therefore limited to the scalars ϵ and γ. Furthermore, if $L_{\mathcal{E}}$ is small enough for a selected ϵ, a sufficiently small $\gamma > 0$ exists such that the initial feasibility implies constraint satisfaction at probability level p_s for all times. Intuitively, sufficiently small $L_{\mathcal{E}}$ means that the difference between $\mathcal{E}(x,u)$ and $\mathcal{E}(x + \Delta x, u + \Delta u)$ must be small for small values Δx, Δu such that the error bound is not allowed to change rapidly. In the case of GP regression using a squared exponential kernel, this relates either to a sufficiently large length-scale parameter or homogeneous data coverage [160].

If problem (54) is not initially feasible due to the confident subset constraint (54g), either the model needs to be refined using additional data or the probability level p_s can be lowered since $p_s \to 0$ typically implies $\mathcal{E}(x,u) \to \{0\}$. While the exact values of L_f, $L_{\mathcal{E}}$, c, and γ are challenging to compute explicitly, the discussion in [60, Sec. 4.3] using $\rho = L$ provides an extensive practical tuning guideline with a statistical verification procedure. Note that conservativeness can further be reduced by using incremental Lyapunov functions [159] instead of a Lipschitz continuity of (51) [60].

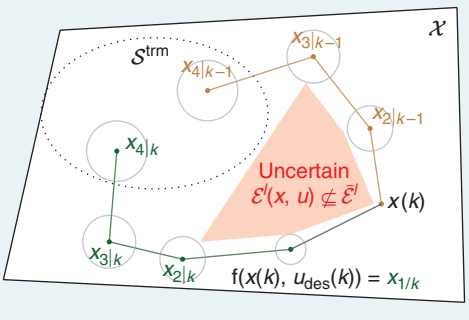


FIGURE 7 A learning enhanced PSF. Uncertain model regions (red) are avoided when planning backup trajectories. An additional safety margin (circles) allows for the compensation of the remaining uncertainty during closed-loop operation.

CONCLUSION

This article provides an introduction to three approaches for constructing safety filters for safety-critical control design and discusses recent research that has sought to unify these techniques. The prospect of bridging the gap between first-principle models and real-world systems through data is a topic at the forefront of research in control theory and applications. We highlight how the three safety filter techniques can be integrated with learning-based models to yield theoretical and practical safety guarantees in the face of model uncertainty. Applications demonstrating each of the safety filter techniques are presented and show that the proposed approaches are promising solutions for real engineering challenges. The design of safety filters blending the three techniques is subject to ongoing investigations and can capitalize on advantages regarding scalability, optimality, and computational efficiency present in each method to produce both performant and robust safety filter designs.

Challenges and Future Research Directions

While we have provided an overview of standard forms and data-driven extensions of HJ reachability, CBFs, and PSFs, there remain several interesting directions for research, both in and outside of a data-driven paradigm. One of the most important open problems is conservative safety interventions, as discussed in the "Approximation of Ideal Safety Filter" section. Such overly cautious interventions typically arise due to poor underapproximations of the maximal control invariant set. The CBF and PSF methodologies explicitly rely on using a set known to be control invariant, either as the zero-superlevel set of the CBF or as the terminal control invariant set in a PSF. If these sets are conservative underapproximations of the maximal control invariant set, the closed-loop behavior of the respective safety filters will be conservative. HJ reachability seeks to address this problem directly by computing the maximal control invariant set, but it does not scale well with the system dimension.

We believe that meaningful steps forward in addressing this challenge will focus on finding permissive underapproximations of the maximal control invariant set in a computationally efficient manner, and promising threads in this vein have recently arisen across the safety filter methodologies. Work from the perspective of HJ reachability has focused on using learning-based methods to approximate the maximal control invariant set of high-dimensional systems [78], [85]. Several approaches have been proposed for synthesizing less conservative CBFs for higher dimensional systems by utilizing system structure [105], convex sums-of-squares programming [102], and data-driven methods [106]. Similarly, data-driven approaches for enlarging the terminal invariant set in PSFs have recently been explored [12], [117]. Perhaps the most promising threads exist at the intersection of the methodologies presented in this work, where computational tools based on HJ reachability support CBF- and PSF-based safety filters [15], [16], [121], [169].

A second open question that is of great interest is bridging the performance gap between the ideal safety filter and the three safety filter methodologies. As seen in "Safety Filter Design Example", the ideal safety filter, though intractable to implement on many real-world systems, greatly outperforms the HJ reachability and CBF- and PSF-based safety filters. The performance of the ideal safety filter is fundamentally enabled by the long prediction horizon used in the optimization problem defining it. In the context of HJ reachability and CBF-based safety filters, future research directions will seek to incorporate a prediction horizon into the safety filter design, as is being explored in the recent work in [97], [129]. In the context of PSFs, if future desired inputs are explicitly known or can be described parametrically, they may be incorporated into the cost function of a PSF. We believe that this may enable improved performance by PSF-based safety filters, which, however, has yet to be studied thoroughly.

This direction of incorporating future inputs also presents an opportunity for data-driven methods. In particular, data-driven approaches that forecast future desired input signals based on a combination of previous experiences and a system's current state could be incorporated into the control synthesis process, enabling improved performance. An example of such a setting is human commands in shared autonomy systems, where prediction or anticipation models of human decisions that leverage data [140], [161], [162] can lead to improved closed-loop performance by a safety filter.

Beyond these challenges facing the core safety filter methodologies, there are a number of other interesting open questions surrounding data-driven safety filters. A primary question regards constructing a general process for the safe collection of data from real-world systems [21], [60], [64], [151]. Such a process will address questions on how to best incorporate prior knowledge of a system, how initial uncertainty should be quantified and subsequently improved, how sampling should be done to guide efficient experimental design, and how safety constraints should be met during the acquisition of data. Each of these individual questions provides a wealth of future research directions that require a unifying theory with the practical limitations of real-world systems.

"Learning With Real-World Data" highlights a second area that we believe presents meaningful future research directions. In particular, data produced by real-world systems is often not immediately amenable to being utilized in a learning algorithm. Instead, the data must often go through various forms of preprocessing to ensure that it is well conditioned for a learning problem. This processing modifies the data and can introduce its own form of uncertainty and residual learning error that should be accounted for in safety filter design. Future research will seek to characterize what processing tools are the most effective for different challenges present in real-world data and provide a rigorous quantification of their impact on the resulting learning error.

Lastly, data-driven models must be computationally efficient to be integrated into safety filters deployed in highly dynamic applications such as legged robots or aircraft control. It is also desirable for models to quickly incorporate incoming measurements and allow model refinement in real time. These challenges are fundamentally at the intersection of theory and practice, and addressing them must consider the tradeoff between model accuracy and expressiveness and computational requirements. A related open question for future work is to explicitly consider changing system dynamics, for instance, due to a system component failure or unexpected disturbances, in the safety filter design. This includes the detection of such events as well as active compensation mechanisms, which, without such detection, may render data-driven models and their corresponding safety filter designs unreliable.

Recent and ongoing research on data-driven safety filters has shown great promise for solutions to some of the largest challenges in ensuring real-world control systems' safety. We believe that the path forward for data-driven safety filter research lies in studying the challenging questions that arise when working with real-world systems producing real-world data while operating in dynamic environments.

ACKNOWLEDGMENT

The first three authors contributed equally to this manuscript.

AUTHOR INFORMATION

Kim P. Wabersich (wabersich@kimpeter.de) received the bachelor's and master's degrees in engineering cybernetics from the University of Stuttgart in Germany in 2015 and 2017, respectively. He received the Ph.D. degree in predictive safety mechanisms at the Institute for Dynamic Systems and Control, ETH Zurich in 2021 and continued his work as a postdoctoral researcher until 2022. He currently works at Bosch Research, 71272 Renningen, Germany, focusing on safety-critical systems with applications in autonomous driving. His research interests include control methods and their intersections with safe reinforcement learning.

Andrew J. Taylor received the B.S. and M.S. degrees in aerospace engineering from the University of Michigan at Ann Arbor, in 2016 and 2017, respectively. He is currently pursuing a Ph.D. degree at Caltech, Pasadena, CA91125 USA, in control and dynamical systems. His research interests include safety-critical control for robotic systems and data-driven control techniques for nonlinear systems. He is a Student Member of IEEE.

Jason J. Choi received the B.S. degree in mechanical engineering from Seoul National University in 2019. He is currently pursuing a Ph.D. degree at the University of California Berkeley, Berkeley, CA94720 USA, in mechanical engineering. His research interests center on optimal control theories for nonlinear and hybrid systems, data-driven methods for safe control, and their applications to robotics and autonomous mobility.

Koushil Sreenath is an associate professor of mechanical engineering at the University of California Berkeley, Berkeley, CA94720 USA. He received the Ph.D. degree in electrical engineering and computer science and the M.S. degree in applied mathematics from the University of Michigan at Ann Arbor, MI, USA, in 2011. He was a postdoctoral scholar at the GRASP Lab at the University of Pennsylvania from 2011 to 2013 and an assistant professor at Carnegie Mellon University from 2013 to 2017. His research interest lies at the intersection of highly dynamic robotics and applied nonlinear control. He received the NSF CAREER Award, Hellman Fellows Award, Best Paper Award at the Robotics: Science and Systems (RSS), and the Google Faculty Research Award in Robotics.

Claire J. Tomlin is the James and Katherine Lau Professor of Engineering and a professor and the chair of the Department of Electrical Engineering and Computer Sciences at the University of California Berkeley (UC Berkeley), Berkeley, CA94720 USA. She was an assistant, associate, and full professor in aeronautics and astronautics at Stanford University from 1998 to 2007, and in 2005, she joined UC Berkeley. She works in the area of control theory and hybrid systems, with applications to air traffic management, unmanned aerial vehicle systems, energy, robotics, and systems biology. She is a MacArthur Foundation Fellow (2006), and in 2017, she was awarded the IEEE Transportation Technologies Award. In 2019, she was elected to the National Academy of Engineering and the American Academy of Arts and Sciences. She is a Fellow of IEEE.

Aaron D. Ames is the Bren Professor of Mechanical and Civil Engineering and Control and Dynamical Systems at Caltech, Pasadena, CA91125 USA. Prior to joining Caltech in 2017, he was an associate professor at Georgia Tech in the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering. He received the B.S. degree in mechanical engineering and the B.A. degree in mathematics from the University of St. Thomas in 2001, and he received the M.A. degree in mathematics and the Ph.D. degree in electrical engineering and computer sciences from the University of California Berkeley (UC Berkeley) in 2006. He served as a postdoctoral scholar in control and dynamical systems at Caltech from 2006 to 2008 and began his faculty career at Texas A&M University in 2008. At UC Berkeley, he was the recipient of the 2005 Leon O. Chua Award for achievement in nonlinear science and the 2006 Bernard Friedman Memorial Prize in Applied Mathematics, and he received the NSF CAREER Award in 2010, the 2015 Donald P. Eckman Award, and the 2019 IEEE CSS Antonio Ruberti Young Researcher Prize. His research interests span the areas of robotics, nonlinear, safety-critical control, and hybrid systems, with a special focus on applications to dynamic robots—both formally and through experimental validation.

Melanie N. Zeilinger is an associate professor at ETH Zurich, 8092 Zurich, Switzerland. She received the Diploma

degree in engineering cybernetics from the University of Stuttgart, Germany in 2006 and the Ph.D. degree with honors in electrical engineering from ETH Zurich, Switzerland, in 2011. From 2011 to 2012 she was a postdoctoral fellow with the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. She was a Marie Curie Fellow and postdoctoral researcher with the Max Planck Institute for Intelligent Systems, Tübingen, Germany, until 2015 and with the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley, CA, USA, from 2012 to 2014. From 2018 to 2019 she was a professor at the University of Freiburg, Germany. Her current research interests include safe learning-based control as well as distributed control and optimization, with applications to robotics and human-in-the-loop control. She is a Member of IEEE.

REFERENCES

[1] O. J. Ayamolowo, P. Manditereza, and K. Kusakana, "Exploring the gaps in renewable energy integration to grid," *Energy Rep.*, vol. 6, pp. 992–999, Dec. 2020, doi: 10.1016/j.egyr.2020.11.086.

[2] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26,754–26,773, Nov. 2017, doi: 10.1109/ACCESS.2017.2773127.

[3] E. Dassau, T. Hennings, J. Fazio, E. Atlas, and M. Phillip, "Closing the loop," *Diabetes Technol. Therapeutics*, vol. 15, no. S1, pp. S29–S39, Feb. 2013, doi: 10.1089/dia.2013.1504.

[4] R. Hovorka et al., "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," *Physiological Meas.*, vol. 25, no. 4, pp. 905–920, Aug. 2004, doi: 10.1088/0967-3334/25/4/010.

[5] P. Englert, N. A. Vien, and M. Toussaint, "Inverse KKT: Learning cost functions of manipulation tasks from demonstrations," *Int. J. Robot. Res.*, vol. 36, nos. 13–14, pp. 1474–1488, Dec. 2017, doi: 10.1177/0278364917745980.

[6] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*.

[7] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proc. IEEE*, vol. 91, no. 7, pp. 986–1001, Jul. 2003, doi: 10.1109/JPROC.2003.814621.

[8] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *Proc. IEEE 56th Conf. Decis. Contr. (CDC)*, Melbourne, VIC, Australia, 2017, pp. 2242–2253, doi: 10.1109/CDC.2017.8263977.

[9] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 462–467, Aug. 2007, doi: 10.3182/20070822-3-ZA-2920.00076.

[10] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. IEEE 18th Eur. Contr. Conf. (ECC)*, Naples, Italy, 2019, pp. 3420–3431, doi: 10.23919/ECC.2019.8796030.

[11] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, CA, USA: Nob Hill Publishing, 2017.

[12] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. IEEE 57th Conf. Decis. Contr. (CDC)*, Miami, FL, USA, 2018, pp. 7130–7135, doi: 10.1109/CDC.2018.8619829.

[13] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, vol. 78. Cham, Switzerland: Springer, 2008.

[14] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6835–6841, doi: 10.1109/CDC45484.2021.9683111.

[15] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6814–6821, doi: 10.1109/CDC45484.2021.9683085.

[16] K. Leung et al., "On infusing reachability-based safety assurance within planning frameworks for human–robot vehicle interactions," *Int. J. Robot. Res.*, vol. 39, nos. 10–11, pp. 1326–1345, Sep. 2020, doi: 10.1177/0278364920950795.

[17] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, New Orleans, LA, USA, 2021, pp. 3882–3889, doi: 10.23919/ACC50511.2021.9483029.

[18] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2638–2651, May 2023, doi: 10.1109/TAC.2022.3209358.

[19] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multirate control: From low-level actuation to high-level planning," *IEEE Trans. Autom. Control*, vol. 67, no. 12, pp. 6627–6640, Dec. 2022, doi: 10.1109/TAC.2022.3184664.

[20] A. Wigren, J. Wågberg, F. Lindsten, A. G. Wills, and T. B. Schön, "Nonlinear system identification: Learning while respecting physical models using a sequential Monte Carlo method," *IEEE Control Syst.*, vol. 42, no. 1, pp. 75–102, Feb. 2022, doi: 10.1109/MCS.2021.3122269.

[21] 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 Trans. Autom. Control*, vol. 64, no. 7, pp. 2737–2752, Jul. 2019, doi: 10.1109/TAC.2018.2876389.

[22] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 3, pp. 269–296, May 2020, doi: 10.1146/annurev-control-090419-075625.

[23] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. Mach. Learn. Res.*, 2020, vol. 120, pp. 708–717.

[24] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, Jan. 2017, doi: 10.1016/j.automatica.2016.08.013.

[25] M. Krstic and M. Bement, "Nonovershooting control of strict-feedback nonlinear systems," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1938–1943, Dec. 2006, doi: 10.1109/TAC.2006.886518.

[26] I. Abel, D. Steeves, M. Krstić, and M. Janković, "Prescribed-time safety design for a chain of integrators," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Atlanta, GA, USA, 2022, pp. 4915–4920, doi: 10.23919/ACC53348.2022.9867700.

[27] C. M. Kellett, "A compendium of comparison function results," *Math. Contr., Signals, Syst.*, vol. 26, no. 3, pp. 339–374, Mar. 2014, doi: 10.1007/s00498-014-0128-8.

[28] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017, doi: 10.1109/TAC.2016.2638961.

[29] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[30] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 604–613, Oct. 1972, doi: 10.1109/TAC.1972.1100085.

[31] D. P. Bertsekas and I. B. Rhodes, "On the minimax reachability of target sets and target tubes," *Automatica*, vol. 7, no. 2, pp. 233–247, Mar. 1971, doi: 10.1016/0005-1098(71)90066-5.

[32] J.-P. Aubin, "A survey of viability theory," *SIAM J. Contr. Optim.*, vol. 28, no. 4, pp. 749–788, 1990, doi: 10.1137/0328044.

[33] M. Bardi et al., *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, vol. 12. Boston, MA, USA: Springer, 1997.

[34] C. Tomlin, J. Lygeros, and S. Sastry, "Synthesizing controllers for nonlinear hybrid systems," in *Proc. Int. Work. Hybrid Syst., Comput. Contr.*, Berlin, Germany: Springer, 1998, pp. 360–373.

[35] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, Mar. 1999, doi: 10.1016/S0005-1098(98)00193-9.

[36] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, Jun. 2004, doi: 10.1016/j.automatica.2004.01.012.

[37] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, Jan. 2006, doi: 10.1016/j.automatica.2005.08.007.

[38] A. Ames, J. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. IEEE 53rd Conf. Decis. Contr. (CDC)*, Los Angeles, CA, USA, 2014, pp. 6271–6278, doi: 10.1109/CDC.2014.7040372.

[39] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, Oct. 1998, doi: 10.1016/S0005-1098(98)00073-9.

[40] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. London, U.K.: Springer-Verlag, 2017.

[41] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005, doi: 10.1109/TAC.2005.851439.
[42] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. 18th Int. Conf. Hybrid Syst., Comput. Contr. (HSCC)*, Seattle, WA, USA, 2015, pp. 11–20, doi: 10.1145/2728606.2728612.
[43] R. Konda, A. D. Ames, and S. Coogan, "Characterizing safety: Minimal control barrier functions from scalar comparison systems," *IEEE Contr. Syst. Lett.*, vol. 5, no. 2, pp. 523–528, Apr. 2021, doi: 10.1109/LCSYS.2020.3003887.
[44] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 108–113, Jan. 2019, doi: 10.1109/LCSYS.2018.2853698.
[45] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, Oct. 2018, doi: 10.1016/j.automatica.2018.07.004.
[46] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *Proc. IEEE 57th Conf. Decis. Contr. (CDC)*, Miami, FL, USA, 2018, pp. 3592–3599, doi: 10.1109/CDC.2018.8619139.
[47] T. Mannucci, E. J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1069–1081, Apr. 2018, doi: 10.1109/TNNLS.2017.2654539.
[48] O. Bastani, "Safe reinforcement learning with nonlinear dynamics via model predictive shielding," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, New Orleans, LA, USA, 2021, pp. 3488–3494, doi: 10.23919/ACC50511.2021.9483182.
[49] 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 *Proc. IEEE 53rd Conf. Decis. Contr. (CDC)*, Los Angeles, CA, USA, 2014, pp. 1424–1431, doi: 10.1109/CDC.2014.7039601.
[50] S. Herbert, J. J. Choi, S. Sanjeev, M. Gibson, K. Sreenath, and C. J. Tomlin, "Scalable learning of safety guarantees for autonomous systems using Hamilton-Jacobi reachability," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi'an, China, 2021, pp. 5914–5920, doi: 10.1109/ICRA48506.2021.9561561.
[51] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using Gaussian processes," in *Proc. IEEE 59th Conf. Decis. Contr. (CDC)*, Jeju, South Korea, 2020, pp. 3699–3704, doi: 10.1109/CDC42340.2020.9303847.
[52] J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," in *Proc. Robot., Sci. Syst. (RSS) XVI*, Bend, OR, USA, 2020, pp. 1–9.
[53] C. Folkestad, Y. Chen, A. D. Ames, and J. W. Burdick, "Data-driven safety-critical control: Synthesizing control barrier functions with Koopman operators," *IEEE Contr. Syst. Lett.*, vol. 5, no. 6, pp. 2012–2017, Dec. 2021, doi: 10.1109/LCSYS.2020.3046159.
[54] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic safety constraints for learned high relative degree system dynamics," in *Proc. Mach. Learn. Res. (PMLR)*, 2020, pp. 781–792.
[55] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," in *Proc. Mach. Learn. Res.*, 2021, vol. 144, pp. 1041–1053.
[56] A. J. Taylor, V. D. Dorobantu, S. Dean, B. Recht, Y. Yue, and A. D. Ames, "Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6469–6476, doi: 10.1109/CDC45484.2021.9683511.
[57] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, New Orleans, LA, USA, 2021, pp. 3683–3690, doi: 10.23919/ACC50511.2021.9483420.
[58] V. Dhiman, M. J. Khojasteh, M. Franceschetti, and N. Atanasov, "Control barriers in Bayesian learning of system dynamics," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 214–229, Jan. 2023, doi: 10.1109/TAC.2021.3137059.
[59] Y. Emam, P. Glotfelter, S. Wilson, G. Notomista, and M. Egerstedt, "Data-driven robust barrier functions for safe, long-term operation," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1671–1685, Jun. 2022, doi: 10.1109/LRA.2022.3216996.
[60] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, Jul. 2021, Art. no. 109597, doi: 10.1016/j.automatica.2021.109597.
[61] A. Didier, K. P. Wabersich, and M. N. Zeilinger, "Adaptive model predictive safety certification for learning-based control," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 809–815, doi: 10.1109/CDC45484.2021.9682832.
[62] S. Muntwiler, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "Distributed model predictive safety certification for learning-based control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5258–5265, Apr. 2021, doi: 10.1016/j.ifacol.2020.12.1205.
[63] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 176–188, Jan. 2022, doi: 10.1109/TAC.2021.3049335.
[64] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proc. 57th IEEE Conf. Decis. Contr. (CDC)*, Miami, FL, USA, 2018, pp. 6059–6066, doi: 10.1109/CDC.2018.8619572.
[65] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 5, pp. 411–444, May 2022, doi: 10.1146/annurev-control-042920-020211.
[66] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, Nov. 1999, doi: 10.1016/S0005-1098(99)00113-2.
[67] M. Nagumo, "Über die Lage der Integralkurven gewöhnlicher differentialgleichungen," in *Proc. Physico-Math. Soc. Jpn.*, in 3rd Series, vol. 24, Jan. 1942, pp. 551–559.
[68] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe online control system upgrades," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Philadelphia, PA, USA, 1998, vol. 6, pp. 3504–3508, doi: 10.1109/ACC.1998.703255.
[69] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability Theory: New Directions*. Springer Science & Business Media, 2011, Berlin, Germany.
[70] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998, doi: 10.1109/9.664154.
[71] A. K. Akametalu, S. Ghosh, J. F. Fisac, and C. J. Tomlin, "A minimum discounted reward Hamilton-Jacobi formulation for computing reachable sets," 2018, *arXiv:1809.00706*.
[72] B. Xue, Q. Wang, N. Zhan, M. Fränzle, and S. Feng, "Reach-avoid differential games based on invariant generation," 2018, *arXiv:1811.03215*.
[73] M. G. Crandall, L. C. Evans, and P.-L. Lions, "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Trans. Amer. Math. Soc.*, vol. 282, no. 2, pp. 487–502, Apr. 1984, doi: 10.1090/S0002-9947-1984-0732102-X.
[74] I. M. Mitchell and J. A. Templeton, "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *Proc. Int. Work. Hybrid Syst., Comput. Contr.*, Berlin, Germany: Springer, 2005, pp. 480–494, doi: 10.1007/978-3-540-31954-2_31.
[75] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[76] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, "Decomposition of reachable sets and tubes for a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3675–3688, Nov. 2018, doi: 10.1109/TAC.2018.2797194.
[77] S. L. Herbert, S. Bansal, S. Ghosh, and C. J. Tomlin, "Reachability-based safety guarantees using efficient initializations," in *Proc. IEEE 58th Conf. Decis. Contr. (CDC)*, Nice, France, 2019, pp. 4810–4816, doi: 10.1109/CDC40024.2019.9029575.
[78] S. Bansal and C. J. Tomlin, "DeepReach: A deep learning approach to high-dimensional reachability," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi'an, China, 2021, pp. 1817–1824, doi: 10.1109/ICRA48506.2021.9561949.
[79] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust tracking with model mismatch for fast and safe planning: An SOS optimization approach," in *Proc. Int. Work. Algorithmic Found. Robot.*, Cham, Switzerland: Springer, 2018, pp. 545–564.
[80] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *Int. J. Robot. Res.*, vol. 39, no. 12, pp. 1419–1469, Aug. 2020, doi: 10.1177/0278364920943266.
[81] I. Hwang, D. M. Stipanović, and C. J. Tomlin, "Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems," in *Proc. Adv. Contr., Commun. Netw., Transp. Syst.*, Boston, MA, USA: Springer-Verlag, 2005, pp. 3–19.
[82] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. Part I: External approximations," *Optim. Methods Softw.*, vol. 17, no. 2, pp. 177–206, 2002, doi: 10.1080/1055678021000012426.

[83] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, Aug. 2014, doi: 10.1109/TRO.2014.2312453.
[84] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," in *Proc. ASME Dyn. Syst. Contr. Conf. (DSCC)*, Park City, UT, USA, 2019, vol. 59162, p. V003T19A010, doi: 10.1115/DSCC2019-9214.
[85] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging Hamilton-Jacobi safety analysis and reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Montreal, QC, Canada, 2019, pp. 8550–8556, doi: 10.1109/ICRA.2019.8794107.
[86] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac, "Safety and liveness guarantees through reach-avoid reinforcement learning," in *Proc. Robot., Sci. Syst. (RSS) XVII*, 2021, pp. 1–12, doi: 10.15607/RSS.2021.XVII.077.
[87] J. Li, D. Lee, S. Sojoudi, and C. J. Tomlin, "Infinite-horizon reach-avoid zero-sum games via deep reinforcement learning," 2022, *arXiv:2203.10142*.
[88] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
[89] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Philadelphia, PA, USA: SIAM, 1990.
[90] E. D. Sontag, "A universal construction of Artstein's theorem on nonlinear stabilization," *Syst. Contr. Lett.*, vol. 13, no. 2, pp. 117–123, Jul. 1989, doi: 10.1016/0167-6911(89)90028-5.
[91] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Physical Syst. (ICCPS)*, Porto, Portugal, 2018, pp. 98–106, doi: 10.1109/ICCPS.2018.00018.
[92] X. Xu et al., "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds," in *Proc. IEEE Conf. Contr. Technol. Appl. (CCTA)*, Kohala Coast, HI, USA, 2017, pp. 1769–1775, doi: 10.1109/CCTA.2017.8062713.
[93] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, Jun. 2017, doi: 10.1109/TRO.2017.2659727.
[94] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, 2018, pp. 2460–2465, doi: 10.1109/ICRA.2018.8460471.
[95] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10,954–10,961, Oct. 2022, doi: 10.1109/LRA.2022.3192634.
[96] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 530–545, Mar. 2021, doi: 10.1109/TCST.2019.2952317.
[97] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi'an, China, 2021, pp. 8352–8358, doi: 10.1109/ICRA48506.2021.9561510.
[98] A. Alan, A. J. Taylor, C. R. He, A. D. Ames, and G. Orosz, "Control barrier functions and input-to-state safety with application to automated vehicles," 2022, *arXiv:2206.03568*.
[99] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proc. Robot., Sci. Syst. (RSS) XIII*, Cambridge, MA, USA, 2017, vol. 13, pp. 1–10.
[100] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of sampled-data systems with control barrier functions via approximate discrete time models," in *Proc. IEEE 61st Conf. Decis. Contr. (CDC)*, Cancú, Mexico, 2022, pp. 7127–7134, doi: 10.1109/CDC51059.2022.9993226.
[101] L. Wang, D. Han, and M. Egerstedt, "Permissive barrier certificates for safe stabilization using sum-of-squares," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Milwaukee, WI, USA, 2018, pp. 585–590, doi: 10.23919/ACC.2018.8431617.
[102] A. Clark, "Verification and synthesis of control barrier functions," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6105–6112, doi: 10.1109/CDC45484.2021.9683520.
[103] H. Dai and F. Permenter, "Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints," 2022, *arXiv:2210.00629*.
[104] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 944–951, Apr. 2022, doi: 10.1109/LRA.2021.3135569.
[105] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames, "Safe backstepping with control barrier functions," in *Proc. IEEE 61st Conf. Decis. Contr. (CDC)*, Cancún, Mexico, 2022, pp. 5775–5782, doi: 10.1109/CDC51059.2022.9992763.
[106] A. Robey et al., "Learning control barrier functions from expert demonstrations," in *Proc. IEEE 59th Conf. Decis. Contr. (CDC)*, Jeju, South Korea, 2020, pp. 3717–3724, doi: 10.1109/CDC42340.2020.9303785.
[107] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural Lyapunov-Barrier functions," in *Proc. Mach. Learn. Res.*, 2022, vol. 164, pp. 1724–1735.
[108] Z. Qin, D. Sun, and C. Fan, "Sablas: Learning safe control for black-box dynamical systems," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1928–1935, Apr. 2022, doi: 10.1109/LRA.2022.3142743.
[109] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Proc. 6th Annu. Conf. Robot Learn. (CoRL)*, Auckland, New Zealand, 2022. [Online]. Available: https://openreview.net/forum?id=4ffLQu_O-Dl
[110] F. Castañeda, H. Nishimura, R. McAllister, K. Sreenath, and A. Gaidon, "In-distribution barrier functions: Self-supervised policy filters that avoid out-of-distribution states," 2023, *arXiv:2301.12012*.
[111] A. Mesbah et al., "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?" in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Atlanta, GA, USA, 2022, pp. 342–357, doi: 10.23919/ACC53348.2022.9867643.
[112] L. Grüne, D. Nešić, and J. Pannek, *Model Predictive Control for Nonlinear Sampled-Data Systems*. Berlin, Germany: Springer-Verlag, 2007.
[113] M. Lazar and M. Tetteroo, "Computation of terminal costs and sets for discrete–time nonlinear MPC," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 141–146, Nov. 2018, doi: 10.1016/j.ifacol.2018.11.006.
[114] R. Amrit, J. B. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annu. Rev. Contr.*, vol. 35, no. 2, pp. 178–186, Dec. 2011, doi: 10.1016/j.arcontrol.2011.10.011.
[115] K. P. Wabersich, F. A. Bayer, M. A. Müller, and F. Allgöwer, "Economic model predictive control for robust periodic operation with guaranteed closed-loop performance," in *Proc. IEEE 18th Eur. Contr. Conf. (ECC)*, Naples, Italy, 2018, pp. 507–513, doi: 10.23919/ECC.2018.8550262.
[116] F. D. Brunner, M. Lazar, and F. Allgöwer, "Stabilizing linear model predictive control: On the enlargement of the terminal set," in *Proc. IEEE 13th Eur. Contr. Conf. (ECC)*, Zürich, Switzerland, 2013, pp. 511–517, doi: 10.23919/ECC.2013.6669436.
[117] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1883–1896, Jul. 2018, doi: 10.1109/TAC.2017.2753460.
[118] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *Proc. Contr. Conf.*, Cambridge, U.K., Sep. 2000, pp. 2319–2327.
[119] M. N. Zeilinger, M. Morari, and C. N. Jones, "Soft constrained model predictive control with robust stability guarantees," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1190–1202, May 2014, doi: 10.1109/TAC.2014.2304371.
[120] C. Feller and C. Ebenbauer, "Relaxed logarithmic barrier function based model predictive control of linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1223–1238, Mar. 2017, doi: 10.1109/TAC.2016.2582040.
[121] S. Tonkens and S. Herbert, "Refining control barrier functions through Hamilton-Jacobi reachability," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 13,355–13,362, doi: 10.1109/IROS47612.2022.9982203.
[122] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Contr. Syst. Lett.*, vol. 1, no. 2, pp. 310–315, Oct. 2017, doi: 10.1109/LCSYS.2017.2710943.
[123] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*, vol. 2, 4th ed. Belmont, MA, USA: Athena Scientific, 2012.
[124] D. Lee and C. J. Tomlin, "Hamilton-Jacobi equations for two classes of state-constrained zero-sum games," 2021, *arXiv:2106.15006*.
[125] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6137–6144, doi: 10.1109/CDC45484.2021.9683174.
[126] M. Davoodi, J. M. Cloud, A. Iqbal, W. J. Beksi, and N. R. Gans, "Safe human-robot coetaneousness through model predictive control barrier functions and motion distributions," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 271–277, Dec. 2021, doi: 10.1016/j.ifacol.2021.11.186.

[127] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Philadelphia, PA, USA, 2022, pp. 286–292, doi: 10.1109/ICRA46639.2022.9812334.
[128] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *IEEE Contr. Syst. Lett.*, vol. 5, no. 3, pp. 1007–1012, Jul. 2021, doi: 10.1109/LCSYS.2020.3008326.
[129] J. Breeden and D. Panagou, "Predictive control barrier functions for online safety critical control," in *Proc. IEEE 61st Conf. Decis. Contr. (CDC)*, 2022, pp. 924–931, doi: 10.1109/CDC51059.2022.9992926.
[130] S. Brüggemann, D. Steeves, and M. Krstic, "Simultaneous lane-keeping and obstacle avoidance by combining model predictive control and control barrier functions," in *Proc. IEEE 61st Conf. Decis. Contr. (CDC)*, 2022, pp. 5285–5290, doi: 10.1109/CDC51059.2022.9992613.
[131] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames, "Online active safety for robotic manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, China, 2019, pp. 173–178, doi: 10.1109/IROS40897.2019.8968231.
[132] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187,249–187,275, Sep. 2020, doi: 10.1109/ACCESS.2020.3025248.
[133] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Denver, CO, USA, 2020, pp. 1399–1405, doi: 10.23919/ACC45564.2020.9147463.
[134] B. T. Lopez, J. E. Slotine, and J. P. How, "Robust adaptive control barrier functions: An adaptive and data-driven approach to safety," *IEEE Contr. Syst. Lett.*, vol. 5, no. 3, pp. 1031–1036, Jul. 2021, doi: 10.1109/LCSYS.2020.3005923.
[135] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A survey on dual control," *Annu. Rev. Contr.*, vol. 45, pp. 107–117, Jun. 2018, doi: 10.1016/j.arcontrol.2017.11.001.
[136] E. Arcari, L. Hewing, M. Schlichting, and M. Zeilinger, "Dual stochastic MPC for systems with parametric and structural uncertainty," in *Proc. Mach. Learn. Res.*, 2020, vol. 120, pp. 894–903.
[137] L. Ljung, *System Identification*. London, U.K.: Springer, 1998.
[138] M. Maiworm, D. Limon, J. M. Manzano, and R. Findeisen, "Stability of Gaussian process learning based output feedback model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 455–461, Nov. 2018, doi: 10.1016/j.ifacol.2018.11.047.
[139] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009.
[140] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7635–7642, Oct. 2021, doi: 10.1109/LRA.2021.3097073.
[141] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Vancouver, BC, Canada, 2018, pp. 1–18.
[142] G. Shi et al., "Neural lander: Stable drone landing control using learned dynamics," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Montreal, QC, Canada, 2019, pp. 9784–9790, doi: 10.1109/ICRA.2019.8794351.
[143] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, Jun. 2004, doi: 10.1016/j.automatica.2004.02.002.
[144] M. Chen and C. J. Tomlin, "Hamilton–Jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 1, no. 1, pp. 333–358, May 2018, doi: 10.1146/annurev-control-060117-104941.
[145] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[146] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE 55th Conf. Decis. Contr. (CDC)*, Las Vegas, NV, USA, 2016, pp. 4661–4666, doi: 10.1109/CDC.2016.7798979.
[147] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler–Lagrange systems," *Automatica*, vol. 103, pp. 390–397, May 2019, doi: 10.1016/j.automatica.2019.01.023.
[148] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for Gaussian process regression with application to safe control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 659–669.
[149] K. P. Wabersich and M. N. Zeilinger, "Nonlinear learning-based model predictive control supporting state and input dependent model uncertainty estimates," *Int. J. Robust Nonlinear Contr.*, vol. 31, no. 18, pp. 8897–8915, Jul. 2021, doi: 10.1002/rnc.5688.
[150] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Pointwise feasibility of Gaussian process-based safety-critical control under model uncertainty," in *Proc. IEEE 60th Conf. Decis. Contr. (CDC)*, Austin, TX, USA, 2021, pp. 6762–6769, doi: 10.1109/CDC45484.2021.9683743.
[151] F. Castañeda, J. J. Choi, W. Jung, B. Zhang, C. J. Tomlin, and K. Sreenath, "Probabilistic safe online learning with control barrier functions," 2022, *arXiv:2208.10733*.
[152] Y. S. Shao, C. Chen, S. Kousik, and R. Vasudevan, "Reachability-based trajectory safeguard (RTS): A safe and fast reinforcement learning safety layer for continuous control," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3663–3670, Apr. 2021, doi: 10.1109/LRA.2021.3063989.
[153] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana Univ. Math. J.*, vol. 33, no. 5, pp. 773–797, 1984, doi: 10.1512/iumj.1984.33.33040.
[154] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," in *Proc. IEEE 58th Conf. Decis. Contr. (CDC)*, Nice, France, 2019, pp. 1758–1765, doi: 10.1109/CDC40024.2019.9030133.
[155] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "A control barrier perspective on episodic learning via projection-to-state safety," *IEEE Contr. Syst. Lett.*, vol. 5, no. 3, pp. 1019–1024, Jul. 2021, doi: 10.1109/LCSYS.2020.3009082.
[156] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, "Safe controller synthesis with tunable input-to-state safe control barrier functions," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 908–913, 2022, doi: 10.1109/LCSYS.2021.3087443.
[157] S. Muntwiler, K. P. Wabersich, L. Hewing, and M. N. Zeilinger, "Data-driven distributed stochastic model predictive control with closed-loop chance constraint satisfaction," in *Proc. IEEE 21st Eur. Contr. Conf. (ECC)*, Delft, The Netherlands, 2021, pp. 210–215, doi: 10.23919/ECC54610.2021.9655214.
[158] D. L. Marruedo, T. Alamo, and E. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proc. IEEE 41st Conf. Decis. Contr. (CDC)*, Las Vegas, NV, USA, 2002, vol. 4, pp. 4619–4624, doi: 10.1109/CDC.2002.1185106.
[159] J. Köhler, M. A. Müller, and F. Allgöwer, "A novel constraint tightening approach for nonlinear robust model predictive control," in *Proc. IEEE Amer. Contr. Conf. (ACC)*, Milwaukee, WI, USA, 2018, pp. 728–734, doi: 10.23919/ACC.2018.8431892.
[160] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[161] S. Löckel, J. Peters, and P. van Vliet, "A probabilistic framework for imitating human race driver behavior," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2086–2093, Apr. 2020, doi: 10.1109/LRA.2020.2970620.
[162] J. F. Fisac et al., "Probabilistically safe robot planning with confidence-based human predictions," in *Proc. Robot., Sci. Syst. (RSS) XIV*, 2018, pp. 1–9.
[163] M. Chen et al., "FaSTrack: A modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Trans. Autom. Control*, vol. 66, no. 12, pp. 5861–5876, Dec. 2021, doi: 10.1109/TAC.2021.3059838.
[164] T. Hsu, J. J. Choi, D. Amin, C. J. Tomlin, S. C. McWherter, and M. Piedmonte, "Towards flight envelope protection for the NASA tiltwing eVTOL flight mode transition using Hamilton-Jacobi reachability," in *Proc. Vertical Flight Soc. Forum 79*, 2023, p. 19, doi: 10.4050/F-0079-2023-18067.
[165] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2897–2904, Apr. 2022, doi: 10.1109/LRA.2022.3144777.
[166] W. Chu and Z. Ghahramani, "Preference learning with Gaussian processes," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bonn, Germany, 2005, pp. 137–144, doi: 10.1145/1102351.1102369.
[167] R. K. Cosner et al., "Safety-aware preference-based learning for safety-critical control," in *Proc. Mach. Learn. Res.*, 2022, vol. 168, pp. 1020–1033.
[168] S. Ross, G. J. Gordon, and J. A. Bagnell, "No-regret reductions for imitation learning and structured prediction," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Sardinia, Italy, 2010, pp. 661–668.
[169] A. Didier, R. C. Jacobs, J. Sieber, K. P. Wabersich and M. N. Zeilinger, "Approximate predictive control barrier functions using neural networks: A computationally cheap and permissive safety filter," in *Proc. IEEE 23rd Eur. Contr. Conf. (ECC)*, Bucharest, Romania, 2023, pp. 1231–1237.