Safety-Critical Control With Input Delay in Dynamic Environment

Tamas G. Molnar, *Member, IEEE*, Adam K. Kiss, Aaron D. Ames, *Fellow, IEEE*, and Gábor Orosz, *Senior Member, IEEE*

***Abstract*— Endowing nonlinear systems with safe behavior is increasingly important in modern control. This task is particularly challenging for real-life control systems that operate in dynamically changing environments. This article develops a framework for safety-critical control in dynamic environments, by establishing the notion of *environmental control barrier functions (ECBFs)*. Importantly, the framework is able to guarantee safety even in the presence of *input delay*, by accounting for the evolution of the environment during the delayed response of the system. The underlying control synthesis relies on predicting the future state of the system and the environment over the delay interval, with robust safety guarantees against prediction errors. The efficacy of the proposed method is demonstrated by a simple adaptive cruise control (ACC) problem and a more complex robotics application on a Segway platform.**

***Index Terms*— Delay systems, dynamic environment, predictive control, robust control, safety-critical control.**

I. Introduction

SAFETY is of great importance in many control systems, including a wide spectrum of applications from automated vehicles (AVs) [1], [2] through robotics [3], [4], [5], [6] and multirobot systems [7], [8], [9], to controlling the spread of infectious diseases [10], [11]. Notably, safety is often affected by a dynamic environment that surrounds the control system. For example, robots must avoid collision with other agents in multirobot systems [12], [13], AVs must drive safely among other road users [14], and robotic manipulators must collaborate safely with their human operator [15], [16], [17].

Strict safety requirements call for theoretical safety guarantees and provably safe controllers. Thus, control synthesis must take into account how the control system interacts with its environment, and it must ensure that environment's evolution does not lead to safety violations. As such, dynamic environments pose a major challenge for safety-critical control.

An important element of this challenge is that the response time of control systems may be commensurate with how fast the environment changes. Response times include sensory, feedback, and actuation delays that arise in practice [18]. The magnitude of the delay depends on the application: it

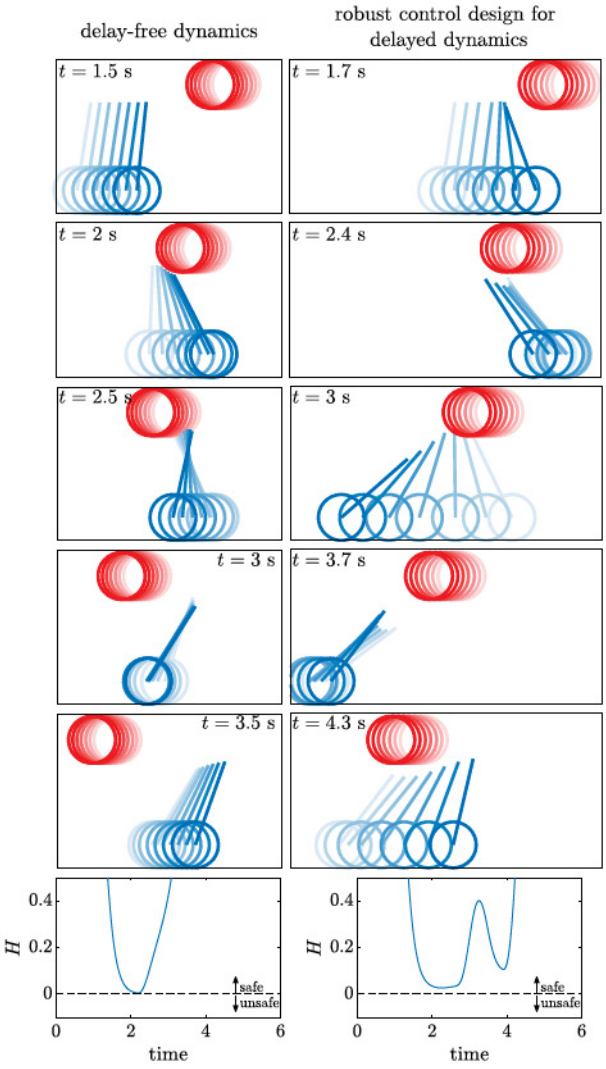

Fig. 1. Proposed safety-critical control framework in high-fidelity simulation of a Segway. The Segway safely avoids a moving obstacle, even when the obstacle's future position is unknown, and there is input delay in the control loop. This is accomplished via the ECBF plotted at the bottom. Observe that the Segway uses different strategies in the delay-free and delayed cases. See video at https://youtu.be/NIImeVnlziM.

Manuscript received 21 September 2022; accepted 26 November 2022. Date of publication 15 December 2022; date of current version 22 June 2023. This research is supported in part by the National Science Foundation (CPS Award #1932091), Aerovironment and Dow (#227027AT), and supported by the NRDI Fund (TKP2020 IES, Grant No. BME-IE-MIFM and TKP2020 NC, Grant No. BME-NC). Recommended by Associate Editor T. Hatanaka. *(Corresponding author: Tamas G. Molnar.)*

Tamas G. Molnar and Aaron D. Ames are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: tmolnar@caltech.edu; ames@caltech.edu).

Adam K. Kiss is with the MTA-BME Lendület Machine Tool Vibration Research Group, Department of Applied Mechanics, Budapest University of Technology and Economics, 1111 Budapest, Hungary (e-mail: kiss_a@mm.bme.hu).

Gábor Orosz is with the Department of Mechanical Engineering and the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: orosz@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2022.3227451.

Digital Object Identifier 10.1109/TCST.2022.3227451

is milliseconds in robotic systems [19], a few tenths of a second in AVs [20], and days in epidemiological models [21]. Delays significantly impact safety in dynamic environments, since by the time the control system responds, the environment may change, and safety may be compromised. To overcome this danger, one must consider how the dynamic—and often uncertain—environment evolves over the delay period, which yields a major challenge in designing provably safe controllers. This article addresses this problem by establishing a framework for safety-critical control that takes dynamic environments and time delays into account explicitly.

A. State of the Art

Formally, safety is often framed as a set invariance problem by requiring the state of the system to evolve within a safe set for all time. The theory of *control barrier functions (CBFs)* provides an elegant solution to achieve this goal [22]. While this theory delivers formal safety guarantees, one shall secure these guarantees in dynamic environments during practical implementation. Several works have built on CBFs to transfer safety-critical controllers from theory to practice, by providing robustness against disturbances [23], [24], [25], [26], [27], measurement uncertainty [28], [29], [30], and model mismatches [31], [32]. Safety in dynamic environments was addressed in [33] and [34] in the context of collision avoidance in multiagent systems by incorporating chance constraints into model predictive control and probabilistic safety barrier certificates, respectively. Furthermore, human assist control was discussed in [35] and [36], in which response to changing environments was handled via time-varying CBFs, including adaptivity to unknown environment parameters and robustness to disturbances.

The safety of time delay systems has been attracting increasing attention. The safety of continuous-time systems with state delay was established by safety functionals in [37] and [38], which were extended to control barrier functionals in [39] and [40]. Discrete-time control systems with input delay were studied in [41] for linear and in [42] for nonlinear dynamics. Linear systems with input delays were addressed in continuous time in [43] and [44] via control barrier and Lyapunov functions. Safety-critical control of continuous-time nonlinear systems with measurement delays was tackled in our works [10], [11] in an application to controlling the spread of COVID-19. These papers leveraged predictor feedback [45], [46], [47], [48] to compensate the delay by predicting the future evolution of the system.

Remarkably, Singletary et al. [42] also relied on predictor feedback to compensate input delay. The underlying theory was established in discrete time by assuming that the system's evolution is predicted accurately. As opposed, here, we consider continuous-time systems and address prediction errors. Parallel to our work, Abel et al. [49] proposed predictor feedback in continuous time for compensating multiple input delays in safety-critical control, wherein input channels with shorter delays were used to keep the system safe until longer delays were compensated. Abel et al. [50] modified the predictor to compensate time-varying input delay and achieve safety. Molnar et al. [51] endowed safety-critical predictor feedback controllers with robustness against disturbances. Yet, these works have not addressed safety in dynamic environments that evolve independently of the control input. This article intends to fill this gap and tackle the challenges arising from the combination of dynamic environments and delays.

In this article, we *explicitly* involve *dynamically changing environments* into the framework for safety-critical control, in order to handle uncertain environments with worst case safety guarantees in a deterministic fashion. Importantly, our framework also allows us to compensate the effect of input delay, which was not addressed in the literature.

B. Contributions

Here, we build on [10] and [11] to establish the theory of safety-critical control for nonlinear continuous-time systems with input delay, operating in dynamically changing environments. Our contributions are threefold.

1) We establish the notion of *environmental control barrier functions (ECBFs)* for delay-free systems to explicitly address scenarios in which safety is affected by a dynamic environment. This notion is particularly useful when the dynamics of the environment are inherently more uncertain than those of the control system.
2) We connect the theories of CBFs and predictor feedback by developing the notions of CBFs and ECBFs for systems with input delay and synthesizing safety-critical controllers via predictor feedback. Predictors require special care in dynamic environments, as the environment's future cannot be predicted accurately. Thus, we make controllers robust against prediction errors.
3) We demonstrate the efficacy of this framework on real-life engineering systems where time delays and dynamic environments both occur, through the examples of adaptive cruise control (ACC) and obstacle avoidance with a Segway.

Fig. 1 illustrates a sample of these results. A Segway is controlled to safely avoid a moving obstacle via the proposed ECBFs in high-fidelity simulation. Without delay in its control loop (left), the Segway pitches backward to go under the obstacle. With input delay (right), the Segway approaches the obstacle, then moves in reverse to make space, and pitches forward to go under it. Remarkably, these safe behaviors emerge from the ECBF automatically, which handles reactive planning in a holistic fashion.

This article is structured as follows. Section II revisits CBFs for delay-free systems. Section III addresses safety in dynamic environments by introducing ECBFs. Section IV extends CBFs and ECBFs to systems with input delays and discusses safety-critical control via predictor feedback with robustness against prediction errors. In these sections, ACC is used as illustrative example, whereas Section V demonstrates the safety-critical control of a Segway by numerical simulations. We conclude our work in Section VI.

II. Preliminaries to Safety-Critical Control

Consider a control-affine system with state $x(t) \in X \subseteq \mathbb{R}^n$ and control input $u(t) \in U \subseteq \mathbb{R}^m$

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

where $f : X \to \mathbb{R}^n$ and $g : X \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous on X. Let $x(0) = x_0 \in X$ be the initial condition. When the input $u = k(x)$ is given by a locally Lipschitz continuous controller $k : X \to U$, system (1) has a unique solution over a time interval $t \in I(x_0)$. For simplicity, we assume $I(x_0) = [0, \infty)$; i.e., the solution exists for all $t \geq 0$.

We consider the system safe if its state is contained within a *safe set* $S \subset X$ for all time. Accordingly, we frame safety-critical control as rendering set S forward invariant under dynamics (1): the controller needs to ensure for all $x_0 \in S$ that $x(t) \in S$, $\forall t \geq 0$. Specifically, we define S as the zero-superlevel set of a continuously differentiable function $h : X \to \mathbb{R}$

$$S = \{x \in X : h(x) \geq 0\} \tag{2}$$

where the selection of h is application-driven.

A. Control Barrier Functions

We ensure the forward invariance of the safe set S by the framework of *CBFs*. First, we briefly revisit the main result in [22] that establishes the definition of CBFs and the theoretical safety guarantees. We use the notation $\|\cdot\|$ for Euclidean norm, and we call a function $\alpha : (-a, b) \to \mathbb{R}$, $a, b > 0$ as extended class $\mathcal{K}$ function, if it is continuous, strictly monotonically increasing and $\alpha(0) = 0$.

Definition 1: Function h is a CBF for (1) if there exists an extended class $\mathcal{K}$ function α, such that for all $x \in S$

$$\sup_{u\in U} \dot{h}(x, u) > -\alpha(h(x)) \tag{3}$$

where

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

is the derivative of h along system (1).

Note that sup becomes max if U is compact. With the CBF definition, [22] establishes formal safety guarantees as follows.

Theorem 1 [22]: If h is a CBF for (1), then any locally Lipschitz continuous controller $u = k(x)$ satisfying

$$\dot{h}(x, u) \geq -\alpha(h(x)) \tag{5}$$

$\forall x \in S$ renders S forward invariant (safe); i.e., it ensures $x_0 \in S \Rightarrow x(t) \in S$, $\forall t \geq 0$.

The proof can be found in [22], and further technical details with discussion about the selection of α are in [52]. Throughout this article, we use the variants of the safety condition (5).

Remark 1: Condition (5) is often used in the context of optimization-based controllers [22]. Given a control input $u_\mathrm{d} = k_\mathrm{d}(x)$ by a desired controller $k_\mathrm{d} : X \to U$, one can modify this input in a minimally invasive fashion to guarantee safety by solving the following quadratic program (QP):

$$\begin{aligned} k(x) = \arg\min_{u\in U} \ & \|u - k_\mathrm{d}(x)\|^2 \\ \text{s.t. } & \dot{h}(x, u) \geq -\alpha(h(x)). \end{aligned} \tag{6}$$

This defines the control law $u = k(x)$ *implicitly*. The feasibility of this QP is guaranteed by the definition of CBFs (Definition 1). However, verifying that a given h is indeed a CBF is nontrivial when there are input constraints ($U \subset \mathbb{R}^m$). An approach to overcome input constraints is the backup set method [17], that relies on the forward integration of the dynamics similar to predictor feedback presented in Section IV. Otherwise, without input bounds ($U = \mathbb{R}^m$), feasibility guarantees can be proven, and the solution to QP (6) can even be expressed *explicitly* based on the Karush–Kuhn–Tucker (KKT) conditions [53] as $k(x) = k_\mathrm{d}(x)$ if $\nabla h(x)g(x) = 0$ and

$$\begin{aligned} k(x) &= k_\mathrm{d}(x) + \max\{-\phi_0(x), 0\}\phi_1^+(x) \\ \phi_0(x) &= \nabla h(x)(f(x) + g(x)k_\mathrm{d}(x)) + \alpha(h(x)) \\ \phi_1(x) &= \nabla h(x)g(x) \end{aligned} \tag{7}$$

if $\nabla h(x)g(x) \neq 0$, where $\phi_1^+(x) = \phi_1^\top(x)/(\phi_1(x)\phi_1^\top(x))$ is the right pseudo-inverse of $\phi_1(x)$. The derivation of (7) is given in Appendix I. Note that if $\nabla h(x)g(x) \neq 0$, $\forall x \in S$, it is often referred to as h has *relative degree* 1 (i.e., the first derivative of h with respect to time is affected by u). For higher relative degrees (when a higher derivative of h is affected by u), there exist systematic methods to construct CBFs from h and guarantee safety; see [54], [55], [56], and [57] for details. An example for such extension is given later in Section V.

III. Safety in Dynamic Environment

So far, we related safety to the state $x(t)$ of the system. Often safety is also affected by the state of the environment, which we characterize by $e(t) \in E \subseteq \mathbb{R}^l$, where e is a continuously differentiable function of time with $\dot{e}(t) \in \mathcal{E} \subseteq \mathbb{R}^l$ and $e(0) = e_0 \in E$. This leads to an *environmental safe set* S_e

$$S_\mathrm{e} = \{(x, e) \in X \times E : H(x, e) \geq 0\} \tag{8}$$

where $H : X \times E \to \mathbb{R}$ is assumed to be continuously differentiable in both arguments.

A. Environmental CBFs

We enforce safety in dynamic environments by introducing the notion of *ECBFs*.

Definition 2: Function H is an ECBF for (1) if there exists an extended class $\mathcal{K}$ function α, such that for all $(x, e) \in S_\mathrm{e}$ and $\dot{e} \in \mathcal{E}$

$$\sup_{u\in U} \dot{H}(x, e, \dot{e}, u) > -\alpha(H(x, e)) \tag{9}$$

where

$$\dot{H}(x, e, \dot{e}, u) = \nabla_x H(x, e)(f(x) + g(x)u) + \nabla_e H(x, e)\dot{e} \tag{10}$$

is the derivative of H along system (1).

ECBFs are a time-dependent extension of CBFs, wherein the dependence on time is considered through the state $e(t)$ of the environment. This will facilitate addressing environment uncertainty in Section III-B. The ECBF condition (9) is directly related to the CBF condition (3), with an additional term in the derivative with respect to time. Further literature on time-varying CBFs can be found, for example, in [35] and [36].

Via the ECBF, an extension of Theorem 1 yields theoretical safety guarantees in dynamic environments, as given below.

Theorem 2: If H is an ECBF for (1), then any locally Lipschitz continuous controller $u = K(x, e, \dot{e})$ satisfying

$$\dot{H}(x, e, \dot{e}, u) \geq -\alpha(H(x, e)) \tag{11}$$

$\forall (x, e) \in S_e$ and $\forall \dot{e} \in \mathcal{E}$ renders S_e forward invariant; i.e., it ensures $(x_0, e_0) \in S_e \Rightarrow (x(t), e(t)) \in S_e$, $\forall t \geq 0$.

Proof: Equation (1) and its environment form the augmented system

$$\dot{z} = F(z) + G(z)v \tag{12}$$

with augmented state z, input v, and dynamics F and G as follows:

$$z = \begin{bmatrix} x \\ e \end{bmatrix}, \quad v = \begin{bmatrix} u \\ \dot{e} \end{bmatrix}, \quad F(z) = \begin{bmatrix} f(x) \\ 0 \end{bmatrix}, \quad G(z) = \begin{bmatrix} g(x) \\ I \end{bmatrix}. \tag{13}$$

For this system, function $H_z : X \times E \to \mathbb{R}$, $H_z(z) = H(x, e)$ is a CBF, since H is an ECBF. Based on Theorem 1, safety is guaranteed with respect to the zero-superlevel set of H_z by

$$\dot{H}_z(z, v) \geq -\alpha(H_z(z)). \tag{14}$$

Substituting the definitions of z, v, F, G, and H_z leads to (11) and proves the statement in Theorem 2. ■

Remark 2: Theorem 2 yields safety-critical controllers of the form $K : X \times E \times \mathcal{E} \to U$, $u = K(x, e, \dot{e})$ that depend on the environment as well through e and $\dot{e}$. For example, a controller based on optimization (specifically, a QP) reads

$$\begin{aligned} K(x, e, \dot{e}) = \arg\min_{u \in U} \ & \|u - K_\mathrm{d}(x, e, \dot{e})\|^2 \\ \text{s.t. } & \dot{H}(x, e, \dot{e}, u) \geq -\alpha(H(x, e)) \end{aligned} \tag{15}$$

analogously to (6), with explicit solution for $U = \mathbb{R}^m$

$$\begin{aligned} K(x, e, \dot{e}) &= K_\mathrm{d}(x, e, \dot{e}) + \max\{-\Phi_0(x, e, \dot{e}), 0\}\Phi_1^+(x, e) \\ \Phi_0(x, e, \dot{e}) &= \nabla_x H(x, e)(f(x) + g(x)K_\mathrm{d}(x, e, \dot{e})) \\ &\quad + \nabla_e H(x, e)\dot{e} + \alpha(H(x, e)) \\ \Phi_1(x, e) &= \nabla_x H(x, e)g(x) \end{aligned} \tag{16}$$

analogously to (7) if $\nabla_x H(x, e)g(x) \neq 0$.

B. Robust Safety in Uncertain Environment

ECBFs rely on the environment's state e and its derivative $\dot{e}$. In practice, these quantities are typically estimated with uncertainty. Thus, now, we robustify safety-critical controllers against uncertainties in the environment. Motivated by the method developed in [30] for handling state uncertainty, we provide robustness based on worst case uncertainty bounds (i.e., in a deterministic fashion). For simplicity, we consider no uncertainty in x, since the environment is typically associated with more uncertainty than the state of the control system.

Consider that the true environment state e and its derivative $\dot{e}$ are not available, only some estimates $\hat{e}$ and $\hat{\dot{e}}$. We assume these estimates have known uncertainty bounds ε_e and $\varepsilon_{\dot{e}}$

$$\|e - \hat{e}\| \leq \varepsilon_e, \quad \left\|\dot{e} - \hat{\dot{e}}\right\| \leq \varepsilon_{\dot{e}}. \tag{17}$$

While it may be nontrivial to find such error bounds, conservative over-approximations of uncertainty bounds are usually available in practice for many perception, measurement, or state estimation algorithms. As such, the approach proposed below is limited to setups with known uncertainty bounds, since safety is guaranteed by considering the worst case scenario.

The main idea is to enforce safety through a conservative lower bound on the unknown expression $\dot{H}(x, e, \dot{e}, u) + \alpha(H(x, e))$ that must be kept nonnegative per Theorem 2. The bound uses the known quantities $\hat{e}$ and $\hat{\dot{e}}$ in the form

$$\begin{aligned} &\dot{H}(x, e, \dot{e}, u) + \alpha(H(x, e)) \\ &\quad \geq \dot{H}(x, \hat{e}, \hat{\dot{e}}, u) + \alpha(H(x, \hat{e})) - C(\varepsilon_e, \varepsilon_{\dot{e}}, u) \geq 0. \end{aligned} \tag{18}$$

This is stated more formally with the specific expression of $C(\varepsilon_e, \varepsilon_{\dot{e}}, u)$ below, after some additional assumptions.

Assume that the following regularity conditions on H hold. Functions $\nabla_x H(x, e)f(x)$, $\nabla_x H(x, e)g(x)$, and $\alpha(H(x, e))$ are Lipschitz continuous in argument e on S_e with Lipschitz coefficients $\mathcal{L}_{\nabla Hf,e}$, $\mathcal{L}_{\nabla Hg,e}$, and $\mathcal{L}_{\alpha\circ H,e}$, whereas $\nabla_e H(x, e)\dot{e}$ is Lipschitz continuous in e and $\dot{e}$ on $S_e \times \mathcal{E}$ with Lipschitz coefficients $\mathcal{L}_{\nabla H\dot{e},e}$ and $\mathcal{L}_{\nabla H\dot{e},\dot{e}}$. This implies

$$\begin{aligned} \nabla_x Hf|_{x,e} - \nabla_x Hf|_{x,\hat{e}} &\geq -\mathcal{L}_{\nabla Hf,e}\|e - \hat{e}\| \\ \left(\nabla_x Hg|_{x,e} - \nabla_x Hg|_{x,\hat{e}}\right)u &\geq -\mathcal{L}_{\nabla Hg,e}\|e - \hat{e}\|\|u\| \\ \nabla_e H\dot{e}|_{x,e,\dot{e}} - \nabla_e H\dot{e}|_{x,\hat{e},\hat{\dot{e}}} &\geq -\mathcal{L}_{\nabla H\dot{e},e}\|e - \hat{e}\| \\ &\quad - \mathcal{L}_{\nabla H\dot{e},\dot{e}}\|\dot{e} - \hat{\dot{e}}\| \\ \alpha \circ H|_{x,e} - \alpha \circ H|_{x,\hat{e}} &\geq -\mathcal{L}_{\alpha\circ H,e}\|e - \hat{e}\|. \end{aligned} \tag{19}$$

This leads to the following sufficient condition for safety.

Proposition 1: If H is an ECBF for (1) and the regularity conditions in (19) hold, then any locally Lipschitz continuous controller $u = K(x, \hat{e}, \hat{\dot{e}})$ satisfying

$$\dot{H}(x, \hat{e}, \hat{\dot{e}}, u) - C(\varepsilon_e, \varepsilon_{\dot{e}}, u) \geq -\alpha\big(H(x, \hat{e})\big) \tag{20}$$

with

$$\begin{aligned} C(\varepsilon_e, \varepsilon_{\dot{e}}, u) = (&\mathcal{L}_{\nabla Hf,e} + \mathcal{L}_{\alpha\circ H,e} + \mathcal{L}_{\nabla H\dot{e},e})\varepsilon_e \\ &+ \mathcal{L}_{\nabla H\dot{e},\dot{e}}\varepsilon_{\dot{e}} + \mathcal{L}_{\nabla Hg,e}\varepsilon_e\|u\| \end{aligned} \tag{21}$$

$\forall (x, \hat{e}) \in S_e$ and $\forall \hat{\dot{e}} \in \mathcal{E}$ renders S_e forward invariant; i.e., it ensures $(x_0, e_0) \in S_e \Rightarrow (x(t), e(t)) \in S_e$, $\forall t \geq 0$.

Proof: The steps of the proof follow those of [30, Theorem 2]: we show that (20) implies (11), and we apply Theorem 2. We relate (20) to (11) by introducing the difference between their corresponding terms. By using (10), we get

$$\begin{aligned} &\dot{H}(x, e, \dot{e}, u) + \alpha(H(x, e)) \\ &\quad = \dot{H}(x, \hat{e}, \hat{\dot{e}}, u) + \alpha(H(x, \hat{e})) \\ &\qquad + \nabla_x Hf|_{x,e} - \nabla_x Hf|_{x,\hat{e}} + \left(\nabla_x Hg|_{x,e} - \nabla_x Hg|_{x,\hat{e}}\right)u \\ &\qquad + \nabla_e H\dot{e}|_{x,e,\dot{e}} - \nabla_e H\dot{e}|_{x,\hat{e},\hat{\dot{e}}} + \alpha \circ H|_{x,e} - \alpha \circ H|_{x,\hat{e}}. \end{aligned} \tag{22}$$

These differences show up in (19). Thus, the regularity conditions (19) on H, the uncertainty bound (17), and conditions (20) and (21) imply (11), which completes the proof. ■

Less conservative problem-specific bounds than (21) also work as long as they imply (11). Furthermore, we highlight that (21) involves the term $\|u\|$. This, when incorporated into an optimization problem, such as (6), leads to a second-order cone program (SOCP) rather than a QP if $\mathcal{L}_{\nabla Hg,e} \neq 0$.

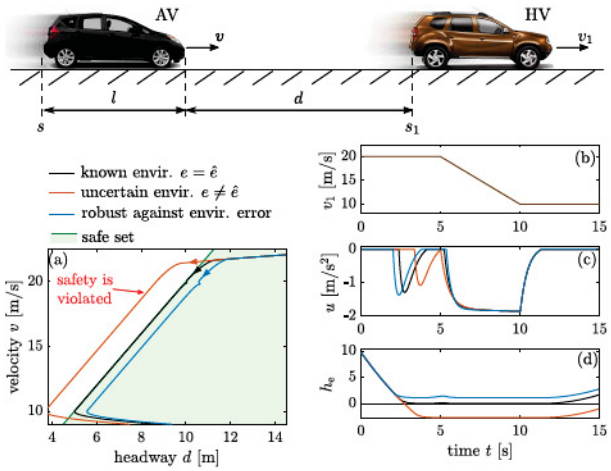

Fig. 2. Strategies for ACC where an AV intends to safely follow an HV; (a) phase portrait, (b) velocity of HV, (c) control input of AV, and (d) ECBF. The HV represents an environment for the AV. In the ideal scenario where the HV's position and speed are accurately known to the AV, controller (15) ensures safety (black). When the HV's position and speed are measured with error (the environment is uncertain), controller (15) violates safety (red). When controller (15) is robustified via constraint (20), safety is maintained even in the presence of environment uncertainty (blue).

Example 1 (ACC): We consider an ACC problem, where an AV intends to follow a human-driven vehicle (HV) without collision; see Fig. 2. This problem has been studied previously without the notion of ECBFs. In [1], polyhedral-controlled invariant sets and finite-state abstraction were used, and in [2] and [22], CBFs were applied. We revisit this problem, use ECBFs to tackle it, and demonstrate that this framework allows us to explicitly take into account uncertainties in the HV's motion. This will also play an essential role in Section IV to extend the resulting safety-critical controller to safe ACC with input delay, which was not addressed in [1], [2], and [22].

We denote the length of the AV by l, the position of its rear bumper by s, and its speed by v, and we model its motion by

$$\underbrace{\begin{bmatrix}\dot{s}\\ \dot{v}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}v\\ -p(v)\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{g(x)} u \tag{23}$$

where $p(v)$ indicates resistance terms. The input u is acceleration command that is assumed to be realized by a low-level controller. The HV's position and speed, denoted by s_1 and v_1, characterize the environment for the AV: $e = s_1$ and $\dot{e} = v_1$.

To avoid collisions, the AV intends to keep its speed v below a safe limit $\bar{\kappa} d$ for a selected $\bar{\kappa} > 0$, where this limit depends on the distance $d = s_1 - s - l$. Thus, we use the ECBF

$$H(x, e) = \bar{\kappa}(s_1 - s - l) - v \tag{24}$$

and a linear extended class $\mathcal{K}$ function $\alpha(h) = \gamma h$ with $\gamma > 0$. For this choice, we have $\nabla_x H(x, e) f(x) = -\bar{\kappa} v + p(v)$, $\nabla_x H(x, e) g(x) = -1$, and $\nabla_e H(x, e) \dot{e} = \bar{\kappa} v_1$.

Substituting these expressions into (10) and the safety condition (11) leads to

$$\bar{\kappa}(v_1 - v) + \gamma\left(\bar{\kappa}(s_1 - s - l) - v\right) + p(v) \geq u. \tag{25}$$

Hence, the AV should not accelerate more than the expression on the left-hand side. This expression resembles the desired acceleration of simple ACC controllers; in fact, for $p(v) = 0$, it is equivalent to the one in [2] with a special choice of feedback gains and range policy. Enforcing (25), for example, through the QP (15), guarantees safety based on Theorem 2.

Fig. 2 shows numerical simulation results with the safety-critical controller for $p(v) = 0.1 + 0.0003\, v^2$, $\gamma = 3$, and $\bar{\kappa} = 2$ (with units in SI). The HV performs constant speed cruising, braking with 2 m/s^2 and constant speed cruising again; see Fig. 2(b). The AV intends to travel at a constant speed higher than the HV's speed with desired controller $K_d(x, e, \dot{e}) = 0$. By applying the QP (15) with constraint (25), the AV is able to slow down safely behind the HV; see Fig. 2 (black curve).

The controller relies on the position and speed of the HV. These can be obtained by onboard sensors, such as radar, LiDAR, cameras, or ultrasonics, or by vehicle-to-vehicle connectivity with the HV. If these quantities are measured with error, safety may be violated. This is demonstrated in Fig. 2 (red color), where the controller relies on the measured values $\hat{e} = \hat{s}_1 = s_1 + 1$ m and $\hat{\dot{e}} = \hat{v}_1 = v_1 + 1$ m/s instead of the true values $e = s_1$ and $\dot{e} = v_1$. Overestimating the position and speed of the HV causes the system to leave the safe set.

The controller can be made robust to such uncertainties in the environment by replacing the safety condition (11) with the robustified constraint (20) in the QP (15). If the HV's position and speed estimates have known error bounds ε_s and ε_v, that is, $|s_1 - \hat{s}_1| \leq \varepsilon_s$ and $|v_1 - \hat{v}_1| \leq \varepsilon_v$, then, after substitution into (20) and using (21), the robustified constraint becomes

$$\bar{\kappa}(v_1 - \varepsilon_v - v) + \gamma\left(\bar{\kappa}(s_1 - \varepsilon_s - s - l) - v\right) + p(v) \geq u \tag{26}$$

where we used the Lipschitz coefficients $\mathcal{L}_{\nabla H f, e} = \mathcal{L}_{\nabla H g, e} = \mathcal{L}_{\nabla H \dot{e}, e} = 0$, $\mathcal{L}_{\alpha \circ H, e} = \gamma \bar{\kappa}$, and $\mathcal{L}_{\nabla H \dot{e}, \dot{e}} = \bar{\kappa}$. In this example, the additional robustifying terms are equivalent to considering the worst case (smallest possible) position and speed for the HV.

The effect of these robustifying terms is shown in Fig. 2 (blue color) for $\varepsilon_s = 1.4$ m and $\varepsilon_v = 1.4$ m/s. The AV is able to safely slow down behind the HV despite the uncertainty in the HV's measured state. Notice that the controller is slightly conservative: the AV stays farther from the boundary of the safe set than in the case without uncertainty.

IV. Safety of Systems With Input Delay

Now, consider the system with input delay $\tau > 0$

$$\dot{x}(t) = f(x(t)) + g(x(t)) u(t - \tau) \tag{27}$$

where f and g are the same as in (1), and u is bounded and continuous almost everywhere (with a potential discontinuity at $t = 0$ when the controller is turned on). We still assume that there exists a unique solution $x(t)$ over $t \geq 0$.

A. Solution of the System and Predictors

To synthesize safety-critical controllers, we ensure that given the state $x(t)$ at time t, the solution of (27) continues to be safe over $[t, t + \tau]$. This property depends on the instantaneous control input $u(t)$ to be synthesized via CBFs and also on the input history over $[t - \tau, t)$ given by $u_t \in \mathcal{B}$

$$u_t(\theta) = u(t + \theta), \quad \theta \in [-\tau, 0). \tag{28}$$

Here, $\mathcal{B}$ denotes the space of functions mapping from $[-\tau, 0)$ to U that are bounded and continuous almost everywhere.

The solution over $[t, t+\tau]$ is characterized by the *semi-flow* $\Psi : [0, \tau] \times X \times \mathcal{B} \to X$ as a function of the state $x(t)$ and as a functional of the input history u_t

$$x(t+\vartheta) = \Psi(\vartheta, x(t), u_t), \quad \vartheta \in [0, \tau]. \tag{29}$$

The semi-flow is obtained by the forward integration of (27)

$$\Psi(\vartheta, x, u_t) = x + \int_0^{\vartheta} \Big(f\big(\Psi(\varphi, x, u_t)\big) + g\big(\Psi(\varphi, x, u_t)\big) u_t(\varphi - \tau) \Big) \mathrm{d}\varphi. \tag{30}$$

Of particular interest will be the state $x(t+\tau)$, that reads

$$x(t+\tau) = \Psi(\tau, x(t), u_t). \tag{31}$$

We remark that, since u is bounded, $u(t)$ does not affect the value of the integral, and thus, u_t is defined over $[-\tau, 0)$. That is, the input history u_t does not include the instantaneous control input $u(t)$. This will allow us to utilize the input history u_t when synthesizing the control input $u(t)$.

Hereinafter, $x(t+\tau) = \Psi(\tau, x(t), u_t)$ is called *predicted state*, and (30) serves as *predictor*. The predicted state will play a key role in safety-critical control. It can be calculated by forward integration of (27) over $[t, t+\tau]$. Explicit expressions are available for linear systems with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$

$$\dot{x}(t) = Ax(t) + Bu(t-\tau) \tag{32}$$

where the predicted state is given by the convolution integral

$$\Psi(\tau, x(t), u_t) = \mathrm{e}^{A\tau} x(t) + \int_0^{\tau} \mathrm{e}^{A(\tau-\vartheta)} B u_t(\vartheta - \tau) \mathrm{d}\vartheta. \tag{33}$$

Note that predictors also exist for systems with time-varying and state-dependent delays, as given in [46], that have also been considered in the context of safety in [50]. While the upcoming theorems are stated for constant delay, they could be extended to varying delays by using the appropriate predictor.

B. CBFs With Input Delay

The following definition generalizes CBFs for systems with input delay in the form (27) with $\tau > 0$.

Definition 3: Function h is a CBF for (27) with $\tau > 0$ if there exists an extended class $\mathcal{K}$ function α, such that for all $x \in S$ and $u_t \in \mathcal{B}$

$$\sup_{u \in U} \dot{h}(x_\mathrm{p}, u) > -\alpha(h(x_\mathrm{p})) \tag{34}$$

where $x_\mathrm{p} = \Psi(\tau, x, u_t)$ with Ψ given by (30).

The definition recovers Definition 1 in the delay-free case, since $x_\mathrm{p} = x$ if $\tau = 0$. With this definition, we guarantee safety analogously to Theorem 1. We assume that safety-critical control starts at $t = 0$. According to (29), $x(\vartheta) = \Psi(\vartheta, x_0, u_0)$, $\vartheta \in [0, \tau]$; that is, the solution over $[0, \tau]$ evolves based on the initial input history u_0, which we cannot prescribe. Therefore, we need the following assumption to ensure safety over $[0, \tau]$.

Assumption 1: The initial history u_0 of the control input satisfies $x(\vartheta) = \Psi(\vartheta, x_0, u_0) \in S$, $\forall \vartheta \in [0, \tau]$.

Now, we are ready to state our main theorem that ensures safety in the presence of the input delay $\tau > 0$.

Theorem 3: If h is a CBF for (27) with $\tau > 0$, then any locally Lipschitz continuous controller $u = k(x_\mathrm{p})$, $x_\mathrm{p} = \Psi(\tau, x, u_t)$ with input history u_t satisfying

$$\dot{h}(x_\mathrm{p}, u) \geq -\alpha(h(x_\mathrm{p})) \tag{35}$$

$\forall x \in S$ and $\forall u_t \in \mathcal{B}$ renders S forward invariant under Assumption 1; i.e., it ensures $x_0 \in S \Rightarrow x(t) \in S$, $\forall t \geq 0$.

Proof: Since Assumption 1 ensures $x(\vartheta) \in S$, $\forall \vartheta \in [0, \tau]$, it is sufficient to prove $x(\tau) \in S \Rightarrow x(t) \in S$, $\forall t \geq \tau$. By differentiation of (30) with respect to ϑ, we have

$$\frac{\mathrm{d}}{\mathrm{d}\vartheta} \Psi(\vartheta, x(t), u_t) = f\big(\Psi(\vartheta, x(t), u_t)\big) + g\big(\Psi(\vartheta, x(t), u_t)\big) u(t + \vartheta - \tau). \tag{36}$$

Furthermore, by noticing $(\mathrm{d}/\mathrm{d}\vartheta) x(t+\vartheta) = (\mathrm{d}/\mathrm{d}t) x(t+\vartheta)$ and by using (29), we get $(\mathrm{d}/\mathrm{d}\vartheta)\Psi(\vartheta, x(t), u_t) = (\mathrm{d}/\mathrm{d}t)\Psi(\vartheta, x(t), u_t)$. Substituting this into (36) and using $\vartheta = \tau$, we get the following delay-free system for $x_\mathrm{p}(t) = \Psi(\tau, x(t), u_t)$:

$$\dot{x}_\mathrm{p}(t) = f(x_\mathrm{p}(t)) + g(x_\mathrm{p}(t)) u(t). \tag{37}$$

For this system, Theorem 1 can be applied, since (34) and (35) hold; thus, we get $x_\mathrm{p}(0) \in S \Rightarrow x_\mathrm{p}(t) \in S$, $\forall t \geq 0$ that is equivalent to $x(\tau) \in S \Rightarrow x(t) \in S$, $\forall t \geq \tau$. ■

Remark 3: As opposed to the delay-free case, the controller in Theorem 3 is no longer a state-feedback controller, but it also depends on the input history u_t through the predicted state $x_\mathrm{p} = \Psi(\tau, x, u_t)$. Furthermore, optimization-based controllers for systems with input delay can be synthesized via Theorem 3 similar to (6). The following QP can be solved if $\tau > 0$:

$$k(x_\mathrm{p}) = \arg\min_{u \in \mathbb{R}^m} \ \|u - k_\mathrm{d}(x_\mathrm{p})\|^2 \quad \text{s.t. } \dot{h}(x_\mathrm{p}, u) \geq -\alpha(h(x_\mathrm{p})). \tag{38}$$

Here, the desired controller $k_\mathrm{d} : X \to U$ may also account for the delay and can potentially depend on the predicted state. The solution to (38) is equivalent to applying the control law (6) of the corresponding delay-free system on the predicted state $x_\mathrm{p} = \Psi(\tau, x, u_t)$. This allows one to extend *explicitly* available delay-free control laws, such as (7), for systems with input delays. However, an explicit expression for k is not always available, especially if additional constraints are added to (6). In such cases, one cannot construct u by separately solving the delay-free QP (6) and calculating the predicted state x_p, but one needs to solve QP (38) directly.

Remark 4: In practice, predicting the future state may not be perfectly accurate. Often, only an estimate $\hat{x}_\mathrm{p}$ of the predicted state x_p is available. Classically, this estimate is provided by the numerical forward integration of (27). Alternatively, state prediction can also be done by more modern tools, such as data-driven methods and machine learning. Theorem 3 guarantees safety for the ideal scenario of accurate prediction, $\hat{x}_\mathrm{p} = x_\mathrm{p}$. However, mismatches between $\hat{x}_\mathrm{p}$ and x_p inevitably occur due to model uncertainties and computation errors [47], and longer prediction (larger delay) typically yields larger *prediction error* $\hat{x}_\mathrm{p} - x_\mathrm{p}$. The effect of prediction errors can be studied via the notion of input-to-state safety [24], [27], as we did in [11], where we showed that the input disturbance

$d = k(\hat{x}_\mathrm{p}) - k(x_\mathrm{p})$ makes a larger set $S_\mathrm{d} \supseteq S$ forward invariant. Alternatively, if the prediction error is bounded and there exists $\varepsilon_x > 0$ such that $\|\hat{x}_\mathrm{p} - x_\mathrm{p}\| \leq \varepsilon_x$, robustness against the prediction error can be provided analogously to Proposition 1 using the following condition:

$$\dot{h}(\hat{x}_\mathrm{p}, u) - (\mathcal{L}_{\nabla hf} + \mathcal{L}_{\alpha oh})\varepsilon_x - \mathcal{L}_{\nabla hg}\varepsilon_x\|u\| \geq -\alpha\big(h(\hat{x}_\mathrm{p})\big) \tag{39}$$

where $\mathcal{L}$ is the Lipschitz coefficient of the subscripted function on S. This method was originally used in [30] to address mismatches between estimated and true states ($\hat{x}$ and x) of delay-free systems. While safety is guaranteed, the additional terms may lead to conservative behavior where the system evolves far away from the safe set boundary. The conservatism depends on the error bound ε_x and the Lipschitz coefficients.

C. Safety With Input Delay in Dynamic Environment

Finally, we consider the scenario when safety needs to be guaranteed for the time delay system (27) in a dynamic environment described by the state $e(t)$ and the environmental safe set S_e. We assume that the state $e(t)$ of the environment is a continuously differentiable function of time.[1]

In Theorem 3, the key step to achieve safety was to predict the system's state over the time interval $[t, t+\tau]$. Now, we make a *prediction of the environment* and rely on its future state $e_\mathrm{p}(t) = e(t+\tau)$. Typically, the future state $e_\mathrm{p}(t)$ depends on the current state $e(t)$, as emphasized by the notation

$$e(t+\vartheta) = \Gamma(\vartheta, e(t)), \quad \vartheta \in [0, \tau] \tag{40}$$

where the map $\Gamma : [0,\tau] \times E \to E$ may be unknown and may involve dependence on other quantities as well.

For this setup, we establish safety by extending the notion of ECBFs to systems with input delay.

Definition 4: Function H is an ECBF for (27) with $\tau > 0$ if there exists an extended class $\mathcal{K}$ function α, such that for all $(x, e) \in S_\mathrm{e}$, $\dot{e} \in \mathcal{E}$ and $u_t \in \mathcal{B}$

$$\sup_{u\in U} \dot{H}(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p}, u) > -\alpha(H(x_\mathrm{p}, e_\mathrm{p})) \tag{41}$$

where $x_\mathrm{p} = \Psi(\tau, x, u_t)$ with Ψ given by (30), while $e_\mathrm{p} = \Gamma(\tau, e)$ with Γ defined by (40).

With this definition, Theorems 2 and 3, that separately guarantee safety in dynamic environment and for input delay, can be integrated into Theorem 4. Again, we make a preliminary assumption that the system is safe over the interval $t \in [0, \tau]$ when safety depends on the initial input history u_0.

Assumption 2: The initial history u_0 of the control input satisfies $(x(\vartheta), e(\vartheta)) = (\Psi(\vartheta, x_0, u_0), \Gamma(\vartheta, e_0)) \in S_\mathrm{e}$, $\forall \vartheta \in [0, \tau]$.

Now, we can state the main theorem to ensure safety for systems with input delay in dynamic environment.

Theorem 4: If H is an ECBF for (27) with $\tau > 0$, then any locally Lipschitz continuous controller $u = K(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p})$, $x_\mathrm{p} = \Psi(\tau, x, u_t)$, and $e_\mathrm{p} = \Gamma(\tau, e)$ with history u_t satisfying

$$\dot{H}(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p}, u) \geq -\alpha(H(x_\mathrm{p}, e_\mathrm{p})) \tag{42}$$

[1]In case of a higher relative degree $r > 1$, the state $e(t)$ of the environment must be r times continuously differentiable. While we omit in-depth discussion about higher relative degrees, an example is shown in Section V.

$\forall (x, e) \in S_\mathrm{e}$, $\forall \dot{e} \in \mathcal{E}$, and $\forall u_t \in \mathcal{B}$ renders S_e forward invariant under Assumption 2; i.e., it ensures $(x_0, e_0) \in S_\mathrm{e} \Rightarrow (x(t), e(t)) \in S_\mathrm{e}$, $\forall t \geq 0$.

Proof: Assumption 2 yields $(x_0, e_0) \in S_\mathrm{e} \Rightarrow (x(\vartheta), e(\vartheta)) \in S_\mathrm{e}$, $\forall \vartheta \in [0, \tau]$; thus, what remains to prove is $(x(\tau), e(\tau)) \in S_\mathrm{e} \Rightarrow (x(t), e(t)) \in S_\mathrm{e}$, $\forall t \geq \tau$. This is equivalent to $(x_\mathrm{p}(0), e_\mathrm{p}(0)) \in S_\mathrm{e} \Rightarrow (x_\mathrm{p}(t), e_\mathrm{p}(t)) \in S_\mathrm{e}$, $\forall t \geq 0$ based on the definitions of x_p and e_p. According to the proof of Theorem 3, $x_\mathrm{p}(t)$ is governed by the delay-free dynamics (37). Hence, Theorem 2 is directly applicable to this delay-free system considering the environment given by e_p. This provides $(x_\mathrm{p}(0), e_\mathrm{p}(0)) \in S_\mathrm{e} \Rightarrow (x_\mathrm{p}(t), e_\mathrm{p}(t)) \in S_\mathrm{e}$, $\forall t \geq 0$ as desired, which completes the proof. ∎

Remark 5: Theorem 4 ultimately leads to controllers that use the state x, the input history u_t, and the state of the environment given by e, $\dot{e}$. An example is the following QP:

$$K(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p}) = \arg\min_{u\in U} \|u - K_\mathrm{d}(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p})\|^2 \quad \text{s.t. } \dot{H}(x_\mathrm{p}, e_\mathrm{p}, \dot{e}_\mathrm{p}, u) \geq -\alpha(H(x_\mathrm{p}, e_\mathrm{p})) \tag{43}$$

with $x_\mathrm{p} = \Psi(\tau, x, u_t)$ and $e_\mathrm{p} = \Gamma(\tau, e)$; see (15) and (38).

Remark 6: In practice, the environment's future state e_p and its derivative $\dot{e}_\mathrm{p}$ are unknown, and we can only provide estimates $\hat{e}_\mathrm{p}$ and $\hat{\dot{e}}_\mathrm{p}$. Robustness against *environment prediction errors* is a significant problem, since the evolution of the environment is typically more uncertain than the dynamics of the control system. Robustness can be addressed similar to Section III-B, as follows. For simplicity, we assume that the dynamics of the control system (27) is well known, and its state is predicted with negligible error ($\hat{x}_\mathrm{p} = x_\mathrm{p}$); otherwise, prediction errors could be overcome based on Remark 4. Then, the approach of Proposition 1 can be applied to achieve robustness against environment prediction errors, via the condition

$$\dot{H}(x_\mathrm{p}, \hat{e}_\mathrm{p}, \hat{\dot{e}}_\mathrm{p}, u) - C(\varepsilon_e, \varepsilon_{\dot{e}}, u) \geq -\alpha\big(H(x_\mathrm{p}, \hat{e}_\mathrm{p})\big) \tag{44}$$

with $C(\varepsilon_e, \varepsilon_{\dot{e}}, u)$ defined in (21), where ε_e and $\varepsilon_{\dot{e}}$ are error bounds satisfying $\|e_\mathrm{p} - \hat{e}_\mathrm{p}\| \leq \varepsilon_e$ and $\|\dot{e}_\mathrm{p} - \hat{\dot{e}}_\mathrm{p}\| \leq \varepsilon_{\dot{e}}$.

Example 2 (ACC With Input Delay): Consider the ACC problem of Example 1, now with input delay τ that represents powertrain delays

$$\underbrace{\begin{bmatrix}\dot{s}(t)\\ \dot{v}(t)\end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix}v(t)\\ -p\big(v(t)\big)\end{bmatrix}}_{f(x(t))} + \underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{g(x(t))} u(t-\tau). \tag{45}$$

For passenger vehicles, the delay τ is around 0.5–1 s [20]; hence, it is not negligible for safety-critical applications.

The effect of the delay is demonstrated in Fig. 3 (black). Simulation results are shown with a large delay $\tau = 1$, zero initial input history, and the parameters of Example 2: $p(v) = 0.1 + 0.0003\, v^2$, $\gamma = 3$, and $\bar{\kappa} = 2$ (with units in SI). If one implements the delay-free control design (15) relying on (25), it fails to keep system (45) safe due to the delay τ. Safety is violated even when the HV cruises at constant speed.

Thus, we use Theorem 4 to ensure safety for $\tau > 0$. We predict the AV's motion by forward integrating (45) over the delay interval $[t, t+\tau]$ using the input history u_t. The resulting predicted state is denoted by $x_\mathrm{p}(t) = [s_\mathrm{p}(t), v_\mathrm{p}(t)]^\top$. Furthermore, we predict the HV's motion by assuming constant

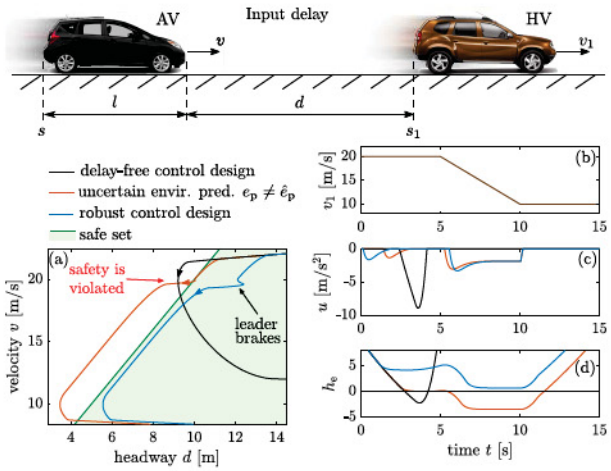

Fig. 3. Strategies for ACC with input delay; (a) phase portrait, (b) velocity of HV, (c) control input of AV, and (d) ECBF. The naïve implementation of the delay-free control design (15) violates safety (black). The controller (43) that relies on predictor feedback enforces safety as long as the prediction of the HV's motion is accurate, and violates safety otherwise (red). When controller (43) is robustified via constraint (44), safety is maintained despite prediction errors (blue).

speed over $[t, t+\tau]$: $\hat{\hat{e}}_{\mathrm{p}}(t) = v_1(t)$ and $\hat{e}_{\mathrm{p}}(t) = s_1(t) + v_1(t)\tau$. The prediction is incorporated into the safety condition (42) to synthesize a control input satisfying

$$\bar{\kappa}\big(v_1 - v_{\mathrm{p}}\big) + \gamma\left(\bar{\kappa}(s_1 + v_1\tau - s_{\mathrm{p}} - l) - v_{\mathrm{p}}\right) + p(v_{\mathrm{p}}) \geq u \tag{46}$$

See (25). Fig. 3 (red color) shows the result of executing the corresponding controller given by QP (43) with desired controller $K_{\mathrm{d}}(x_{\mathrm{p}}, e_{\mathrm{p}}, \dot{e}_{\mathrm{p}}) = 0$ and constraint (46). The controller maintains safety as long as the HV travels at constant speed, and the prediction about HV's future motion is accurate ($\hat{\hat{e}}_{\mathrm{p}}(t) = \dot{e}_{\mathrm{p}}(t)$ and $\hat{\hat{e}}_{\mathrm{p}}(t) = \dot{e}_{\mathrm{p}}(t)$). Then, safety is violated once the HV starts to slow down, and the prediction no longer matches the true future motion of the HV ($\hat{e}_{\mathrm{p}}(t) \neq e_{\mathrm{p}}(t)$ and $\hat{\hat{e}}_{\mathrm{p}}(t) \neq \dot{e}_{\mathrm{p}}(t)$). While one can argue that the constant speed prediction is overly simplistic and more sophisticated predictions exist, the HV's future motion is inherently uncertain. Hence, we need to robustify the controller against this uncertainty.

Predicting the HV's future motion with constant speed leads to a time-varying prediction error that depends on the velocity profile $v_1(t)$. Assuming that the HV's acceleration is limited to a range $[-a_{\min}, a_{\max}]$, we have the following physical bounds for the environment prediction error: $\varepsilon_{\dot{\mathbf{e}}} = \bar{a}\tau$ and $\varepsilon_{\mathrm{e}} = \bar{a}\tau^2/2$ with $\bar{a} = \max\{a_{\min}, a_{\max}\}$. Then, the robustified condition (44) leads to the form

$$\bar{\kappa}\big(v_1 - \bar{a}\tau - v_{\mathrm{p}}\big) + \gamma\left(\bar{\kappa}(s_1 + v_1\tau - \bar{a}\tau^2/2 - s_{\mathrm{p}} - l) - v_{\mathrm{p}}\right) + p(v_{\mathrm{p}}) \geq u \tag{47}$$

See (46). Besides, it can be shown that replacing $\bar{a}$ with $a_{\min}$ in (47) also implies (42). This provides a problem-specific bound that is less conservative than (47) if $a_{\min} < a_{\max}$.

Fig. 3 (blue curve) shows simulation results for controller (43) with the robustified constraint (47) and $a_{\min} = a_{\max} = 2.5\ \mathrm{m/s^2}$. By Theorem 2, the controller ensures safety even with input delay, in a dynamic, uncertain environment. The price of robustness is slight conservatism: the system does not reach the safe set boundary but keeps a small distance, since the controller uses the 2.5-m/s^2 braking limit instead of the actual 2-m/s^2 braking over the 1-s delay interval.

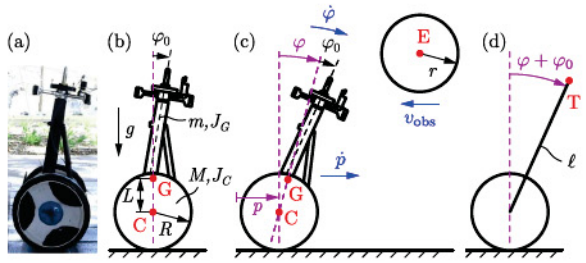

Fig. 4. (a) Ninebot E+ Segway platform. (b) Its mechanical model illustrated in equilibrium position. (c) Segway in motion aiming to avoid a moving obstacle. (d) Simplified representation of the Segway.

V. Case Study: Control of a Segway

Now, we apply the theoretical constructions of this article to a real-life robotic system: we consider the control of a Ninebot E+ Segway platform [58] shown in Fig. 4(a). We intend to drive the Segway, so that it safely avoids a moving obstacle, even when the obstacle position is uncertain, and there is a delay in the control loop. We conduct numerical simulations of the Segway's motion using a high-fidelity dynamical model.

We describe the planar motion of the Segway by its mechanical model in Fig. 4(b) and (c). Fig. 4(b) depicts the Segway in equilibrium, where the center of mass (CoM) of its frame (point G) is above the wheel center (point C). Note that the frame is asymmetric, and its axis is tilted in equilibrium at an offset angle φ_0. Fig. 4(c) shows the Segway in motion during obstacle avoidance, and Fig. 4(d) depicts its simplified representation.

Our goal is to drive the Segway forward with a desired speed $\dot{p}_{\mathrm{d}}$ while avoiding a moving, circular obstacle centered at $[e, y]^\top$ (point E in Fig. 4) with radius r. The obstacle represents the environment of the Segway. We intend to control the Segway such that its tip—point T in Fig. 4, located at distance ℓ from the wheel center—does not collide with the obstacle. The obstacle moves horizontally with constant speed v_{obs}: $e = e_0 - tv_{\mathrm{obs}}$, $\dot{e} = -v_{\mathrm{obs}}$, and $\dot{y} = 0$. For numerical case study, we use $\dot{p}_{\mathrm{d}} = 1$ m/s, $r = 0.2$ m, $v_{\mathrm{obs}} = 0.5$ m/s, $e_0 = 1$ m, and $y = 1.0418$ m (for this value, point T is located 0.05 m above the bottom of the obstacle when the Segway is in equilibrium). First, we consider safety-critical control by neglecting the time delay that may arise in the Segway's control loop; then, we address the effects of delay.

A. Safety-Critical Control in Dynamic Environment

We describe the Segway dynamics by the wheel center position p and pitch angle φ as two-degrees-of-freedom planar system with general coordinates $q = [p, \varphi]^\top \in \mathcal{Q}$ and velocities $\dot{q} = [v, \omega]^\top \in \mathbb{R}^2$. The state becomes $x = [p, \varphi, v, \omega]^\top \in X$, the configuration space is $\mathcal{Q} = \mathbb{R} \times [0, 2\pi]$, and the state space is $X = \mathcal{Q} \times \mathbb{R}^2$. The control input $u \in \mathbb{R}$ is the voltage applied on the motors at the wheels. The dynamics are governed by

$$\begin{bmatrix} \dot{p} \\ \dot{\varphi} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \omega \\ f_v(\varphi, v, \omega) \\ f_\omega(\varphi, v, \omega) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_v(\varphi) \\ g_\omega(\varphi) \end{bmatrix} u. \tag{48}$$

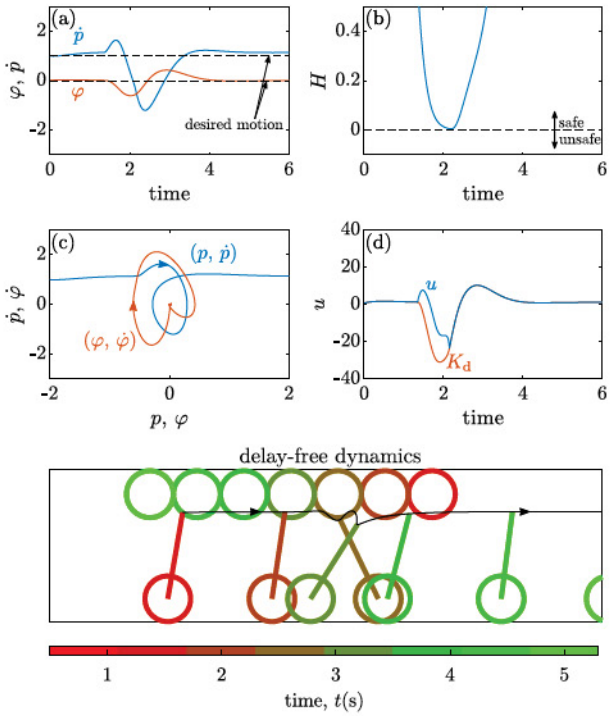

Fig. 5. Safety-critical control of the Segway to avoid a moving obstacle, with the delay-free dynamics (48) and known obstacle position. The Segway safely avoids the obstacle with a controller that satisfies (53). (a) States, (b) ECBF candidate, (c) phase portrait, and (d) control input.

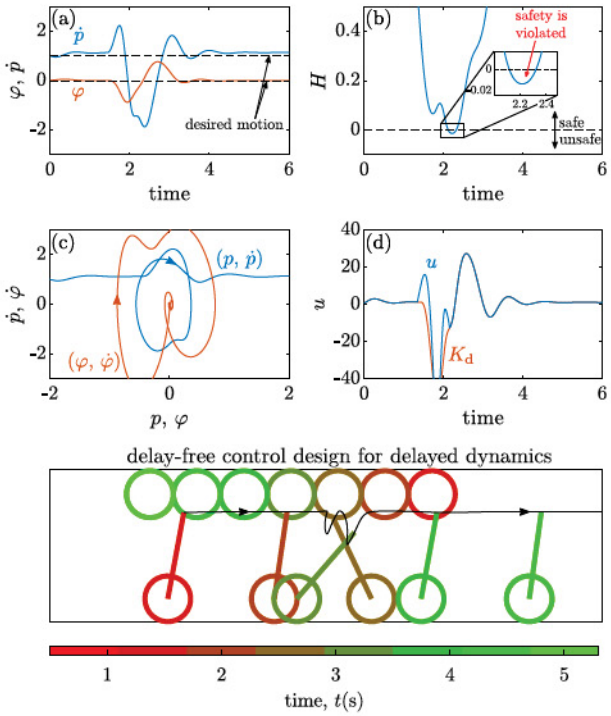

Fig. 6. Safety-critical control of the Segway to avoid a moving obstacle. The dynamics (54) involve an input delay. The naïve implementation of the delay-free control design based on (53) fails to avoid the obstacle. (a) States, (b) ECBF candidate, (c) phase portrait, and (d) control input.

For the derivation of this equation and the detailed expressions of f_v, f_ω, g_v, and g_ω, please refer to Appendix II-A. The model parameters were identified in [58] and are listed in Table I.

We track the desired speed $\dot{p}_d$ by the desired controller

$$K_d(x, e, \dot{e}) = K_{\dot{p}}(\dot{p} - \dot{p}_d) + K_\varphi \varphi + K_{\dot{\varphi}} \dot{\varphi} \tag{49}$$

with gains $K_{\dot{p}} = 8$ Vs/m, $K_\varphi = 40$ V/rad, and $K_{\dot{\varphi}} = 10$ Vs/rad, that also stabilizes the Segway to the upright position. To avoid the moving obstacle, we construct the ECBF candidate

$$H(x, e) = d^\top d - r^2 \qquad d = \begin{bmatrix} p + \ell \sin(\varphi + \varphi_0) - e \\ R + \ell \cos(\varphi + \varphi_0) - y \end{bmatrix} \tag{50}$$

where d points from the obstacle center to the Segway's tip.

We seek to maintain safety with respect to the environmental safe set (8) using Theorem 2. However, H is not a valid ECBF, since $\nabla_x H(x, e) g(x) = 0$, and $\dot{H}$ is independent of the input u. Thus, we use a dynamic extension of the ECBF based on [54]. We define the *extended ECBF*

$$H_e(x, e, \dot{e}) = \dot{H}(x, e, \dot{e}) + \gamma_e H(x, e) \tag{51}$$

with $\gamma_e > 0$, whose derivative depends on the control input u

$$\begin{aligned} \dot{H}_e(x, e, \dot{e}, \ddot{e}, u) = {} & \nabla_x \dot{H}(x, e, \dot{e})(f(x) + g(x)u) \\ & + \nabla_e \dot{H}(x, e, \dot{e})\dot{e} + \nabla_{\dot{e}} \dot{H}(x, e, \dot{e})\ddot{e} \\ & + \gamma_e \dot{H}(x, e, \dot{e}). \end{aligned} \tag{52}$$

With this choice, $H_e(x, e, \dot{e}) \geq 0$ is equivalent to (11) in Theorem 2 considering a linear class $\mathcal{K}$ function with gradient γ_e. Thus, safety is achieved if H_e is kept nonnegative for all time, which can be enforced if $H_e(x_0, e_0, \dot{e}_0) \geq 0$ and

$$\dot{H}_e(x, e, \dot{e}, \ddot{e}, u) \geq -\alpha\big(H_e(x, \hat{e}, \dot{e})\big) \tag{53}$$

See Theorem 2. Notice that the second derivative $\ddot{e}$ shows up.

We implement a QP-based controller similar to (15), with desired controller (49) and constraint (53) using linear class $\mathcal{K}$ function $\alpha(H_e) = \gamma H_e$ with $\gamma = 7.5\,\mathrm{s}^{-1}$ and $\gamma_e = 7.5\ \mathrm{s}^{-1}$. For known obstacle position, the performance of the controller is demonstrated in Fig. 5, with snapshots of the motion at the bottom and its characteristics at the top. Fig. 5(a) shows that the Segway tracks the desired velocity ($\dot{p} \approx \dot{p}_d$) in upright position ($\varphi \approx 0$) until it has to evade the obstacle. Fig. 5(b) indicates that the obstacle is safely avoided, as H is positive for all time. Fig. 5(c) shows the corresponding phase portrait, whereas Fig. 5(d) depicts the desired and actual control inputs.

B. Safety-Critical Control With Input Delay

Now, we consider the dynamics with input delay $\tau > 0$ arising from sensory, feedback, and actuation latencies

$$\begin{bmatrix} \dot{p}(t) \\ \dot{\varphi}(t) \\ \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \omega(t) \\ f_v(\varphi(t), v(t), \omega(t)) \\ f_\omega(\varphi(t), v(t), \omega(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_v(\varphi(t)) \\ g_\omega(\varphi(t)) \end{bmatrix} u(t - \tau). \tag{54}$$

See (48). The effect of the delay is illustrated in Fig. 6. Here, the same delay-free control design is used as in Fig. 5, but the dynamics are subject to the input delay $\tau = 0.1$ s. Although the Segway realizes a stable motion, the delay leads to safety violation: the Segway collides with the obstacle [H becomes negative in Fig. 6(b)]. While collision could be avoided by buffering the obstacle, formal safety guarantees no longer hold with delay. Moreover, the control input is much larger with delay than without delay; see Figs. 5(d) and 6(d). Such large

TABLE I
PARAMETERS OF THE SEGWAY MODEL

Description	Parameter	Value	Unit
gravitational acceleration	g	9.81	m/s^2
radius of wheels	R	0.195	m
mass of wheels	M	2×2.485	kg
mass moment of inertia of wheels	J_C	2×0.0559	kgm^2
distance of wheel center and frame CoM	L	0.169	m
distance of wheel center and frame tip	ℓ	0.75	m
mass of frame	m	44.798	kg
mass moment of inertia of frame	J_G	3.836	kgm^2
offset angle	φ_0	0.138	rad
torque constant of motors	K_m	2×1.262	Nm/V
damping constant of motors	b_t	2×1.225	Ns
combined parameters	m_0	52.710	kg
	J_0	5.108	kgm^2
	a	0.6768	m
	b	4.7274	-
	c	68.5205	1/s^2
	κ	0.9713	Vs/m
	A	1.1605	m/s^2/V
	B	0.3344	m/s^2/V
	C	2.3355	1/s^2/V
	D	1.7147	1/s^2/V

inputs are undesired, as safety-critical control could become infeasible with input bounds. To overcome the unsafe behavior, the delay needs to be incorporated into the control design.

The input delay can be tackled via predictor feedback, using Theorem 4. We assume that the state x_p is accurately predicted ($\hat{x}_p = x_p$), while the predictions e_p, $\dot{e}_p$, and $\ddot{e}_p$ of the environment are uncertain. Hence, the controller relies on estimates $\hat{e}_p$, $\hat{\dot{e}}_p$, and $\hat{\ddot{e}}_p$ and their error bounds $\|e_p - \hat{e}_p\| \le \varepsilon_e$, $\|\dot{e}_p - \hat{\dot{e}}_p\| \le \varepsilon_{\dot{e}}$, and $\|\ddot{e}_p - \hat{\ddot{e}}_p\| \le \varepsilon_{\ddot{e}}$. Analogously to (44), we use the robustified safety constraint

$$\dot{H}_e(x_p, \hat{e}_p, \hat{\dot{e}}_p, \hat{\ddot{e}}_p, u) - C(\varepsilon_e, \varepsilon_{\dot{e}}, \varepsilon_{\ddot{e}}, u) \ge -\alpha\big(H_e(x_p, \hat{e}_p, \hat{\dot{e}}_p)\big) \tag{55}$$

with

$$\begin{aligned} C(\varepsilon_e, \varepsilon_{\dot{e}}, u) = {} & (\mathcal{L}_{\nabla H_e f, e} + \mathcal{L}_{\alpha \circ H_e, e} + \mathcal{L}_{\nabla H_e \dot{e}, e} + \mathcal{L}_{\nabla H_e \ddot{e}, e})\varepsilon_e \\ & + (\mathcal{L}_{\nabla H_e f, \dot{e}} + \mathcal{L}_{\alpha \circ H_e, \dot{e}} + \mathcal{L}_{\nabla H_e \dot{e}, \dot{e}} + \mathcal{L}_{\nabla H_e \ddot{e}, \dot{e}})\varepsilon_{\dot{e}} \\ & + \mathcal{L}_{\nabla H_e \ddot{e}, \ddot{e}}\varepsilon_{\ddot{e}} + (\mathcal{L}_{\nabla H_e g, e}\varepsilon_e + \mathcal{L}_{\nabla H_e g, \dot{e}}\varepsilon_{\dot{e}})\|u\| \end{aligned} \tag{56}$$

See (21). Here, $\mathcal{L}$ denotes the Lipschitz coefficient of the subscripted function with respect to the argument at the end of the subscript. These coefficients were determined based on the expressions of the Segway dynamics; see Appendix II-B.

Fig. 7 shows the implementation of the corresponding QP-based controller, similar to (43), with desired controller (49) applied on the predicted state and with constraint (55). The true future of the environment, given by $\ddot{e}_p = 0$, $\dot{e}_p = -v_{obs}$, and $e_p = e - v_{obs}\tau$, is unknown to the controller. Instead, the controller relies on the prediction $\hat{\ddot{e}}_p = 0$, $\hat{\dot{e}}_p = -(v_{obs} - \Delta v)$, and $\hat{e}_p = e - (v_{obs} - \Delta v)\tau$. That is, the speed of the obstacle is underestimated by $\Delta v = 0.05$ m/s. The controller is robustified against the prediction error using the error bounds $\varepsilon_{\ddot{e}} = 0$, $\varepsilon_{\dot{e}} = 0.055$ m/s, and $\varepsilon_e = \tau\varepsilon_{\dot{e}} = 0.0055$ m. With the proposed robust controller, the Segway safely executes the obstacle avoidance task, despite the delay in the control loop and the uncertainty in the obstacle's future position. This is achieved with a qualitatively different motion than in the delay-free case. For zero delay in Fig. 5, the Segway pitches backward to go under the obstacle. For nonzero delay in Fig. 7, the Segway moves in reverse to get away from the obstacle, and then pitches forward to go under it. Notably, this behavior is automatically generated by ECBF, and with provable guarantees of safety.

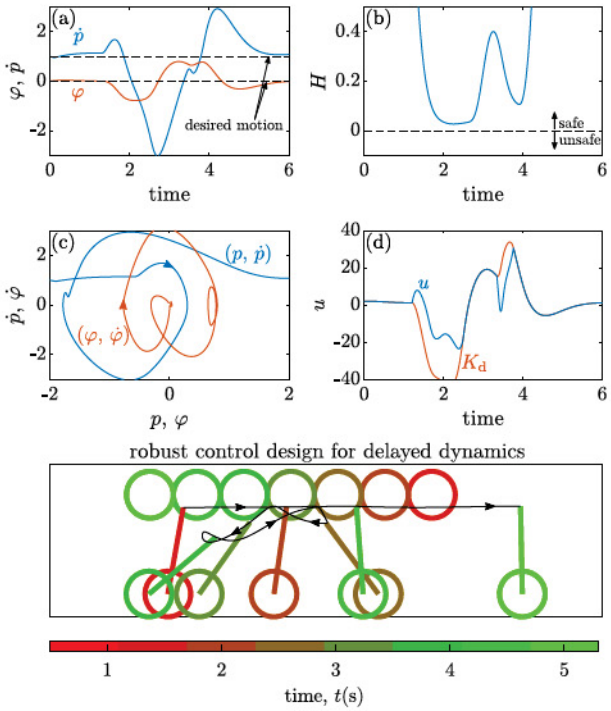

Fig. 7. Safety-critical control of the Segway to avoid a moving obstacle. The dynamics (54) involve an input delay that is compensated via predictor feedback. The controller is designed using (55), taking into account prediction errors. The Segway successfully avoids the obstacle despite the delay and the uncertain future motion of the obstacle. (a) States, (b) ECBF candidate, (c) phase portrait, and (d) control input.

VI. CONCLUSION

We have discussed safety-critical control for systems with input delay that operate in dynamically evolving environment. We have provided formal safety guarantees and proofs thereof. We have established a method for safe control synthesis by proposing ECBFs and integrating them with predictor feedback. We have strengthened the underlying safety condition to provide robustness against uncertain environments, in which the future of the environment cannot be predicted accurately, but bounds on the related prediction error are known. The resulting control design uses worst case uncertainty bounds and is provably safe. We have demonstrated the method by an ACC problem, where the motion of another vehicle creates an uncertain environment, and by a Segway controller that avoids moving obstacles. Our future work includes the analysis of prediction errors, control of systems with both state and input delays, and use of control barrier functionals acting on delayed states.

APPENDIX I
KKT CONDITIONS

This appendix shows the derivation of solution (6) to the QP (7) without input constraints ($U = \mathbb{R}^m$). Let $\Delta k(x) = k(x) - k_d(x)$, and consider $\dot{h}(x, u)$ in (4) and $\phi_0(x)$, $\phi_1(x)$ in (7) with $\phi_1(x) \ne 0$. We can restate (6) as follows:

$$\begin{aligned} \Delta k(x) = \arg\min_{\Delta u \in \mathbb{R}^m} \ & \|\Delta u\|^2 \\ \text{s.t. } & \phi_0(x) + \phi_1(x)\Delta u \ge 0. \end{aligned} \tag{57}$$

This optimization problem has convex objective and affine constraint; hence, the *KKT conditions* [53] provide necessary and sufficient conditions for optimality. The KKT conditions imply that there exists a Lagrange multiplier $\mu : X \to \mathbb{R}$, such that $\mu(x)$ and $\Delta k(x)$ satisfy

$$\mu(x) \geq 0 \tag{58}$$
$$\Delta k(x) = \mu(x)\phi_1^\top(x) \tag{59}$$
$$\phi_0(x) + \phi_1(x)\Delta k(x) \geq 0 \tag{60}$$
$$\mu(x)(\phi_0(x) + \phi_1(x)\Delta k(x)) = 0 \tag{61}$$

which are referred to as dual feasibility, stationary, primal feasibility, and complementary slackness conditions, respectively.

We decompose the dual feasibility condition (58) into two cases: $\mu(x) = 0$ and $\mu(x) > 0$. For $\mu(x) = 0$, the stationary condition (59) gives

$$\Delta k(x) = 0. \tag{62}$$

With the primal feasibility condition (60), this leads to

$$\phi_0(x) \geq 0. \tag{63}$$

For $\mu(x) > 0$, the complementary slackness condition (61) implies

$$\phi_0(x) + \phi_1(x)\Delta k(x) = 0. \tag{64}$$

Recall that $\phi_1(x) \in \mathbb{R}^n$ is a nonzero vector with right pseudo-inverse $\phi_1^+(x) = \phi_1^\top(x)/(\phi_1(x)\phi_1^\top(x))$, and $\phi_0(x) \in \mathbb{R}$ is a scalar. Then, we can express $\Delta k(x)$ from (64) as follows:

$$\Delta k(x) = -\phi_0(x)\phi_1^+(x). \tag{65}$$

Furthermore, we can show that $\phi_0(x) < 0$ holds by expressing $\phi_0(x)$ from (64) and substituting the stationary condition (59)

$$\phi_0(x) = -\phi_1(x)\Delta k(x) = -\mu(x)\phi_1(x)\phi_1^\top(x) < 0 \tag{66}$$

where we used that $\mu(x) > 0$ and $\phi_1(x)\phi_1^\top(x) > 0$.

In summary, for $\mu(x) = 0$, we have $\Delta k(x) = 0$ and $\phi_0(x) \geq 0$, while $\mu(x) > 0$ implies $\Delta k(x) = -\phi_0(x)\phi_1^+(x)$ and $\phi_0(x) < 0$. These can be written as follows:

$$\Delta k(x) = \begin{cases} 0, & \text{if } \phi_0(x) \geq 0 \\ -\phi_0(x)\phi_1^+(x), & \text{if } \phi_0(x) < 0 \end{cases} \tag{67}$$

or more compactly as follows [10]:

$$\Delta k(x) = \max\{-\phi_0(x), 0\}\phi_1^+(x). \tag{68}$$

Since $k(x) = k_\mathrm{d}(x) + \Delta k(x)$, we finally obtain (7) as the solution to the QP (6).

Appendix II
Technical Details of the Segway Application

Here, we derive the governing equations of the Segway model described in Section V, using Lagrange equations of the second kind. This reproduces the model in [58]. Then, we describe the ECBF and the corresponding Lipschitz coefficients for the obstacle avoidance task.

A. Segway Dynamics

The Segway's mechanical model is shown in Fig. 4. This planar model contains two rigid bodies: the frame and the wheels. The two wheels are considered to be identical; hence, they are treated together with their combined mass and inertia, while the voltage and torque at the two motors are assumed to be the same. We denote the center of the wheels by point C, the CoM of the frame by point G, their distance by L, and the wheel radius by R. We measure the pitch angle, such that $\varphi = 0$ in equilibrium, where G is located above C. Note that the frame is asymmetric, and the frame axis is not vertical in equilibrium, but it has an offset angle φ_0.

Assuming the wheels are rolling without slipping, the angular velocities ω_w and ω_f of the wheel and the frame and the velocities v_C and v_G of points C and G can be calculated by

$$\begin{aligned} \omega_\mathrm{w} &= \dot{p}/R, \quad \omega_\mathrm{f} = \dot{\varphi} \\ v_\mathrm{C} &= \begin{bmatrix} \dot{p} \\ 0 \end{bmatrix}, \quad v_\mathrm{G} = \begin{bmatrix} \dot{p} + L\dot{\varphi}\cos\varphi \\ -L\dot{\varphi}\sin\varphi \end{bmatrix}. \end{aligned} \tag{69}$$

Then, with the mass M and mass moment of inertia J_C of the wheels and the mass m and mass moment of inertia J_G of the frame, the kinetic energy of the Segway is

$$\begin{aligned} T &= \frac{1}{2}Mv_\mathrm{C}^2 + \frac{1}{2}J_\mathrm{C}\omega_\mathrm{w}^2 + \frac{1}{2}mv_\mathrm{G}^2 + \frac{1}{2}J_\mathrm{G}\omega_\mathrm{f}^2 \\ &= \frac{1}{2}m_0\dot{p}^2 + mL\dot{p}\dot{\varphi}\cos\varphi + \frac{1}{2}J_0\dot{\varphi}^2 \end{aligned} \tag{70}$$

where $m_0 = m + M + J_\mathrm{C}/R^2$ and $J_0 = mL^2 + J_\mathrm{G}$. The potential energy of the Segway is

$$U = mgL\cos\varphi. \tag{71}$$

The power of the total driving torque M_d exerted by the two motors at the wheels can be expressed as follows:

$$P = M_\mathrm{d}(\omega_\mathrm{w} - \omega_\mathrm{f}) = Q_p\dot{p} + Q_\varphi\dot{\varphi} \tag{72}$$

yielding the general forces $Q_p = M_\mathrm{d}/R$ and $Q_\varphi = -M_\mathrm{d}$.

The driving torque M_d can be related to the voltage u of the motors. We regard the voltage as control input, obtained from the following motor model:

$$\begin{aligned} u &= R_\mathrm{a}i + K_\mathrm{b}(\omega_\mathrm{w} - \omega_\mathrm{f}) \\ M_\mathrm{d} &= K_\mathrm{t}i \end{aligned} \tag{73}$$

where i is the armature current, R_a is the armature resistance, K_b is the back electromagnetic field constant, and K_t is the torque constant of the motors. This implies the driving torque

$$M_\mathrm{d} = K_\mathrm{m}u - b_\mathrm{t}(\dot{p} - R\dot{\varphi}) \tag{74}$$

with constants $K_\mathrm{m} = K_\mathrm{t}/R_\mathrm{a}$ and $b_\mathrm{t} = K_\mathrm{t}K_\mathrm{b}/(R_\mathrm{a}R)$.

With these preliminaries, we write Lagrange's equations

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial \dot{p}} - \frac{\partial T}{\partial p} + \frac{\partial U}{\partial p} &= Q_p \\ \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial \dot{\varphi}} - \frac{\partial T}{\partial \varphi} + \frac{\partial U}{\partial \varphi} &= Q_\varphi \end{aligned} \tag{75}$$

which, after substitution, lead to

$$\begin{aligned} m_0\ddot{p} + mL\cos\varphi\ddot{\varphi} - mL\sin\varphi\dot{\varphi}^2 &= \frac{M_\mathrm{d}}{R} \\ mL\cos\varphi\ddot{p} + J_0\ddot{p} - mgL\sin\varphi &= -M_\mathrm{d}. \end{aligned} \tag{76}$$

Ultimately, we obtain the equations of motion in the form

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu \tag{77}$$

with the inertia matrix $D(q)$, Coriolis and gravity terms included in $H(q,\dot{q})$, and input matrix B

$$D(q) = \begin{bmatrix} m_0 & mL\cos\varphi \\ mL\cos\varphi & J_0 \end{bmatrix}, \quad B = \begin{bmatrix} K_\mathrm{m}/R \\ -K_\mathrm{m} \end{bmatrix}$$
$$H(q,\dot{q}) = \begin{bmatrix} -mL\sin\varphi\dot{\varphi}^2 + b_\mathrm{t}/R(\dot{p} - R\dot{\varphi}) \\ -mgL\sin\varphi - b_\mathrm{t}(\dot{p} - R\dot{\varphi}) \end{bmatrix}. \tag{78}$$

The equations of motion can be rearranged to the first-order control-affine form (1)

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)H(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ D^{-1}(q)B \end{bmatrix} u \tag{79}$$

which leads to

$$\begin{bmatrix} \dot{p} \\ \dot{\varphi} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \omega \\ f_v(\varphi, v, \omega) \\ f_\omega(\varphi, v, \omega) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_v(\varphi) \\ g_\omega(\varphi) \end{bmatrix} u \tag{80}$$

See (48). The expressions of the drift terms are

$$f_v(\varphi, v, \omega) = \frac{a\omega^2 \sin\varphi - g\sin\varphi\cos\varphi}{b - \cos^2\varphi} - \kappa g_v(\varphi)(v - R\omega)$$
$$f_\omega(\varphi, v, \omega) = \frac{c\sin\varphi - \omega^2\sin\varphi\cos\varphi}{b - \cos^2\varphi} - \kappa g_\omega(\varphi)(v - R\omega) \tag{81}$$

whereas those of the control matrix read

$$g_v(\varphi) = \frac{A + B\cos\varphi}{b - \cos^2\varphi}, \quad g_\omega(\varphi) = -\frac{C + D\cos\varphi}{b - \cos^2\varphi} \tag{82}$$

with parameters

$$a = \frac{J_0}{mL}, \quad b = \frac{m_0 J_0}{m^2 L^2}, \quad c = \frac{m_0 g}{mL}, \quad \kappa = \frac{b_\mathrm{t}}{K_\mathrm{m}}$$
$$A = \frac{K_\mathrm{m} J_0}{m^2 L^2 R}, \quad B = \frac{K_\mathrm{m}}{mL}, \quad C = \frac{K_\mathrm{m} m_0}{m^2 L^2}, \quad D = \frac{K_\mathrm{m}}{mLR}. \tag{83}$$

The values of all parameters are listed in Table I. These were identified for the Ninebot E+ Segway platform in [58].

B. Expression of the ECBF and Its Lipschitz Coefficients

Now, we give the detailed expressions of the extended ECBF in (51) and the corresponding Lipschitz coefficients in (56). The ECBF candidate in (50) is of the form

$$H(x,e) = h_0(x) + h_1(x)e + e^2 \tag{84}$$

with coefficients

$$h_0(x) = \big(p + \ell\sin(\varphi + \varphi_0)\big)^2 + \big(R + \ell\cos(\varphi + \varphi_0) - y\big)^2 - r^2$$
$$h_1(x) = -2\big(p + \ell\sin(\varphi + \varphi_0)\big). \tag{85}$$

Then, the extended ECBF in (51) becomes

$$H_\mathrm{e}(x,e,\dot{e}) = H_0(x) + H_1(x)e + h_1(x)\dot{e} + \gamma_\mathrm{e}e^2 + 2e\dot{e} \tag{86}$$

where

$$H_0(x) = \nabla_p h_0(x) v + \nabla_\varphi h_0(x)\omega + \gamma_\mathrm{e} h_0(x)$$
$$H_1(x) = \nabla_p h_1(x) v + \nabla_\varphi h_1(x)\omega + \gamma_\mathrm{e} h_1(x). \tag{87}$$

Notice that h_0 and h_1 depend on the states p and φ only, whose derivatives are independent of the control input u.

The Lipschitz coefficients in (56) belong to the functions

$$\begin{aligned} \nabla_x H_\mathrm{e} f(x,e,\dot{e}) &= C_0(x) + C_1(x)e + C_2(x)\dot{e} \\ \nabla_x H_\mathrm{e} g(x,e,\dot{e}) &= C_3(x) + C_4(x)e \\ \nabla_e H_\mathrm{e}\dot{e}(x,e,\dot{e}) &= H_1(x)\dot{e} + 2\gamma_\mathrm{e}e\dot{e} + 2\dot{e}^2 \\ \nabla_{\dot{e}} H_\mathrm{e}\ddot{e}(x,e,\dot{e},\ddot{e}) &= h_1(x)\ddot{e} + 2e\ddot{e} \\ \alpha \circ H_\mathrm{e}(x,e,\dot{e}) &= \gamma H_0(x) + \gamma H_1(x)e + \gamma h_1(x)\dot{e} \\ &\quad + \gamma\gamma_\mathrm{e}e^2 + 2\gamma e\dot{e} \end{aligned} \tag{88}$$

where

$$\begin{aligned} C_0(x) &= \nabla_x H_0(x) f(x) \\ C_1(x) &= \nabla_x H_1(x) f(x) \\ C_2(x) &= \nabla_x h_1(x) f(x) \\ C_3(x) &= \nabla_x H_0(x) g(x) \\ C_4(x) &= \nabla_x H_1(x) g(x). \end{aligned} \tag{89}$$

For example, to identify the Lipschitz coefficients of $\nabla_e H_\mathrm{e}\dot{e}$, we can write

$$\begin{aligned} &\nabla_e H_\mathrm{e}\dot{e}(x,e,\dot{e}) - \nabla_e H_\mathrm{e}\dot{e}(x,\hat{e},\hat{\dot{e}}) \\ &\quad = H_1(x)(e - \hat{e}) + 2\gamma_\mathrm{e}(e\dot{e} - \hat{e}\hat{\dot{e}}) + 2\big(\dot{e}^2 - \hat{\dot{e}}^2\big) \\ &\quad = H_1(x)(e - \hat{e}) + 2\gamma_\mathrm{e}\dot{e}(e - \hat{e}) + 2\gamma_\mathrm{e}\hat{e}(\dot{e} - \hat{\dot{e}}) \\ &\qquad + 2\hat{e}(\dot{e} - \hat{\dot{e}}) + 2\dot{e}(\dot{e} - \hat{\dot{e}}) \\ &\quad \geq -\big(|H_1(x)| + 2\gamma_\mathrm{e}\max_{\dot{e}\in D_{\dot{e}}}|\dot{e}|\big)|e - \hat{e}| \\ &\qquad - \big(2\gamma_\mathrm{e}|\hat{e}| + 2|\hat{\dot{e}}| + 2\max_{\dot{e}\in D_{\dot{e}}}|\dot{e}|\big)|\dot{e} - \hat{\dot{e}}|. \end{aligned} \tag{90}$$

Hence, the corresponding Lipschitz coefficients are

$$\begin{aligned} \mathcal{L}_{\nabla H_\mathrm{e}\dot{e},e} &= |H_1(x)| + 2\gamma_\mathrm{e}\max_{\dot{e}\in D_{\dot{e}}}|\dot{e}| \\ \mathcal{L}_{\nabla H_\mathrm{e}\dot{e},\dot{e}} &= 2\gamma_\mathrm{e}|\hat{e}| + 2|\hat{\dot{e}}| + 2\max_{\dot{e}\in D_{\dot{e}}}|\dot{e}|. \end{aligned} \tag{91}$$

Here, we considered that the unknown environment state derivative $\dot{e}$ is restricted to a domain $D_{\dot{e}} \subseteq \mathcal{E}$ to get local Lipschitz coefficients. Similarly, the unknown environment state e and acceleration $\ddot{e}$ can also be restricted to some domains $D_e \subseteq E$ and $D_{\ddot{e}} \subseteq \mathbb{R}^l$. In the case of the Segway, we assumed that the obstacle's position and the velocity are restricted to $D_e = [-3, 3]$ m and $D_{\dot{e}} = [-0.55, 0.55]$ m/s (while its acceleration was known to be zero).

After similar calculation, the list of the remaining Lipschitz coefficients is

$$\begin{aligned} \mathcal{L}_{\nabla H_\mathrm{e} f,e} &= |C_1(x)| \\ \mathcal{L}_{\nabla H_\mathrm{e} f,\dot{e}} &= |C_2(x)| \\ \mathcal{L}_{\nabla H_\mathrm{e} g,e} &= |C_4(x)| \\ \mathcal{L}_{\nabla H_\mathrm{e} g,\dot{e}} &= 0 \\ \mathcal{L}_{\nabla H_\mathrm{e}\ddot{e},e} &= 2\max_{\ddot{e}\in D_{\ddot{e}}}|\ddot{e}| \\ \mathcal{L}_{\nabla H_\mathrm{e}\ddot{e},\dot{e}} &= 0 \end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\nabla H_e \ddot{e},\ddot{e}} &= |h_1(x)| + 2|\hat{e}| \\ \mathcal{L}_{\alpha \circ H_e,e} &= \gamma\,|H_1(x)| + \gamma\,\gamma_e\big(|\hat{e}| + \max_{e\in D_e}\ |e|\big) + 2\gamma \max_{\dot{e}\in D_{\dot{e}}}\ |\dot{e}| \\ \mathcal{L}_{\alpha \circ H_e,\dot{e}} &= \gamma\,|h_1(x)| + 2\gamma\,|\hat{e}|. \end{aligned} \tag{92}$$

Note that these coefficients may depend on the state x or the estimates $\hat{e}$, $\hat{\dot{e}}$, and $\hat{\ddot{e}}$ to reduce conservatism, while they are independent of the unknown values e, $\dot{e}$, and $\ddot{e}$.

REFERENCES

[1] P. Nilsson et al., "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1294–1307, Jul. 2016.

[2] C. R. He and G. Orosz, "Safety guaranteed connected cruise control," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 549–554.

[3] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," 2021, *arXiv:2103.14252*.

[4] J. Tordesillas, B. T. Lopez, and J. P. How, "FASTER: Fast and safe trajectory planner for flights in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1934–1940.

[5] 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, Oct. 2020.

[6] J. Nubert, J. Kohler, V. Berenz, F. Allgower, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3050–3057, Apr. 2020.

[7] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 617–632, Mar. 2016.

[8] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 310–315, Oct. 2017.

[9] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 1991–1996.

[10] A. D. Ames, T. G. Molnar, A. W. Singletary, and G. Orosz, "Safety-critical control of active interventions for COVID-19 mitigation," *IEEE Access*, vol. 8, pp. 188454–188474, 2020.

[11] T. G. Molnar, A. W. Singletary, G. Orosz, and A. D. Ames, "Safety-critical control of compartmental epidemiological models with measurement delays," *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1537–1542, Nov. 2021.

[12] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, and C. Melchiorri, "Edge-weighted consensus-based formation control strategy with collision avoidance," *Robotica*, vol. 33, no. 2, pp. 332–347, Feb. 2015.

[13] M. Santillo and M. Jankovic, "Collision free navigation with interacting, non-communicating obstacles," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1637–1643.

[14] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, pp. 187–210, May 2018.

[15] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human–robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, Apr. 2016.

[16] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfe, and C. Secchi, "Safety barrier functions for human–robot interaction with industrial manipulators," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 2565–2570.

[17] 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)*, Nov. 2019, pp. 173–178.

[18] G. Stépán, *Retarded Dynamical Systems: Stability and Characteristic Functions*. Harlow, U.K.: Longman, 1989.

[19] T. T. Andersen, H. B. Amor, N. A. Andersen, and O. Ravn, "Measuring and modelling delays in robot manipulators for temporally precise control using machine learning," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 168–175.

[20] X. A. Ji, T. G. Molnar, A. A. Gorodetsky, and G. Orosz, "Bayesian inference for time delay systems with application to connected automated vehicles," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3259–3264.

[21] F. Casella, "Can the COVID-19 epidemic be controlled on the basis of daily test reports?" *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 1079–1084, Jul. 2021.

[22] 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.

[23] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, Oct. 2018.

[24] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 108–113, Jan. 2019.

[25] 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. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 6814–6821.

[26] L. Zheng, R. Yang, J. Pan, and H. Cheng, "Safe learning-based tracking control for quadrotors under wind disturbances," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 3638–3643.

[27] 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 Control Syst. Lett.*, vol. 6, pp. 908–913, 2022.

[28] R. Takano and M. Yamakita, "Robust constrained stabilization control using control Lyapunov and control barrier function in the presence of measurement noises," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2018, pp. 300–305.

[29] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, Aug. 2021, Art. no. 109688.

[30] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Proc. Mach. Learn. Res.*, vol. 155, J. Kober, F. Ramos, and C. Tomlin, Eds. 2021, pp. 654–670.

[31] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2460–2465.

[32] J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," in *Proc. Robotics, Sci. Syst.*, 2020, pp. 1–9.

[33] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for MAVs in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 776–783, Apr. 2019.

[34] W. Luo, W. Sun, and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates," in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 372–383.

[35] M. Igarashi, I. Tezuka, and H. Nakamura, "Time-varying control barrier function and its application to environment-adaptive human assist control," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 735–740, 2019.

[36] I. Tezuka and H. Nakamura, "Time-varying obstacle avoidance by using high-gain observer and input-to-state constraint safe control barrier function," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 391–396, 2020.

[37] G. Orosz and A. D. Ames, "Safety functionals for time delay systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 4374–4379.

[38] A. K. Kiss, T. G. Molnar, D. Bachrathy, A. D. Ames, and G. Orosz, "Certifying safety for nonlinear time delay systems via safety functionals: A discretization based approach," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1055–1060.

[39] W. Liu, Y. Bai, L. Jiao, and N. Zhan, "Safety guarantee for time-delay systems with disturbances by control barrier functionals," *Sci. China Inf. Sci.*, pp. 1–15, 2021.

[40] A. K. Kiss, T. G. Molnar, A. D. Ames, and G. Orosz, "Control barrier functionals: Safety-critical control for time delay systems," 2022, *arXiv:2206.08409*.

[41] Z. Liu, L. Yang, and N. Ozay, "Scalable computation of controlled invariant sets for discrete-time linear systems with input delays," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 4722–4728.

[42] A. Singletary, Y. Chen, and A. D. Ames, "Control barrier functions for sampled-data systems with input delays," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 804–809.

[43] M. Jankovic, "Control barrier functions for constrained control of linear systems with input delay," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 3316–3321.

[44] I. Abel, M. Jankovic, and M. Krstić, "Constrained stabilization of multi-input linear systems with distinct input delays," *IFAC-PapersOnLine*, vol. 52, no. 2, pp. 82–87, 2019.
[45] M. Krstic, *Delay Compensation for Nonlinear, Adaptive, and PDE Systems*. Basel, Switzerland: Birkhäuser, 2009.
[46] N. Bekiaris-Liberis and M. Krstic, *Nonlinear Control Under Nonconstant Delays*. Philadelphia, PA, USA: SIAM, 2013.
[47] I. Karafyllis and M. Krstic, *Predictor Feedback for Delay Systems: Implementations and Approximations*. Basel, Switzerland: Birkhäuser, 2017.
[48] W. Michiels and S.-I. Niculescu, *Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach*. Philadelphia, PA, USA: SIAM, 2007.
[49] I. Abel, M. Janković, and M. Krstić, "Constrained control of input delayed systems with partially compensated input delays," in *Proc. Dyn. Syst. Control Conf.*, vol. 84270. New York, NY, USA: American Society of Mechanical Engineers, 2020, p. 8.
[50] I. Abel, M. Krstić, and M. Janković, "Safety-critical control of systems with time-varying input delay," *IFAC-PapersOnLine*, vol. 54, no. 18, pp. 169–174, 2021.
[51] T. G. Molnar, A. Alan, A. K. Kiss, A. D. Ames, and G. Orosz, "Input-to-state safety with input delay in longitudinal vehicle control," 2022, *arXiv:2205.14567*.
[52] R. Konda, A. D. Ames, and S. Coogan, "Characterizing safety: Minimal control barrier functions from scalar comparison systems," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 523–528, Apr. 2021.
[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[54] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 322–328.
[55] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 474–479.
[56] M. Sarkar, D. Ghose, and E. A. Theodorou, "High-relative degree stochastic control Lyapunov and barrier functions," 2020, *arXiv:2004.03856*.
[57] C. Wang, Y. Meng, Y. Li, S. L. Smith, and J. Liu, "Learning control barrier functions with high relative degree for safety-critical control," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 1459–1464.
[58] 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-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 98–106.

Tamas G. Molnar (Member, IEEE) received the B.Sc. degree in mechatronics engineering and the M.Sc. and Ph.D. degrees in mechanical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2013, 2015, and 2018, respectively.

He was a Post-Doctoral Researcher with the University of Michigan, Ann Arbor, MI, USA, from 2018 to 2020. Since 2020, he has been a Post-Doctoral Fellow with the California Institute of Technology, Pasadena, CA, USA. His research interests include nonlinear dynamics and control, safety-critical control, and time delay systems with applications to connected automated vehicles, robotic systems, and machine tool vibrations.

Adam K. Kiss received the B.Sc. and M.Sc. degrees in mechanical engineering from the Budapest University of Technology and Economics (BME), Budapest, Hungary, in 2013 and 2015, where he is currently pursuing the Ph.D. degree.

He is a Research Assistant with the MTA-BME Lendület Machine Tool Vibration Research Group, Department of Applied Mechanics, BME. His current research interests include nonlinear dynamics, safety-critical control, and time delay systems with applications to machine tool vibrations and connected automated vehicles.

Aaron D. Ames (Fellow, IEEE) received the B.S. degree in mechanical engineering and the B.A. degree in mathematics from the University of St. Thomas, Saint Paul, MN, USA in 2001, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 2006.

From 2006 to 2008, he served as a Post-Doctoral Scholar in control and dynamical systems with the California Institute of Technology (Caltech), Pasadena, CA, USA. In 2008, he began his faculty career at Texas A&M University, College Station, TX, USA. He was an Associate Professor with the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Since 2017, he has been a Bren Professor of mechanical and civil engineering and control and dynamical systems with Caltech. His research interests include 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.

Dr. Ames was a recipient of the 2005 Leon O. Chua Award for Achievement in Nonlinear Science and the 2006 Bernard Friedman Memorial Prize in Applied Mathematics from the University of California, Berkeley. He received the NSF CAREER award in 2010, the 2015 Donald P. Eckman Award, and the 2019 IEEE CSS Antonio Ruberti Young Researcher Prize.

Gábor Orosz (Senior Member, IEEE) received the M.Sc. degree in engineering physics from the Budapest University of Technology, Budapest, Hungary, in 2002, and the Ph.D. degree in engineering mathematics from the University of Bristol, Bristol, U.K., in 2006.

He held postdoctoral positions at the University of Exeter, Exeter, U.K., and the University of California, Santa Barbara, CA, USA. In 2010, he joined the University of Michigan, Ann Arbor, MI, USA, where he is currently an Associate Professor in mechanical engineering and in civil and environmental engineering. His research interests include nonlinear dynamics and control, time delay systems, and machine learning with applications to connected and automated vehicles, traffic flow, and biological networks.