

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, *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 must operate safely in dynamically changing environments. This paper develops a framework for safety-critical control in dynamic environments, by establishing the notion of *environmental control barrier functions (ECBFs)*. 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 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 modern control systems. Safety-critical control covers a wide spectrum of applications ranging from automated vehicles [1], [2] through robotics [3]–[6] and multi-robot systems [7]–[9], to controlling the spread of infectious diseases [10], [11]. Notably, in many applications safety is affected by a dynamically changing environment that surrounds the control system. For example, robots must avoid collision with other agents in multi-robot systems [12], [13], automated vehicles must be safe with respect to other road users [14], and robotic manipulators must collaborate safely with their human operator [15]–[17].

Such strict safety requirements call for theoretical safety guarantees and provably safe controllers. To achieve safety, control synthesis must take into account how the control system interacts with its environment. As the environment of

This research is supported in part by the National Science Foundation (CPS Award #1932091), Aerovironment and Dow (#227027AT), and supported by the NRD1 Fund (TKP2020 IES, Grant No. BME-IE-MIFM and TKP2020 NC, Grant No. BME-NC).

Tamás G. Molnár 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, Budapest 1111, Hungary (email: kiss.a@mm.bme.hu).

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

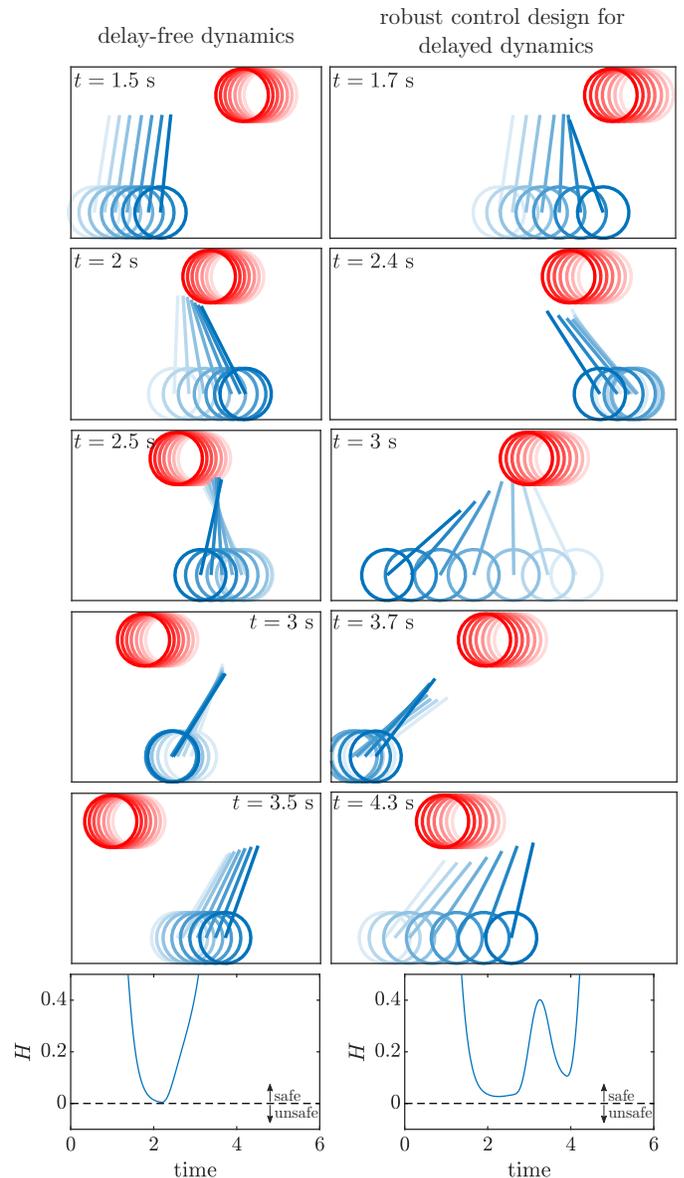


Fig. 1. The proposed safety-critical control framework implemented in high-fidelity simulation of a Segway. The Segway is able to safely avoid a moving obstacle, even when the obstacle’s future position is unknown and there is input delay in the control loop. This is accomplished via the environmental control barrier function (ECBF) plotted at the bottom.

many engineering systems changes over time, one must ensure that this dynamic 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 arising in practice [18]. The magnitude of the delay depends on the application: it is milliseconds in robotic systems [19], a few tenths of a second in connected automated vehicles [20] and days in epidemiological models [21]. Delays have significant impact on safety in the context of dynamic environments: it may occur that by the time the control system responds, the environment changes and safety is compromised. To overcome this danger, one needs to consider how the dynamic – and often uncertain – environment evolves over the delay period. This yields a major challenge in designing provably safe controllers. This paper aims to address this problem by establishing a framework for safety-critical control in which dynamic environments and time delays are explicitly taken into account.

A. State of the Art

Formally, safety is often framed as a set invariance problem by requiring that the state of the system evolves within a safe set for all time. The theory of *control barrier functions (CBFs)* provides an elegant solution to achieve this goal and maintain safety [22]. While this theory delivers the required 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]–[27], measurement uncertainty [28]–[30] and model mismatches [31], [32]. These strategies significantly facilitate controller deployment, however, their formulation has not yet considered dynamic environments explicitly but is restricted to quasi-static environments. For the first time, this paper *explicitly* involves *dynamically changing environments* into the mathematical framework for safety-critical control.

On the other hand, the safety of time delay systems has also attracted increasing attention in the recent literature. The safety of continuous-time systems with state delay was established by safety functionals in [33], [34], which were extended to control barrier functionals in [35]. Discrete-time control systems with input delay were studied in [36] for linear and in [37] for nonlinear dynamics. Linear control systems with input delays were investigated in continuous time in [38], [39] via control barrier and control 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. In these papers, we leveraged predictor feedback [40]–[43] to compensate the delay by predicting the future evolution of the system. Parallel to our work, [44], [45] used predictor feedback to compensate multiple and time-varying input delays with robustness against prediction errors. Yet, these works have not addressed safety-critical scenarios in dynamic environments that evolve independently of the control input being synthesized. This paper intends to fill this gap and

tackle the challenges arising from the combination of dynamic environments and delays.

B. Contributions

Here we build on [10], [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 develop the notions of CBFs and ECBFs for systems with input delay, and synthesize safety-critical controllers via predictor feedback. The use of predictors requires special care in dynamic environments, since the environment’s future is not controlled and cannot be predicted accurately. We make controllers robust against prediction errors, especially those related to the future of the dynamically changing environment.
3. We demonstrate the efficacy of this framework on real-life engineering systems: adaptive cruise control and obstacle avoidance with a Segway platform.

Figure 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 backwards to go under the obstacle. With input delay (right) the Segway first approaches the obstacle, then moves in reverse to make space, and finally pitches forward to go under it. Remarkably, these safe behaviors emerge from the ECBF automatically, which handles reactive planning in a holistic fashion.

The paper 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, adaptive cruise control 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 input $u(t) \in U \subseteq \mathbb{R}^m$:

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

where $f : X \rightarrow \mathbb{R}^n$ and $g : X \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz continuous on X that is an open and connected subset of \mathbb{R}^n . Consider the initial condition $x(0) = x_0 \in X$. When the control input $u = k(x)$ is given by a locally Lipschitz continuous controller $k : X \rightarrow U$, then system (1) has a unique solution over a time interval $t \in I(x_0)$. For simplicity of exposition, 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 0-superlevel set of a continuously differentiable function $h : X \rightarrow \mathbb{R}$:

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

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

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

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

where

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

is the derivative of h along system (1).

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*

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

$\forall x \in S$ 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 [46]. Throughout the paper, we use 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 $u_d = k_d(x)$ by a desired controller $k_d : X \rightarrow U$, one can modify this input in a minimally invasive fashion to guarantee safety by solving the following quadratic program (QP):

$$\begin{aligned} k(x) = \underset{u \in U}{\operatorname{argmin}} \quad & \|u - k_d(x)\|^2 \\ \text{s.t.} \quad & \dot{h}(x, u) \geq -\alpha(h(x)). \end{aligned} \quad (6)$$

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 particular choice of h is indeed a CBF is nontrivial when there are input constraints ($U \subset \mathbb{R}^m$). If there are no 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 KKT conditions [47] as

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

where $\phi_1^+(x) = \phi_1^T(x)/(\phi_1(x)\phi_1^T(x))$ is the right pseudoinverse of $\phi_1(x)$. The derivation of (7) is given in Appendix I. Note that for $\phi_1^+(x)$ to exist, one needs $\nabla h(x)g(x) \neq 0, \forall x \in S$. This 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 [48]–[51] 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 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 $e(0) = e_0 \in E$. Then the safe set modifies to an *environmental safe set* S_e :

$$S_e = \{(x, e) \in X \times E : H(x, e) \geq 0\}, \quad (8)$$

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

A. Environmental Control Barrier Functions

We enforce safety in dynamic environments by introducing the notion of *environmental control barrier functions (ECBFs)*.

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

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

where

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

is the derivative of H along system (1).

With this definition, 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*

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

$\forall (x, e) \in S_e$ 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. We couple (1) with its environment into the augmented system

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

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

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

For this system, function $H_z : X \times E \rightarrow \mathbb{R}$, $H_z(z) = H(x, e)$ is a CBF, since H is an ECBF. Hence, safety can be guaranteed with respect to the 0-superlevel set of H_z based on Theorem 1 via the condition

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

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 $u = K(x, e, \dot{e})$ with control law $K : X \times E \times \mathcal{E} \rightarrow U$. That is, the control input depends on the environment as well through e and \dot{e} which need to be measured. For example, a controller based on optimization (specifically, a QP) reads

$$K(x, e, \dot{e}) = \underset{u \in U}{\operatorname{argmin}} \quad \|u - K_d(x, e, \dot{e})\|^2 \quad (15)$$

s.t. $\dot{H}(x, e, \dot{e}, u) \geq -\alpha(H(x, e)),$

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

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

analogously to (7).

B. Robust Safety in Uncertain Environment

ECBFs rely on the environment state e and its derivative \dot{e} . In practice, these quantities are typically estimated with uncertainty. Therefore, now we robustify safety-critical controllers against uncertainties in the environment. Motivated by the method developed in [30] for 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 typically the environment is associated with more uncertainty than the state of the control system.

Consider that 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 $\varepsilon_{\dot{e}}$:

$$\|e - \hat{e}\| \leq \varepsilon_e, \quad \|\dot{e} - \hat{\dot{e}}\| \leq \varepsilon_{\dot{e}}. \quad (17)$$

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 according to Theorem 2. The bound uses the known quantities \hat{e} and $\hat{\dot{e}}$ in the form:

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

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. The 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 $\mathcal{L}_{\nabla H f, e}$, $\mathcal{L}_{\nabla H g, e}$ and $\mathcal{L}_{\alpha \circ H, e}$, whereas $\nabla_e H(x, e)\dot{e}$ is Lipschitz continuous in arguments 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:

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

Then, the following sufficient condition for safety can be constructed.

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*

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

with

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

$\forall (x, \hat{e}) \in S_e$ 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 Theorem 2 in [30]. We prove the Proposition by showing that (20) implies (11) and by applying Theorem 2. We relate (20) to (11) by introducing the difference between their corresponding terms. Applying (10) to express both $\dot{H}(x, e, \dot{e}, u)$ and $\dot{H}(x, \hat{e}, \hat{\dot{e}}, u)$, we get

$$\begin{aligned} \dot{H}(x, e, \dot{e}, u) + \alpha(H(x, e)) \\ = \dot{H}(x, \hat{e}, \hat{\dot{e}}, u) + \alpha(H(x, \hat{e})) \\ + \nabla_x H f|_{x, e} - \nabla_x H f|_{x, \hat{e}} + (\nabla_x H g|_{x, e} - \nabla_x H g|_{x, \hat{e}})u \\ + \nabla_e H \dot{e}|_{x, e, \dot{e}} - \nabla_e H \dot{e}|_{x, \hat{e}, \hat{\dot{e}}} + \alpha \circ H|_{x, e} - \alpha \circ H|_{x, \hat{e}}. \end{aligned} \quad (22)$$

These differences show up on the left-hand side of (19). Thus, the regularity conditions (19) on H , the uncertainty bound (17) and condition (20, 21) sufficiently provide (11), which completes the proof. ■

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

Example 1 (Adaptive Cruise Control). We consider an adaptive cruise control (ACC) problem, where an automated vehicle (AV) intends to follow a human-driven vehicle (HV) without collision; see Fig. 2. This problem was studied in [1], [2] without the notion of ECBFs. We revisit this problem and use ECBFs to tackle it, which will play an essential role to

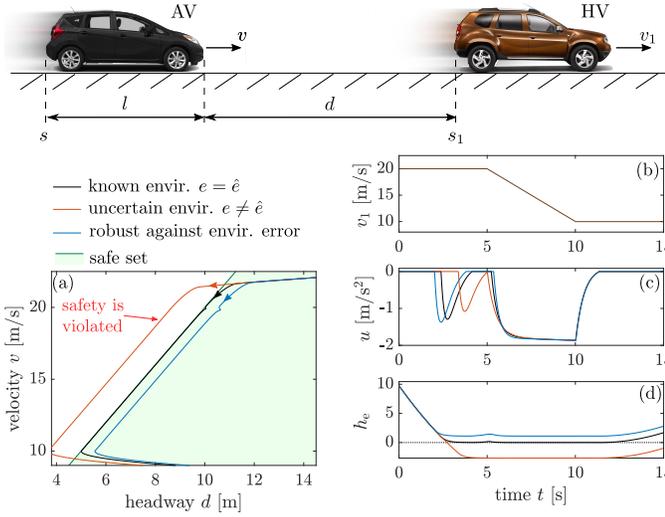


Fig. 2. Strategies for adaptive cruise control where an automated vehicle (AV) intends to safely follow a human-driven vehicle (HV). The HV represents an environment for the AV. In the ideal scenario where the HV's position and speed are accurately known to the AV, controller (15) ensures safety (black). 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).

extend the resulting safety-critical controller to safe ACC with input delay in Section IV.

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 longitudinal motion by

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

where $p(v)$ indicates resistance terms. The input u is the commanded acceleration that is assumed to be realized by a low-level controller. We denote the HV's position and speed by s_1 and v_1 . These quantities 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 $\bar{V}(d) = \bar{\kappa}d$ for a selected $\bar{\kappa} > 0$, where this limit depends on the distance $d = s_1 - s - l$. Thus, we use the ECBF:

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

and select a linear 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, while using (10), into the safety condition (11) leads to

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

This implies that the AV should not accelerate more than how much the expression on the left-hand side dictates. 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 $p(v) = 0.1 + 0.0003v^2$, $\gamma = 3$ and $\bar{\kappa} = 2$ (with units in SI). The speed of the HV is presented in panel (b): it performs constant speed cruising, braking with 2 m/s^2 and constant speed cruising again. 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 the constraint (25), the AV is able to slow down safely behind the HV; see the black curve.

The controller relies on the position and speed of the HV. These can be obtained by on-board sensors such as radar, lidar, cameras or ultrasonics, or by means of vehicle-to-vehicle connectivity with the HV. If these quantities are measured with error, safety may be violated. This is demonstrated by red color in Fig. 2, where the controller relies on the measured value $\hat{e} = \hat{s}_1 = s_1 + 1 \text{ m}$ and $\hat{\dot{e}} = \hat{v}_1 = v_1 + 1 \text{ m/s}$ instead of the true values $e = s_1$ and $\dot{e} = v_1$. That is, both the position and speed of the HV are overestimated, which 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

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

where we used the Lipschitz coefficients $\mathcal{L}_{\nabla H f, e} = \mathcal{L}_{\nabla H g, e} = \mathcal{L}_{\nabla H \dot{e}, e} = 0$, $\mathcal{L}_{\alpha \circ H, e} = \gamma \bar{\kappa}$, $\mathcal{L}_{\nabla H \dot{e}, \dot{e}} = \bar{\kappa}$. 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 by blue color in Fig. 2 for $\varepsilon_s = 1.4 \text{ m}$ and $\varepsilon_v = 1.4 \text{ m/s}$. 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 $\tau > 0$:

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

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 $u(t)$ that will be synthesized via CBFs and also on the control input over $[t - \tau, t]$ given by the input history $u_t \in \mathcal{B}$:

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

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* Ψ as a function of the state $x(t)$ and as a functional of the input history u_t :

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

The semi-flow $\Psi : [0, \tau] \times X \times \mathcal{B} \rightarrow X$ can be obtained by the forward integration of (27) as follows:

$$\begin{aligned} & \Psi(\vartheta, x, u_t) \\ &= x + \int_0^\vartheta \left(f(\Psi(\varphi, x, u_t)) + g(\Psi(\varphi, x, u_t))u_t(\varphi - \tau) \right) d\varphi. \end{aligned} \quad (30)$$

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

$$x(t + \tau) = \Psi(\tau, x(t), u_t). \quad (31)$$

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 a 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 may also be available for linear systems

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

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, where the predicted state is given by the convolution integral

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

B. Control Barrier Functions 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 control barrier function (CBF) for (27) with $\tau > 0$ if there exists an extended class \mathcal{K} function α such that for all $x \in S$ and $u_t \in \mathcal{B}$

$$\sup_{u \in U} \dot{h}(x_p, u) \geq -\alpha(h(x_p)), \quad (34)$$

where $x_p = \Psi(\tau, x, u_t)$ with Ψ given by (30).

Notice that the definition recovers Definition 1 for the delay-free case, since $x_p = x$ if $\tau = 0$. We use this definition to 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 the system is safe 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 $u = k(x_p)$, $x_p = \Psi(\tau, x, u_t)$ with input history u_t satisfying*

$$\dot{h}(x_p, u) \geq -\alpha(h(x_p)), \quad (35)$$

$\forall x \in S$ 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

$$\begin{aligned} \frac{d}{d\vartheta} \Psi(\vartheta, x(t), u_t) &= f(\Psi(\vartheta, x(t), u_t)) \\ &+ g(\Psi(\vartheta, x(t), u_t))u(t + \vartheta - \tau). \end{aligned} \quad (36)$$

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

$$\dot{x}_p(t) = f(x_p(t)) + g(x_p(t))u(t). \quad (37)$$

For this system, Theorem 1 can be applied since (34, 35) hold, thus we get $x_p(0) \in S \Rightarrow x_p(t) \in S$, $\forall t \geq 0$ 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 feedback of the predicted state $x_p = \Psi(\tau, x, u_t)$. Furthermore, optimization-based controllers for systems with input delay can be synthesized via Theorem 3 similarly to (6). The following QP can be solved if $\tau > 0$:

$$\begin{aligned} k(x_p) &= \underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \|u - k_d(x_p)\|^2 \\ \text{s.t. } & \dot{h}(x_p, u) \geq -\alpha(h(x_p)). \end{aligned} \quad (38)$$

Here the desired controller $k_d : X \rightarrow U$ 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 $x_p = \Psi(\tau, x, u_t)$. 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}_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, $\hat{x}_p = x_p$. However, mismatches between \hat{x}_p and x_p inevitably occur

due to model uncertainties and computation errors [42]. Robustness against the *prediction error* $\hat{x}_p - x_p$ can be provided analogously to Proposition 1 using the following condition:

$$\dot{h}(\hat{x}_p, u) - (\mathcal{L}_{\nabla h f} + \mathcal{L}_{\alpha \circ h})\varepsilon_x - \mathcal{L}_{\nabla h g}\varepsilon_x \|u\| \geq -\alpha(h(\hat{x}_p)), \quad (39)$$

where ε_x is the prediction error bound, $\|\hat{x}_p - x_p\| \leq \varepsilon_x$, and \mathcal{L} is the Lipschitz coefficient of the subscripted function on S . Moreover, safety can also be studied under the notion of input-to-state safety [24], [27] as we did in [11], where it was shown that without robustification the input disturbance $d = k(\hat{x}_p) - k(x_p)$ may make a larger set $S_d \supseteq S$ forward invariant.

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 . In Theorem 3, the key step to achieve safety was to predict the state of the system over the time interval $[t, t+\tau]$. Similarly, this section will rely on a prediction for the environment. For simplicity of exposition, we assume that the future state of the environment is determined by the current state in the form:

$$\begin{aligned} e(t + \vartheta) &= \Gamma(\vartheta, e(t)), \quad \vartheta \in [0, \tau]. \\ \dot{e}(t + \vartheta) &= \dot{\Gamma}(\vartheta, e(t), \dot{e}(t)), \quad \vartheta \in [0, \tau]. \end{aligned} \quad (40)$$

where the maps $\Gamma : [0, \tau] \times E \rightarrow E$ and $\dot{\Gamma} : [0, \tau] \times E \times \mathcal{E} \rightarrow \mathcal{E}$ may be unknown. We call $e(t + \tau) = \Gamma(\tau, e(t))$ and $\dot{e}(t + \tau) = \dot{\Gamma}(\tau, e(t), \dot{e}(t))$ as the *prediction of the environment*.

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

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

$$\sup_{u \in U} \dot{H}(x_p, e_p, \dot{e}_p, u) \geq -\alpha(H(x_p, e_p)), \quad (41)$$

where $x_p = \Psi(\tau, x, u_t)$ with Ψ given by (30), while $e_p = \Gamma(\tau, e)$, $\dot{e}_p = \dot{\Gamma}(\tau, e, \dot{e})$ with $\Gamma, \dot{\Gamma}$ defined by (40).

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

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

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 $u = K(x_p, e_p, \dot{e}_p)$,*

$x_p = \Psi(\tau, x, u_t)$, $e_p = \Gamma(\tau, e)$, $\dot{e}_p = \dot{\Gamma}(\tau, e, \dot{e})$ with history u_t satisfying

$$\dot{H}(x_p, e_p, \dot{e}_p, u) \geq -\alpha(H(x_p, e_p)), \quad (42)$$

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

Proof. Assumption 2 yields $(x_0, e_0) \in S_e \Rightarrow (x(\vartheta), e(\vartheta)) \in S_e$, $\forall \vartheta \in [0, \tau]$, thus what remains to prove is $(x(\tau), e(\tau)) \in S_e \Rightarrow (x(t), e(t)) \in S_e$, $\forall t \geq \tau$. This is equivalent to $(x_p(0), e_p(0)) \in S_e \Rightarrow (x_p(t), e_p(t)) \in S_e$, $\forall t \geq 0$ based on the definitions of x_p and e_p . According to the proof of Theorem 3, $x_p(t)$ 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 $(x_p(0), e_p(0)) \in S_e \Rightarrow (x_p(t), e_p(t)) \in S_e$, $\forall t \geq 0$ as desired, which completes the proof. ■

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

$$\begin{aligned} K(x_p, e_p, \dot{e}_p) &= \underset{u \in U}{\operatorname{argmin}} \|u - K_d(x_p, e_p, \dot{e}_p)\|^2 \\ \text{s.t.} \quad &\dot{H}(x_p, e_p, \dot{e}_p, u) \geq -\alpha(H(x_p, e_p)), \end{aligned} \quad (43)$$

with $x_p = \Psi(\tau, x, u_t)$, $e_p = \Gamma(\tau, e)$ and $\dot{e}_p = \dot{\Gamma}(\tau, e, \dot{e})$, cf. (15, 38).

Remark 6. In practice, the future state of the environment given by e_p is unknown, we can only potentially provide an estimate \hat{e}_p with corresponding estimate $\hat{\dot{e}}_p$ for the derivative. Robustness with respect to *environment prediction errors* is a significant problem since the evolution of the environment is often more uncertain than the dynamics of the control system. Robustness can be addressed similarly to Section III-B, as discussed below. For simplicity, we assume that the dynamics of the control system (27) is well-known and its state can be predicted with negligible error ($\hat{x}_p = x_p$); 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:

$$\dot{H}(x_p, \hat{e}_p, \hat{\dot{e}}_p, u) - C(\varepsilon_e, \varepsilon_{\dot{e}}, u) \geq -\alpha(H(x_p, \hat{e}_p)), \quad (44)$$

with $C(\varepsilon_e, \varepsilon_{\dot{e}}, u)$ defined in (21), where ε_e and $\varepsilon_{\dot{e}}$ are the error bounds on the environment prediction, satisfying $\|e_p - \hat{e}_p\| \leq \varepsilon_e$ and $\|\dot{e}_p - \hat{\dot{e}}_p\| \leq \varepsilon_{\dot{e}}$.

Example 2 (Adaptive Cruise Control with input delay). Consider the adaptive cruise control problem outlined in Example 1, now with input delay:

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

where τ represents powertrain delays. For passenger vehicles, this delay is around 0.5–1 s [20], hence it is not negligible for safety-critical applications.

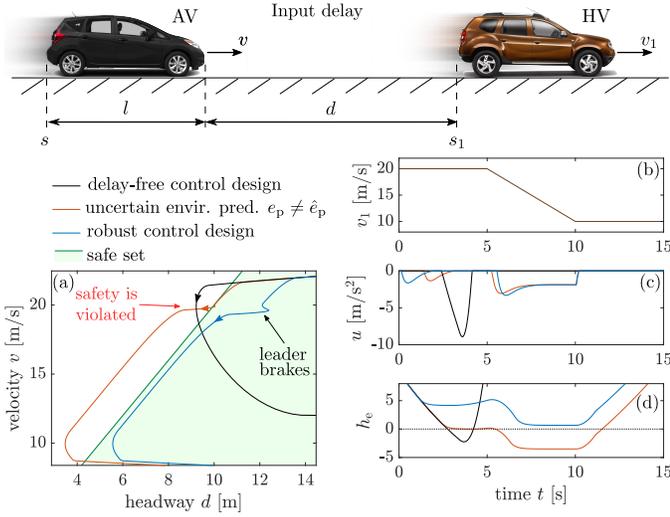


Fig. 3. Strategies for adaptive cruise control with input delay. The naïve implementation of the delay-free control design (15) violates safety (black). The controller (43) that relies on predictor feedback enforces safety as long as the prediction of the HV’s motion is accurate, and violates safety otherwise (red). When controller (43) is robustified via constraint (44), safety is maintained despite prediction errors (blue).

The effect of this delay is demonstrated in Fig. 3 by black color. Simulation results are shown with a large delay $\tau = 1$, zero initial input history, and the same parameters as in Example 2: $p(v) = 0.1 + 0.0003v^2$, $\gamma = 3$ 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 violations happen even when the HV drives with constant speed.

Thus, we use Theorem 4 to maintain 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 $x_p(t) = [s_p(t), v_p(t)]^T$. Furthermore, we predict the HV’s motion by assuming constant speed over $[t, t + \tau]$: $\hat{e}_p(t) = v_1(t)$ and $\hat{e}_p(t) = s_1(t) + v_1(t)\tau$. These predictions can be incorporated into the safety condition (42) to synthesize a control input satisfying

$$\bar{\kappa}(v_1 - v_p) + \gamma(\bar{\kappa}(s_1 + v_1\tau - s_p - l) - v_p) + p(v_p) \geq u, \quad (46)$$

cf. (25).

Red color in Fig. 3 shows the result of executing the corresponding controller given by QP (43) with desired controller $K_d(x_p, e_p, \dot{e}_p) = 0$ and constraint (46). The controller is able to maintain safety as long as the HV travels at constant speed and the corresponding prediction accurately captures HV’s future motion ($\hat{e}_p(t) = \dot{e}_p(t)$ and $\hat{e}_p(t) = \dot{e}_p(t)$). Then, safety is violated once the HV starts to slow down and the prediction is no longer equal to the true future motion of the HV ($\hat{e}_p(t) \neq \dot{e}_p(t)$ and $\hat{e}_p(t) \neq \dot{e}_p(t)$). 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 with respect to this uncertainty.

Assuming that the HV’s acceleration is limited to a range $[-a_{\min}, a_{\max}]$, we have the following physical bounds for the

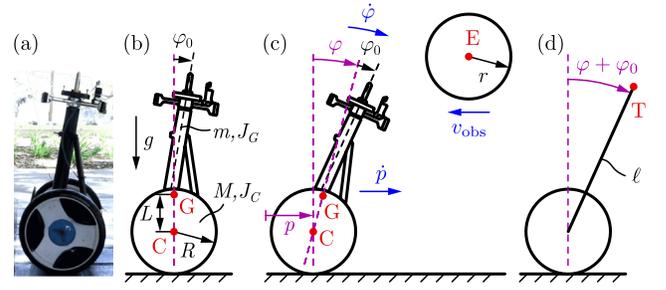


Fig. 4. (a) The Ninebot E+ Segway platform. (b) Its mechanical model illustrated in equilibrium position. (c) The Segway in motion aiming to avoid a moving obstacle. (d) Simplified representation of the Segway.

environment prediction error: $\varepsilon_e = \bar{a}\tau$ and $\varepsilon_e = \bar{a}\tau^2/2$ with $\bar{a} = \max\{a_{\min}, a_{\max}\}$. Then the robustified condition (44) leads to the form

$$\bar{\kappa}(v_1 - \bar{a}\tau - v_p) + \gamma(\bar{\kappa}(s_1 + v_1\tau - \bar{a}\tau^2/2 - s_p - l) - v_p) + p(v_p) \geq u, \quad (47)$$

cf. (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 $a_{\min} < a_{\max}$.

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

V. CASE-STUDY: CONTROL OF A SEGWAY

Now we apply the theoretical constructions of this paper to a real-life robotic system: we consider the control of a Ninebot E+ Segway platform [52] 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 characterize the planar motion of the Segway using its position p and pitch angle φ via the mechanical model shown in Fig. 4(b,c). Fig. 4(b) illustrates the Segway in equilibrium, where the center of mass of its frame (point G) is located above the the wheel center (point C). Notice 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, executing the obstacle avoidance, and Fig. 4(d) illustrates a simplified representation of the wheel and frame.

Our goal is to drive the Segway forward with a desired speed \dot{p}_d while avoiding a moving, circular obstacle centered at $[e, y]^T$ (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

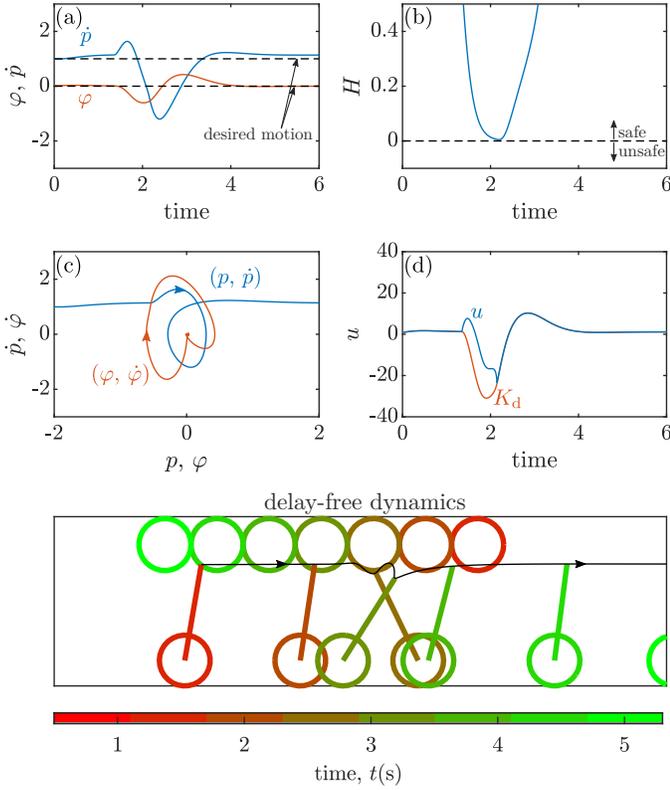


Fig. 5. Safety-critical control of the Segway to avoid a moving obstacle. The dynamics (48) are delay-free and the obstacle position is known. Using a controller that satisfies (53), the Segway successfully and safely avoids the obstacle.

v_{obs} : $e = e_0 - tv_{\text{obs}}$, $\dot{e} = -v_{\text{obs}}$, $\dot{y} = 0$. For numerical case-study, we consider $\dot{p}_d = 1 \text{ m/s}$, $r = 0.2 \text{ m}$, $v_{\text{obs}} = 0.5 \text{ m/s}$, $e_0 = 1 \text{ m}$ and $y = 1.0418 \text{ m}$ (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 delays.

A. Safety-Critical Control in Dynamic Environment

We describe the dynamics of the Segway using its position p and pitch angle φ as a planar, two-degrees-of-freedom system with general coordinates $q = [p, \varphi]^T \in \mathcal{Q}$. With the general velocities $\dot{q} = [v, \omega]^T \in \mathbb{R}^2$, the state becomes $x = [p, \varphi, v, \omega]^T \in X$. Note that 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 is the voltage applied on the motors at the wheels, $u \in \mathbb{R}$. The dynamics are governed by

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

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 [52] and are listed in Table I.

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

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

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

$$H(x, e) = d^T d - r^2, \quad (50)$$

$$d = \begin{bmatrix} p + \ell \sin(\varphi + \varphi_0) - e \\ R + \ell \cos(\varphi + \varphi_0) - y \end{bmatrix},$$

where d is the vector pointing from the obstacle center to the tip of the Segway.

We seek to maintain safety with respect to the environmental safe set (8) using Theorem 2. However, H is not a valid ECBF since \dot{H} does not depend on the input u since $\nabla_x H(x, e)g(x) = 0$. Therefore, we consider a dynamic extension of the ECBF based on [48]. We define the following *extended environmental control barrier function*:

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

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

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

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

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

according to Theorem 2. Notice that this condition involves the second derivative \ddot{e} as well.

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 $\gamma = 7.5 \text{ s}^{-1}$ and $\gamma_e = 7.5 \text{ s}^{-1}$. The performance of the controller is demonstrated in Fig. 5 for a scenario with known obstacle position. Snapshots of the motion are illustrated at the bottom whereas the characteristics of the motion are quantified at the top. Panel (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. Panel (b) indicates that the obstacle is safely avoided as H is positive for all time. Panel (c) shows the corresponding phase portrait, whereas panel (d) indicates 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:

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

cf. (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 $\tau = 0.1 \text{ s}$. 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

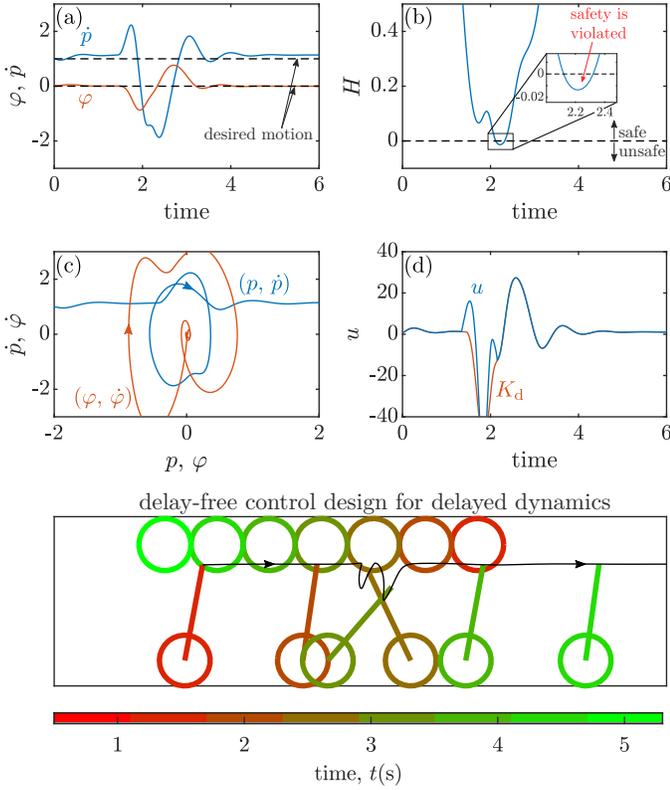


Fig. 6. Safety-critical control of the Segway to avoid a moving obstacle. The dynamics (54) involve an input delay. The naïve implementation of the delay-free control design based on (53) fails to avoid the obstacle.

buffering the obstacle, formal safety guarantees no longer hold with delay. Moreover, the control input is much larger for the case with input delay than without delay, cf. Fig. 5(d) and Fig. 6(d). Although this example did not consider physical bounds on the control input, large inputs are undesired, and safety-critical control could become infeasible when considering 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 $\|e_p - \hat{e}_p\| \leq \varepsilon_e$, $\|\dot{e}_p - \hat{\dot{e}}_p\| \leq \varepsilon_{\dot{e}}$ and $\|\ddot{e}_p - \hat{\ddot{e}}_p\| \leq \varepsilon_{\ddot{e}}$. Analogously to (44), we use the robustified safety constraint

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

with

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

cf. (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 detailed expressions of the Segway dynamics, and they are listed in 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 is given by $\ddot{e}_p = 0$, $\dot{e}_p = -v_{\text{obs}}$ and $e_p = e - v_{\text{obs}}\tau$, which is unknown to the controller. Instead, the controller relies on the prediction $\hat{\ddot{e}}_p = 0$, $\hat{\dot{e}}_p = -(v_{\text{obs}} - \Delta v)$ and $\hat{e}_p = e - (v_{\text{obs}} - \Delta v)\tau$. That is, we consider that the speed of the obstacle is underestimated by $\Delta v = 0.05$ m/s. The controller is robustified with respect to 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 is able to safely execute the obstacle avoidance task, despite the delay in the control loop and the uncertainty in the obstacle's future position. The Segway achieves this with a qualitatively different motion than in the delay-free case. For zero delay in Fig. 5, the Segway pitches backwards to go under the obstacle. For nonzero delay in Fig. 7, the Segway first moves in reverse to get away from the obstacle, then pitches forward to go under it. Notably, this behavior is automatically generated by the control barrier function, and with provable guarantees of safety.

VI. CONCLUSIONS

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 environmental control barrier functions 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 utilizes worst-case uncertainty bounds and is provably safe. We have demonstrated the method by an adaptive cruise control problem where the motion of other road participants create an uncertain environment, and by a Segway controller executing a moving obstacle avoidance task. Our future work includes further analysis of prediction errors, control of systems with both state and input delays, and considering control barrier functionals acting on delayed states.

APPENDIX I KKT CONDITIONS

This appendix shows the derivation of the solution (6) to the quadratic program (7) for the case without input constraints ($U = \mathbb{R}^m$). Let us define $\Delta k(x) = k(x) - k_d(x)$ and consider the expressions of $\dot{h}(x, u)$ in (4) and $\phi_0(x)$, $\phi_1(x)$ in (7) with $\phi_1(x) \neq 0$. Then, we can restate (6) as

$$\begin{aligned} \Delta k(x) &= \underset{\Delta u \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|\Delta u\|^2 \\ \text{s.t.} \quad &\phi_0(x) + \phi_1(x)\Delta u \geq 0. \end{aligned} \quad (57)$$

This optimization problem has convex objective and affine constraint, hence the *Karush-Kuhn-Tucker (KKT) conditions* [47] provide the necessary and sufficient conditions for optimality. The KKT optimality conditions imply that there

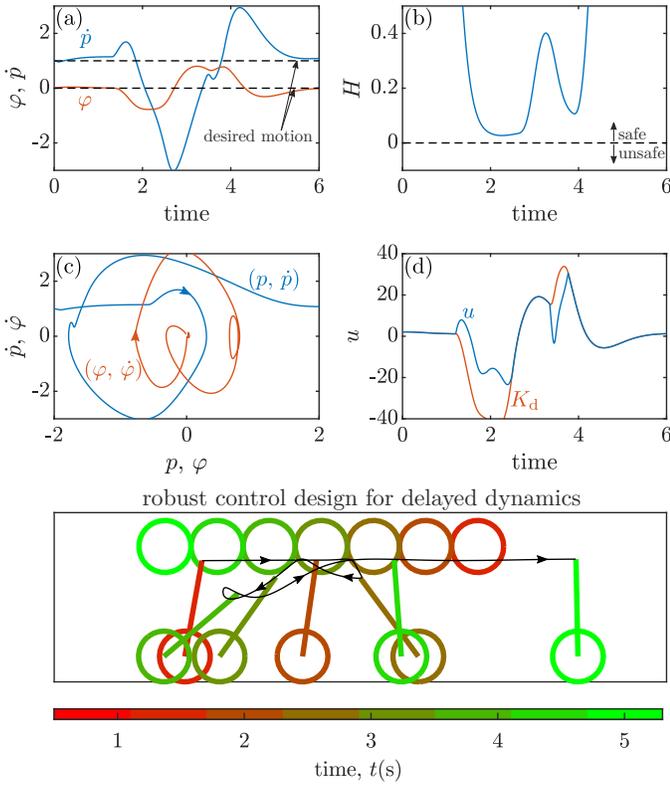


Fig. 7. Safety-critical control of the Segway to avoid a moving obstacle. The dynamics (54) involve an input delay that is compensated via predictor feedback. The controller is designed based on (55), taking into account prediction errors. The Segway successfully accomplishes the obstacle avoidance task despite the delay and the uncertain future motion of the obstacle.

exists a Lagrange multiplier $\mu : X \rightarrow \mathbb{R}$ such that $\mu(x)$ and $\Delta k(x)$ satisfy

$$\mu(x) \geq 0, \quad (58)$$

$$\Delta k(x) = \mu(x)\phi_1^\top(x), \quad (59)$$

$$\phi_0(x) + \phi_1(x)\Delta k(x) \geq 0, \quad (60)$$

$$\mu(x)(\phi_0(x) + \phi_1(x)\Delta k(x)) = 0, \quad (61)$$

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

$$\Delta k(x) = 0, \quad (62)$$

and with the primal feasibility condition (60) this leads to

$$\phi_0(x) \geq 0. \quad (63)$$

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

$$\phi_0(x) + \phi_1(x)\Delta k(x) = 0. \quad (64)$$

Recall that $\phi_0(x)$ is a scalar ($\phi_0(x) \in \mathbb{R}$), $\phi_1(x)$ is a vector ($\phi_1(x) \in \mathbb{R}^n$), $\phi_1(x)$ is nonzero, and its right pseudoinverse is denoted by $\phi_1^+(x) = \phi_1^\top(x)/(\phi_1(x)\phi_1^\top(x))$. Then, we can express $\Delta k(x)$ from (64) as

$$\Delta k(x) = -\phi_0(x)\phi_1^+(x). \quad (65)$$

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

$$\phi_0(x) = -\phi_1(x)\Delta k(x) = -\mu(x)\phi_1(x)\phi_1^\top(x) < 0, \quad (66)$$

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

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

or more compactly as [10]

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

Since $k(x) = k_d(x) + \Delta k(x)$, we finally obtain (7) as the solution to the quadratic program (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 [52]. 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 center of mass (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 position, where G is located above C. Note that since the frame is asymmetric, 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

$$\begin{aligned} \omega_w &= \dot{p}/R, & \omega_f &= \dot{\varphi}, \\ v_C &= \begin{bmatrix} \dot{p} \\ 0 \end{bmatrix}, & v_G &= \begin{bmatrix} \dot{p} + L\dot{\varphi} \cos \varphi \\ -L\dot{\varphi} \sin \varphi \end{bmatrix}. \end{aligned} \quad (69)$$

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

$$\begin{aligned} T &= \frac{1}{2}Mv_C^2 + \frac{1}{2}J_C\omega_w^2 + \frac{1}{2}mv_G^2 + \frac{1}{2}J_G\omega_f^2 \\ &= \frac{1}{2}m_0\dot{p}^2 + mL\dot{p}\dot{\varphi} \cos \varphi + \frac{1}{2}J_0\dot{\varphi}^2, \end{aligned} \quad (70)$$

where $m_0 = m + M + J_C/R^2$ and $J_0 = mL^2 + J_G$. The potential energy of the Segway is

$$U = mgL \cos \varphi. \quad (71)$$

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

$$P = M_d(\omega_w - \omega_f) = Q_p \dot{p} + Q_\varphi \dot{\varphi}, \quad (72)$$

yielding the general forces $Q_p = M_d/R$ and $Q_\varphi = -M_d$.

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:

$$\begin{aligned} u &= R_a i + K_b(\omega_w - \omega_f), \\ M_d &= K_t i, \end{aligned} \quad (73)$$

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

$$M_d = K_m u - b_t(\dot{p} - R\dot{\varphi}), \quad (74)$$

with constants $K_m = K_t/R_a$ and $b_t = K_t K_b/(R_a R)$.

With these preliminaries, we can write Lagrange's equations

$$\begin{aligned} \frac{d}{dt} \frac{\partial T}{\partial \dot{p}} - \frac{\partial T}{\partial p} + \frac{\partial U}{\partial p} &= Q_p, \\ \frac{d}{dt} \frac{\partial T}{\partial \dot{\varphi}} - \frac{\partial T}{\partial \varphi} + \frac{\partial U}{\partial \varphi} &= Q_\varphi, \end{aligned} \quad (75)$$

which, after substitution, lead to

$$\begin{aligned} m_0 \ddot{p} + mL \cos \varphi \ddot{\varphi} - mL \sin \varphi \dot{\varphi}^2 &= \frac{M_d}{R}, \\ mL \cos \varphi \ddot{p} + J_0 \ddot{\varphi} - mgL \sin \varphi &= -M_d. \end{aligned} \quad (76)$$

Ultimately, we obtain the equations of motion in the form

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

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

$$\begin{aligned} D(q) &= \begin{bmatrix} m_0 & mL \cos \varphi \\ mL \cos \varphi & J_0 \end{bmatrix}, \quad B = \begin{bmatrix} K_m/R \\ -K_m \end{bmatrix}, \\ H(q, \dot{q}) &= \begin{bmatrix} -mL \sin \varphi \dot{\varphi}^2 + b_t/R(\dot{p} - R\dot{\varphi}) \\ -mgL \sin \varphi - b_t(\dot{p} - R\dot{\varphi}) \end{bmatrix}. \end{aligned} \quad (78)$$

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

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

which leads to

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

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

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

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
	m_0	52.710	kg
	J_0	5.108	kgm ²
	a	0.6768	m
	b	4.7274	-
	c	68.5205	1/s ²
combined parameters	κ	0.9713	Vs/m
	A	1.1605	m/s ² /V
	B	0.3344	m/s ² /V
	C	1.7147	1/s ² /V
	D	2.3355	1/s ² /V

whereas those of the control matrix read

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

with parameters

$$\begin{aligned} a &= \frac{J_0}{mL}, \quad b = \frac{m_0 J_0}{m^2 L^2}, \quad c = \frac{m_0 g}{mL}, \quad \kappa = \frac{b_t}{K_m}, \\ A &= \frac{K_m J_0}{m^2 L^2 R}, \quad B = \frac{K_m}{mL}, \quad C = \frac{K_m}{mLR}, \quad D = \frac{K_m m_0}{m^2 L^2}. \end{aligned} \quad (83)$$

The values of all parameters are listed in Table I. These were identified for the Ninebot E+ Segway platform in [52].

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

$$H(x, e) = h_0(x) + h_1(x)e + e^2, \quad (84)$$

with coefficients

$$\begin{aligned} h_0(x) &= (p + \ell \sin(\varphi + \varphi_0))^2 \\ &\quad + (R + \ell \cos(\varphi + \varphi_0) - y)^2 - r^2, \\ h_1(x) &= -2(p + \ell \sin(\varphi + \varphi_0)). \end{aligned} \quad (85)$$

Then, the extended ECBF in (51) becomes

$$H_e(x, e, \dot{e}) = H_0(x) + H_1(x)e + h_1(x)\dot{e} + \gamma_e e^2 + 2e\dot{e}, \quad (86)$$

where

$$\begin{aligned} H_0(x) &= \nabla_p h_0(x)v + \nabla_\varphi h_0(x)\omega + \gamma_e h_0(x), \\ H_1(x) &= \nabla_p h_1(x)v + \nabla_\varphi h_1(x)\omega + \gamma_e h_1(x). \end{aligned} \quad (87)$$

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

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

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 $\nabla_e H_e \dot{e}$, we can write

$$\begin{aligned}
&\nabla_e H_e \dot{e}(x, e, \dot{e}) - \nabla_e H_e \dot{e}(x, \hat{e}, \hat{\dot{e}}) \\
&= H_1(x)(e - \hat{e}) + 2\gamma_e(e\dot{e} - \hat{e}\hat{\dot{e}}) + 2(\dot{e}^2 - \hat{\dot{e}}^2) \\
&= H_1(x)(e - \hat{e}) + 2\gamma_e \dot{e}(e - \hat{e}) + 2\gamma_e \hat{\dot{e}}(\dot{e} - \hat{\dot{e}}) \\
&\quad + 2\hat{\dot{e}}(\dot{e} - \hat{\dot{e}}) + 2\dot{e}(\dot{e} - \hat{\dot{e}}) \\
&\geq -(|H_1(x)| + 2\gamma_e \max_{\dot{e} \in D_{\dot{e}}} |\dot{e}|)|e - \hat{e}| \\
&\quad - (2\gamma_e |\hat{\dot{e}}| + 2|\hat{\dot{e}}| + 2 \max_{\dot{e} \in D_{\dot{e}}} |\dot{e}|)|\dot{e} - \hat{\dot{e}}|,
\end{aligned} \tag{90}$$

hence the corresponding Lipschitz coefficients are

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

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 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

$$\begin{aligned}
\mathcal{L}_{\nabla H_e f, e} &= |C_1(x)|, \\
\mathcal{L}_{\nabla H_e f, \dot{e}} &= |C_2(x)|, \\
\mathcal{L}_{\nabla H_e g, e} &= |C_4(x)|, \\
\mathcal{L}_{\nabla H_e g, \dot{e}} &= 0, \\
\mathcal{L}_{\nabla H_e \ddot{e}, e} &= 2 \max_{\ddot{e} \in D_{\ddot{e}}} |\ddot{e}|, \\
\mathcal{L}_{\nabla H_e \ddot{e}, \dot{e}} &= 0, \\
\mathcal{L}_{\nabla H_e \ddot{e}, \ddot{e}} &= |h_1(x)| + 2|\hat{\ddot{e}}|, \\
\mathcal{L}_{\alpha \circ H_e, e} &= \gamma |H_1(x)| + \gamma\gamma_e (|\hat{\dot{e}}| + \max_{e \in D_e} |e|) + 2\gamma \max_{\dot{e} \in D_{\dot{e}}} |\dot{e}|, \\
\mathcal{L}_{\alpha \circ H_e, \dot{e}} &= \gamma |h_1(x)| + 2\gamma |\hat{\dot{e}}|.
\end{aligned} \tag{92}$$

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, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.
- [2] C. R. He and G. Orosz, "Safety guaranteed connected cruise control," in *21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 549–554.
- [3] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," *arXiv preprint*, no. arXiv:2103.14252, 2021.
- [4] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 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," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.
- [6] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.
- [7] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination planner for multi-robot networks using Lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, 2016.
- [8] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [9] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *57th IEEE Conference on Decision and Control*, 2018, pp. 1991–1996.
- [10] A. D. Ames, T. G. Molnár, A. W. Singletary, and G. Orosz, "Safety-critical control of active interventions for COVID-19 mitigation," *IEEE Access*, vol. 8, pp. 188 454–188 474, 2020.
- [11] T. G. Molnár, A. W. Singletary, G. Orosz, and A. D. Ames, "Safety-critical control of compartmental epidemiological models with measurement delays," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1537–1542, 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, pp. 332–347, 2014.
- [13] M. Santillo and M. Jankovic, "Collision free navigation with interacting, non-communicating obstacles," in *American Control Conference*, 2021, pp. 1637–1643.
- [14] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, 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 Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2016.
- [16] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *18th European Control Conference*, 2019, pp. 2565–2570.
- [17] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames, "Online active safety for robotic manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 173–178.
- [18] G. Stépán, *Retarded Dynamical Systems: Stability and Characteristic Functions*. Longman, UK, 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 *IEEE 14th International Conference on Machine Learning and Applications*, 2015, pp. 168–175.
- [20] X. A. Ji, T. G. Molnár, A. A. Gorodetsky, and G. Orosz, "Bayesian inference for time delay systems with application to connected automated vehicles," in *24th IEEE International Conference on Intelligent Transportation Systems*, 2021.
- [21] F. Casella, "Can the COVID-19 epidemic be controlled on the basis of daily test reports?" *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1079–1084, 2021.
- [22] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

- [23] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.
- [24] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019.
- [25] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," *arXiv preprint*, no. arXiv:2104.02808, 2021.
- [26] L. Zheng, R. Yang, J. Pan, and H. Cheng, "Safe learning-based tracking control for quadrotors under wind disturbances," in *2021 American Control Conference*, 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 Systems Letters*, 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 *Conference on Control Technology and Applications*, 2018, pp. 300–305.
- [29] A. Clark, "Control barrier functions for stochastic systems," *arXiv preprint*, no. arXiv:2003.03498, 2019.
- [30] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," *arXiv preprint*, no. arXiv:2010.16001, 2020.
- [31] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *International Conference on Robotics and Automation*, 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 *Robotics: Science and Systems*, 2020.
- [33] G. Orosz and A. D. Ames, "Safety functionals for time delay systems," in *American Control Conference*, 2019, pp. 4374–4379.
- [34] 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 *American Control Conference*, 2021, pp. 1055–1060.
- [35] W. Liu, Y. Bai, L. Jiao, and N. Zhan, "Safety guarantee for time-delay systems with disturbances by control barrier functionals," *Science China Information Sciences*, pp. 1–15, 2021.
- [36] Z. Liu, L. Yang, and N. Ozay, "Scalable computation of controlled invariant sets for discrete-time linear systems with input delays," *arXiv:2003.04953*, 2020.
- [37] A. Singletary, Y. Chen, and A. D. Ames, "Control barrier functions for sampled-data systems with input delays," *arXiv:2005.06418*, 2020.
- [38] M. Jankovic, "Control barrier functions for constrained control of linear systems with input delay," in *American Control Conference*, 2018, pp. 3316–3321.
- [39] 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.
- [40] M. Krstic, *Delay Compensation for Nonlinear, Adaptive, and PDE Systems*. Birkhäuser, 2009.
- [41] N. Bekiaris-Liberis and M. Krstic, *Nonlinear Control Under Nonconstant Delays*. SIAM, 2013.
- [42] I. Karafyllis and M. Krstic, *Predictor feedback for delay systems: Implementations and approximations*. Basel: Birkhäuser, 2017.
- [43] W. Michiels and S.-I. Niculescu, *Stability and stabilization of time-delay systems: An eigenvalue-based approach*. SIAM, 2007.
- [44] I. Abel, M. Janković, and M. Krstić, "Constrained control of input delayed systems with partially compensated input delays," in *Dynamic Systems and Control Conference*, vol. 84270. American Society of Mechanical Engineers, 2020, p. V001T04A006.
- [45] 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.
- [46] R. Konda, A. D. Ames, and S. Coogan, "Characterizing safety: Minimal control barrier functions from scalar comparison systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 523–528, 2021.
- [47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [48] Q. Nguyen and K. Sreenath, "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016, pp. 322–328.
- [49] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *58th IEEE Conference on Decision and Control*, 2019, pp. 474–479.
- [50] M. Sarkar, D. Ghose, and E. A. Theodorou, "High-relative degree stochastic control Lyapunov and barrier functions," *arXiv preprint*, no. arXiv: 2004.03856, 2020.
- [51] C. Wang, Y. Li, Y. Meng, S. L. Smith, and J. Liu, "Learning control barrier functions with high relative degree for safety-critical control," *arXiv preprint*, no. arXiv:2011.10721, 2020.
- [52] 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 *ACM/IEEE 9th International Conference on Cyber-Physical Systems*, 2018, pp. 98–106.



Tamas G. Molnar received his B.Sc. degree in Mechatronics Engineering, M.Sc. and Ph.D. degrees in Mechanical Engineering from the Budapest University of Technology and Economics, Hungary, in 2013, 2015 and 2018. He held post-doctoral position at the University of Michigan, Ann Arbor between 2018 and 2020. Since 2020 he is a postdoctoral fellow at the California Institute of Technology, Pasadena. 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 his B.Sc. and M.Sc. degrees in mechanical engineering from the Budapest University of Technology and Economics (BME) in 2013 and 2015, where he is currently pursuing the Ph.D. degree. He is currently a research assistant at the MTA-BME Lendület Machine Tool Vibration Research Group and at the 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 is the Bren Professor of Mechanical and Civil Engineering and Control and Dynamical Systems at Caltech. 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 & Computer Engineering. He received a B.S. in Mechanical Engineering and a B.A. in Mathematics from the University of St. Thomas in 2001, and he received a M.A. in Mathematics and a Ph.D. in Electrical Engineering and Computer Sciences from 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.



Gabor Orosz received the M.Sc. degree in Engineering Physics from the Budapest University of Technology, Hungary, in 2002 and the Ph.D. degree in Engineering Mathematics from University of Bristol, UK, in 2006. He held postdoctoral positions at the University of Exeter, UK, and at the University of California, Santa Barbara. In 2010, he joined the University of Michigan, Ann Arbor 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 reinforcement learning with applications to connected and automated vehicles, traffic flow, and biological networks.