Annual Reviews in Control 57 (2024) 100947

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Safety-critical control for autonomous systems: Control barrier functions via reduced-order models

Max H. Cohen [a,*], Tamas G. Molnar [b], Aaron D. Ames [a]

[a] Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, 91125, USA
[b] Department of Mechanical Engineering, Wichita State University, Wichita, KS, 67260, USA

ARTICLE INFO

Keywords:
Safety-critical control
Control barrier functions
Reduced-order models
Autonomous systems

ABSTRACT

Modern autonomous systems, such as flying, legged, and wheeled robots, are generally characterized by high-dimensional nonlinear dynamics, which presents challenges for model-based safety-critical control design. Motivated by the success of reduced-order models in robotics, this paper presents a tutorial on constructive safety-critical control via reduced-order models and control barrier functions (CBFs). To this end, we provide a unified formulation of techniques in the literature that share a common foundation of constructing CBFs for complex systems from CBFs for much simpler systems. Such ideas are illustrated through formal results, simple numerical examples, and case studies of real-world systems to which these techniques have been experimentally applied.

1. Introduction

The control stack for modern autonomous systems – from legged robots to self-driving vehicles – typically consists of a complex interconnection of decision-making, planning, and control modules, all of which may leverage different model representations to strike a balance between computational efficiency, model uncertainty, and satisfaction of system-level specifications. Among the various specifications that such autonomous systems must satisfy, safety – informally thought of as requiring a system never to do anything "bad" – is often given precedence, as the violation of specifications deemed to be safety-critical could result in undesirable behavior. Over the past decade, control barrier functions (CBFs) (Ames, Coogan, Egerstedt, Notomista, Sreenath, & Tabuada, 2019; Ames, Grizzle, & Tabuada, 2014; Ames, Xu, Grizzle, & Tabuada, 2017; Xu, Tabuada, Grizzle, & Ames, 2015) have emerged as a powerful tool for designing controllers that ensure the safety of autonomous systems. Despite their success, constructing CBFs for high-dimensional autonomous systems remains an open challenge since their dynamics may be nontrivial or not even known.

To address these challenges, there has been recent interest in constructing CBFs for complex autonomous systems based on reduced-order models (ROMs) — lower-dimensional representations that are rich enough to capture the high-level behavior of the full-order system but that are simple enough to synthesize safety-critical controllers (Molnar & Ames, 2023b; Molnar, Cosner, Singletary, Ubellacker, & Ames, 2022; Singletary, Klingebiel, Bourne, Browning, Tokumaru, & Ames, 2021). This approach has demonstrated success in controlling seemingly complex systems, such as underactuated and dynamic robotic systems, in a computationally efficient manner, and naturally integrates into the existing control stack present in many autonomous systems.

In this paper, we provide a self-contained introduction and detailed overview of CBF techniques based on ROMs. Here, we highlight the theoretical foundations of this approach and illustrate its applications across different domains through a collection of case studies. Before diving into this discussion, however, we first review current state-of-the-art techniques in the field of safety-critical control and motivate the techniques covered and perspective taken in this tutorial.

1.1. The different flavors of control barrier functions

The property of safety is often formalized using the framework of set invariance (Blanchini & Miani, 2008) in which a system is said to be safe if its trajectories remain within a desirable set of the state space (Ames et al., 2019). That is, a closed-loop system is safe if there exists an invariant set that does not intersect with a set of states deemed by the user to be dangerous. Such an invariant set is referred to as a *safe set*.

By moving from invariant sets to controlled invariant sets – those that can be rendered forward invariant through the application of a feedback controller – this notion of safety may also be applied to systems with control inputs. Control designs in which safety is a

* Corresponding author.
E-mail address: maxcohen@caltech.edu (M.H. Cohen).

https://doi.org/10.1016/j.arcontrol.2024.100947
Received 23 November 2023; Accepted 4 March 2024
Available online 15 March 2024

high-priority requirement are often referred to as *safety-critical* controllers. Among the various tools that have emerged to address safety-critical control, including, but not limited to, model predictive control (MPC) (Borrelli, Bemporad, & Morari, 2017; Hewing, Wabersich, Menner, & Zeilinger, 2020), reachability analysis (Bansal, Chen, Herbert, & Tomlin, 2017; Mitchell, Bayen, & Tomlin, 2005), and symbolic control (Belta, Yordanov, & Gol, 2017; Tabuada, 2009), CBFs (Ames et al., 2019, 2017) have demonstrated success in synthesizing safety-critical controllers for high-dimensional nonlinear systems.

Since the introduction of CBFs (Ames et al., 2014, 2017) (see Ames et al. (2019) for a more in-depth survey on the history of CBFs), there has been a large body of work in developing various types of CBFs for different classes of systems and control objectives. Given that CBFs are a model-based tool, and that most models are coarse representations of the underlying system, many of these developments have been motivated by controlling systems subject to uncertainty (Wabersich, Taylor, Choi, Sreenath, Tomlin, Ames, & Zeilinger, 2023). These include, for example, robust CBFs for systems with unstructured uncertainty (Alan, Taylor, He, Ames, & Orosz, 2023; Alan, Taylor, He, Orosz, & Ames, 2022; Jankovic, 2018; Kolathaya & Ames, 2019), adaptive CBFs for systems with parametric uncertainty (Cohen & Belta, 2023; Lopez, Slotine, & How, 2021; Taylor & Ames, 2020), data-driven CBFs for systems with unknown dynamics (Brunke, Zhou, & Schoellig, 2022; Dhiman, Khojasteh, Franceschetti, & Atanasov, 2023; Taylor, Singletary, Yue, & Ames, 2020), and stochastic CBFs for systems with stochastic dynamics (Cosner, Culbertson, Taylor, & Ames, 2023; Santoyo, Dutreix, & Coogan, 2021).

Other lines of work have developed classes of CBFs to account for different assumptions on systems' actuation and sensing capabilities. For example, measurement-robust (Dean, Taylor, Cosner, Recht, & Ames, 2020) and observer-based CBFs (Agrawal & Panagou, 2023; Wang & Xu, 2022) have been developed to design safety-critical controllers for systems with measurement uncertainty, whereas event-triggered (Long & Wang, 2022; Taylor, Ong, Cortés, & Ames, 2021; Xiao, Belta, & Cassandras, 2023; Yang, Belta, & Tron, 2019) and sampled-data CBFs (Breeden, Garg, & Panagou, 2021; Ghaffari, Abel, Ricketts, Lerner, & Krstić, 2018; Taylor, Dorobantu, Cosner, Yue & Ames, 2022) have been developed to enforce safety when one may only update control inputs at discrete instances in time. Variants of CBFs have also been developed to address more nuanced notions of safety including input-to-state safety (ISSf) (Alan et al., 2022; Kolathaya & Ames, 2019) and finite/fixed/prescribed-time safety (Abel, Steeves, Krstić, & Janković, 2023; Garg & Panagou, 2021; Polyakov & Krstic, 2023), whereas others have been used to enforce satisfaction of more general temporal logic specifications (Cohen, Serlin, Leahy & Belta, 2023; Lindemann & Dimarogonas, 2019; Srinivasan & Coogan, 2021).

1.2. Constructive methods for control barrier functions

Although much attention has been given to defining different classes of CBFs for various systems and control objectives of interest, relatively less attention has been given to the construction of such CBFs. As a result, there exists a plethora of different types of CBFs, but a lack of constructive techniques required to obtain such CBFs in the first place. This lack of constructive techniques often limits the applicability of CBFs to relatively simple low-dimensional systems. Motivated by these limitations, researchers have begun to investigate constructive techniques for safety-critical control and CBFs.

A central challenge in constructing a CBF is finding a scalar function whose time derivative directly depends on the system's control input and whose zero superlevel set defines a controlled invariant subset of the state space. This challenge highlights the crucial distinction between a *safe* set and a *constraint* set. The former is a controlled invariant set that does not intersect with the set of failure states. The latter is simply the set of states deemed by the user to not be in violation of a given safety constraint. These sets need not coincide and, in general, they do not. For example, in robot motion planning problems, the "distance to the obstacle" function – depending only on the robot's position – defines the obstacle-free space (constraint set) but is not a CBF (i.e., it does not yield a safe set) unless the derivatives of the position directly depend on the control inputs.

The challenges mentioned above are related to the *relative degree* of the function – the number of times it must be differentiated along the system dynamics until the input appears – defining the safety constraint. A popular approach to address such challenges is through the use of *extended*, also called exponential (Nguyen & Sreenath, 2016) or high-order (Xiao & Belta, 2022), CBFs, which have roots in work on non-overshooting control (Krstić & Bement, 2006). Here, one differentiates a high relative degree constraint function until the control input appears and then enforces CBF-like conditions upon its highest-order derivative. Such an approach has demonstrated success in safety-critical control of high-dimensional systems (Xiao, Cassandras & Belta, 2023), but also faces challenges in verifying the satisfaction of CBF-like conditions (Tan, Cortez, & Dimarogonas, 2022).

Some limitations of extended CBFs have been addressed by leveraging the structure present in certain classes of systems. For example, constructive CBF techniques have been developed for robotic systems (Cortez & Dimarogonas, 2022; Cortez, Oetomo, Manzie & Choong, 2021; Cortez, Verginis & Dimarogonas, 2021; Singletary, Kolathaya & Ames, 2022) by exploiting structural properties of their dynamics. Other approaches have sought to extend Lyapunov backstepping (Krstić, Kanellakopoulus, & Kokotović, 1995) to CBFs for systems in strict-feedback form (Taylor, Ong, Molnar & Ames, 2022).

Other works have sought to address the limitations outlined above by leveraging implicitly defined CBFs, often constructed by propagating forward the dynamics of the system in a receding-horizon fashion (Breeden & Panagou, 2023) under a "backup" (Chen, Jankovic, Santillo, & Ames, 2021; Gurriet, Mote, Singletary, Nilsson, Feron, & Ames, 2020) or performance-based policy (Breeden & Panagou, 2022). Such approaches have close connections with MPC, and, indeed, one may also leverage MPC techniques to construct CBFs in a receding horizon manner (Wabersich et al., 2023; Wabersich & Zeilinger, 2022). Although powerful, these techniques often require additional online computation that may prohibit their use for real-time control of high-dimensional systems.

To address these limitations, alternative approaches seek to shift the computational burden of constructing a CBF offline where one may leverage powerful optimization tools to build a CBF. For example, sum-of-squares programming has been used to construct CBFs for systems with polynomial dynamics (Clark, 2021, 2022; Dai & Permenter, 2023; Zhao, Ghabcheloo, Cheng, Abdi, & Hovakimyan, 2023). Other works have sought to bridge the gap between reachability analysis and CBFs (Choi, Lee, Li, How, Sreenath, Herbert, & Tomlin, 2023; Choi, Lee, Sreenath, Tomlin, & Herbert, 2021; Tonkens & Herbert, 2022), and illustrate that a CBF for a general class of nonlinear systems can be constructed from the value function of a particular discounted optimal control problem. Although promising, these techniques are limited by the computation needed to solve sum-of-squares programs or compute value functions over a grid, both of which scale poorly with the state dimension.

The computational challenges in constructing CBFs using offline optimization have motivated the use of learning-based techniques to learn CBFs from data. Such approaches model the CBF using a suitable class of function approximators, such as neural networks, and train such a model to satisfy the criteria of a CBF either directly (Dawson, Gao, & Fan, 2023; Dawson, Qin, Gao, & Fan, 2022; So, Serlin, Mann, Gonzales, Rutledge, Roy, & Fan, 2023) or by using data from expert demonstrations (Lindemann, Hu, Robey, Zhang, Dimorogonas, Tu, & Matni, 2020; Robey, Hu, Lindemann, Zhang, Dimorogonas, Tu, & Matni, 2020). These learning-based approaches empirically perform well but also face the challenge of verifying if the trained model satisfies CBF conditions for safety, which may preclude their application to systems where safety must be rigorously certified.

1.3. Control barrier functions via reduced-order models

Modern autonomous systems, such as flying, legged, and wheeled robots, are generally characterized by high-dimensional nonlinear dynamics. Although CBF-based controllers may, in principle, be applied to such systems, this first requires constructing a CBF for a complex high-dimensional nonlinear system — a task that many of the aforementioned methods struggle with. Rather than directly constructing a CBF for a complicated system, an alternative approach is to construct a CBF for a much simpler system, and then attempt to relate the inputs that enforce safety of this simpler system back to the inputs of the original system. That is, one may use a *reduced-order* representation of the original, full-order, dynamics for the purpose of control design, and then refine such a controller for the full-order system provided its dynamic behavior is sufficiently captured by the reduced-order model.

Such control designs, despite leveraging simple models, have demonstrated success in different areas of robotics. In mobile robotics, single integrator (Zhao & Sun, 2017) and unicycle models (Luca, Oriolo, & Vendittelli, 2001) are often used as the basis for control designs of more complicated nonholonomic systems. In legged robotics, reduced-order models such as the spring-loaded inverted pendulum (Raibert, 1986), linear inverted pendulum (Kajita, Kanehiro, Kaneko, Fujiwara, Yokoi, & Hirukawa, 2002), and hybrid-linear inverted pendulum (Xiong & Ames, 2022) have demonstrated continued success in controlling walking robots with high-dimensional nonlinear dynamics.

Inspired by their success in robotics, there has been recent interest in using reduced-order models for safety-critical control design. In the context of CBFs, such ideas were introduced in Singletary et al. (2021), Singletary, Kolathaya et al. (2022) where CBFs designed for simple kinematic models were used to generate safe velocity commands to be tracked by more complicated robotic systems, such as drones (Singletary et al., 2021) and manipulators (Singletary, Kolathaya et al., 2022). Such control designs were formalized in Molnar et al. (2022) by illustrating that the combination of a CBF for a reduced-order model and a Lyapunov function certifying tracking of the reduced-order trajectory may be used to establish safety of the full-order system. Further extensions and applications of this approach have been reported in Kim, Lee, and Ames (2023), Molnar and Ames (2023b), Singletary, Guffey, Molnar, Sinnet, and Ames (2022). Although not explicitly framed as safety-critical control based on reduced-order models, CBF backstepping (Taylor, Ong et al., 2022) shares with these approaches the ability to construct CBFs for complicated systems from CBFs for simple models.

1.4. Objective of this paper

The primary objective of this paper is to provide a tutorial presentation of CBF techniques based on reduced-order models. In doing so, we present a unified formulation of techniques in the literature that share a common foundation of constructing CBFs for complex systems from CBFs for much simpler systems. These ideas are illustrated through formal results, simple numerical examples, and high-level overviews of more complicated applications. The majority of the stated theoretical results have already been established, in one form or another, in the various works cited herein. For illustrative purposes, the proofs of selected results are provided in the Appendix. Other results are new but are also minor extensions or combinations of existing results. For completeness, the proofs of such results are also collected in the Appendix. All the numerical examples presented in this tutorial can be reproduced using open-source code available on Github.[1]

[1] https://github.com/maxhcohen/ReducedOrderModelCBFs.jl

1.5. Organization and outline

The remainder of this paper is organized as follows.

In Section 2, we provide a self-contained introduction to safety-critical control via CBFs. First, we review the characterization of safety via set invariance (Blanchini & Miani, 2008) and barrier functions (Ames et al., 2017) and then discuss how such ideas may be extended to design safety-critical controllers using CBFs. Next, we discuss how CBFs may be extended to disturbed systems using the framework of ISSf (Alan et al., 2023, 2022; Kolathaya & Ames, 2019), leading to the synthesis of robust safety-critical controllers. Finally, we review the concept of a smooth safety filter (Cohen, Ong, Bahati & Ames, 2023) — a class of differentiable CBF-based controllers that will play an important role in synthesizing CBFs via ROMs.

In Section 3, we begin our exposition on the construction of CBFs via ROMs. Here, we first discuss some of the technical challenges in constructing CBFs for high-dimensional systems and then outline various classes of systems whose structure facilitates the synthesis of CBFs using ROMs.

In Section 4, we present our first constructive technique for CBF synthesis, which exploits the idea of CBF backstepping as originally developed in Taylor, Ong et al. (2022). We demonstrate how this approach applies to general classes of systems whose dynamics may be interpreted as a layered control architecture and compare this backstepping approach with existing high-order CBF approaches.

In Section 5, we demonstrate how CBF backstepping may be specialized to robotic systems whose dynamics also exhibit a particularly useful cascaded structure. When such a system is fully actuated, we illustrate how one may directly apply the backstepping approach presented in Section 4 to generate CBFs. We then extend this approach, combining it with the notion of an *energy-based* CBF (Singletary, Kolathaya et al., 2022), which further exploits the structure of the robot dynamics to construct CBFs. Finally, using ideas inspired by those from (Spong, 1994), we show how CBFs may be constructed for certain classes of underactuated robotic systems.

In Section 6, we illustrate how previous constructions can be understood as combining a CBF for a ROM with a Lyapunov function certifying tracking of the ROM by the full-order dynamics. Such an approach relaxes many of the structural requirements imposed in the previous sections and replaces them with the, perhaps, less strict requirement of the existence of a tracking controller. Moreover, we demonstrate how this approach leads to the paradigm of *model-free* safety-critical control (Molnar et al., 2022) in which one need not directly rely on the full-order dynamics to construct safety-critical controllers.

In Section 7, we revisit more complex application examples from the literature that leverage the constructive CBF techniques outlined in previous sections. These examples include safety-critical control of fixed-wing aircraft, flying, legged and wheeled robots, manipulators, and heavy-duty trucks — both in simulation and hardware experiments.

In Section 8, we highlight the limitations of the paradigms presented in this tutorial and provide our perspective on open research directions.

2. A primer on safety-critical control

2.1. Notation

We use $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$ to denote the set of natural numbers, real numbers, nonnegative real numbers, and positive real numbers, respectively. The notation $\mathbb{R}^n$ denotes the n-dimensional Euclidean vector space. Given a vector $\mathbf{x} \in \mathbb{R}^n$ we write $\mathbf{x}^\top \in \mathbb{R}^{1\times n}$ to denote its transpose and $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^\top \mathbf{y}$ to denote the inner product between vectors. Given a continuously differentiable scalar function $h : \mathbb{R}^n \to \mathbb{R}$ we denote the gradient of h as $\nabla h : \mathbb{R}^n \to \mathbb{R}^n$. We use $L_{\mathbf{f}} h(\mathbf{x}) := \nabla h(\mathbf{x}) \cdot (\mathbf{x})$ to denote the *Lie derivative* of a continuously differentiable scalar function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$. The same definition applies

when taking the Lie derivative of h along a matrix-valued function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ whose columns can be thought of as vector fields on $\mathbb{R}^n$. For a continuously differentiable function $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ we use $\frac{\partial \mathbf{k}}{\partial \mathbf{x}} \in \mathbb{R}^{m\times n}$ to denote the Jacobian matrix of $\mathbf{k}$ evaluated at $\mathbf{x} \in \mathbb{R}^n$. A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to be an *extended class* $\mathcal{K}_\infty$ *function*, denoted by $\alpha \in \mathcal{K}_\infty^e$, if $\alpha(0) = 0$, α is strictly increasing, and $\lim_{s\to\pm\infty} \alpha(s) = \pm\infty$. A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be class $\mathcal{K}_\infty$ function, denoted by $\alpha \in \mathcal{K}_\infty$, if $\alpha(0) = 0$, α is strictly increasing and $\lim_{s\to\infty} \alpha(s) = \infty$. We use $\mathrm{ReLU}(x) := \max\{0, x\}$ to denote the ReLU activation function. For a manifold $\mathcal{Q}$, we use $T_{\mathbf{q}}\mathcal{Q}$ to denote the tangent space to $\mathcal{Q}$ at a point $\mathbf{q} \in \mathcal{Q}$ and $T\mathcal{Q}$ to denote the tangent bundle. We use $\|\mathbf{x}\|$ to denote the Euclidean norm of a vector $\mathbf{x} \in \mathbb{R}^n$ and $\|\mathbf{x}\|_C := \inf_{\mathbf{y}\in C} \|\mathbf{x} - \mathbf{y}\|$ to denote the distance between a vector $\mathbf{x} \in \mathbb{R}^n$ and a set $C \subset \mathbb{R}^n$. Given a function $h : \mathbb{R}^n \to \mathbb{R}$ and set $C \subset \mathbb{R}$ we denote the *restriction* of h to C by $h|_C : C \to \mathbb{R}$. For a closed set $C \subset \mathbb{R}^n$, we use ∂C to denote its boundary and $\mathrm{Int}(C)$ to denote its interior. We use $\mathbf{0}$ to denote a vector or matrix of zeros of appropriate dimension and $\mathbf{I}$ to denote an identity matrix of appropriate dimension, where all dimensions will be made clear from the context.

2.2. Foundations of safety-critical control

In this subsection, we outline the foundations of safety-critical control based on the fundamental notion of set invariance. We begin by considering the dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz vector field. Then, for each initial condition $\mathbf{x}_0 \in \mathbb{R}^n$, the dynamics in (1) generate a unique continuously differentiable trajectory $\mathbf{x} : I(\mathbf{x}_0) \to \mathbb{R}^n$ defined on some maximal interval of existence $I(\mathbf{x}_0) \subseteq \mathbb{R}_{\geq 0}$ satisfying:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t)) \\ \mathbf{x}(0) &= \mathbf{x}_0, \end{aligned} \tag{2}$$

for all $t \in I(\mathbf{x}_0)$ (Khalil, 2002, Ch. 3).

The main property of (1) studied in this paper is *safety*, which is formalized by requiring trajectories of (1) to remain within a safe set $C \subset \mathbb{R}^n$ at all times.

Definition 1 (*Safety (Ames et al., 2019)*)**.** A set $C \subset \mathbb{R}^n$ is said to be *forward invariant* for (1) if for each initial condition $\mathbf{x}_0 \in C$, the resulting trajectory $\mathbf{x} : I(\mathbf{x}_0) \to \mathbb{R}^n$ satisfies $\mathbf{x}(t) \in C$ for all $t \in I(\mathbf{x}_0)$. System (1) is said to be *safe* on a set $C \subset \mathbb{R}^n$ if C is forward invariant.

Necessary and sufficient conditions for set invariance, and thus safety, can be characterized using the notion of tangent cones[2] (Bony, 1969; Brezis, 1970; Nagumo, 1942; Redheffer, 1972). Informally, the tangent cone $\mathcal{T}_C(\mathbf{x}) \subset \mathbb{R}^n$ to a closed set $C \subset \mathbb{R}^n$ at a point $\mathbf{x} \in \mathbb{R}^n$ is the set of all vectors $\mathbf{v} \in \mathbb{R}^n$ emanating from $\mathbf{x}$ such that if one were to move infinitesimally along $\mathbf{v}$, then one would remain in C. Hence, for $\mathbf{x} \in \mathrm{Int}(C)$ we have $\mathcal{T}_C(\mathbf{x}) = \mathbb{R}^n$, whereas for $\mathbf{x} \notin C$ we have $\mathcal{T}_C(\mathbf{x}) = \emptyset$, implying the tangent cone is nontrivial only on the boundary of C. The above ideas can be formalized concisely using the following definition:

$$\mathcal{T}_C(\mathbf{x}) := \left\{ \mathbf{v} \in \mathbb{R}^n \ : \ \liminf_{\delta\to 0^+} \frac{\|\mathbf{x} + \delta\mathbf{v}\|_C}{\delta} = 0 \right\}. \tag{3}$$

The following result, known as *Nagumo's Theorem*, leverages tangent cones to provide necessary and sufficient conditions for safety.

[2] For a general closed set C one may define various classes of tangent cones, all of which coincide when C is convex. Examples include the Bouligand tangent cone (Bouligand, 1932) and the Clarke tangent cone. In this tutorial, our definition corresponds to the Bouligand tangent cone.

Theorem 1 (*Nagumo's Theorem (Nagumo, 1942)*)**.** *A closed set $C \subset \mathbb{R}^n$ is forward invariant for (1) if and only if for all $\mathbf{x} \in \partial C$:*

$$\mathbf{f}(\mathbf{x}) \in \mathcal{T}_C(\mathbf{x}). \tag{4}$$

Intuitively, Nagumo's Theorem states that C is forward invariant if and only if the vector field characterizing (1) points into or is tangent to C for each point on the boundary of C. Modern proofs of Nagumo's Theorem can be found in Blanchini and Miani (2008, Ch. 4) and Abraham, Marsden, and Ratiu (1983, Ch. 4). Unfortunately, obtaining a closed-form expression to (3) for general closed sets C is often not possible, making the general version of Nagumo's Theorem challenging to apply in practice. To obtain more practical conditions for safety, we must restrict the class of sets whose invariance we wish to certify. Throughout this paper, we focus on sets of the form:

$$\begin{aligned} C &= \{\mathbf{x} \in \mathbb{R}^n \ : \ h(\mathbf{x}) \geq 0\}, \\ \partial C &= \{\mathbf{x} \in \mathbb{R}^n \ : \ h(\mathbf{x}) = 0\}, \\ \mathrm{Int}(C) &= \{\mathbf{x} \in \mathbb{R}^n \ : \ h(\mathbf{x}) > 0\}, \end{aligned} \tag{5}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Before illustrating how such sets yield convenient representations of tangent cones, we require the notion of a *regular value*.

Definition 2 (*Regular value (Abraham et al., 1983)*)**.** A real number $a \in \mathbb{R}$ is said to be a *regular value* of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ if $\nabla h(\mathbf{x}) \neq \mathbf{0}$ whenever $h(\mathbf{x}) = a$.

When C is defined as in (5) and zero is a regular value of h, the tangent cone is straightforward to compute.

Lemma 1 (*Abraham et al., 1983*)**.** *Consider a set $C \subset \mathbb{R}^n$ as in (5) and suppose that zero is a regular value of h. Then:*

$$\mathcal{T}_C(\mathbf{x}) = \{\mathbf{v} \in \mathbb{R}^n \ : \ \nabla h(\mathbf{x}) \cdot \mathbf{v} \geq 0\}, \quad \forall \mathbf{x} \in \partial C. \tag{6}$$

This characterization of tangent cones leads to the following useful corollary of Nagumo's Theorem.

Corollary 1. *Let the conditions of Lemma 1 hold. Then, C is forward invariant for (1) if and only if:*

$$h(\mathbf{x}) = 0 \implies \dot{h}(\mathbf{x}) = L_{\mathbf{f}} h(\mathbf{x}) \geq 0. \tag{7}$$

Note that when zero is not a regular value of h, the condition in (7) does not necessarily imply the forward invariance of C since, in such a situation, the tangent cone does not coincide with (6).

The preceding developments serve as the foundation for *barrier functions* — Lyapunov-like functions that can be used to verify the safety (rather than stability) of nonlinear systems.

Definition 3 (*Barrier function (Xu et al., 2015)*)**.** A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ defining a set $C \subset \mathbb{R}^n$ as in (5) is said to be a *barrier function* for (1) on C if zero is a regular value of h and there exists $\alpha \in \mathcal{K}_\infty^e$ such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\dot{h}(\mathbf{x}) = L_{\mathbf{f}} h(\mathbf{x}) \geq -\alpha(h(\mathbf{x})). \tag{8}$$

Note that since $\alpha(0) = 0$, the condition in (8) implies that in (7), thereby providing a suitable generalization of invariance conditions beyond just the boundary of C. Intuitively, the condition in (8) requires the system to "slow down" as it approaches the boundary of C and stop once it reaches the boundary. Although our definition of a barrier function requires zero to be a regular value of h, this is not strictly necessary. Indeed, the use of an extended class $\mathcal{K}_\infty$ function in conjunction with requiring inequality (8) to hold at points outside of C enables one to dispense with this regularity condition and establish forward invariance using the comparison lemma (Konda, Ames, & Coogan, 2021), providing further generalizations of classical invariance tools. An additional benefit of requiring inequality (8) to hold on a set larger

than $\mathcal{C}$ – in our case, all of $\mathbb{R}^n$ – is that such a condition not only enforces invariance of $\mathcal{C}$, but also attractivity of $\mathcal{C}$. That is, $\mathcal{C}$ is *asymptotically stable*[3] for (1) with $V(\mathbf{x}) = \text{ReLU}(-h(\mathbf{x}))$ as a Lyapunov function.

Theorem 2 (*Xu et al., 2015*)**.** *If $h : \mathbb{R}^n \to \mathbb{R}$ is a barrier function for (1) on a set $\mathcal{C} \subset \mathbb{R}^n$ as in (5), then $\mathcal{C}$ is forward invariant. Moreover, if $\mathcal{C}$ is compact or the vector field $\mathbf{f}$ in (1) is forward complete, then $\mathcal{C}$ is asymptotically stable.*

In the above result, the requirement that (8) holds on all of $\mathbb{R}^n$ is made only for ease of exposition — Theorem 2 and almost all other barrier-related results presented in this tutorial can be generalized to hold on a subset $\mathcal{D} \subseteq \mathbb{R}^n$ such that $\mathcal{C} \subset \mathcal{D}$. Finally, we note that the characterization of set invariance via barrier functions is tight in the sense that, under certain conditions, the existence of a barrier function is necessary and sufficient for forward invariance.

Theorem 3 (*Xu et al., 2015*)**.** *Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function defining a compact set $\mathcal{C} \subset \mathbb{R}^n$ as in (5) and assume zero is a regular value of h. Then, $\mathcal{C}$ is forward invariant for (1) if and only if $h|_{\mathcal{C}} : \mathcal{C} \to \mathbb{R}$ is a barrier function for (1) on $\mathcal{C}$.*

The preceding generalizations of set invariance via barrier functions play an important role in synthesizing controllers enforcing safety, discussed in the following section.

2.3. Control barrier functions

In the previous subsection, we laid the foundation for safety-critical control using the language of set invariance and illustrated how barrier functions provide a useful tool for verifying safety properties of dynamical systems. In this section, we focus our attention on control systems of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{9}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz vector field modeling the *drift* of the system, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is a locally Lipschitz mapping characterizing the *control directions*, and $\mathbf{u} \in \mathbb{R}^m$ is the control input. Defining a notion of safety for a control system, such as in (9), rather than a closed-loop system, such as in (1), requires some modifications. Definition 1 cannot be directly applied to (9) since the trajectories of (9) cannot be determined, in general, until one specifies a control input $\mathbf{u}$. The definition of safety for (9) is captured via the notion of *controlled invariance*.

Definition 4 (*Controlled Invariance (Blanchini & Miani, 2008)*)**.** A set $\mathcal{C} \subset \mathbb{R}^n$ is said to be *feedback controlled invariant* for (9) if there exists a locally Lipschitz feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ such that $\mathcal{C}$ is forward invariant for the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}) := \mathbf{f}_{\text{cl}}(\mathbf{x}). \tag{10}$$

Rather than verifying that an a priori designed controller renders $\mathcal{C}$ forward invariant using the barrier conditions outlined in the previous subsection, our objective in this subsection is to provide a general methodology to design controllers that enforce safety by construction. Towards this objective, the aforementioned barrier conditions suggest designing such a controller so as to satisfy:

$$\underbrace{L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{k}(\mathbf{x})}_{L_{\mathbf{f}_{\text{cl}}}h(\mathbf{x})} \geq -\alpha(h(\mathbf{x})), \tag{11}$$

[3] Note that forward invariance is a necessary condition for asymptotic stability of a set. Thus, barrier functions can also be seen as generalizing Lyapunov functions certifying stability of equilibrium points to Lyapunov functions certifying stability of sets.

implying that such a controller enforces safety of the closed-loop system by Theorem 2. This observation motivates the concept of a *control barrier function* (CBF).

Definition 5 (*Control Barrier Function (Ames et al., 2017)*)**.** A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ defining a set $\mathcal{C} \subset \mathbb{R}^n$ as in (5) is said to be a *control barrier function* for (9) on $\mathcal{C}$ if there exists $\alpha \in \mathcal{K}_\infty^e$ such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{u}\in\mathbb{R}^m} \dot{h}(\mathbf{x},\mathbf{u}) = \sup_{\mathbf{u}\in\mathbb{R}^m} \left\{L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u}\right\} > -\alpha(h(\mathbf{x})). \tag{12}$$

In contrast to Definition 3, we do not explicitly require zero to be a regular value of h in the above definition since this property implicitly follows from the strict inequality in (12). Further motivation behind the use of this strict inequality is presented in Remark 1, and concerns the continuity of controllers synthesized from CBFs. The existence of a CBF implies that for each $\mathbf{x} \in \mathbb{R}^n$ there exists an input $\mathbf{u} \in \mathbb{R}^m$ enforcing the inequality:

$$L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} > -\alpha(h(\mathbf{x})).$$

To use such inputs to enforce safety, we must be able to stitch them together into a locally Lipschitz feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ satisfying (11). Fortunately, the existence of a CBF implies the existence of such a controller.

Theorem 4 (*Ames et al., 2017*)**.** *If $h : \mathbb{R}^n \to \mathbb{R}$ is a CBF for (9) on a set $\mathcal{C} \subset \mathbb{R}^n$ as in (5), then $\mathcal{C}$ is feedback controlled invariant. Furthermore, if a locally Lipschitz feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ satisfies (11) for all $\mathbf{x} \in \mathbb{R}^n$, then $\mathcal{C}$ is forward invariant for (10).*

Although the above theorem guarantees the existence of a controller enforcing safety, it does not explicitly state how to construct one. The most popular approach to constructing CBF-based controllers is to incorporate (11) as a constraint in an optimization problem parameterized by the system state. That is, the controller $\mathbf{x} \mapsto \mathbf{k}(\mathbf{x})$ is itself an optimization problem that returns, for each $\mathbf{x}$, a control input $\mathbf{u} = \mathbf{k}(\mathbf{x})$ satisfying (11). This approach is motivated by the fact that such an inequality defines an affine constraint on the control input, implying $\mathbf{k}(\mathbf{x})$ can often be cast as a quadratic program (QP) that, in many situations, admits a closed-form solution.

Perhaps the greatest utility of this QP-based perspective is the ability to use CBFs as a *safety filter* for a desired control policy $\mathbf{k}_{\text{d}} : \mathbb{R}^n \to \mathbb{R}^m$ whose safety has not yet been established. Often, it is desirable to modify such a controller in a minimally invasive fashion while guaranteeing safety. This leads to the instantiation of safety-critical controllers via the following optimization problem:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^m}{\text{argmin}} \quad & \tfrac{1}{2}\|\mathbf{u} - \mathbf{k}_{\text{d}}(\mathbf{x})\|^2 \\ \text{subject to} \quad & L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})), \end{aligned} \tag{13}$$

which is a QP whose closed-form solution can be obtained by defining:

$$\begin{aligned} a(\mathbf{x}) &:= L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{k}_{\text{d}}(\mathbf{x}) + \alpha(h(\mathbf{x})) \\ b(\mathbf{x}) &:= \|L_{\mathbf{g}}h(\mathbf{x})\|^2, \end{aligned} \tag{14}$$

and applying the Karush-Kuhn Tucker conditions (Boyd & Vandenberghe, 2004) to yield (Alan et al., 2023):

$$\begin{aligned} \mathbf{k}(\mathbf{x}) &= \mathbf{k}_{\text{d}}(\mathbf{x}) + \lambda(a(\mathbf{x}), b(\mathbf{x}))L_{\mathbf{g}}h(\mathbf{x})^\top \\ \lambda(a,b) &:= \begin{cases} 0 & b \leq 0 \\ \text{ReLU}(-a/b) & b > 0, \end{cases} \end{aligned} \tag{15}$$

where λ is the Lagrange multiplier associated with the constraint in (13). Note that, by (15), the controller in (13) allows the desired controller $\mathbf{k}_{\text{d}}$ to be applied so long as it satisfies the barrier condition (11), and provides a minimal correction to $\mathbf{k}_{\text{d}}$ when such a condition is not satisfied. Importantly, the closed-form expression to the QP (13) in (15) obviates the need explicitly solve an optimization problem in

the control loop, which enables the deployment of such controllers on hardware with limited computational resources. Although this closed-form expression is only valid for a single CBF, whereas, in practice, one must often consider multiple CBFs, one often can combine multiple CBFs into one, allowing one to leverage the closed form solution even for arbitrarily complicated safety specifications (Molnar and Ames, 2023a).

Remark 1 (*Strict inequality*)**.** One may note that in (8) and (11) we have used a nonstrict inequality, whereas in the definition of a CBF (12) we have opted for a *strict* inequality. This difference is subtle but plays an important role in ensuring Lipschitz continuity of CBF-based controllers (Jankovic, 2018). In short, the strict inequality preserves Lipschitz continuity of CBF-based controllers at points where $L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0}$ (see Sepulchre, Janković, and Kokotović (1997, Ch. 3.5.3) for a similar discussion in the context of control Lyapunov functions). Such points arise often in practice. For example, any compact safe set[4] will contain points such that $L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0}$. Note that, as a result, one may use a nonstrict inequality in (12) if $L_{\mathbf{g}}h(\mathbf{x}) \neq \mathbf{0}$ for all $\mathbf{x} \in \mathbb{R}^n$. Finally, we note that the strict inequality is a property of the dynamics and safe set irrespective of any particular controller — its purpose is to restrict the class of functions that may serve as a CBF to those that can be used to synthesize a locally Lipschitz feedback controller.

Although constructing a controller given a CBF can be done systematically, constructing a CBF is often more challenging. To determine if a candidate CBF h – a continuously differentiable function defining (5) – is indeed a CBF, one must verify that (12) holds for each $\mathbf{x} \in \mathbb{R}^n$. To do so, one may compute the supremum in (12):

$$\sup_{\mathbf{u}\in\mathbb{R}^m} \left\{ L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \right\} = \begin{cases} \infty & L_{\mathbf{g}}h(\mathbf{x}) \neq \mathbf{0} \\ L_{\mathbf{f}}h(\mathbf{x}) & L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0} \end{cases}$$

and verify that the above result is strictly greater than $-\alpha(h(\mathbf{x}))$. This simplifies to verifying that:

$$L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0} \implies L_{\mathbf{f}}h(\mathbf{x}) > -\alpha(h(\mathbf{x})),$$

for all $\mathbf{x} \in \mathbb{R}^n$. Intuitively, the CBF condition (12) is a scalar inequality, which, when $L_{\mathbf{g}}h(\mathbf{x}) \neq \mathbf{0}$, is always possible to satisfy by simply taking $\mathbf{u}$ as large or small as necessary. When $L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0}$, however, one must rely on the unforced dynamics of the system – captured via $\mathbf{f}$ – to satisfy the CBF inequality. This discussion is formalized via the following lemma.

Lemma 2. *A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is a CBF for (9) on $\mathcal{C}$ if and only if zero is a regular value of h and for all $\mathbf{x} \in \mathbb{R}^n$:*

$$L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0} \implies L_{\mathbf{f}}h(\mathbf{x}) > -\alpha(h(\mathbf{x})). \tag{16}$$

Remark 2 (*Input Constraints*)**.** Lemma 2 provides necessary and sufficient conditions for h to be a CBF when the control input is *unconstrained*, that is, when $\mathbf{u}$ may take any value in $\mathbb{R}^m$. When additional inputs bounds are present in the sense that $\mathbf{u}$ may only take values in a strict subset $\mathcal{U} \subset \mathbb{R}^m$, Lemma 2 provides necessary[5], but not necessarily sufficient conditions that h must satisfy to be a CBF. For ease of exposition, this tutorial will focus on the construction of CBFs *without* additional input bounds. Many of the approaches discussed herein may be extended to include input bounds through the use of backup CBFs (Chen et al., 2021; Gurriet et al., 2020), with more details on the unification of backup CBFs and ROMs discussed in Molnar and Ames (2023b).

[4] If $\mathcal{C}$ is compact and h is continuously differentiable, then h achieves a local maximum over $\mathcal{C}$. At such a local maximum the gradient of h must vanish, implying $L_{\mathbf{g}}h$ will also vanish.

[5] If h is not a CBF without input bounds, then it certainly will not be with input bounds.

For relatively simple systems, Lemma 2 provides a simple condition that one may check to certify that a continuously differentiable function h defining a set $\mathcal{C}$ as in (5) is indeed a CBF. The following example demonstrates such a procedure for a canonical example in the CBF literature: the inverted pendulum.

Example 1 (*Inverted Pendulum*)**.** We now consider the example of an inverted pendulum with state $\mathbf{x} = (\theta, \dot{\theta})$ and dynamics:

$$\underbrace{\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \dot{\theta} \\ \frac{g}{l}\sin(\theta) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \mathbf{u},$$

where $\theta \in \mathbb{R}$ denotes the angular position of the pendulum, g the acceleration due to gravity, l the length of the pendulum, and m the mass of the pendulum. We establish a safety-critical controller for the inverted pendulum by following the corresponding example in Alan et al. (2022). Our objective is to design a controller for the above system that keeps the pendulum upright in the sense that its angular position satisfies $|\theta| \leq \bar{\theta}$ for some $\bar{\theta} \in \mathbb{R}_{>0}$.

To achieve this objective, we propose the CBF candidate:

$$h(\mathbf{x}) = \bar{\theta}^2 - \theta^2 - \frac{1}{2}(\dot{\theta} + \theta)^2,$$

which defines a candidate safe set $\mathcal{C} \subset \mathbb{R}^2$ as in (5). Note that if $(\theta, \dot{\theta}) \in \mathcal{C}$, then $|\theta| \leq \bar{\theta}$ since:

$$h(\mathbf{x}) \geq 0 \implies \bar{\theta}^2 - \theta^2 \geq \frac{1}{2}(\dot{\theta} + \theta)^2 \geq 0 \implies \theta^2 \leq \bar{\theta}^2.$$

Hence, enforcing forward invariance of $\mathcal{C}$ ensures that $|\theta(t)| \leq \bar{\theta}$ for all t. To check if h is a CBF we first compute:

$$\nabla h(\mathbf{x}) = \begin{bmatrix} -2\theta - (\dot{\theta} + \theta) \\ -(\dot{\theta} + \theta) \end{bmatrix},$$

and verify that zero is a regular value of h by investigating the solution set of the linear system:

$$\nabla h(\mathbf{x}) = \mathbf{0} \iff \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The matrix in the above linear system is positive definite, thus the only solution is $(\theta, \dot{\theta}) = \mathbf{0}$. Since the only point where the gradient of h vanishes is at the origin, which does not lie on the boundary of $\mathcal{C}$, zero is a regular value of h. To use Lemma 2 and verify h as a CBF, we must analyze the behavior of $\dot{h}$ when $L_{\mathbf{g}}h(\mathbf{x}) = 0$. To this end, we note that:

$$L_{\mathbf{g}}h(\mathbf{x}) = -\frac{\dot{\theta} + \theta}{ml^2} = 0 \implies \dot{\theta} + \theta = 0.$$

Hence, when $L_{\mathbf{g}}h(\mathbf{x}) = 0$, we also have:

$$L_{\mathbf{f}}h(\mathbf{x}) = \begin{bmatrix} -2\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \frac{g}{l}\sin(\theta) \end{bmatrix} = -2\theta\dot{\theta} = 2\theta^2,$$

and $h(\mathbf{x}) = \bar{\theta}^2 - \theta^2$, implying that:

$$L_{\mathbf{f}}h(\mathbf{x}) + \alpha(h(\mathbf{x})) = 2\theta^2 + \alpha(\bar{\theta}^2 - \theta^2).$$

By taking $\alpha(s) = \alpha_0 s$ as a linear extended class $\mathcal{K}_\infty$ function, we see that:

$$L_{\mathbf{f}}h(\mathbf{x}) + \alpha(h(\mathbf{x})) = (2 - \alpha_0)\theta^2 + \alpha_0\bar{\theta}^2 > 0,$$

for all $\mathbf{x} \in \mathbb{R}^2$ for any $\alpha_0 \in (0, 2]$, implying that (12) holds for all $\mathbf{x} \in \mathbb{R}^2$ and, consequently, that h is a CBF for the inverted pendulum. To accomplish the objective of keeping the pendulum upright, we synthesize a safety filter $\mathbf{k} : \mathbb{R}^2 \to \mathbb{R}$ using the QP in (13) with a nominal policy of $\mathbf{k}_{\mathrm{d}}(\mathbf{x}) = 0$ and $\alpha_0 = 1$ whose closed-form solution is given by (15). The closed-loop vector field of the pendulum under the influence of the safety filter and the corresponding safe set is provided in Fig. 1.

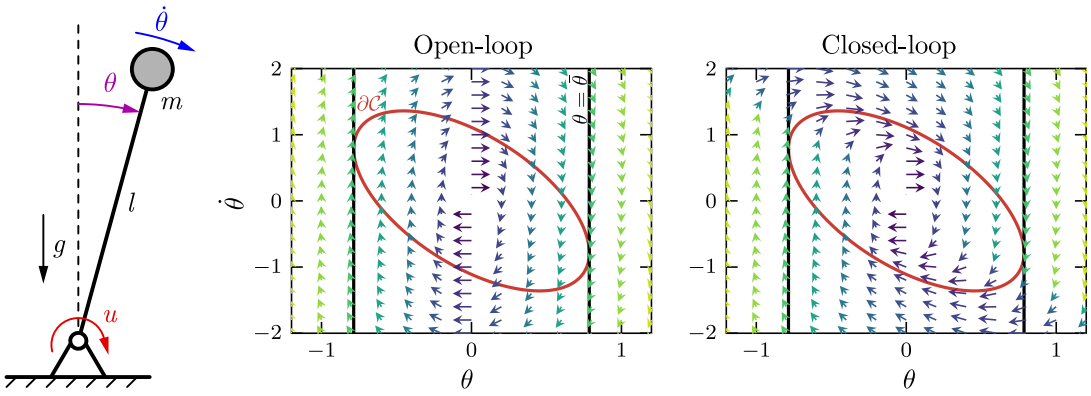

Fig. 1. Vector field of the inverted pendulum in Example 1 without any controller (left) and with the safety filter from (15) (right). In each plot, the red ellipse denotes the boundary of $\mathcal{C}$, the black vertical lines denote $|\theta| = \bar{\theta} = \frac{\pi}{4}$, and the arrows of varying color illustrate the system vector field. The varying colors of the arrows characterize the magnitude of each vector, with lighter colors corresponding to larger magnitudes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The previous example illustrates the procedure required to construct a CBF for relatively simple systems. In Example 1, our CBF was different than the safety constraint $\bar{\theta}^2 - \theta^2 \geq 0$ we wished to satisfy and contained additional terms that depended on both the position and velocity of the pendulum. For relatively simple systems, such as the inverted pendulum, appending such terms to the original safety requirement to obtain a CBF can often be done through intuition or trial-and-error. For more complex high-dimensional systems, however, constructing such a "handcrafted" CBF by carefully blending various states of the system into a single scalar function may be intractable.

Motivated by these challenges, the primary objective of this paper is to outline a comprehensive methodology for systematically constructing CBFs for high-dimensional nonlinear systems based on reduced-order models. Ultimately, this methodology enables one to construct CBFs for complex systems from CBFs for much simpler systems, such as the inverted pendulum outlined above. Before presenting such constructions, we discuss in the following section how the results of the present section can be extended to handle uncertainties.

2.4. Robust safety-critical control

In the previous subsections, we discussed notions of safety for dynamical and control systems, implicitly assuming that the dynamics governing the system are fully known. In reality, however, any system will be affected by unmodeled dynamics and disturbances, which begs the question: how do safety properties degrade in the presence of uncertainties, and how may we design controllers so as to mitigate the effects of such uncertainties? In this subsection, we discuss robust variants of CBFs via the notion of *input-to-state safety* (ISSf) (Alan et al., 2023, 2022; Kolathaya & Ames, 2019), which provides an answer to this question.

Our starting point is the uncertain control affine system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{u} + \mathbf{d}), \tag{17}$$

where $\mathbf{d} \in \mathbb{R}^m$ is a disturbance. As the disturbance enters the dynamics through the same channels as the control input, the disturbance is said to be *matched*, implying that, if the disturbance were known, it could simply be canceled by the control input. Given a locally Lipschitz feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ and a piecewise continuous disturbance signal $t \mapsto \mathbf{d}(t)$, we obtain the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{k}(\mathbf{x}) + \mathbf{d}(t)), \tag{18}$$

which, for each initial condition $\mathbf{x}_0 \in \mathbb{R}^n$, admits a piecewise continuously differentiable solution $\mathbf{x} : I(\mathbf{x}_0, \mathbf{d}(\cdot)) \to \mathbb{R}^n$ defined on some maximal interval of existence $I(\mathbf{x}_0, \mathbf{d}(\cdot)) \subseteq \mathbb{R}_{\geq 0}$.

In what follows, we assume bounded disturbance:

$$\|\mathbf{d}\|_\infty := \sup_{t \geq 0} \|\mathbf{d}(t)\| \leq \delta, \tag{19}$$

with some $\delta \geq 0$. Given this bound on $\mathbf{d}$, we introduce a family of *inflated* safe sets:

$$C_\delta := \{\mathbf{x} \in \mathbb{R}^n \ : \ h_\delta(\mathbf{x}) \geq 0\}, \tag{20}$$

parameterized by δ, where:

$$h_\delta(\mathbf{x}) := h(\mathbf{x}) + \gamma(\delta), \tag{21}$$

for a $\gamma \in \mathcal{K}_\infty$ to be specified shortly. Our notion of safety for (18) is captured via the notion of ISSf.

Definition 6 (*Input-to-State Safety (Kolathaya & Ames, 2019)*)**.** System (18) is said to be *input-to-state safe* (ISSf) on a set $\mathcal{C} \subset \mathbb{R}^n$ as in (5) if there exists a $\gamma \in \mathcal{K}_\infty$ such that for all $\delta \geq 0$ and all $t \mapsto \mathbf{d}(t)$ satisfying (19) the set $\mathcal{C}_\delta \subset \mathbb{R}^n$ as in (20) is forward invariant for (18).

The ISSf property implies a graceful degradation of safety in the presence of uncertainties — potential safety violations are bounded by the magnitude of such uncertainties. Similar to previous subsections, controllers enforcing such a safety property may be constructed using CBFs.

Definition 7 (*Issf Control Barrier Function (Alan et al., 2022)*)**.** A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ defining a set $\mathcal{C} \subset \mathbb{R}^n$ as in (20) is said to be an *input-to-state safe CBF* (ISSf-CBF) for (17) on $\mathcal{C}$ if there exist $\alpha \in \mathcal{K}_\infty^e$ and $\varepsilon \in \mathbb{R}_{>0}$ such that for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \left\{ L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x}) \mathbf{u} \right\} > -\alpha(h(\mathbf{x})) + \frac{1}{\varepsilon} \|L_\mathbf{g} h(\mathbf{x})\|^2. \tag{22}$$

The main difference between CBFs and ISSf-CBFs is the inclusion of $\frac{1}{\varepsilon}\|L_\mathbf{g} h(\mathbf{x})\|^2$ in (22), which imposes a stronger condition on the control input. This term serves to mitigate the impact of uncertainties via the tuning parameter $\varepsilon > 0$ as shown in the following result.

Theorem 5 (*Alan et al., 2022*)**.** *If $h : \mathbb{R}^n \to \mathbb{R}$ is an ISSf-CBF for (17) on a set $\mathcal{C} \subset \mathbb{R}^n$ as in (5), then any locally Lipschitz controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ satisfying:*

$$L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x}) \mathbf{k}(\mathbf{x}) \geq -\alpha(h(\mathbf{x})) + \frac{1}{\varepsilon} \|L_\mathbf{g} h(\mathbf{x})\|^2, \tag{23}$$

renders the closed-loop system (18) ISSf on $\mathcal{C}$ with:

$$\gamma(\delta) = -\alpha^{-1}\left(-\frac{\varepsilon \delta^2}{4}\right). \tag{24}$$

According to (24), the inflated set C_δ can be brought as close as desired to the original safe set C by decreasing ϵ, with $C_\delta \to C$ in the limit as $\epsilon \to 0$. Although, in principle, one can take ϵ as close to zero as desired, doing so generally imposes a stronger condition on the control input, requiring larger control effort, which may not be achievable in practice. Similar to CBFs, the ISSf-CBF condition (23) can be interpreted as an affine constraint that the control input must satisfy, leading to the construction of ISSf enforcing controllers via QPs as in (13). Note that when the uncertainties $\mathbf{d}$ are matched, as in (17), and $L_\mathbf{g} h(\mathbf{x}) = \mathbf{0}$, neither the control input nor uncertainties may impact the system, implying the criterion for constructing CBFs in Lemma 2 also applies to ISSf-CBFs.

2.5. Smooth safety filters

In what follows, many of our results will require smooth (differentiable as many times as necessary) CBF controllers. This may seem restrictive since the vast majority of CBF controllers – including the ones discussed in this tutorial thus far – are computed as the solution to an optimization problem and are inherently nonsmooth. However, when the problem data itself is smooth (i.e., the dynamics $\mathbf{f}, \mathbf{g}$, CBF h, and extended class $\mathcal{K}_\infty$ function α), it is always possible to construct a smooth CBF controller.

Lemma 3 (*Cohen, Ong et al., 2023*)**.** *Consider system (9) with $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ smooth functions and let $h : \mathbb{R}^n \to \mathbb{R}$ be a smooth CBF for (9) on a set $C \subset \mathbb{R}^n$ as in (5) with a smooth $\alpha \in \mathcal{K}_\infty^e$. Then, there exists a smooth feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ such that (11) holds for all $\mathbf{x} \in \mathbb{R}^n$.*

The class of smooth controllers considered in this tutorial inherit the same structure as the closed-form QP controller (15):

$$\mathbf{k}(\mathbf{x}) = \mathbf{k}_\mathrm{d}(\mathbf{x}) + \lambda(a(\mathbf{x}), b(\mathbf{x})) L_\mathbf{g} h(\mathbf{x})^\top, \tag{25}$$

where $\mathbf{k}_\mathrm{d} : \mathbb{R}^n \to \mathbb{R}^m$ is a nominal controller and a and b are as in (14). Any smooth controller of the form (25) satisfying the CBF inequality (11) is said to be a *smooth safety filter*. The fact that the QP controller in (13) is nonsmooth stems from the presence of the ReLU activation function in the Lagrange multiplier λ in (15), which has the interpretation of "activating" the safety filter when the nominal controller fails to guarantee satisfaction of the CBF constraint in (11). This nonsmoothness can be removed by modifying the Lagrange multiplier λ using various "smooth universal formulas" such as (Cohen, Ong et al., 2023):

$$\begin{aligned}
\lambda(a,b) &= \begin{cases} 0 & b = 0 \\ \frac{-a+\sqrt{a^2+\sigma b^2}}{b} & b \neq 0 \end{cases} && \text{(Sontag)} \\
\lambda(a,b) &= \begin{cases} 0 & b = 0 \\ \frac{-a+\sqrt{a^2+\sigma b^2}}{2b} & b \neq 0 \end{cases} && \text{(Half-Sontag)} \\
\lambda(a,b) &= \begin{cases} 0 & b \leq 0 \\ \sigma \log\left(1 + e^{-\frac{a}{\sigma b}}\right) & b > 0 \end{cases} && \text{(Softplus)} \\
\lambda(a,b) &= \begin{cases} 0 & b \leq 0 \\ \sigma \frac{\mathrm{pdf}_{\mathcal{N}(0,1)}\left(\frac{a}{\sigma b}\right)}{\mathrm{cdf}_{\mathcal{N}(0,1)}\left(\frac{a}{\sigma b}\right)} & b > 0 \end{cases} && \text{(Gaussian),}
\end{aligned} \tag{26}$$

where $\sigma > 0$ and $\mathrm{pdf}_{\mathcal{N}(0,1)}(\cdot)$ and $\mathrm{cdf}_{\mathcal{N}(0,1)}(\cdot)$ denote the probability density function and cumulative distribution function of a zero-mean Gaussian distribution with unit variance (Ong & Cortes, 2019). Each of these functions can be shown to be smooth on the set[6]:

$$S = \{(a,b) \in \mathbb{R}^2 \ : \ a > 0 \text{ or } b > 0\},$$

and may be interpreted as a smooth over-approximation of the original Lagrange multiplier from (15) as illustrated in Fig. 2. The safety properties of these smooth universal formulas – including how closely they may approximate the QP-based controller (13) – can be established using the techniques introduced in Cohen, Ong et al. (2023).

Remark 3. In the context of control Lyapunov functions (CLFs), it is often stated that Sontag's formula (Sontag, 1989) is smooth everywhere except possibly the origin, where one can generally only guarantee continuity under the small control property (Sepulchre et al., 1997, Ch. 3.5.3). However, this phenomenon is unique to CLFs and does not arise in the context of CBFs provided one is willing to use a strict inequality in (12). Indeed, as discussed in Remark 1, to guarantee even continuity of CBF or CLF based controllers, one must generally use a strict inequality in the definition of a CBF/CLF, otherwise, the controller may not be continuous when $b = 0$. This follows from the observation that $\lambda(a,0) = 0$ and the limit of $\lambda(a,b)$ as b approaches zero is zero under the condition that $b = 0 \implies a > 0$, where λ is any of the formulas from (15) and (26). In contrast, if one only requires $b = 0 \implies a \geq 0$ this limit may not exist. Now, when using a CLF, the strict inequality does not hold at the origin since CLFs are positive definite, and thus one requires an additional property to guarantee continuity, which comes in the form of the small control property. However, in the context of CBFs, under the presumption that zero is a regular value of h, which implicitly holds when defining a CBF as in (12), the strict inequality holds for all $\mathbf{x} \in \mathbb{R}^n$, which ensures continuity of the QP-based controller at all points and smoothness of the other formulas at all points.

As each of the formulas in (26) is an over-approximation of the Lagrange multiplier from (15), the resulting smooth safety filter in (25) enforces *strict* satisfaction of the CBF inequality (11), which will become important when constructing CBFs from reduced-order models. Our discussion on smooth safety filters is formalized in the following result.

Theorem 6 (*Cohen, Ong et al., 2023*)**.** *Let the conditions of Lemma 3 hold. Then, for each $\lambda : \mathbb{R}^2 \to \mathbb{R}$ in (26), the controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ in (25) is smooth and satisfies:*

$$L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x}) \mathbf{k}(\mathbf{x}) > -\alpha(h(\mathbf{x})), \tag{27}$$

for all $\mathbf{x} \in \mathbb{R}^n$ and therefore renders the set $C \subset \mathbb{R}^n$ from (5) forward invariant for the closed-loop system.

3. Reduced-order models and layered control architectures

In this section, we begin our formal presentation of synthesizing CBFs via reduced-order models (ROMs). First, we motivate our eventual constructions by discussing the challenges associated with synthesizing CBFs for high-dimensional systems. We then introduce various classes of control systems that may be interpreted as *layered* control architectures. These include, for example, robotic systems, where the dynamics of higher layers act as ROMs for the lower layer dynamics, the states of which are, in turn, viewed as control inputs to the aforementioned ROM.

3.1. Challenges in constructing CBFs

Our main focus in this tutorial is on high-dimensional nonlinear control systems whose dynamics may be viewed as a *layered architecture* in which states of lower layers are viewed as control inputs for higher layers. This perspective is motivated by the fact that constructing CBFs for high-dimensional systems may be challenging — such CBFs must generally take into account the behavior of the full-order dynamics to ensure safety. As demonstrated throughout this tutorial, these challenges can often be overcome by exploiting the layered structure

[6] In Cohen, Ong et al. (2023) this set was originally taken as a subset of $\mathbb{R} \times \mathbb{R}_{\geq 0}$ since, in the context of CBFs, $b := \|L_\mathbf{g} h(\mathbf{x})\|^2 \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$, but can be extended to a subset of $\mathbb{R}^2$ to discuss smoothness of (26) independent of their relation to CBFs.

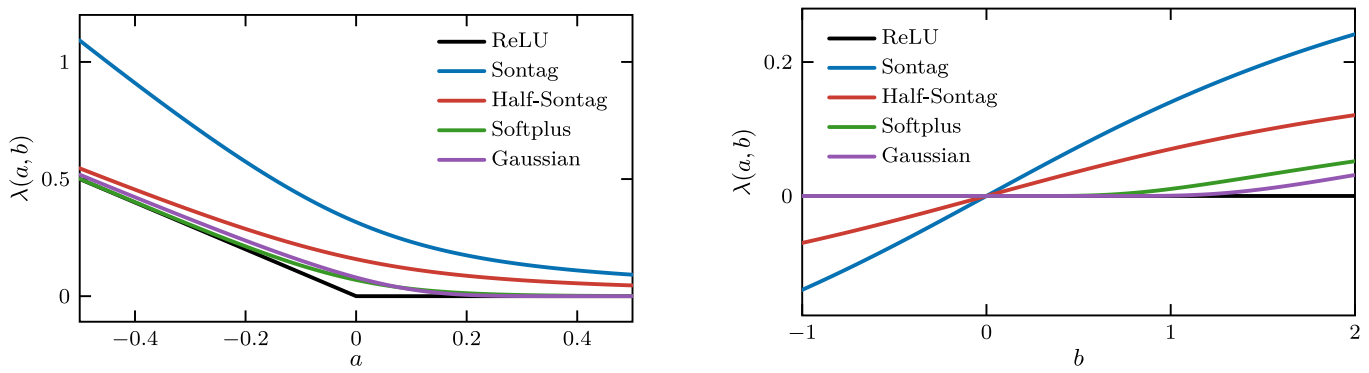

Fig. 2. Smooth universal formulas for safety-critical control compared to the ReLU function associated with quadratic programs. The left plot illustrates the variation of $\lambda(a,b)$ with respect to a for a fixed $b>0$ while the right plot illustrates the variation of $\lambda(a,b)$ with respect to b for a fixed $a>0$ for each of the formulas in (26).

present in many relevant systems to recursively construct a CBF for a complex system from a CBF for a much simpler one.

Many of the challenges associated with constructing CBFs are often related to the *relative degree* of a function $h : \mathbb{R}^n \to \mathbb{R}$ defining a candidate safe set as in (5).

Definition 8 (*Relative degree*)**.** A smooth function $h : \mathbb{R}^n \to \mathbb{R}$ is said to have *relative degree* $r \in \mathbb{N}$ for (9) on a set $\mathcal{D} \subseteq \mathbb{R}^n$ if:

1. $L_{\mathbf{g}} L_{\mathbf{f}}^{r-i} h(\mathbf{x}) = \mathbf{0}$ for all $\mathbf{x} \in \mathcal{D}$ and $i \in \{2, \dots, r\}$;
2. $L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} h(\mathbf{x}) \neq \mathbf{0}$ for some $\mathbf{x} \in \mathcal{D}$,

where higher-order Lie derivatives are defined as:

$$L_{\mathbf{f}}^0 h(\mathbf{x}) := h(\mathbf{x}), \qquad L_{\mathbf{f}}^i h(\mathbf{x}) := \frac{\partial L_{\mathbf{f}}^{i-1} h}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}),$$
$$L_{\mathbf{g}} L_{\mathbf{f}} h(\mathbf{x}) := \frac{\partial L_{\mathbf{f}} h}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}), \qquad L_{\mathbf{g}} L_{\mathbf{f}}^i h(\mathbf{x}) := \frac{\partial L_{\mathbf{f}}^i h}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}).$$

If the second condition holds for all $\mathbf{x} \in \mathcal{D}$, then h is said to have *uniform* relative degree $r \in \mathbb{N}$ for (9) on $\mathcal{D}$.

When h has uniform relative degree 1 for (9) on $\mathbb{R}^n$, i.e., if $L_{\mathbf{g}} h(\mathbf{x}) \neq \mathbf{0}$ for all $\mathbf{x} \in \mathbb{R}^n$, then h is a CBF for (9) (with $\mathbf{u} \in \mathbb{R}^m$) since it is always possible to pick $\mathbf{u} \in \mathbb{R}^m$ as large or small as necessary to satisfy (12). When h has relative degree 1, but not uniform relative degree 1, h is a CBF for (9) provided $L_{\mathbf{f}} h(\mathbf{x}) > -\alpha(h(\mathbf{x}))$ whenever $L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0}$. When h has relative degree larger than 1, then $L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0}$ for all $\mathbf{x} \in \mathbb{R}^n$ and h is unlikely to be a CBF for (9) unless the unforced dynamics of the system are already safe in the sense that $L_{\mathbf{f}} h(\mathbf{x}) > -\alpha(h(\mathbf{x}))$ for all $\mathbf{x} \in \mathbb{R}^n$. Thus, the ability to construct a CBF for a given system is tightly coupled to the construction of a relative degree one function whose zero superlevel set contains the set of states deemed to be safe.

Example 2 (*Double Integrator*)**.** We illustrate many of the ideas presented in this tutorial using the simplest possible example of a higher-dimensional system — the double integrator with state $\mathbf{x} = (\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^N$ and dynamics:

$$\underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \boldsymbol{\xi} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \mathbf{u}. \tag{28}$$

Here, $\mathbf{q} \in \mathbb{R}^n$ represents the position/configuration of the system and $\boldsymbol{\xi} \in \mathbb{R}^p$ captures the velocity. Often, one desires to design a feedback controller for (28) so that the resulting configuration trajectory $t \mapsto \mathbf{q}(t)$ satisfies $\mathbf{q}(t) \in C_0$ for all $t \geq 0$, where $C_0 \subset \mathbb{R}^n$ is a configuration constraint set that may, for example, capture the obstacle-free space in a collision avoidance problem. We assume this set may be characterized as the zero superlevel set of a continuously differentiable function $h_0 : \mathbb{R}^n \to \mathbb{R}$ as:

$$C_0 = \{\mathbf{q} \in \mathbb{R}^n : h_0(\mathbf{q}) \geq 0\}.$$

Given the objective of keeping the configuration in the above set, and the ability of CBFs to render such sets forward invariant, one may be tempted to simply take $h(\mathbf{x}) = h_0(\mathbf{q})$ and $C = C_0 \times \mathbb{R}^p$ as a CBF candidate and corresponding safe set for (28). Yet, this function may not serve as a CBF for (28), in general, since it has a relative degree larger than one:

$$L_{\mathbf{g}} h(\mathbf{x}) = \underbrace{\begin{bmatrix} \nabla h_0(\mathbf{q})^\top & \mathbf{0} \end{bmatrix}}_{\nabla h(\mathbf{x})^\top} \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} = \mathbf{0}.$$

To remedy this, one must choose h to additionally depend on $\boldsymbol{\xi}$, which could be done in a similar fashion to Example 1 so that h has relative degree one and defines a set C such that rendering C forward invariant is sufficient to ensure satisfaction of the original configuration constraint in C_0.

The previous example, although extremely simple, underscores one of the primary challenges,[7] in constructing CBFs: a CBF, in general, must depend on all of the states of the system. For the double integrator in Example 2 it is often possible to construct a relative degree one function containing all of the system states to serve as CBF whose corresponding safe set contains the configuration constraint set of interest, as was done in Example 1 for the inverted pendulum. For more complex systems, however, capturing all of the states necessary to ensure safety in a single scalar function may be intractable. In the remainder of this tutorial, we outline various methodologies to systematically build CBFs for complex systems using ROMs — lower dimensional representations of the original system that capture its high-level dynamics, but that are simple enough to construct CBFs for. Naturally, such methodologies require more structure than is present in the general control affine system (9) considered thus far. As hinted at earlier, these constructions are applicable to systems admitting a layered architecture in which the dynamics of higher layers act as ROMs for the lower-layer dynamics, the states of which are viewed as control inputs to the higher-layer dynamics. In the remainder of this section, we outline relevant classes of dynamics that satisfy such structural assumptions.

3.2. Multi-layer cascaded dynamics

The first layered control architecture we consider is the two-layer cascaded control system:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\boldsymbol{\xi} \\ \dot{\boldsymbol{\xi}} &= \mathbf{f}_1(\mathbf{q}, \boldsymbol{\xi}) + \mathbf{g}_1(\mathbf{q}, \boldsymbol{\xi})\mathbf{u}. \end{aligned} \tag{29}$$

where $\mathbf{q} \in \mathbb{R}^n$ represents the state of the top layer, $\boldsymbol{\xi} \in \mathbb{R}^p$ represents the states of the bottom layer, $\mathbf{u} \in \mathbb{R}^m$ is the control input, and $\mathbf{f}_0$:

[7] The other primary challenge is verifying (12) when $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$.

$\mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_0 : \mathbb{R}^n \to \mathbb{R}^{n\times p}$, $\mathbf{f}_1 : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^p$, $\mathbf{g}_1 : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^{p\times m}$ are locally Lipschitz mappings capturing the dynamics of the multi-layered system. For many physical systems of interest, $\mathbf{q}$ may represent the system's position/configuration and ξ is the system's velocity, implying the top-layer dynamics:

$$\dot{\mathbf{q}} = \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\xi \tag{30}$$

capture the kinematics of the system. Note that by defining $\mathbf{x} := (\mathbf{q}, \xi) \in \mathbb{R}^n \times \mathbb{R}^p = \mathbb{R}^N$, we may write (29) in standard control affine form:

$$\underbrace{\begin{bmatrix}\dot{\mathbf{q}}\\ \dot{\xi}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\xi\\ \mathbf{f}_1(\mathbf{q}, \xi),\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{g}_1(\mathbf{q}, \xi)\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}\mathbf{u}, \tag{31}$$

cf. (28). Here, we view (30) as a ROM, with state $\mathbf{q}$ and control input ξ, for the multi-layered system (29) with the ultimate objective of building a CBF for the corresponding control affine system (31) from a CBF for the ROM (30).

For ease of exposition, most of our discussion will focus on cascaded dynamics with two-layers as in (29); however, the approaches we discuss are also applicable to more general multi-layer systems:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\xi_1\\ \dot{\xi}_1 &= \mathbf{f}_1(\mathbf{q}, \xi_1) + \mathbf{g}_1(\mathbf{q}, \xi_1)\xi_2\\ \dot{\xi}_2 &= \mathbf{f}_2(\mathbf{q}, \xi_1, \xi_2) + \mathbf{g}_2(\mathbf{q}, \xi_1, \xi_2)\xi_3\\ &\vdots\\ \dot{\xi}_r &= \mathbf{f}_r(\mathbf{q}, \xi_1, \xi_2, \dots, \xi_r) + \mathbf{g}_r(\mathbf{q}, \xi_1, \xi_2, \dots, \xi_r)\mathbf{u},\end{aligned} \tag{32}$$

with an arbitrary number of layers $r \in \mathbb{N}$. In traditional control-theoretic literature, such systems are said to be in *strict feedback form* and can also be put into general control affine form (9) with state $\mathbf{x} = (\mathbf{q}, \xi_1, \dots, \xi_r)$ as:

$$\underbrace{\begin{bmatrix}\dot{\mathbf{q}}\\ \dot{\xi}_1\\ \vdots\\ \dot{\xi}_r\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\xi_1\\ \mathbf{f}_1(\mathbf{q}, \xi_1) + \mathbf{g}_1(\mathbf{q}, \xi_1)\xi_2\\ \vdots\\ \mathbf{f}_r(\mathbf{q}, \xi_1, \xi_2, \dots, \xi_r)\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{g}_r(\mathbf{q}, \xi_1, \xi_2, \dots, \xi_r)\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}\mathbf{u}.$$

3.3. Robotic systems

A particularly relevant class of systems whose dynamics exhibit a layered structure is mechanical systems, which can be used to model the majority of robotic systems. To introduce the dynamics of such systems, let $\mathbf{q} \in \mathcal{Q}$ denote the generalized configuration of a mechanical system with n degrees of freedom, where $\mathcal{Q} \subseteq \mathbb{R}^n$ is the configuration manifold. The dynamics of such systems are modeled using the Euler–Lagrange equations:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u}, \tag{33}$$

where $\dot{\mathbf{q}} \in T_{\mathbf{q}}\mathcal{Q}$ is the generalized velocity, $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ represents gravitational and other potential effects, and $\mathbf{B} \in \mathbb{R}^{n\times m}$ is the actuation matrix. By defining $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \subseteq \mathbb{R}^{2n}$, the above dynamics may be cast in control affine form (9) as:

$$\underbrace{\begin{bmatrix}\dot{\mathbf{q}}\\ \ddot{\mathbf{q}}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\dot{\mathbf{q}}\\ -\mathbf{D}(\mathbf{q})^{-1}\Big(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})\Big)\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}\mathbf{0}\\ \mathbf{D}(\mathbf{q})^{-1}\mathbf{B}\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}\mathbf{u}. \tag{34}$$

When $m = n$ and $\mathbf{B}$ is invertible, the robotic system (33) is said to be *fully actuated*, otherwise it is said to be *underactuated*. The dynamics in (33) are also a special case of the two-layer cascaded system in (29), which can be recovered by defining:

$$\begin{aligned}\mathbf{f}_0(\mathbf{q}) &= \mathbf{0}, \quad \mathbf{f}_1(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{D}(\mathbf{q})^{-1}\Big(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})\Big),\\ \mathbf{g}_0(\mathbf{q}) &= \mathbf{I}, \quad \mathbf{g}_1(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{D}(\mathbf{q})^{-1}\mathbf{B},\end{aligned}$$

which implies that the ROM for the full-order robotic system (33) takes the form of a single integrator:

$$\dot{\mathbf{q}} = \xi, \tag{35}$$

where the generalized velocity is viewed as a control input.

Although the structure of (33) dictates that its ROM is a single integrator, one may also employ more general ROMs. In particular, one may consider more general ROMs for (33) of the form:

$$\dot{\mathbf{q}} = \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\xi, \tag{36}$$

with control input $\xi \in \mathbb{R}^p$, where $\mathbf{f}_0 : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g}_0 : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ capture the dynamics of the ROM. For example, (36) may be used to represent unicycle-like dynamics:

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}}_{\dot{\mathbf{q}}} = \underbrace{\begin{bmatrix}\cos(\theta) & 0\\ \sin(\theta) & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{g}_0(\mathbf{q})}\underbrace{\begin{bmatrix}v\\ \omega\end{bmatrix}}_{\xi},$$

where $(x, y) \in \mathbb{R}^2$ denotes planar position, $\theta \in [0, 2\pi)$ heading, $v \in \mathbb{R}$ forward velocity, and $\omega \in \mathbb{R}$ angular velocity. For ease of exposition, our presentation regarding robotic systems will focus on the single integrator ROM, and we will indicate how various results can be modified to account for more general ROMs, such as those described by (36).

4. Safe backstepping

Backstepping is a recursive control design tool that has demonstrated success in constructing control Lyapunov functions (CLFs) (Freeman & Kokotović, 1992; Krstić et al., 1995) for nonlinear systems that possess a layered structure (29). The main idea behind backstepping is to treat the states of lower layers as "virtual" control inputs to the top layer, and then design a virtual controller for the top layer that would accomplish the given control objective. However, as this controller is only "virtual", in the sense that it cannot be directly applied to the top layer, one must "backstep" through the dynamics to reach the actual control input. This backstepping process often requires differentiating through the virtual controllers designed at intermediate layers until the original input is reached. Once this input is reached, the control objective reduces to enforcing convergence of the bottom layer states to the aforementioned virtual controller, which, ultimately, leads to the satisfaction of the original control objective. As this procedure implies the existence of a controller satisfying the control objective for the overall system, this enables the construction of a certificate function, such as a CLF, that certifies the ability of the system to complete the given control objective. Thus, backstepping may be interpreted as a procedure to generate a certificate function for a potentially complex high-dimensional system from a certificate function for a much simpler lower-dimensional system.

In principle, there is nothing preventing one from applying a similar methodology to safety-critical control, rather than stabilization. Yet, backstepping has only recently been explored in the context of CBFs (Taylor, Ong et al., 2022) despite the fact that CBFs, in their modern form, have existed for almost a decade (Ames et al., 2014; Xu et al., 2015). The reason, perhaps, for this delayed adoption of backstepping in the context of CBFs may be due to the emphasis in the CBF literature on optimization-based controllers, which are generally nonsmooth. Other reasons may be the development of viable alternatives, such as extended CBFs (Nguyen & Sreenath, 2016; Xiao & Belta, 2019, 2022), that construct CBF-like functions for high-dimensional systems. In the remainder of this section, we demonstrate how recent results on smooth CBF-based controllers (Cohen, Ong et al., 2023; Ong & Cortes, 2019), such as those outlined in Section 2.5, provide a pathway towards the development of CBF backstepping and illustrate the advantages of such an approach over existing methods that construct CBFs for high-order systems.

4.1. Backstepping with control barrier functions

Now we revisit backstepping in the context of safety-critical control with CBFs (Taylor, Ong et al., 2022). As a first step, we consider the top layer in (30) as a ROM, where $\boldsymbol{\xi}$ – the state of the bottom layer – is viewed as a "virtual" control input to the top layer. We wish to design this input to render:

$$C_0 := \{\mathbf{q} \in \mathbb{R}^n \ : \ h_0(\mathbf{q}) \geq 0\}, \tag{37}$$

for some continuously differentiable $h_0 : \mathbb{R}^n \to \mathbb{R}$, forward invariant for the top layer. To this end, we assume that h_0 is a CBF for this ROM in the sense that:

$$\sup_{\boldsymbol{\xi} \in \mathbb{R}^p} \left\{ L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \boldsymbol{\xi} \right\} > -\alpha(h_0(\mathbf{q})),$$

for all $\mathbf{q} \in \mathbb{R}^n$ for some $\alpha \in \mathcal{K}_\infty^e$. Provided $\mathbf{f}_0$, $\mathbf{g}_0$, h_0, and α are smooth, Theorem 6 then implies the existence of a smooth controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ satisfying:

$$L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \mathbf{k}_0(\mathbf{q}) > -\alpha(h_0(\mathbf{q})), \tag{38}$$

for all $\mathbf{q} \in \mathbb{R}^n$. This controller may be designed, for example, using the formulas in (25) and (26). The interpretation of (38) is that setting $\boldsymbol{\xi} = \mathbf{k}_0(\mathbf{q})$ would ensure the forward invariance of C_0 for the top-level dynamics if we could directly control $\boldsymbol{\xi}$.

As we cannot directly control $\boldsymbol{\xi}$, however, we must backstep through $\mathbf{k}_0$ to determine the inputs $\mathbf{u}$ that drive $\boldsymbol{\xi}$ to $\mathbf{k}_0(\mathbf{q})$. Hence, the problem of constructing a CBF for the full-order system is reduced to that of tracking the output of the ROM. For the full-order dynamics in (29), we leverage $\mathbf{k}_0$ to propose the CBF candidate:

$$h(\mathbf{q}, \boldsymbol{\xi}) := h_0(\mathbf{q}) - \frac{1}{2\mu} \|\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q})\|^2, \tag{39}$$

with parameter $\mu \in \mathbb{R}_{>0}$, which is used to define the safe set for the full-order system:

$$C = \{(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^{n+p} \ : \ h(\mathbf{q}, \boldsymbol{\xi}) \geq 0\}. \tag{40}$$

Importantly, note that $(\mathbf{q}, \boldsymbol{\xi}) \in C$ implies $\mathbf{q} \in C_0$ since $h_0(\mathbf{q}) \geq h(\mathbf{q}, \boldsymbol{\xi})$ for all $(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^{n+p}$. Therefore, rendering C forward invariant for the full-order dynamics ensures that $\mathbf{q}(t) \in C_0$ for all $t \in I(\mathbf{q}_0, \boldsymbol{\xi}_0)$.

To determine if the candidate CBF in (39) is indeed a CBF for the full-order dynamics in (29) with state $\mathbf{x} = (\mathbf{q}, \boldsymbol{\xi})$, we recall from Lemma 2 that one need only to consider the system behavior when $L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0}$. To this end, we compute:

$$\nabla h(\mathbf{x}) = \begin{bmatrix} \nabla h_0(\mathbf{q}) + \frac{1}{\mu} \frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})^\top (\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q})) \\ -\frac{1}{\mu}(\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q})), \end{bmatrix}$$

and

$$L_{\mathbf{g}} h(\mathbf{x}) = -\frac{1}{\mu}(\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q}))^\top \mathbf{g}_1(\mathbf{q}, \boldsymbol{\xi}),$$

noting that, if $\mathbf{g}_1(\mathbf{q}, \boldsymbol{\xi})$ is pseudo-invertible for all $(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^{n+p}$, then:

$$L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0} \implies \boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q}) = \mathbf{0} \implies h(\mathbf{x}) = h_0(\mathbf{q}).$$

Thus, when $L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0}$, we have:

$$\begin{aligned} L_{\mathbf{f}} h(\mathbf{x}) &= L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \boldsymbol{\xi} \\ &= L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \mathbf{k}_0(\mathbf{q}) \\ &> -\alpha(h_0(\mathbf{q})) \\ &= -\alpha(h(\mathbf{x})), \end{aligned}$$

which implies that h is a CBF for the full-order dynamics by Lemma 2. This is formalized via the following theorem, which captures the main result with regard to CBF backstepping.

Theorem 7 (*Taylor, Ong et al., 2022*). *Consider the two-layer dynamics in (29), the constraint set $C_0 \subset \mathbb{R}^n$ in (37), and suppose there exists a continuously differentiable controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ and $\alpha \in \mathcal{K}_\infty^e$ satisfying (38). If $\mathbf{g}_1(\mathbf{q}, \boldsymbol{\xi})$ is pseudo-invertible for all $(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^{n+p}$, then $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ as defined in (39) is a CBF for the corresponding control affine system (31) on the set $C \subset \mathbb{R}^n \times \mathbb{R}^p$ as in (40).*

The preceding theorem facilitates the construction of CBFs for high-dimensional nonlinear systems that exhibit a layered structure as in (29). Although these constructions have been presented for the special case of a two-layered system, similar to Lyapunov backstepping (Krstić et al., 1995), this approach may be recursively used to construct a CBF for a system with an arbitrary number $r \in \mathbb{N}$ of layers (Taylor, Ong et al., 2022) as defined in (32). The following examples illustrate the steps needed to construct a CBF using backstepping on the double integrator from Example 2.

Example 3 (*Double integrator*)**.** Consider a one-dimensional double integrator with dynamics of the form (29), where $\mathbf{q} = x \in \mathbb{R}$ represents the position and $\boldsymbol{\xi} = v \in \mathbb{R}$ represents velocity, while $\mathbf{x} = (x, v)$ is the full-order state. Let the objective of designing a feedback controller be to keep the system's position x in the interval $[-1, 1] \subset \mathbb{R}$. This objective can be formalized by requiring the system's position to remain in the set:

$$C_0 = \{x \in \mathbb{R} \ : \ h_0(x) = 1 - x^2 \geq 0\}.$$

Recall from Example 2, however, that this function may not serve as a CBF for the full-order system (31) since, with $h(\mathbf{x}) = h_0(x)$, we have $L_{\mathbf{g}} h(\mathbf{x}) = 0$.

To remedy this, we take a backstepping-based approach, where we view the top-layer dynamics:

$$\dot{x} = \underbrace{0}_{f_0(x)} + \underbrace{1}_{g_0(x)} \times v$$

as a ROM with control input v. To check if h_0 is a CBF for the ROM, we compute:

$$L_{g_0} h_0(x) = -2x,$$

so that when $L_{g_0} h_0(x) = 0$, we have $x = 0$ and:

$$L_{f_0} h_0(x) + \alpha(h_0(x)) = \alpha(1 - x^2) = \alpha(1) > 0.$$

Hence, by Lemma 2, h_0 is a CBF for the ROM for any $\alpha \in \mathcal{K}_\infty^e$, which for simplicity, we take as $\alpha(s) = s$. As h_0 is a CBF for the ROM, then, by Theorem 6, there exists a smooth controller $k_0 : \mathbb{R} \to \mathbb{R}$ satisfying (38). Furthermore, since $g_1(x, v) = 1$ is invertible, the function:

$$h(\mathbf{x}) = h(x, v) = h_0(x) - \frac{1}{2\mu}\left(v - k_0(x)\right)^2,$$

is a CBF for the full-order dynamics on the set:

$$C = \{(x, v) \in \mathbb{R}^2 \ : \ h(x, v) \geq 0\}, \tag{41}$$

by Theorem 7.

This safe set is illustrated for different values of μ in Fig. 3, where the smooth controller k_0 is defined as in (25) with λ chosen as the Softplus universal formula (26) with $\sigma = 0.1$ and $k_d(x) = 0$. Note that as μ is increased, the safe set C approaches the original constraint set C_0 at the cost of including larger velocities, which may require compensation with larger control efforts.

Example 4 (*Obstacle Avoidance (Taylor, Ong et al., 2022)*)**.** We now continue Example 2 but present the details of constructing a CBF for an obstacle avoidance problem, which is used as an opportunity to illustrate the effect of the smooth safety filter on the corresponding CBF. This example was previously presented in the context of safe backstepping in Taylor, Ong et al. (2022). As demonstrated in Example 2, any function that depends only on position is not a CBF for the double integrator. Yet, by viewing a single integrator $\dot{\mathbf{q}} = \boldsymbol{\xi}$ as a reduced-order representation of the full-order double integrator dynamics, we may

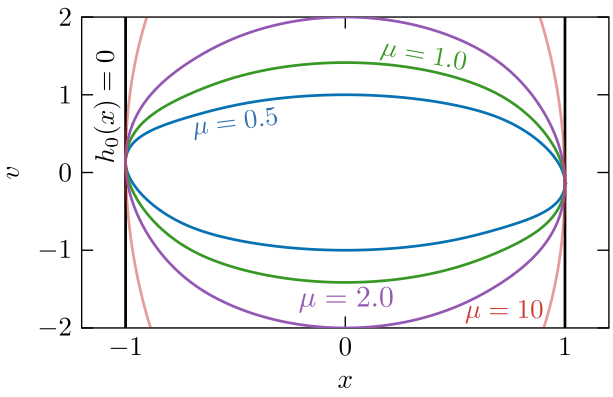

Fig. 3. Safe set constructed for the one-dimensional double integrator via backstepping. Here, the colored curves represent the zero level set of h as defined in (39) for various μ, where k_0 is constructed using the Softplus universal formula from (26) with $\sigma = 0.1$. Note that as μ is increased the resulting safe set approaches the original constraint set C_0 from (37). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

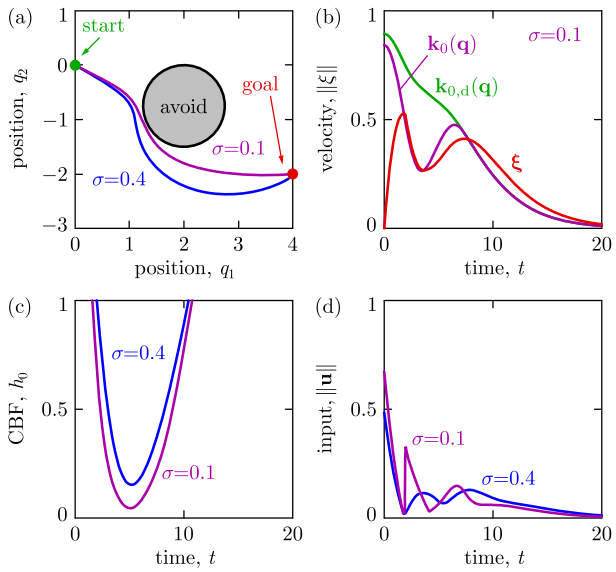

Fig. 4. Results of the double integrator obstacle avoidance scenario from Example 4. (a) The trajectories of the double integrator's position, (b) its velocities, (c) the values of the safety constraint h_0 along the system's trajectory, and (d) the norm of the control input over time.
Source: This figure has been adapted from Taylor, Ong et al. (2022)

still design a controller that uses a CBF constructed from the function characterizing the distance to the obstacle:

$$h_0(\mathbf{q}) = \frac{1}{2}\left(\|\mathbf{q} - \mathbf{q}_\mathrm{o}\|^2 - R_\mathrm{o}^2\right),$$

where $\mathbf{q}_\mathrm{o} \in \mathbb{R}^2$ is the obstacle's center and $R_\mathrm{o} \in \mathbb{R}_{>0}$ is its radius, which is a valid CBF for the single integrator.

To construct a CBF for the double integrator from its reduced-order single integrator model, we leverage the safe backstepping approach outlined in this section. First, we construct a smooth safety filter $\mathbf{k}_0 : \mathbb{R}^2 \to \mathbb{R}^2$ for the single integrator via (25), where λ is chosen as the Gaussian smooth universal formula (26) and $\alpha(s) = s$, which filters out unsafe controls from the desired reduced-order controller $\mathbf{k}_{0,\mathrm{d}}(\mathbf{q}) := K_\mathrm{p}(\mathbf{q}_\mathrm{g} - \mathbf{q})$, where $\mathbf{q}_\mathrm{g} \in \mathbb{R}^2$ is a goal location and $K_\mathrm{p} \in \mathbb{R}_{>0}$ is a gain. This smooth safety filter is then used to construct a CBF for the double integrator using (39) with $\mu = 1$. Finally, the CBF is used to synthesize a QP-based safety filter $\mathbf{k} : \mathbb{R}^4 \to \mathbb{R}^2$ for the full-order system using (13).

The results of this procedure are displayed in Fig. 4 that is repeated from (Taylor, Ong et al., 2022). Simulations are shown for various choices of σ in the smooth universal formula (26). Note that as σ approaches zero, the behavior of the smooth safety filter approaches that of a QP controller, where λ depends on the ReLU activation function, leading to less smooth control signals.

4.2. Comparison to extended control barrier functions

Control barrier backstepping may be interpreted as a systematic methodology to construct a CBF for a high-dimensional system from a high relative degree safety constraint $h_0(\mathbf{q}) \geq 0$ that depends only on the states of the top layer, the end result of which is a *relative degree one* CBF $h(\mathbf{q}, \xi_1, \dots, \xi_r)$ for a higher dimensional control system. The construction of this CBF requires only a CBF for the top layer of (32) and a few controllability assumptions, namely that each $\mathbf{g}_i$ for $i \in \{1, \dots, r\}$ is pseudo-invertible.

This approach is similar in spirit to other high-order CBF techniques that build a relative degree one CBF-like function from a high relative degree safety constraint $h_0(\mathbf{q}) \geq 0$ defining a set $C_0 \subset \mathbb{R}^n$ as in (37), but have important technical differences as we discuss next. Such approaches are typically predicated on constructing an *extended* CBF (also referred to as an *exponential* (Nguyen & Sreenath, 2016) or *high order* (Xiao & Belta, 2019, 2022) CBF) by computing the derivative of h_0 along the system vector fields until the control input appears, reminiscent of classical input–output linearization. For example, when considering the two-layer cascaded system (29), h_0 has relative degree two, thus one computes:

$$h(\mathbf{x}) = L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q})\xi + \alpha_0 h_0(\mathbf{q}), \tag{42}$$

where $\alpha_0 \in \mathbb{R}_{>0}$ and $\mathbf{x} = (\mathbf{q}, \xi)$, as an extended CBF candidate, which now has relative degree one and defines a set $C \subset \mathbb{R}^n \times \mathbb{R}^p$ as its zero superlevel set.

Note, however, that unlike the backstepping-based approach, $\hat{C}_0 = C_0 \times \mathbb{R}^p$ is not a subset of C and one must instead consider the intersection $\hat{C}_0 \cap C \subset \mathbb{R}^n \times \mathbb{R}^p$ as the candidate safe set of interest. To guarantee safety, this extended CBF must then satisfy:

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \left\{ L_\mathbf{f} h(\mathbf{x}) + L_\mathbf{g} h(\mathbf{x})\mathbf{u} \right\} > -\alpha(h(\mathbf{x})), \tag{43}$$

for all $\mathbf{x} \in \hat{C}_0 \cap C$ for some $\alpha \in \mathcal{K}_\infty^e$, which can be used to develop feedback controllers enforcing forward invariance of $\hat{C}_0 \cap C$. Similar to CBFs, the satisfaction of (43) can also be verified by checking that $L_\mathbf{f} h(\mathbf{x}) > -\alpha(h(\mathbf{x}))$ whenever $L_\mathbf{g} h(\mathbf{x}) = \mathbf{0}$. Unfortunately, as illustrated in the following example (Cohen & Belta, 2023; Tan et al., 2022), an extended CBF satisfying (43) may not exist even for relatively simple safety constraints.

Example 5 (*Cohen & Belta, 2023*)**.** We now consider the same system and safety constraint h_0 and corresponding constraint set C_0 as in Example 3, but attempt to construct a safe set using an extended CBF rather than using backstepping. Since h_0 has relative degree larger than one based on Example 2, we calculate the extended CBF candidate in (42):

$$h(\mathbf{x}) = -2xv + \alpha_0 - \alpha_0 x^2,$$

which defines a set C as its zero superlevel set, and a candidate safe set as $\hat{C}_0 \cap C$ with $\hat{C}_0 = C_0 \times \mathbb{R}$. This candidate safe set for different choices of α_0 is illustrated in Fig. 5. Similar to Example 3, one may force $\hat{C}_0 \cap C$ closer to $\hat{C}_0$ by increasing α_0.

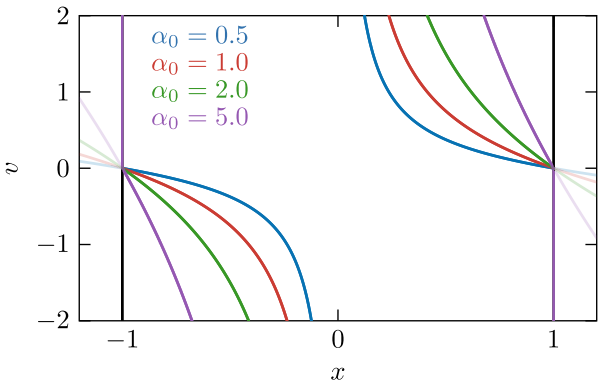

Fig. 5. Safe set constructed for the one-dimensional double integrator using the extended CBF approach. Here, the colored curves represent the boundary of $\hat{C}_0 \cap C$ for different choices of α_0, the black lines denote the boundary of $\hat{C}_0$, and the transparent curves of corresponding color denote the boundary of C for different choices of α_0. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To check if h satisfies the criteria in (43) for all $\mathbf{x} \in \hat{C}_0 \cap C$, we must ensure that $L_\mathbf{f} h(\mathbf{x}) + \alpha(h(\mathbf{x})) > 0$ whenever $L_\mathbf{g} h(\mathbf{x}) = 0$. To this end, we compute:

$$L_\mathbf{g} h(\mathbf{x}) = \underbrace{\begin{bmatrix} -2v - 2\alpha_0 x & -2x \end{bmatrix}}_{\nabla h(\mathbf{x})^\top} \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} = -2x,$$

noting that $L_\mathbf{g} h(\mathbf{x}) = 0$ implies $x = 0$. Hence, when $L_\mathbf{g} h(\mathbf{x}) = 0$, we also have:

$$L_\mathbf{f} h(\mathbf{x}) + \alpha(h(\mathbf{x})) = -2v^2 + \alpha(\alpha_0),$$

implying (43) only holds at points such that:

$$v^2 < \frac{\alpha(\alpha_0)}{2}.$$

That is, when $x = 0$, (43) only holds provided the magnitude of the velocity is bounded above by a function of α_0 and α. In practice, one may tune α_0 and α so that (43) is only violated for arbitrarily large velocities, yet, such points will still be contained in $\hat{C}_0 \cap C$ (see Fig. 5), implying (43) does not hold for all $\mathbf{x} \in \hat{C}_0 \cap C$ and, consequently, that h is not an extended CBF.

The previous example demonstrates that one must take care when using the extended CBF methodology, as seemingly benign safety constraints may generate a function that cannot serve as an extended CBF no matter the choice of extended class $\mathcal{K}_\infty$ functions. The consequence of this is that controllers synthesized from such invalid extended CBFs may not be well-defined even in the case when the control input is unconstrained. In contrast, the backstepping methodology outlined above produces, by construction, a relative degree one function that is guaranteed to be a CBF for the full-order system. The price to pay for this correct-by-construction approach is that it requires the full-order dynamics to exhibit a particular cascaded structure. In the following subsection, we extend this approach to a more general class of cascaded systems.

4.3. Mixed relative degree backstepping

Another advantage of CBF backstepping over existing high order CBF approaches is the ability to handle layered systems with a *mixed relative degree* — that is, systems where inputs may enter not only at the bottom layer as in (32), but also at intermediate layers. Such mixed relative degree systems with two layers take the form:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0^\xi(\mathbf{q})\xi + \mathbf{g}_0^\mathbf{u}(\mathbf{q})\mathbf{u}_0 \\ \dot{\xi} &= \mathbf{f}_1(\mathbf{q}, \xi) + \mathbf{g}_1^\mathbf{u}(\mathbf{q}, \xi)\mathbf{u}_1, \end{aligned} \tag{44}$$

where $\mathbf{x} = (\mathbf{q}, \xi) \in \mathbb{R}^n \times \mathbb{R}^p$ is the system state, $\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1) \in \mathbb{R}^{m_0} \times \mathbb{R}^{m_1}$ is the control input, and $\mathbf{f}_0 : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{g}_0^\xi : \mathbb{R}^n \to \mathbb{R}^{n \times p}$, $\mathbf{g}_0^\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^{n \times m_0}$, $\mathbf{f}_1 : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^p$, $\mathbf{g}_1^\mathbf{u} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^{p \times m_1}$ characterize the dynamics. Similar to the previous layered architecture, this system admits a control affine representation (9) as:

$$\underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\xi} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0^\xi(\mathbf{q})\xi \\ \mathbf{f}_1(\mathbf{q}, \xi) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{g}_0^\mathbf{u}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_1^\mathbf{u}(\mathbf{q}, \xi) \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \underbrace{\begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \end{bmatrix}}_{\mathbf{u}}. \tag{45}$$

For this system, we consider a function $h_0 : \mathbb{R}^n \to \mathbb{R}$ on the top layer states defining a constraint set $C_0 \subset \mathbb{R}^n$ as in (37). The mixed relative degree characterization of (44) follows from the fact that the safety constraint h_0 may have different relative degrees with respect to different components of the control vector $\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1)$. To address this challenge, we suppose the existence of smooth feedback controllers $\mathbf{k}_0^\xi : \mathbb{R}^n \to \mathbb{R}^p$, $\mathbf{k}_0^\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$ and $\alpha \in \mathcal{K}_\infty^e$ satisfying:

$$L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0^\xi} h_0(\mathbf{q}) \mathbf{k}_0^\xi(\mathbf{q}) + L_{\mathbf{g}_0^\mathbf{u}} h_0(\mathbf{q}) \mathbf{k}_0^\mathbf{u}(\mathbf{q}) > -\alpha(h_0(\mathbf{q})), \tag{46}$$

for all $\mathbf{q} \in \mathbb{R}^n$. With the above condition, we propose the CBF candidate (Taylor, Ong et al., 2022):

$$h(\mathbf{q}, \xi) = h_0(\mathbf{q}) - \frac{1}{2\mu} \|\xi - \mathbf{k}_0^\xi(\mathbf{q})\|^2, \tag{47}$$

which is used to define a candidate safe set C as in (40). Once again, note that $(\mathbf{q}, \xi) \in C$ implies $\mathbf{q} \in C_0$ since $h_0(\mathbf{q}) \geq h(\mathbf{q}, \xi)$ for all $(\mathbf{q}, \xi) \in \mathbb{R}^n \times \mathbb{R}^p$. With these conditions, we may state the following result formalizing the construction of CBFs for mixed relative degree systems.

Theorem 8 (*Taylor, Ong et al., 2022*)**.** *Consider the dynamics in (44), the set $C_0 \subset \mathbb{R}^n$ in (37), and suppose there exist smooth feedback controllers $\mathbf{k}_0^\xi : \mathbb{R}^n \to \mathbb{R}^p$, $\mathbf{k}_0^\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$ and $\alpha \in \mathcal{K}_\infty^e$ satisfying (46). If $g_1^\mathbf{u}(\mathbf{q}, \xi)$ is pseudo-invertible for all $(\mathbf{q}, \xi) \in \mathbb{R}^{n+p}$, then $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ as defined in (47) is a CBF for the corresponding control affine system (45) on the set $C \subset \mathbb{R}^n \times \mathbb{R}^p$ as in (40).*

The proof of this theorem largely follows the same procedure as that of Theorem 7 and is provided in the Appendix for completeness. Similar to (32), one may recursively apply Theorem 8 to construct CBFs for mixed relative degree systems with an arbitrary number of layers:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0^\xi(\mathbf{q})\xi_1 + \mathbf{g}_0^\mathbf{u}(\mathbf{q})\mathbf{u}_0 \\ \dot{\xi}_1 &= \mathbf{f}_1(\mathbf{q}, \xi_1) + \mathbf{g}_1^\xi(\mathbf{q}, \xi_1)\xi_2 + \mathbf{g}_1^\mathbf{u}(\mathbf{q}, \xi_1)\mathbf{u}_1 \\ &\vdots \\ \dot{\xi}_r &= \mathbf{f}_r(\mathbf{q}, \xi_1, \ldots, \xi_r) + \mathbf{g}_r^\mathbf{u}(\mathbf{q}, \xi_1, \ldots, \xi_r)\mathbf{u}_r. \end{aligned} \tag{48}$$

Example 6 (*Unicycle (Taylor, Ong et al., 2022)*)**.** A classic example of a mixed-relative degree system is the unicycle:

$$\begin{aligned} \dot{x} &= v\cos(\psi) \\ \dot{y} &= v\sin(\psi) \\ \dot{\psi} &= \omega, \end{aligned}$$

where $(x, y) \in \mathbb{R}^2$ denote planar position, $\psi \in \mathbb{R}$ the heading angle, $v \in \mathbb{R}$ the linear velocity, and $\omega \in \mathbb{R}$ the angular velocity. Here, the state is $\mathbf{x} := (x, y, \psi)$ while the control input is $\mathbf{u} := (v, \omega) = (u_0, u_1)$. As written, the above dynamics are not in the form of (44), but can be transformed into such a system with a few modifications. First, we define:

$$\mathbf{q} := \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \xi := \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix},$$

which implies that:

$$\begin{aligned} \dot{\mathbf{q}} &= \xi u_0 := \mathbf{v}, \\ \dot{\xi} &= \begin{bmatrix} -\xi_2 \\ \xi_1 \end{bmatrix} u_1. \end{aligned}$$

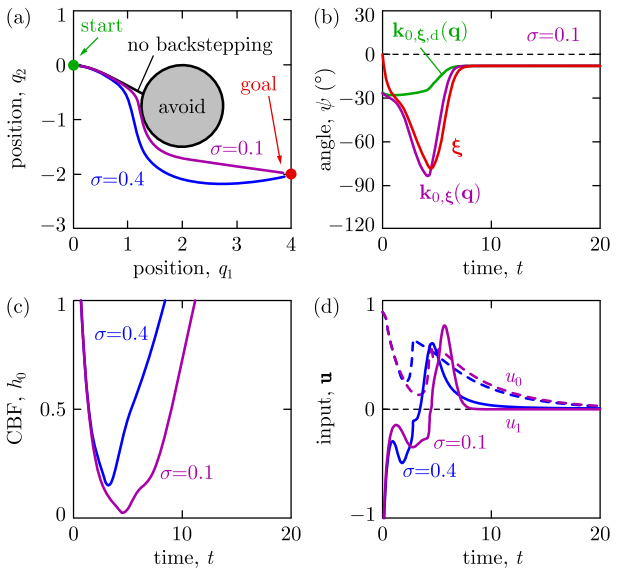

Fig. 6. Simulation results for the unicycle from Example 6. Each plot has a similar interpretation to those in Fig. 4. This figure has been adapted from Taylor, Ong et al. (2022).

where $\mathbf{v}$ denotes the planar velocity vector. Note that, as opposed to (44), the first equation is not affine w.r.t. (ξ, u_0) but is affine in $\mathbf{v}$. Thus we conduct backstepping by viewing the single integrator with input $\mathbf{v}$ as a reduced-order model for the unicycle, and by converting $\mathbf{v}$ to (ξ, u_0).

Our control objective for this system is the same as that in Example 4: we wish to design a controller that enforces convergence of the position to a goal location while avoiding an obstacle. This obstacle avoidance task can be captured using the same safety constraint h_0 as in Example 4. We then synthesize a smooth safety filter $\mathbf{k}_0 : \mathbb{R}^2 \to \mathbb{R}^2$ for the single integrator using the same approach as in Example 4, which outputs safe velocity commands $\mathbf{v} = \mathbf{k}_0(\mathbf{q})$. To use such commands in backstepping, we decompose $\mathbf{v} = \mathbf{k}_0(\mathbf{q})$ into $\xi = \mathbf{k}_0^{\xi}(\mathbf{q})$ and $u_0 = k_0^{\mathbf{u}}(\mathbf{q})$ as:

$$\mathbf{k}_0(\mathbf{q}) = \underbrace{\frac{\mathbf{k}_0(\mathbf{q})}{\|\mathbf{k}_0(\mathbf{q})\|}}_{\mathbf{k}_0^{\xi}(\mathbf{q})} \underbrace{\|\mathbf{k}_0(\mathbf{q})\|}_{k_0^{\mathbf{u}}(\mathbf{q})},$$

which is valid so long as $\mathbf{k}_0(\mathbf{q}) \neq \mathbf{0}$. Then the desired value $\mathbf{k}_0^{\xi}(\mathbf{q})$ of ξ is used to construct a CBF for the full-order system as in (47). This CBF is subsequently used to synthesize a safety filter $\mathbf{k} : \mathbb{R}^3 \to \mathbb{R}^2$ for the unicycle equipped with the desired controller:

$$\mathbf{k}_{\mathrm{d}}(\mathbf{x}) = \begin{bmatrix} K_{\mathrm{p}}\|\mathbf{q} - \mathbf{q}_{\mathrm{g}}\| \\ -K_{\psi}\big(\sin(\psi) - \sin(\psi_0(\mathbf{q}))\big) \end{bmatrix},$$

where $K_{\mathrm{p}}, K_{\psi} \in \mathbb{R}_{>0}$ are gains and $\psi_0 : \mathbb{R}^2 \to \mathbb{R}$, defined by $\mathbf{k}_0^{\xi}(\mathbf{q}) = \begin{bmatrix}\cos(\psi_0(\mathbf{q})) & \sin(\psi_0(\mathbf{q}))\end{bmatrix}^{\top}$, computes the desired heading angle. The results of applying such a controller $\mathbf{u} = \mathbf{k}(\mathbf{x})$ to the unicycle are provided in Fig. 6, where all extended class $\mathcal{K}_\infty$ functions involved are chosen as the identity function.

5. Constructive safety for robotic systems

We now turn our attention to a special case of the cascaded control systems considered in the previous section — robotic systems with dynamics in (33). These dynamics comply with the structure outlined in Section 4, implying the developed backstepping results may be applied to (33) by converting such systems into the form of (29) as detailed in Section 3. However, given the relevance of CBFs in the context of robotics, and the fact that (33) possess certain structural properties that further facilitate the construction of CBFs, we outline in this section how the previous developments may be specialized to robotic systems.

As in the previous section, we wish to design a feedback controller for the full-order system that keeps the system inside a subset of the configuration space:

$$C_0 := \{\mathbf{q} \in \mathcal{Q} \ : \ h_0(\mathbf{q}) \geq 0\}, \tag{49}$$

where $h_0 : \mathcal{Q} \to \mathbb{R}$ is a continuously differentiable configuration constraint. Although we wish to keep the configuration in C_0, such an objective may not be possible without taking into account the full-order dynamics (33). That is, similar to Example 2, C_0 is unlikely to be a controlled invariant set for (33) since for $h(\mathbf{x}) = h_0(\mathbf{q})$ we would have:

$$L_{\mathbf{g}}h(\mathbf{x}) = \begin{bmatrix}\nabla h_0(\mathbf{q})^{\top} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{0} \\ \mathbf{D}(\mathbf{q})^{-1}\mathbf{B}\end{bmatrix} = \mathbf{0},$$

for all $\mathbf{x} \in T\mathcal{Q}$. In what follows, we outline various approaches to construct CBFs for the full-order dynamics (33) from the configuration constraint (49) under different assumptions regarding the system's actuation capability.

5.1. Safe backstepping for robotic systems

To remedy that h_0 is not a CBF, we first follow the backstepping-based approach outlined in the previous section, where we suppose the existence of a continuously differentiable controller $\mathbf{k}_0 : \mathcal{Q} \to \mathbb{R}^n$ satisfying:

$$\nabla h_0(\mathbf{q}) \cdot \mathbf{k}_0(\mathbf{q}) > -\alpha(h_0(\mathbf{q})), \tag{50}$$

for all $\mathbf{q} \in \mathcal{Q}$. Similar to Section 4, we think of (35) as a reduced-order model for the full-order system (33) with input $\xi \in \mathbb{R}^n$ and $\mathbf{k}_0$ representing a controller we would apply to the reduced-order dynamics if we could simply set $\dot{\mathbf{q}} = \mathbf{k}_0(\mathbf{q})$. Thus, $\mathbf{k}_0$ may be interpreted as a *desired velocity* that we wish the full-order system to track. This controller is used to construct the *energy-based CBF candidate*:

$$h(\mathbf{q}, \dot{\mathbf{q}}) = h_0(\mathbf{q}) - \frac{1}{\mu}V(\mathbf{q}, \dot{\mathbf{q}}), \tag{51}$$

where $\mu \in \mathbb{R}_{>0}$ and:

$$V(\mathbf{q}, \dot{\mathbf{q}}) := \frac{1}{2}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^{\top}\mathbf{D}(\mathbf{q})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})), \tag{52}$$

whose form is inspired by that of the system's kinetic energy. This energy-based CBF candidate defines:

$$C := \{(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \ : \ h(\mathbf{q}, \dot{\mathbf{q}}) \geq 0\}, \tag{53}$$

as a candidate safe set, which ensures that $\mathbf{q} \in C_0$ whenever $(\mathbf{q}, \dot{\mathbf{q}}) \in C$ since $h_0(\mathbf{q}) \geq h(\mathbf{q}, \dot{\mathbf{q}})$ for all $(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q}$. Verifying this CBF candidate requires checking the behavior of $\dot{h}$ when $L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0}$, where $\mathbf{g} : T\mathcal{Q} \to \mathbb{R}^{n\times m}$ is defined as in (34) and $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$. To this end, we compute:

$$\frac{\partial h}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}) = -\frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^{\top}\mathbf{D}(\mathbf{q}),$$

noting that:

$$L_{\mathbf{g}}h(\mathbf{x}) = \underbrace{\begin{bmatrix}\frac{\partial h}{\partial \mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) & \frac{\partial h}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}})\end{bmatrix}}_{\nabla h(\mathbf{x})^{\top}}\underbrace{\begin{bmatrix}\mathbf{0} \\ \mathbf{D}(\mathbf{q})^{-1}\mathbf{B}\end{bmatrix}}_{\mathbf{g}(\mathbf{x})}$$
$$= -\frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^{\top}\mathbf{B}.$$

Thus, when (33) is fully actuated, we have:

$$L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0} \implies \dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}) = \mathbf{0} \implies h(\mathbf{q}, \dot{\mathbf{q}}) = h_0(\mathbf{q}),$$

so that, when $L_{\mathbf{g}}h(\mathbf{x}) = \mathbf{0}$, we have:

$$L_{\mathbf{f}}h(\mathbf{x}) = \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} = \nabla h_0(\mathbf{q}) \cdot \mathbf{k}_0(\mathbf{q}) > -\alpha(h_0(\mathbf{q}))$$
$$= -\alpha(h(\mathbf{q}, \dot{\mathbf{q}})),$$

which implies that h is a CBF for the corresponding control affine dynamics (34) by Lemma 2. The preceding discussion is formalized in the following lemma.

Lemma 4. *Consider system (33), a configuration constraint $h_0 : \mathcal{Q} \to \mathbb{R}$ defining a set $C_0 \subset \mathcal{Q}$ as in (49), and suppose there exists a continuously differentiable function $\mathbf{k}_0 : \mathcal{Q} \to \mathbb{R}$ satisfying (50). If (33) is fully actuated, then $h : T\mathcal{Q} \to \mathbb{R}$ as in (51) is a CBF for the corresponding control affine system (34) on $C \subset T\mathcal{Q}$ as in (53).*

Remark 4. The preceding result can also be applied to reduced-order models other than the single integrator in (35), such as the general control affine ROM in (36). To construct a CBF for (33) from this reduced-order model, however, one must modify (50) to:

$$\nabla h_0(\mathbf{q}) \cdot \underbrace{(\mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\mathbf{k}_0(\mathbf{q}))}_{:=\mathbf{f}_{0,\mathrm{cl}}(\mathbf{q})} > -\alpha(h_0(\mathbf{q})),$$

and (52) to:

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}(\dot{\mathbf{q}} - \mathbf{f}_{0,\mathrm{cl}}(\mathbf{q}))^\top \mathbf{D}(\mathbf{q})(\dot{\mathbf{q}} - \mathbf{f}_{0,\mathrm{cl}}(\mathbf{q})).$$

Example 7 (*Double pendulum*)**.** To illustrate the systematic construction of CBFs for robotic systems, we apply the results of this subsection to a fully actuated double pendulum with configuration $\mathbf{q} = (\theta_1, \theta_2)$ denoting the angular position of the first θ_1 and second θ_2 link. Our objective is to design a feedback controller that keeps the x-component of Cartesian position (x, y) of the pendulum's tip within a certain range $|x| \leq \bar{x}$. To this end, we first define $\mathbf{p} : \mathcal{Q} \to \mathbb{R}^2$ associating to each configuration $\mathbf{q} \in \mathcal{Q}$ the Cartesian position of the pendulum's tip as:

$$\mathbf{p}(\mathbf{q}) = l_1 \begin{bmatrix} \sin(\theta_1) \\ -\cos(\theta_1) \end{bmatrix} + l_2 \begin{bmatrix} \sin(\theta_1 + \theta_2) \\ -\cos(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}.$$

Denoting by $p_x(\mathbf{q}) = x$, we propose:

$$h_0(\mathbf{q}) = \bar{x}^2 - p_x(\mathbf{q})^2,$$

as a configuration constraint defining the configuration constraint set $C_0 \subset \mathcal{Q}$ as in (49), which we use as a CBF to define a smooth safety filter $\mathbf{k}_0 : \mathcal{Q} \to \mathbb{R}^2$ as in (25) for the single integrator reduced-order model (35) using the Softplus universal formula (26) with $\sigma = 0.1$ and $\alpha(s) = s$. This system is fully actuated, hence:

$$h(\mathbf{q}, \dot{\mathbf{q}}) = h_0(\mathbf{q}) - \frac{1}{2\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \mathbf{D}(\mathbf{q})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})),$$

is a CBF for the full-order dynamics (34) by Lemma 4. This CBF is then used to construct a QP-based safety filter (13) for the corresponding control affine system (34) and nominal controller $\mathbf{k}_\mathrm{d}(\mathbf{q}, \dot{\mathbf{q}}) = -\dot{\mathbf{q}}$ that adds damping to the system. To demonstrate the effectiveness of this CBF, we simulate the system from an upright position with the objective of bringing the pendulum to a downward position while keeping the pendulum within the safe set, the results of which are provided in Fig. 7. Note that the pendulum initially falls towards the boundary of the safe set, stops itself before crossing the boundary, and then allows the tip of the pendulum to slide along the boundary of the safe set until reaching a downward position.

5.2. Energy-based control barrier functions

At this point, one could directly use h from (51) as a CBF for the control affine representation of the robot dynamics (34); however, such an approach presents certain limitations. In particular, such an approach requires computing the vector fields $\mathbf{f}$ and $\mathbf{g}$ in (34), requiring inversions of the inertia matrix $\mathbf{D}$, which may be costly for high-dimensional robotic systems. In what follows, we demonstrate how one may directly leverage (33) without first converting such dynamics into control affine form to compute controllers enforcing safety. Such constructions are facilitated by the formal notion of an energy-based CBF.

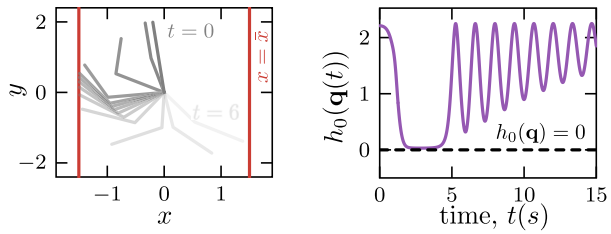

Fig. 7. Simulation results corresponding to the double pendulum from Example 7. The left plot illustrates the evolution of the pendulum in Cartesian space, where the red lines denote the boundary of the configuration constraint set, while the right plot illustrates the value of the configuration constraint along the system's trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Definition 9. The continuously differentiable function $h : T\mathcal{Q} \to \mathbb{R}$ defined as in (51) that defines a set $C \subset T\mathcal{Q}$ as in (53) is said to be an *energy-based control barrier function* for (33) on C if there exists $\alpha \in \mathcal{K}^e_\infty$ such that for all $(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q}$

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \left\{ \frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \left[\mathbf{D}(\mathbf{q}) \frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) + \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{u} \right] + \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} \right\} > -\alpha(h(\mathbf{q}, \dot{\mathbf{q}})).$$

By defining:

$$\begin{aligned} a(\mathbf{q}, \dot{\mathbf{q}}) :=& \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} + \frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \left[\mathbf{D}(\mathbf{q}) \frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} \right. \\ &\left. + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) + \mathbf{G}(\mathbf{q}) \right] + \alpha(h(\mathbf{q}, \dot{\mathbf{q}})), \\ b(\mathbf{q}, \dot{\mathbf{q}}) :=& \frac{1}{\mu^2} \|(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \mathbf{B}\|^2, \end{aligned} \tag{54}$$

the validity of an energy-based CBF candidate may be assessed using the same approach as for standard CBFs. Namely, h is an energy-based CBF provided that:

$$b(\mathbf{q}, \dot{\mathbf{q}}) = 0 \implies a(\mathbf{q}, \dot{\mathbf{q}}) > 0.$$

When $\mathbf{k}_0 : \mathcal{Q} \to \mathbb{R}^n$ and $\alpha \in \mathcal{K}^e_\infty$ satisfy (50), and (33) is fully actuated, the above condition holds since:

$$b(\mathbf{q}, \dot{\mathbf{q}}) = 0 \implies (\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \mathbf{B} = \mathbf{0} \implies \dot{\mathbf{q}} = \mathbf{k}_0(\mathbf{q}),$$

so that when $b(\mathbf{q}, \dot{\mathbf{q}}) = 0$, we have:

$$\begin{aligned} a(\mathbf{q}, \dot{\mathbf{q}}) =& \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} + \alpha(h(\mathbf{q}, \dot{\mathbf{q}})) \\ =& \nabla h_0(\mathbf{q}) \cdot \mathbf{k}_0(\mathbf{q}) + \alpha(h_0(\mathbf{q})) > 0, \end{aligned}$$

where the second equality follows from $\dot{\mathbf{q}} = \mathbf{k}_0(\mathbf{q})$ and the inequality from (50). With the above calculations, we have the following result regarding the construction of energy-based CBFs.

Lemma 5. *Let the assumptions of Lemma 4 hold. Then, $h : T\mathcal{Q} \to \mathbb{R}$ as defined in (51) is an energy-based CBF for (33) on the set $C \subset T\mathcal{Q}$ as defined in (53).*

Although the above result formalizes the construction of energy-based CBFs, we have yet to show that they may be used to synthesize controllers enforcing safety. The following theorem shows that this is indeed the case.

Theorem 9. *If $h : T\mathcal{Q} \to \mathbb{R}$ is an energy-based CBF for (33) on a set $C \subset T\mathcal{Q}$ as in (5), the any locally Lipschitz controller $\mathbf{k} : T\mathcal{Q} \to \mathbb{R}^m$ satisfying:*

$$\begin{aligned} &\frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \left[\mathbf{D}(\mathbf{q}) \frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) \right. \\ &\left. + \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) \right] + \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} \geq -\alpha(h(\mathbf{q}, \dot{\mathbf{q}})), \end{aligned} \tag{55}$$

for all $(\mathbf{q}, \dot{\mathbf{q}}) \in TQ$ *renders* C *forward invariant for the closed-loop system* (33) *with* $\mathbf{u} = \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})$.

The proof of this result, presented in the Appendix, exploits the following property of robotic systems in (33).

Property 1. *The inertia and Coriolis matrices in* (33) *satisfy the skew-symmetric property:*

$$\mathbf{v}^\top(\dot{\mathbf{D}}(\mathbf{q}, \dot{\mathbf{q}}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}))\mathbf{v} = 0, \tag{56}$$

for all $(\mathbf{q}, \dot{\mathbf{q}}) \in TQ$ *and any* $\mathbf{v} \in \mathbb{R}^n$.

Once an energy-based CBF has been constructed, a controller satisfying (55) may be synthesized by incorporating (55) as a constraint into an optimization problem to instantiate the safety filter:

$$\begin{aligned} \min_{\mathbf{u}\in\mathbb{R}^m} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_\mathrm{d}(\mathbf{q}, \dot{\mathbf{q}})\|^2 \\ \text{s.t.} \quad & \frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \Big[\mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) \\ & + \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{u} \Big] + \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} \geq -\alpha(h(\mathbf{q}, \dot{\mathbf{q}})) \end{aligned} \tag{57}$$

where $\mathbf{k}_\mathrm{d} : TQ \to \mathbb{R}^m$ is a desired control policy, whose closed-form solution is given similarly to (15) by:

$$\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{k}_\mathrm{d}(\mathbf{q}, \dot{\mathbf{q}}) - \frac{1}{\mu}\lambda\left(a(\mathbf{q}, \dot{\mathbf{q}}), b(\mathbf{q}, \dot{\mathbf{q}})\right)\mathbf{B}^\top(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})),$$

where $a : TQ \to \mathbb{R}$ and $b : TQ \to \mathbb{R}$ are defined as in (54), and $\lambda : \mathbb{R}^2 \to \mathbb{R}$ is defined with the ReLU activation function as in (15). This controller no longer contains the inverse of the inertia matrix $\mathbf{D}$. Another advantage of directly leveraging the robot dynamics in (33) is that this approach enables the use of safety-enforcing controllers other than the QP-based controller in (57). For example, when $\alpha \in \mathcal{K}_\infty^e$ is Lipschitz continuous with Lipschitz constant $\ell \in \mathbb{R}_{>0}$ and (33) is fully actuated, one can verify that:

$$\begin{aligned} \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = & \mathbf{B}^{-1}\Big[\mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) + \mathbf{G}(\mathbf{q}) \\ & + \mu\nabla h_0(\mathbf{q}) - \frac{\gamma}{2}\mathbf{D}(\mathbf{q})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})) \Big], \end{aligned} \tag{58}$$

satisfies (55) for any $\gamma \geq \ell$.

Remark 5. The energy-based CBFs outlined in this section are a generalization of those originally introduced in Singletary, Kolathaya et al. (2022). In particular, earlier notions of such CBFs are recovered by taking $\mathbf{k}_0(\mathbf{q}) = \mathbf{0}$ in (51) to obtain:

$$h(\mathbf{q}, \dot{\mathbf{q}}) = h_0(\mathbf{q}) - \frac{1}{2\mu}\dot{\mathbf{q}}^\top\mathbf{D}(\mathbf{q})\dot{\mathbf{q}}. \tag{59}$$

A limitation of the above CBF candidate becomes evident when verifying if (59) is indeed a CBF via Lemma 2. When (33) is fully actuated, we have:

$$L_\mathbf{g}h(\mathbf{x}) = -\frac{1}{\mu}\dot{\mathbf{q}}^\top\mathbf{B} = \mathbf{0} \implies \dot{\mathbf{q}} = \mathbf{0},$$

implying that when $L_\mathbf{g}h(\mathbf{x}) = \mathbf{0}$, we also have:

$$L_\mathbf{f}h(\mathbf{x}) + \alpha(h(\mathbf{x})) = \nabla h_0(\mathbf{q}) \cdot \dot{\mathbf{q}} + \alpha(h_0(\mathbf{q})) = \alpha(h_0(\mathbf{q})),$$

which is only *strictly* greater than zero on the interior of the safe set and is thus not a CBF on any set[8] $D \supseteq C$. Although, in principle, one may relax the strict inequality in Definition 5 to a nonstrict one so that (59) may serve as a CBF on C, the lack of the strict satisfaction of (12) may lead to controllers that are discontinuous when $\dot{\mathbf{q}} = \mathbf{0}$.

[8] Recall that although Definition 5 requires (12) to hold for all $\mathbf{x} \in \mathbb{R}^n$, one may also require (12) to only hold on a set D containing C.

5.3. Underactuated robotic systems

The previous results in this section formalize the construction of CBFs for fully actuated robotic systems and illustrate that when the control input is unconstrained, it is always possible to construct a CBF for the full-order dynamics (33) by simply building a CBF for a reduced-order model. These results are not surprising given that fully actuated systems are feedback equivalent to double integrators — a class of systems for which CBFs can be readily constructed as detailed in Section 4. The construction of CBFs becomes more challenging when (33) is underactuated; however, under certain assumptions, similar approaches to those outlined thus far may still be employed with the help of ideas introduced in Spong (1994) (see also (Tedrake, 2023, Ch. 3)). To introduce these ideas, we rewrite (33) as:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\mathbf{u}, \tag{60}$$

where $\mathbf{D}$ and $\mathbf{B}$ are as in (33) and $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) := \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$ collects the Coriolis and gravitational terms from (33). We now suppose that (60) is underactuated (i.e., $m < n$) and that the configuration can be partitioned into actuated $\mathbf{q}_1 \in Q_1 \subset \mathbb{R}^{n_1}$ and passive $\mathbf{q}_2 \in Q_2 \subset \mathbb{R}^{n_2}$ components in the sense that $\ddot{\mathbf{q}}_1$ may be directly influenced by the control input while $\ddot{\mathbf{q}}_2$ may only be indirectly influenced through the evolution of $\mathbf{q}_1$. Under this assumption, we may represent the dynamics as:

$$\underbrace{\begin{bmatrix} \mathbf{D}_{11}(\mathbf{q}) & \mathbf{D}_{12}(\mathbf{q}) \\ \mathbf{D}_{21}(\mathbf{q}) & \mathbf{D}_{22}(\mathbf{q}) \end{bmatrix}}_{\mathbf{D}(\mathbf{q})} \underbrace{\begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \end{bmatrix}}_{\ddot{\mathbf{q}}} + \underbrace{\begin{bmatrix} \mathbf{H}_1(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{H}_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}}_{\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})} = \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}} \mathbf{u}, \tag{61}$$

where $\mathbf{D}_{11}(\mathbf{q}) \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{D}_{22}(\mathbf{q}) \in \mathbb{R}^{n_2 \times n_2}$ are uniformly positive definite since $\mathbf{D}$ is as well. We now suppose that our configuration constraint set $C_0 \subset Q$ can be characterized as the zero superlevel set of a continuously differentiable function $h_0 : Q \to \mathbb{R}$ as in (49) that depends only on either the actuated or passive components of the configuration. For example, if our component of interest is $\mathbf{q}_1$ – the actuated component – we assume that:

$$C_0 = \{\mathbf{q} \in Q : h_{0,1}(\mathbf{q}_1) \geq 0\}, \tag{62}$$

whereas if our component of interest is $\mathbf{q}_2$ – the passive component – we assume that:

$$C_0 = \{\mathbf{q} \in Q : h_{0,2}(\mathbf{q}_2) \geq 0\}, \tag{63}$$

where $h_{0,i} : Q_i \to \mathbb{R}$, $i \in \{1, 2\}$ is continuously differentiable. Our objective is now to use the decomposition in (61) to derive a new set of equations that depends only on the acceleration of one of the components of the configuration, depending on the configuration constraint.

We begin with the simpler situation in which our configuration constraint depends on the actuated components of the configuration. Our objective is to derive an equivalent representation of (60) that depends only on $\ddot{\mathbf{q}}_1$. To this end, we note that since $\mathbf{D}_{22}(\mathbf{q})$ is invertible, we may use the second equation in (61) to solve for $\ddot{\mathbf{q}}_2$ as:

$$\ddot{\mathbf{q}}_2 = -\mathbf{D}_{22}(\mathbf{q})^{-1}\left[\mathbf{D}_{21}(\mathbf{q})\ddot{\mathbf{q}}_1 + \mathbf{H}_2(\mathbf{q}, \dot{\mathbf{q}})\right]. \tag{64}$$

This expression may now be substituted back into the first equation to obtain:

$$\bar{\mathbf{D}}_1(\mathbf{q})\ddot{\mathbf{q}}_1 + \bar{\mathbf{H}}_1(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}_1\mathbf{u}, \tag{65}$$

which depends only on $\ddot{\mathbf{q}}_1$, where

$$\begin{aligned} \bar{\mathbf{D}}_1(\mathbf{q}) &:= \mathbf{D}_{11}(\mathbf{q}) - \mathbf{D}_{12}(\mathbf{q})\mathbf{D}_{22}(\mathbf{q})^{-1}\mathbf{D}_{21}(\mathbf{q}), \\ \bar{\mathbf{H}}_1(\mathbf{q}, \dot{\mathbf{q}}) &:= \mathbf{H}_1(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{D}_{12}(\mathbf{q})\mathbf{D}_{22}(\mathbf{q})^{-1}\mathbf{H}_2(\mathbf{q}, \dot{\mathbf{q}}). \end{aligned}$$

Note that $\bar{\mathbf{D}}_1$ is simply the Schur complement of $\mathbf{D}$ and is symmetric and positive definite since $\mathbf{D}$ is as well (Spong, 1994). Given the dynamics in (65), we propose the CBF candidate:

$$\begin{aligned} h(\mathbf{q}, \dot{\mathbf{q}}) = & h_{0,1}(\mathbf{q}_1) \\ & - \frac{1}{2\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{D}}_1(\mathbf{q})(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1)), \end{aligned} \tag{66}$$

where $\mu \in \mathbb{R}_{>0}$ and $\mathbf{k}_{0,1} : \mathcal{Q}_1 \to \mathbb{R}^{n_1}$ is a continuously differentiable controller satisfying:

$$\nabla h_{0,1}(\mathbf{q}_1) \cdot \mathbf{k}_{0,1}(\mathbf{q}_1) > -\alpha(h_{0,1}(\mathbf{q}_1)), \tag{67}$$

for all $\mathbf{q}_1 \in \mathcal{Q}_1$ for some $\alpha \in \mathcal{K}^e_\infty$. This CBF candidate may be used to define a candidate safe set $\mathcal{C} \subset T\mathcal{Q}$ for the robotic system as in (53). The following theorem illustrates that this function is a CBF for the control affine representation of this underactuated robotic system.

Theorem 10. *Consider system (61) and a configuration constraint set $\mathcal{C}_0 \subset \mathcal{Q}$ as in (62). Provided $\mathbf{B}_1 \in \mathbb{R}^{n_1 \times m}$ is pseudo-invertible and $\mathbf{k}_{0,1} : \mathcal{Q}_1 \to \mathbb{R}^{n_1}$ satisfies (67), then the function $h : T\mathcal{Q} \to \mathbb{R}$ as defined in (66) is a CBF for the corresponding control affine system (34).*

A proof of this theorem is provided in the Appendix and follows a similar argument to the results of Section 5.1. Note that, under the assumption that $\mathbf{B}_1$ is pseudo-invertible, system (65) effectively acts as a fully actuated system since one may directly command any desired $\ddot{\mathbf{q}}_1$ to achieve the control objective, and is reminiscent of the collocated feedback linearization method outlined in Spong (1994).

The fact that we may construct a CBF for the actuated subsystem in (61) under similar assumptions to those in the previous section should not be too surprising. A more interesting situation, however, arises when our configuration constraint is a function of the passive components of the configuration as in (63). Under the following condition, a similar approach to that just introduced may be used to construct a CBF from a configuration constraint on the passive components of the configuration.

Definition 10 (*Spong, 1994*)**.** System (61) is said to *strongly inertially coupled* on a set $\mathcal{D} \subset \mathcal{Q}$ if $\mathbf{D}_{21}(\mathbf{q})$ is pseudo-invertible for all $\mathbf{q} \in \mathcal{D}$.

Provided the above condition is satisfied, we may rewrite the first equation in (61) in terms of $\ddot{\mathbf{q}}_2$ by first solving the second equation in (61) for $\ddot{\mathbf{q}}_1$ to obtain:

$$\ddot{\mathbf{q}}_1 = -\mathbf{D}_{21}(\mathbf{q})^\dagger \left[\mathbf{D}_{22}(\mathbf{q})\ddot{\mathbf{q}}_2 + \mathbf{H}_2(\mathbf{q}, \dot{\mathbf{q}})\right],$$

where $\mathbf{D}_{21}(\mathbf{q})^\dagger$ denotes the pseudo-inverse of $\mathbf{D}_{21}(\mathbf{q})$. The above expression can then be substituted into the first equation in (61) to obtain:

$$\bar{\mathbf{D}}_2(\mathbf{q})\ddot{\mathbf{q}}_2 + \bar{\mathbf{H}}_2(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}_1 \mathbf{u}, \tag{68}$$

where

$$\begin{aligned}\bar{\mathbf{D}}_2(\mathbf{q}) :=& \mathbf{D}_{12}(\mathbf{q}) - \mathbf{D}_{11}(\mathbf{q})\mathbf{D}_{21}(\mathbf{q})^\dagger \mathbf{D}_{22}(\mathbf{q}) \\ \bar{\mathbf{H}}_2(\mathbf{q}, \dot{\mathbf{q}}) :=& \mathbf{H}_1(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{D}_{11}(\mathbf{q})\mathbf{D}_{21}(\mathbf{q})^\dagger \mathbf{H}_2(\mathbf{q}, \dot{\mathbf{q}}),\end{aligned}$$

which now depends only on $\ddot{\mathbf{q}}_2$, and is a valid representation of (61) on the set where (61) is strongly inertially coupled. As discussed in Spong (1994), $\bar{\mathbf{D}}_2$ also has full rank on the set where the strong inertial coupling condition holds. Given the dynamics in (68), we propose the CBF candidate:

$$h(\mathbf{q}, \dot{\mathbf{q}}) = h_{0,2}(\mathbf{q}_2) - \frac{1}{2\mu} \left\| \bar{\mathbf{D}}_2(\mathbf{q})(\dot{\mathbf{q}}_2 - \mathbf{k}_{0,2}(\mathbf{q}_2)) \right\|^2 \tag{69}$$

where $\mu \in \mathbb{R}_{>0}$ and $\mathbf{k}_{0,2} : \mathcal{Q}_2 \to \mathbb{R}^{n_2}$ is a continuously differentiable controller satisfying:

$$\nabla h_{0,2}(\mathbf{q}_2) \cdot \mathbf{k}_{0,2}(\mathbf{q}_2) > -\alpha(h_{0,2}(\mathbf{q}_2)), \tag{70}$$

for all $\mathbf{q}_2 \in \mathcal{Q}_2$ for some $\alpha \in \mathcal{K}^e_\infty$. As in the previous case, this CBF candidate may be used to define a candidate safe set $\mathcal{C} \subset T\mathcal{Q}$ for the robotic system as in (53). Now, under the additional assumption that (61) is strongly inertially coupled on $\mathcal{C}_0$, Theorem 10 may be extended to construct a CBF from a configuration constraint that depends on the passive components of the configuration.

Theorem 11. *Consider system (61) and a configuration constraint set $\mathcal{C}_0 \subset \mathcal{Q}$ as in (63). Provided $\mathbf{B}_1 \in \mathbb{R}^{n_1 \times m}$ is pseudo-invertible, $\mathbf{k}_{0,2} : \mathcal{Q}_2 \to \mathbb{R}^{n_2}$ satisfies (67), and (61) is strongly inertially coupled on $\mathcal{C}_0$, then the function $h : T\mathcal{Q} \to \mathbb{R}$ as defined in (69) is a CBF for the corresponding control affine system (34).*

The above theorem, whose proof follows the same steps as those in the proof of Theorem 10, is, effectively, an extension of the non-collocated feedback linearization method from (Spong, 1994) to safety-critical control. The following example illustrates how one may apply these results to a classic underactuated robotic system.

Example 8 (*Cartpole*)**.** We now demonstrate the design of CBFs for underactuated robotic systems using an example borrowed from (Singletary, Kolathaya et al., 2022), which involves designing a safety-critical controller for the cartpole system as illustrated in Fig. 8. The configuration of this system is given by $\mathbf{q} = (x, \theta)$, where $x \in \mathbb{R}$ is the position of the cart and $\theta \in [0, 2\pi)$ the angular position of the pole, and the input corresponds to a force applied to the cart. The dynamics are of the form (33) with:

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} m_c + m_p & m_p l \cos(\theta) \\ m_p l \cos(\theta) & m_p l^2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$
$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & -m_p l \dot{\theta} \sin(\theta) \\ 0 & 0 \end{bmatrix}, \quad \mathbf{G}(\mathbf{q}) = \begin{bmatrix} 0 \\ m_p g l \sin(\theta) \end{bmatrix}$$

where $m_c \in \mathbb{R}_{>0}$ denotes the mass of the cart, $m_p \in \mathbb{R}_{>0}$ denotes the mass of the pole, $l \in \mathbb{R}_{>0}$ denotes the length of the pole, and $g \in \mathbb{R}_{>0}$ is the acceleration due to gravity. These dynamics may also be represented as in (61) with x and θ corresponding to the actuated and passive components of the configuration, respectively, implying one may directly influence $\ddot{x}$ via control inputs, whereas $\ddot{\theta}$ may only be indirectly influenced by actuating the cart. Our control objective is to constrain the angular position of the pole to lie within $\theta \in [\frac{5\pi}{6}, \frac{7\pi}{6}]$, which may be expressed as the safety constraint:

$$h_0(\theta) = \left(\frac{\pi}{6}\right)^2 - (\theta - \pi)^2,$$

where $\theta = \pi$ corresponds to the pole being upright, which defines a configuration constraint set $\mathcal{C}_0 \subset \mathcal{Q}$ as in (63). As our safety constraint depends only on θ, we attempt to rewrite the cartpole dynamics as in (68). To do so, we must ensure that the cartpole dynamics are strongly inertially coupled, at least on $\mathcal{C}_0$, which follows from the fact that $D_{21}(\mathbf{q}) = m_p l \cos(\theta)$ is only zero for $\theta = \pm\pi/2$ and is not contained in $\mathcal{C}_0$. Hence, we represent the cartpole dynamics as in (68) for all $\mathbf{q} \in \mathcal{C}_0$ with:

$$\begin{aligned}\bar{D}_2(\mathbf{q}) =& m_p l \cos(\theta) - \frac{(m_c + m_p) m_p l^2}{m_p l \cos(\theta)} \\ \bar{H}_2(\mathbf{q}, \dot{\mathbf{q}}) =& -m_p l \dot{\theta}^2 \sin(\theta) - \frac{(m_c + m_p)(m_p g l \sin(\theta))}{m_p l \cos(\theta)},\end{aligned}$$

which are valid so long as $\cos(\theta) \neq 0$. With this representation of the dynamics, we form our CBF candidate as in (69), where $k_{0,2} : [0, 2\pi) \to \mathbb{R}$ is constructed using the Softplus universal formula from Section 2.5. Since the dynamics are strongly inertially coupled on $\mathcal{C}_0$ and $B_1 = 1$ is invertible, the function h from (69) is a CBF for the control-affine representation of this system (34). This CBF is used to construct a QP-based safety filter $\mathbf{k}$ as in (13) for the nominal controller:

$$k_d(\mathbf{q}, \dot{\mathbf{q}}) = -K_\theta(\theta - \theta_d(t)) - K_{\dot{\theta}}\dot{\theta},$$

where $K_\theta, K_{\dot{\theta}} \in \mathbb{R}_{>0}$ are gains, which attempts to track a desired trajectory $\theta_d : \mathbb{R}_{\geq 0} \to \mathbb{R}$ for the pole's angular position. The results of applying this safety filter to the cartpole are provided in Fig. 8. Note that the desired pole position lies outside of $\mathcal{C}_0$ so that the performance objective is directly in conflict with the safety objective. Despite this, and the fact that one cannot directly actuate the angular position of the pole, safety is guaranteed through the careful construction of a CBF.

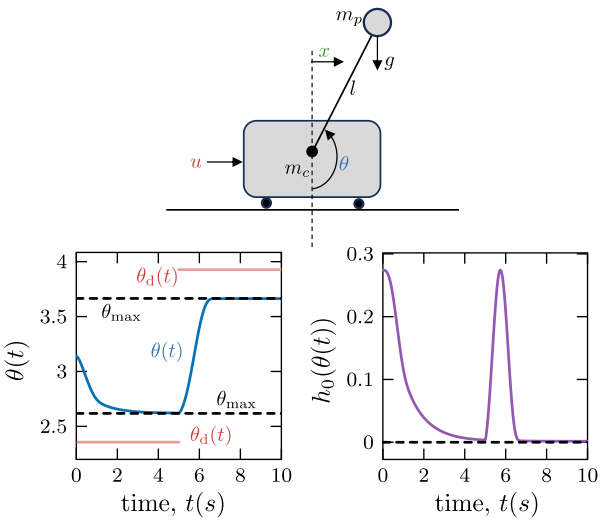

Fig. 8. Results of the cartpole simulation from Example 8. Here, the left plot displays the evolution of the pole's position and the right plot illustrates the evolution of the configuration constraint along the trajectory of the system, both of which demonstrate the resulting safe behavior.

6. Stable tracking of safe reduced order models

In the previous sections, we outlined various methodologies to construct CBFs for high-dimensional systems with cascaded dynamics. Although these approaches enable the systematic construction of CBFs for relevant classes of systems, they are heavily model-dependent in the sense that one must leverage the full-order dynamics of the system to compute controllers enforcing safety. In practice, such models may be imperfect or may be computationally intensive to compute, limiting their use in controllers that must run in real time. Moreover, in many situations, one may not even have direct access to the control input for the full-order system, and may only be able to pass reference commands to black-box modules within the existing autonomy stack that compute such control inputs.

In this section, we present a suite of techniques to address these aforementioned challenges. Such techniques are, in a certain sense, a generalization of the ideas introduced thus far and enable the application of these ideas to more complex systems, but also lead to a fundamentally different approach to safety-critical control. Our developments here are facilitated by the realization that the paradigm of safety-critical control based on ROMs can be understood as certifying the ability of the full-order system to track a suitably designed ROM. Earlier, we implicitly combined a CBF for a ROM with a Lyapunov-like function to produce a CBF for the overall system. In this section, we make such an idea more explicit.

The benefit of making this unification of barrier and Lyapunov functions explicit lies in the ability to decouple the design of the safety-critical control architecture from the full-order model. This decoupling leads to a notion of *model-free* safety-critical control in the sense that the safety-critical component of the control architecture may be designed and implemented independent of the full-order dynamics. Safety of the full-order dynamics can then be guaranteed so long as such dynamics track commands generated by the ROM. The synthesis of such tracking controllers may require knowledge of the full-order dynamics; however, tracking controllers for many relevant classes of systems, such as those in robotics, are well established and may be readily applied within this model-free safety-critical control paradigm to enforce safety.

6.1. Lyapunov-certified tracking

To illustrate the ideas introduced earlier in a more general context, consider again the two-layered system from (29), which may also be written in standard control affine form (9) with state $\mathbf{x} = (\mathbf{q}, \boldsymbol{\xi})$ as noted in (31). As we did earlier, we consider the top-level dynamics:

$$\dot{\mathbf{q}} = \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\boldsymbol{\xi},$$

as a reduced-order representation of the full-order system for which we wish to design a smooth controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ that would enforce safety of the ROM if its dynamics were directly controllable. Rather than leveraging $\mathbf{k}_0$ to backstep through these dynamics to compute a safe controller, here we consider the existence of a *tracking controller* $\mathbf{k} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ that is able drive the state $\boldsymbol{\xi}$ to $\mathbf{k}_0(\mathbf{q})$. Accordingly, we assume that there exists a Lyapunov function $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}_{\geq 0}$ for the full-order dynamics:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})\boldsymbol{\xi}\\ \dot{\boldsymbol{\xi}} &= \mathbf{f}_1(\mathbf{q}, \boldsymbol{\xi}) + \mathbf{g}_1(\mathbf{q}, \boldsymbol{\xi})\mathbf{k}(\mathbf{q}, \boldsymbol{\xi}),\end{aligned}$$

satisfying:

$$\gamma_1 \|\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q})\|^2 \leq V(\mathbf{q}, \boldsymbol{\xi}) \leq \gamma_2 \|\boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q})\|^2 \tag{71a}$$

$$\dot{V}(\mathbf{q}, \boldsymbol{\xi}) = L_{\mathbf{f}} V(\mathbf{x}) + L_{\mathbf{g}} V(\mathbf{x})\mathbf{k}(\mathbf{x}) \leq -\gamma V(\mathbf{q}, \boldsymbol{\xi}), \tag{71b}$$

for positive constants $\gamma_1, \gamma_2, \gamma > 0$. This Lyapunov function certifies the ability of the full-order dynamics to track commands generated by the reduced dynamics, represented as the outputs of the reduced-order controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$.

To see how this tracking controller and corresponding Lyapunov function may be used to establish safety of the overall system, we write the top layer dynamics from (29) as:

$$\dot{\mathbf{q}} = \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0(\mathbf{q})(\mathbf{k}_0(\mathbf{q}) + \mathbf{d}), \tag{72}$$

where:

$$\mathbf{d} := \boldsymbol{\xi} - \mathbf{k}_0(\mathbf{q}), \tag{73}$$

is the tracking error for the full order system, which is treated as a disturbance that must be rejected by the top layer to ensure safety. To account for this disturbance, we now require $\mathbf{k}_0$ to satisfy:

$$L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q})\mathbf{k}_0(\mathbf{q}) \geq -\alpha h_0(\mathbf{q}) + \frac{1}{\varepsilon}\|L_{\mathbf{g}_0} h_0(\mathbf{q})\|^2, \tag{74}$$

where $h_0 : \mathbb{R}^n \to \mathbb{R}$ defines the set $C_0 \subset \mathbb{R}^n$ as in (37) and $\alpha, \varepsilon > 0$. That is, rather than requiring h_0 to be a CBF for the top layer dynamics, we now require h_0 to be an *ISSf-CBF* (see Section 2.4) for the top layer. Following a similar procedure as before, we now define:

$$h(\mathbf{q}, \boldsymbol{\xi}) = h_0(\mathbf{q}) - \frac{1}{\mu\gamma_1} V(\mathbf{q}, \boldsymbol{\xi}), \tag{75}$$

as a candidate barrier function for the closed-loop system, which defines the candidate safe set:

$$C = \{(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^n \times \mathbb{R}^p \ : \ h(\mathbf{q}, \boldsymbol{\xi}) \geq 0\}, \tag{76}$$

as its zero superlevel set. As V is positive definite, we have $h(\mathbf{q}, \boldsymbol{\xi}) \geq 0 \implies h_0(\mathbf{q}) \geq 0$ so that enforcing forward invariance of C in (76) is sufficient to ensure that $h_0(\mathbf{q}(t)) \geq 0$. The following theorem provides conditions under which h is a barrier function for the closed-loop system.

Theorem 12. *Consider the dynamics in (29), the constraint set $C_0 \subset \mathbb{R}^n$ in (37), and suppose there exists a continuously differentiable controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ and positive constants $\alpha, \varepsilon > 0$ satisfying (74). Furthermore, suppose there exists a tracking controller $\mathbf{k} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ and Lyapunov function $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}_{\geq 0}$ satisfying (71) for positive constants $\gamma_1, \gamma_2, \gamma > 0$. Provided:*

$$\gamma \geq \alpha + \frac{\varepsilon\mu}{4}, \tag{77}$$

then $C \subset \mathbb{R}^n \times \mathbb{R}^p$ as defined in (76) is forward invariant for the closed-loop control affine system (31) with $\mathbf{u} = \mathbf{k}(\mathbf{q}, \boldsymbol{\xi})$.

The previous theorem, whose proof is provided in the Appendix, states that, with good enough tracking performance, safety may be enforced on the full-order dynamics by simply tracking the outputs of a safe ROM. The condition in (77) requires that the rate of convergence of the tracking error – captured via γ – must be larger than the rate at which the ROM may approach the boundary of the constraint set – captured via α. For a fixed tracking controller, one may satisfy (77) by designing an appropriate ROM by decreasing α, which limits how quickly the ROM may approach the boundary of the constraint set, and decreasing ε, which corresponds to robustifying the ROM to larger tracking errors. Hence, for a fixed tracking controller satisfying (71), one may always ensure safety at the cost of using a more conservative ROM.

As argued earlier, the benefit of the preceding result is that the safety-critical portion of the control architecture only relies on the reduced-order dynamics. As opposed, the results from earlier sections established the existence of CBF for the full-order system, the dynamics of which one must ultimately leverage to synthesize a controller enforcing safety. Here, one may instead leverage an existing tracking controller that may already be integrated into the system's autonomy stack to track commands produced by the reduced-order controller and guarantee safety.

These safety guarantees, of course, are conditioned on the ability of such a tracking controller to perfectly track reference commands. In practice, however, perfect tracking – the satisfaction of (71b) – is often not achievable and instead, our tracking controller $\mathbf{k}$ may only achieve:

$$\dot{V}(\mathbf{q},\xi) \leq -\gamma V(\mathbf{q},\xi) + \delta, \tag{78}$$

for positive constants $\gamma, \delta > 0$. That is, the tracking controller enforces *input-to-state-stability* (ISS) of the tracking error dynamics rather than exponential stability as in (71b). The inability of the full-order dynamics to perfectly track the reduced-order model leads us to consider the modified barrier candidate:

$$h(\mathbf{q},\xi) = h_0(\mathbf{q}) - \frac{1}{\mu\gamma_1}\left(V(\mathbf{q},\xi) - \frac{\delta}{\alpha}\right), \tag{79}$$

which defines a candidate safe set $\mathcal{C}$ as in (76). Compared to (75), the above barrier candidate inflates the original safe set proportional to δ to account for imperfect tracking. The following result illustrates that under similar conditions to those in Theorem 12, this tracking controller enforces ISSf of the overall system with respect to the ISSf barrier function (79).

Theorem 13. *Consider the dynamics in (29), the constraint set $\mathcal{C}_0 \subset \mathbb{R}^n$ in (37), and suppose there exists a continuously differentiable controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ and positive constants $\alpha, \varepsilon > 0$ satisfying (74). Furthermore, suppose there exists a tracking controller $\mathbf{k} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ and Lyapunov function $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}_{\geq 0}$ satisfying (78) and (71a) for positive constants $\gamma_1, \gamma_2, \gamma, \delta > 0$. Provided (77) holds, then $\mathcal{C} \subset \mathbb{R}^n \times \mathbb{R}^p$ as defined in (76), with $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ from (79), is forward invariant for the closed-loop control affine system (31) with $\mathbf{u} = \mathbf{k}(\mathbf{q},\xi)$.*

The proof of this result follows the same steps as those employed in the proof of Theorem 12. As this result establishes forward invariance of an inflated safe set, rather than the original safe set defined by (75), it effectively establishes ISSf of the full-order dynamics. Note that for both Theorems 12 and 13 the parameters of the ROM and tracking controller must satisfy the same condition (77); however, the safe sets for each of these results – characterized as the zero superlevel sets of (75) and (79), respectively – are different. Compared to (75), the safe set defined by (79) is inflated by an additional margin proportional to δ/α. One can bring the resulting inflated safe set closer to the original safe set by increasing α, resulting in a more "aggressive" ROM; however, to guarantee ISSf, the increase in α must be compensated for with larger γ, which requires the tracking controller to enforce faster convergence of the system to commanded references. Furthermore, by increasing robustness through decreasing ε, one may take larger values of μ in (77), making the corresponding forward invariant set given by (79) closer to the original constraint set given by h_0. Before proceeding, we illustrate how one may apply these results with the help of the following example.

Example 9 (*Planar Segway*)**.** We demonstrate the model-free safety-critical control paradigm by using the example of a Segway control problem from (Molnar et al., 2022). Consider the planar Segway model in Fig. 10(a) with configuration $\mathbf{q} = (p, \varphi) \in \mathcal{Q} = \mathbb{R} \times [0, 2\pi)$ including the position p and pitch angle φ of the Segway. We seek to drive the Segway with a desired speed $\dot{p}_{\mathrm{d}}$ until reaching a wall at position $p_{\max}$ where the Segway must stop automatically such that $p \leq p_{\max}$. The dynamics of the Segway are given by (33) with $u \in \mathbb{R}$ being the voltage on the Segway's motors and:

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} m_0 & mL\cos\varphi \\ mL\cos\varphi & J_0 \end{bmatrix}, \quad \mathbf{G}(\mathbf{q}) = \begin{bmatrix} 0 \\ -mgL\sin\varphi \end{bmatrix},$$
$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} b_{\mathrm{t}}/R & -b_{\mathrm{t}} - mL\dot{\varphi}\sin\varphi \\ -b_{\mathrm{t}} & b_{\mathrm{t}}R \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} K_{\mathrm{m}}/R \\ -K_{\mathrm{m}} \end{bmatrix},$$

where R and L are geometric dimensions, m, m_0, J_0 are mass and inertia parameters, g is acceleration from gravity, while b_{t} and K_{m} are motor parameters, all given in Molnar et al. (2022). Note that although these dynamics are in the form of (33), they are underactuated, which complicates the backstepping-like methods developed in previous sections.

To address this challenge, we proceed to leverage the model-free safety-critical control approach developed in this section, where we use the single integrator $\dot{\mathbf{q}} = \xi$ as a ROM to provide safety against collision with the wall, with desired controller $\mathbf{k}_{0,\mathrm{d}}(\mathbf{q}) = \begin{bmatrix} \dot{p}_{\mathrm{d}} & 0 \end{bmatrix}^\top$ and CBF:

$$h_0(\mathbf{q}) = p_{\max} - p,$$

that satisfies $L_{\mathbf{g}_0} h_0(\mathbf{q}) \neq \mathbf{0}$. This CBF is then used to construct a smooth safety filter $\mathbf{k}_0 : \mathcal{Q} \to \mathbb{R}^2$ as in Section 2.5 for the ROM. The output of this smooth safety filter represents a safe velocity for the Segway: the robot may travel with the desired speed $\dot{p}_{\mathrm{d}}$ until getting close to the wall, where it must reduce its speed according to its distance from the wall. The safe velocity can be tracked by an on-board controller designed for the full system (33) that also stabilizes the Segway upright:

$$\mathbf{k}(\mathbf{q},\dot{\mathbf{q}}) = K_{\dot{p}}(\dot{p} - k_0(\mathbf{q})) + K_{\varphi}\varphi + K_{\dot{\varphi}}\dot{\varphi}. \tag{80}$$

with gains $K_{\dot{p}}$, K_{φ}, $K_{\dot{\varphi}}$, where $k_0(\mathbf{q})$ is the first component of $\mathbf{k}_0(\mathbf{q})$ and represents a safe forward velocity. This controller satisfies the conditions of Theorem 13 using:

$$V(\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{2}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top \mathbf{D}(\mathbf{q})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})),$$

as an ISS Lyapunov function, wherein the constants γ and δ from (78) may be determined using a similar analysis to that performed in Molnar et al. (2022).

The results of applying this controller to the Segway for different choices of gains in (80) and different choices of α and ε used in synthesizing the smooth safety filter $\mathbf{k}_0$ are provided in Fig. 9. In particular, the left and right columns in Fig. 9 illustrate the behavior of the system for $K_{\dot{p}} = 50$ and $K_{\dot{p}} = 30$, respectively, for different choices of α and ε. Here, safety is maintained for larger $K_{\dot{p}}$, resulting in larger γ in (78), whereas safety is violated for small values of $K_{\dot{p}}$. Intuitively, larger values of $K_{\dot{p}}$ allow the full-order dynamics to respond faster to commands generated by the ROM and maintain safety (cf. (77)). This highlights the fact that, although the controller (80) ultimately applied to this system does not directly leverage the full-order Segway dynamics, tuning this tracking controller to enforce safety may require exploiting model knowledge. In practice, however, it may not be possible to modify an existing tracking controller to satisfy (77) as it may represent a "black-box" module already be integrated

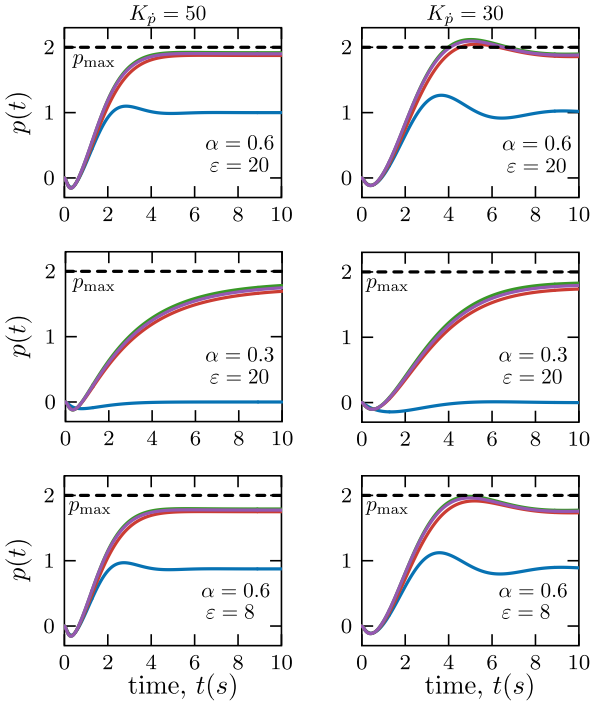

Fig. 9. Model-free safety-critical control of the planar Segway from Example 9. The plots display the evolution of the Segway's position generated by the controller in (80) with $K_{\dot{p}} = 50$ (left) and $K_{\dot{p}} = 30$ (right) for different choices of α and ε. The curves of different colors represent the trajectories under different smooth safety filters for the ROM, where the colors have the same interpretation as in Fig. 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

into the system's autonomy stack. In such a situation, one can only tune the behavior of the reduced-order model via α and ε, to satisfy the conditions required by (77). The effect of changing α for the two tracking controllers is illustrated in the middle row of Fig. 9, where the tracking controller that originally did not enforce safety ($K_{\dot{p}} = 30$) maintains safety with a lower value of α. Intuitively, decreasing α causes the reduced-order model to approach the boundary of the constraint set more slowly, requiring less aggressive tracking by the full-order dynamics to ensure safety. Alternatively, one may tune the reduced-order model by decreasing ε (bottom row of Fig. 9), which effectively adds an additional robustness margin to the reduced-order model, causing it to stop short of the original constraint boundary.

6.2. Safely tracking nonsmooth ROMs

Thus far, the safety-critical control via ROM paradigm has relied on the use of *smooth* ROMs, implying that one must leverage the smooth safety filters from Section 2.5 to design a safe ROM controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$. Although these smooth safety filters can be tuned to approximate the QP-based safety filter from (13) arbitrarily closely, in practice, such controllers tend to be more conservative than their QP counterparts. Our restriction to smooth controllers at the ROM level was necessary in our backstepping approach since such controllers were explicitly used to define a CBF for the full-order system, which must be continuously differentiable[9]. Smoothness also played an important role in the previous subsection wherein we explicitly combined a ROM

[9] Note that nonsmooth versions of CBFs do exist (Glotfelter, Cortés, & Egerstedt, 2017; Usevitch, Garg, & Panagou, 2020) and have been used to address multiple safety constraints (Glotfelter, Cortés, & Egerstedt, 2020).

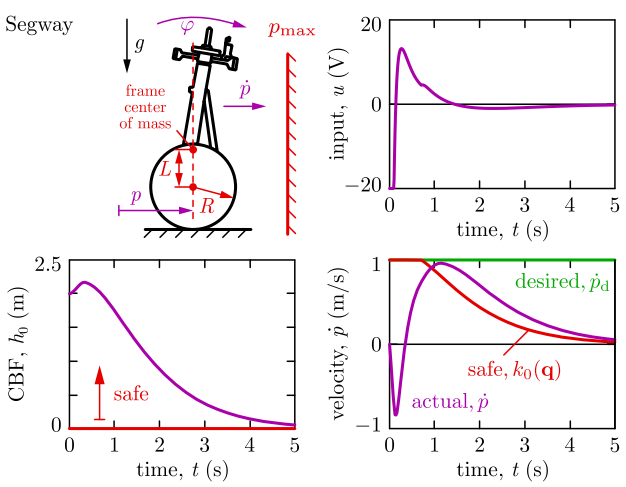

Fig. 10. Model-free safety-critical control of a Segway in simulation, with results from (Molnar et al., 2022). A planar Segway model is controlled to stop in front of a wall, by the help of a CBF-based safe velocity command and a velocity-tracking controller.

CBF and a smooth Lyapunov function to build a CBF for the full-order system; however, as shown in this subsection, the existence of a smooth Lyapunov function is not necessary to establish such results.

We now relax this smoothness requirement, which facilitates the use of QP-based controllers for the ROM, by assuming that the tracking error $\mathbf{d}$ is bounded as:

$$\|\mathbf{d}\|^2 \leq M e^{-\gamma t} + \delta, \tag{81}$$

for nonnegative constants $M, \gamma, \delta \geq 0$. This bound reflects the ability of the full-order system to exponentially track the reduced-order model up to a bound δ and is analogous to the ISS condition in (78), albeit without the explicit use of a Lyapunov function. One may set various constants in (81) equal to zero to reflect the tracking capabilities of the full-order system: $\delta = 0$ reflects perfect tracking and $M = 0$ reflects bounded, but not convergent tracking. Rather than building a barrier function for the full-order system from a Lyapunov function, we directly utilize (81) to propose the time-varying barrier candidate:

$$h(\mathbf{q}, \boldsymbol{\xi}, t) = h_0(\mathbf{q}) - \frac{M}{\mu} e^{-\gamma t} + \frac{\varepsilon\delta}{4\alpha}, \tag{82}$$

for a positive constant $\mu > 0$, which defines the time-varying safe set:

$$C(t) := \{(\mathbf{q}, \boldsymbol{\xi}) \in \mathbb{R}^n \times \mathbb{R}^p \ : \ h(\mathbf{q}, \boldsymbol{\xi}, t) \geq 0\}, \tag{83}$$

associating to each time t a set $C(t) \subset \mathbb{R}^n \times \mathbb{R}^p$ of safe states. The following theorem shows that, under similar conditions to the preceding results, h as in (82) is an ISSf barrier function for the closed-loop system.

Theorem 14. *Consider the dynamics in (29), the constraint set $C_0 \subset \mathbb{R}^n$ in (37), and suppose there exists a controller $\mathbf{k}_0 : \mathbb{R}^n \to \mathbb{R}^p$ and positive constants $\alpha, \varepsilon > 0$ satisfying (74). Furthermore, suppose there exists a tracking controller $\mathbf{k} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ enforcing the tracking error bound in (81) for constants $M, \gamma, \delta \geq 0$. Provided that (77) holds then $C(t) \subset \mathbb{R}^n \times \mathbb{R}^p$ as defined in (83) is forward invariant for the corresponding closed-loop control affine system (31) with $\mathbf{u} = \mathbf{k}(\mathbf{q}, \boldsymbol{\xi})$.*

For completeness, the proof of this theorem is provided in the Appendix. The following example shows how the preceding results allow for leveraging a QP-based controller for the ROM from Example 9.

Example 10 (*Planar Segway*). We now return to Example 9, where we seek to use a QP-based controller (13) for the ROM rather than

a smooth safety filter. The QP solution (15) leads to the following safety-critical controller for the ROM:

$$\mathbf{k}_0(\mathbf{q}) = \begin{bmatrix} k_0(\mathbf{q}) \\ 0 \end{bmatrix}, \quad k_0(\mathbf{q}) = \min\{\dot{p}_\mathrm{d}, \alpha(p_\mathrm{max} - p) - \tfrac{1}{\epsilon}\},$$

with $\alpha > 0$. Although this controller is nonsmooth, we may leverage the same exact tracking controller (80) as in the previous example, and leverage Theorem 14 to establish safety of the full-order dynamics.

Fig. 10 shows the corresponding simulation results from (Molnar et al., 2022). The Segway's motion is safe, as established by Theorem 14. Once again, the safe velocity expression does not use the full model (33), but only exploits the underlying multi-layer structure with a corresponding trivial ROM that has no parameters. This ultimately leads to a *model-free* method with a simple explicit "min" formula to provide safety for a robotic system. Meanwhile, the tracking controller does not involve the expressions in the model (33) either, however, as discussed in Example 9, appropriate selection of the gains $K_{\dot{p}}$, K_φ, $K_{\dot{\varphi}}$ may require model information. Furthermore, when directly tuning the gains of the tracking controller is not feasible, one may directly modify the parameters of the reduced-order model to ensure safety as demonstrated in Example 9.

7. Case studies

Thus far, we have introduced a variety of different CBF techniques based on the idea of leveraging ROMs to extend a CBF for a simple system to one for a complex system. In each of our illustrations of these techniques, we have chosen relatively simple examples that are just rich enough to capture the main ideas introduced herein. Yet, the motivation for introducing such ideas in the first place was to provide a viable pathway to safety-critical control of complex, high-dimensional autonomous systems.

The safety-critical controllers established above through the use of CBF theory have been implemented on a wide variety of such systems, and, in this section, we revisit more complex application examples from the literature that use these methods. These examples include safety-critical control of fixed-wing aircraft, flying, legged and wheeled robots, manipulators, and heavy-duty trucks — both in simulation and hardware experiments.

7.1. Run-time assurance on fixed-wing aircraft

We demonstrate the application of safe backstepping with CBFs by revisiting the work in Molnar et al. (2024), wherein a fixed-wing aircraft was controlled in a safety-critical fashion with the objective of preventing collision with other aircraft or entry into a restricted airspace bounded by a "geofence". The overall control pipeline is illustrated in Fig. 11. The aircraft uses a desired flight controller, that tracks a trajectory with stable flight, and a run-time assurance (RTA) system, that overrides this desired flight controller whenever necessary for collision avoidance and geofencing. The RTA is formulated as a safety filter using CBFs constructed by backstepping.

The controller synthesis is based on a kinematic model, that is used to design acceleration and angular velocity commands for the aircraft in a provably safe fashion. This model has a multi-layer cascaded structure similar to (48):

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v}(\boldsymbol{\zeta}), \\ \dot{\boldsymbol{\zeta}} &= \mathbf{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}, \phi, A_\mathrm{T}, Q), \\ \dot{\phi} &= f_\phi(\boldsymbol{\zeta}, \phi, Q, P), \end{aligned}$$

with state $\mathbf{x} = (\mathbf{r}, \boldsymbol{\zeta}, \phi) \in \mathbb{R}^7$ and input $\mathbf{u} = (A_\mathrm{T}, P, Q) \in \mathbb{R}^3$; see detailed description in Molnar et al. (2024). According to this model, the position $\mathbf{r} \in \mathbb{R}^3$ evolves according to the expression of the velocity $\mathbf{v}$, given by the state $\boldsymbol{\zeta} \in \mathbb{R}^3$ that includes speed, pitch angle and yaw angle. The evolution of $\boldsymbol{\zeta}$ depends on the roll angle $\phi \in \mathbb{R}$, the longitudinal acceleration $A_\mathrm{T} \in \mathbb{R}$ and the angular velocity $Q \in \mathbb{R}$ about the right axis of the aircraft (related to pitching up or down), where A_T and Q are viewed as control inputs. Finally, the evolution of the last state ϕ involves the angular velocity P about the front axis (related to rolling), which is considered to be the third control input. Overall, the dynamics have a 3-layer cascaded structure, where inputs enter at the second and third layers. Importantly, the right-hand side functions $\mathbf{f}_{\boldsymbol{\zeta}}$ and f_ϕ are affine in the control inputs A_T, P, Q and in certain expressions of the states.

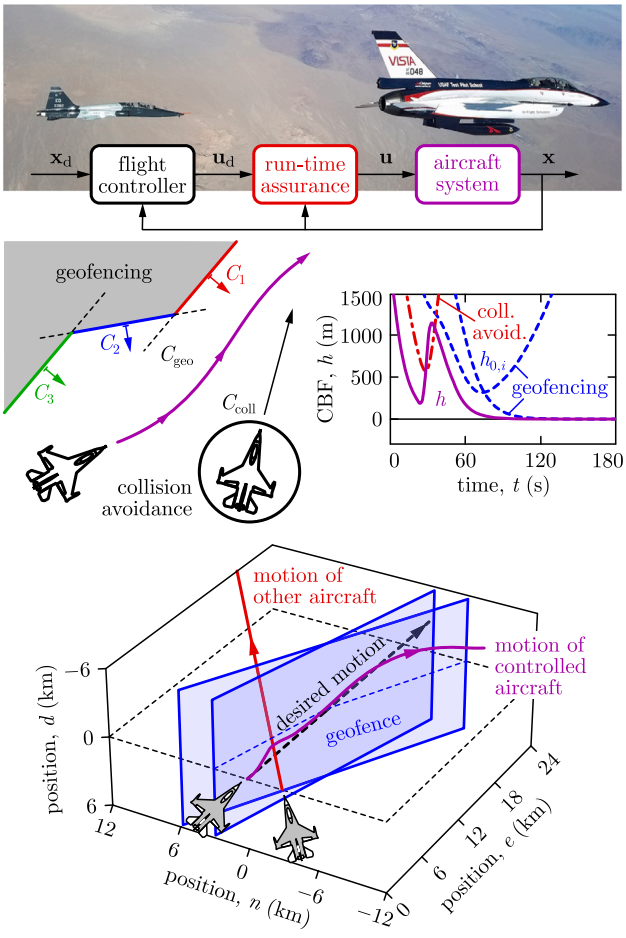

Fig. 11. Run-time assurance on fixed-wing aircraft to guarantee safety with respect to collision avoidance and geofencing. The results – repeated from Molnar et al. (2024) – demonstrate that safety-critical flight controllers, which use backstepping-based CBFs and leverage the multi-layer structure of the underlying dynamics, are able to generate maneuvers to prevent collision with other aircraft and entry into restricted airspace.

This structure can be exploited to synthesize a CBF via backstepping for use in collision avoidance and geofencing. For collision avoidance, consider the distance:

$$h_{0,i}(\mathbf{r}) = \|\mathbf{r} - \mathbf{r}_i\| - \rho_i,$$

between the controlled aircraft and multiple other aircraft with index i, whose position is $\mathbf{r}_i \in \mathbb{R}^3$, while $\rho_i > 0$ are collision radii. For geofencing, the distance between the aircraft and a planar geofence boundary with position $\mathbf{r}_i$ and normal vector $\mathbf{n}_i$ can be utilized:

$$h_{0,i}(\mathbf{r}) = \mathbf{n}_i^\top(\mathbf{r} - \mathbf{r}_i) - \rho_i,$$

where index i refers to multiple geofence constraints, that is, geofences with more complex geometry. These functions can be combined into a single CBF candidate and used to construct the CBF h via backstepping. This process takes multiple steps; the details are found in Molnar et al. (2024).

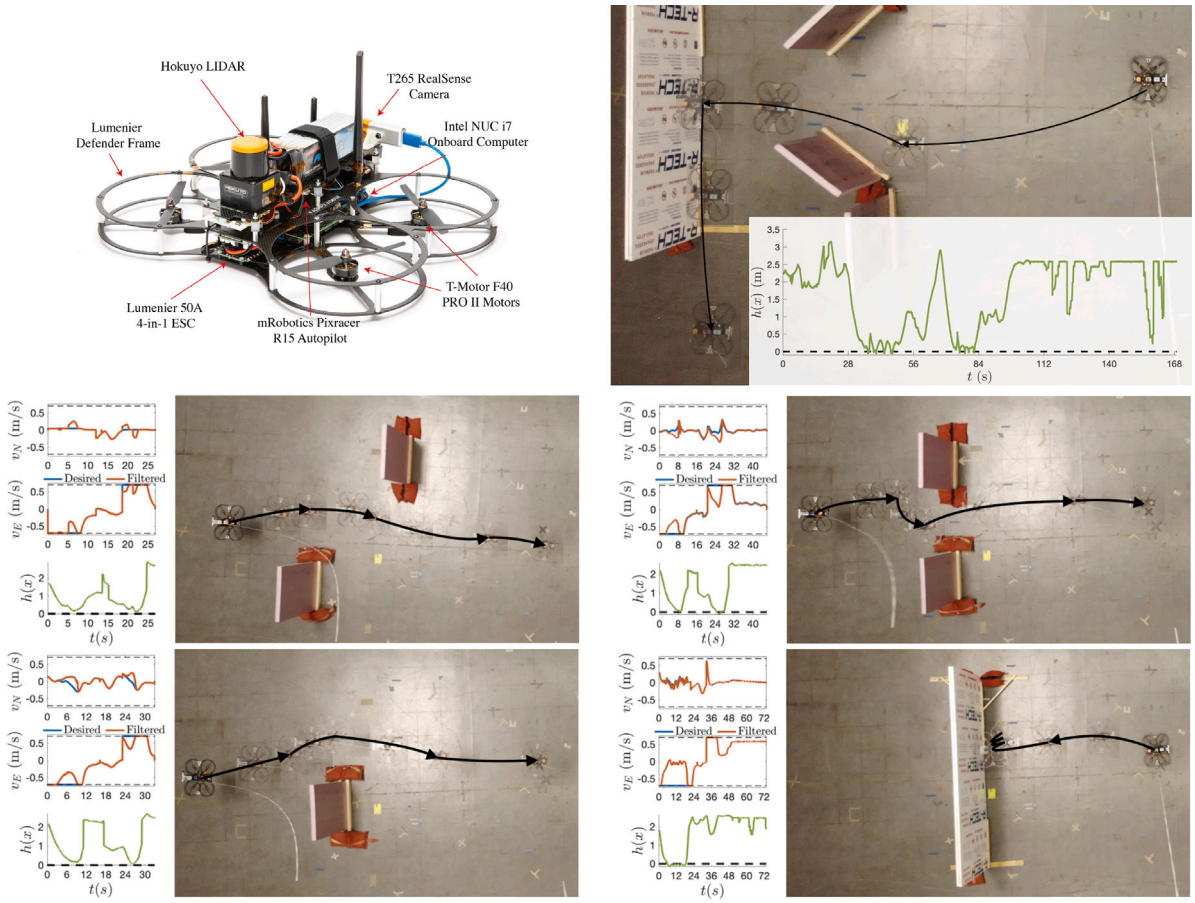

Fig. 12. Safety-critical indoor flight tests with a quadrotor (Singletary et al., 2021). The quadrotor is controlled to traverse obstacle courses with various obstacle arrangements while maintaining a collision-free flight. The single integrator is used as ROM for the quadrotor's dynamics, while the distance from the obstacle is considered as the CBF. By incorporating these into a safety filter, safe velocity commands are computed, which are then tracked by the onboard flight controller. The end result is collision-free motion in each scenario.

The CBF can be used in the QP-based controller (15) to achieve safety-critical behavior. The resulting motion is demonstrated in Fig. 11 by the simulation of simultaneous collision avoidance and geofencing scenario. The controlled aircraft seeks to track a straight trajectory, and its run-time assurance system intervenes to guarantee safety. The aircraft first accelerates, pitches up, and turns left to avoid collision with the other aircraft, and then it is forced to turn right to avoid crossing the two geofence boundaries. This behavior is generated by the backstepping-based CBF h, which was kept nonnegative throughout the motion. As a result, the three position-based CBF candidates $h_{0,i}$ are also kept nonnegative, which indicates that the underlying maneuvers are executed with guaranteed safety.

7.2. Safety-critical control of quadrotors

Next, we illustrate safe behavior on another important class of aircraft: quadrotors. We revisit the results of Singletary et al. (2021), where the techniques discussed in Section 6 were first demonstrated by hardware experiments on drones. The quadrotor shown in Fig. 12 was utilized in indoor flight tests to traverse obstacle fields with various obstacle arrangements (see bottom panels). In each scenario, the drone used an onboard flight controller to track velocity commands. To obtain these commands, first, a desired velocity was provided by a high-level desired controller. Then, using a single integrator as a ROM of the full quadrotor dynamics, a safety filter modified the desired velocity to a safe velocity command. The CBF underlying this safety filter was the distance between the quadrotor and the obstacle. Tracking of the resulting velocity resulted in collision-free flight, as the theory in Section 6 suggests.

Importantly, safety filters can also be implemented to prevent a human pilot from crashing a drone. The flight tests in Singletary et al. (2021) also demonstrated a case where a human was piloting the drone manually. These experimental results are shown in the top right panel of Fig. 12. Here, a human pilot provides the desired velocity commands for traversing the field such that the drone is actively driven *towards* the obstacles. Yet, even when the human pilot intends to hit the obstacles, the safety filter intervenes and prevents a collision. As such, human pilots usually provide high-level commands for robotic systems like this drone, hence a high-level safety filter – operating based on ROMs and CBFs – is suitable for keeping the system safe.

7.3. Safe flying, legged and wheeled robots

The control strategy discussed for quadrotors can be extended to a wide range of robotic systems. We demonstrate this by revisiting the results from (Molnar et al., 2022) where flying, legged, and wheeled robots were controlled via the same approach: stable tracking of safe ROMs. This approach leverages the fact that many robotic systems have multi-layer structures in their dynamics, where the top layer captures the relationship between the configuration and velocity of robots while the bottom layer relates velocities to forces or torques. As such, the top-level dynamics can be viewed as ROMs describing the evolution of the configuration. If safety is captured by a set C_0 in the configuration space (that is the case e.g. for collision avoidance), then CBFs for these ROMs can be used to find safe velocity commands, which can be tracked by existing on-board controllers that make the robot fly, walk or drive. This yields a simple method to guarantee safety of various

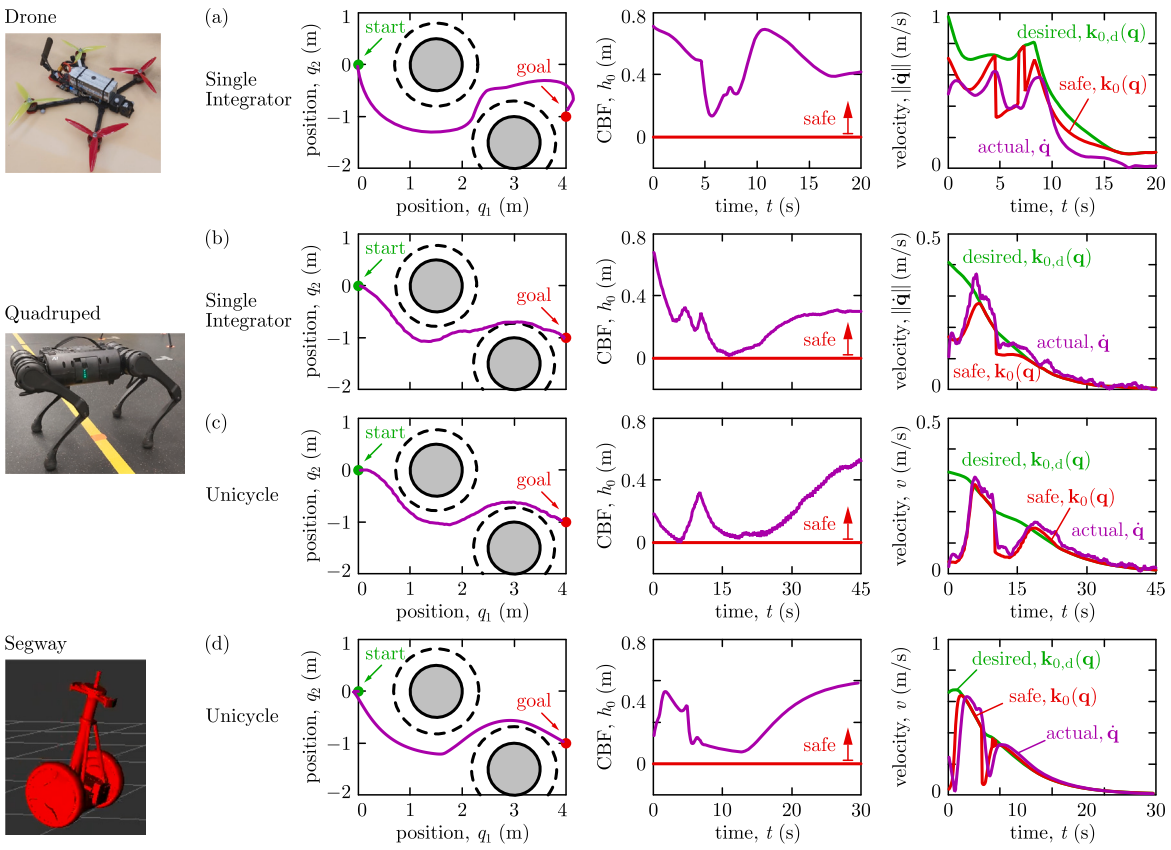

Fig. 13. Illustration of the model-free safety-critical control paradigm from Molnar et al. (2022). An obstacle avoidance task is executed on three fundamentally different systems: flying, legged, and wheeled robots. Each robot is controlled safely based on reduced-order (i.e., single integrator or unicycle) kinematics, by calculating safe velocity commands using CBFs and tracking these commands using on-board flight, walking, and driving controllers. (a) Hardware experiments on Drone, (b,c) hardware experiments on Quadruped, (d) high-fidelity simulations on Segway.

robots. Moreover, as was highlighted for quadrotors, the ROMs are often trivial equations with no parameters, like the single integrator in (35). Such ROMs lead to simple geometric expressions for the safe velocity, regardless of how complex the full model is. We refer to this approach as *model-free* safety-critical control.

The model-free safety-critical control paradigm is illustrated in Fig. 13. Three fundamentally different robots – a custom-built racing drone, a Unitree A1 quadruped, and a Ninebot E+ Segway – are controlled with the model-free approach to accomplish a reach-avoid task similar to that in Fig. 4. Using single integrator or unicycle reduced-order kinematics, CBF-based safe velocity expressions are computed for each robot, which are commanded as a reference signal to be tracked by the controller that flies the drone (established in Singletary, Swann, Chen and Ames (2022)), locomotes the quadruped (developed in Ubellacker, Csomay-Shanklin, Molnar, and Ames (2021)) and drives the Segway (described in Gurriet et al. (2020), Molnar et al. (2022)), respectively. The velocity tracking error, observed in the right panels, satisfies the bound (81), thus safety can be established according to Theorem 14. Indeed, safe behavior was observed in hardware experiments (drone and quadruped) and high-fidelity simulations (segway), as indicated by the positive value of the CBF h_0 of the reduced-order kinematics. Note that these results from Molnar et al. (2022) did not include the robustness term with ϵ in (74) (i.e., $\epsilon \to \infty$ was taken), hence a different variant of Theorem 14 with more restrictive assumptions was required to prove safety. We will highlight the relevance of robustness terms in the upcoming subsections where CBFs are used on industrial manipulators and heavy-duty vehicles.

7.4. CBFs in collaborative robotics

In the previous case study (Molnar et al., 2022), we demonstrated how ROMs may be used to develop safety-critical controllers for a variety of robotic systems, including legged robots. In the context of safe legged locomotion, this approach leveraged the system's existing control architecture, developed in Ubellacker et al. (2021), and allowed to control a rather complex robotic system by simply passing safe reference commands, generated by models such as a single integrator or unicycle, to the existing architecture. In the present case study, we further explore how CBFs may integrate into a system's overall autonomy stack in the context of collaborative legged locomotion (Kim, Lee et al., 2023) as portrayed in Fig. 14.

Here, the objective is for a team of holonomically constrained robots, in this case, a team of quadrupeds, to collaborate and safely navigate around obstacles before arriving at a goal location. These holonomic constraints could represent, for example, a payload that these robots seek to transport, which constrains the team's overall formation. To complete this task, the control architecture is broken down into three layers, each leveraging a more detailed model of the interconnected robotic system. The top layer represents each quadruped as a double integrator and leverages CBFs to simultaneously enforce the holonomic constraints and obstacle avoidance. The outputs of the top layer are thus safe position and velocity trajectories that also respect the holonomic constraints imposed on the full-order dynamics. The middle layer seeks to bridge the gap between these reduced-order trajectories and the full-order dynamics by representing the robotic team as an interconnection of single rigid bodies (SRBs). At this level, the outputs of the top layer are used as reference commands for the center of mass of each SRB, which are tracked by a model predictive controller that

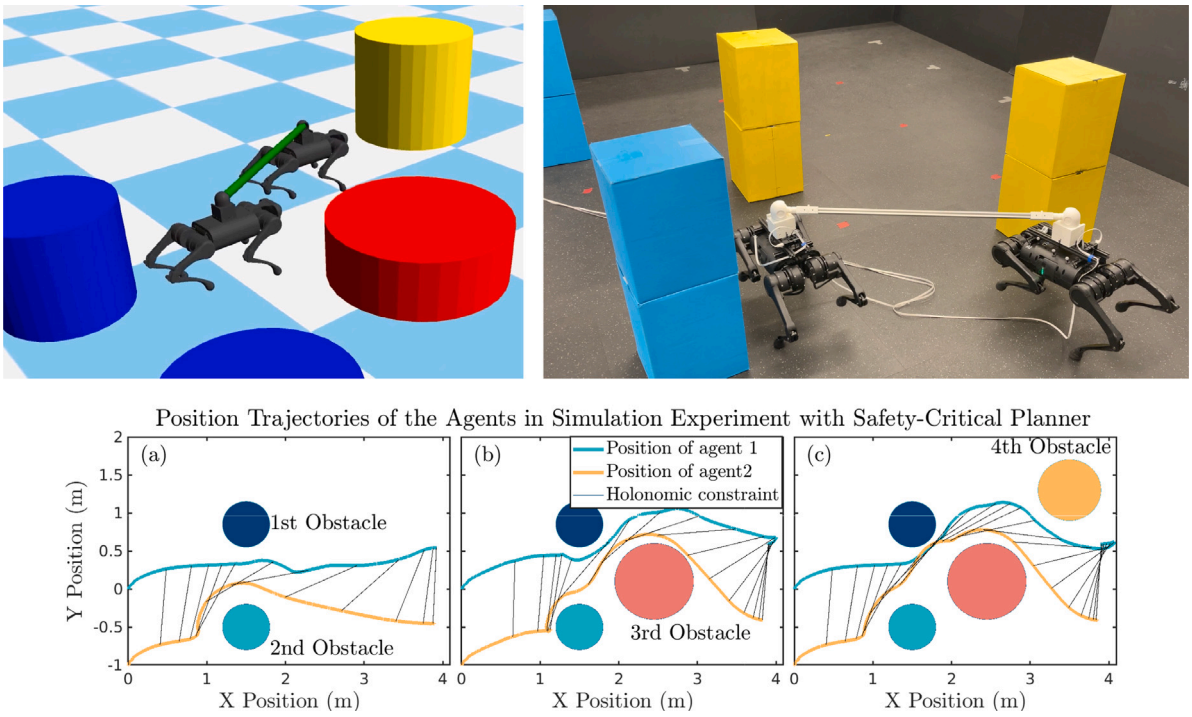

Fig. 14. Simulation and hardware results corresponding the to collaborative locomotion case study, originally reported in Kim, Lee et al. (2023).

outputs ground reaction forces (GRFs). These GRFs are input to the bottom layer, which leverages a high-fidelity model of each quadruped and a virtual constraint-based QP controller (Hamed, Kim, & Pandala, 2020; Kim, Fawcett, Ramidi, Ames & Hamed, 2023) to generate torque inputs that impose the commanded GRFs and track the safe position and velocity trajectories generated by higher layers.

The control architecture outlined above was implemented on a pair of Unitree A1 quadrupeds in both simulation and experimentally (Kim, Lee et al., 2023), where the objective is for a pair of interconnected quadrupeds to navigate around obstacles to a goal location. As shown in Fig. 14, in both simulation and hardware, the interconnected robotic system successfully navigates through simple (Fig. 14a) and cluttered environments (Fig. 14c). This is achieved by decomposing the control architecture into multiple layers and reasoning about both the system's holonomic constraints – representing the interconnection of the robots – and safety constraints at each layer using different model representations. Ultimately, this decomposition enables the implementation of safe and real-time collaborative locomotion.

7.5. Collision-free food preparation with manipulators

Next, we showcase the efficacy of utilizing CBFs and ROMs in the context of safe robotic manipulation. In particular, we present a real-world industrial application, reported in Singletary et al. (2022), wherein a manipulator is employed in a kitchen for automated food preparation that must be executed in a collision-free manner. The manipulator, shown in Fig. 15, is a Miso Robotics Flippy2 robot. This robot is intended to manipulate kitchen equipment in order to pick up, deep fry, and dispense food while avoiding collision with its environment. Executing such behaviors requires sophisticated motion plans, which are computed for various environmental factors and initial conditions. Many of the required motion plans are similar trajectories with only slight deviations, accounting for the fact that food baskets may move and deform slightly, workers may push the equipment, or the robot may have a slightly different initial configuration. Therefore, rather than replanning a trajectory in each slightly different situation, it is more efficient to use a CBF-based safety filter to modify a nominal trajectory online and provide formal safety guarantees.

Importantly, the manipulator has an efficient low-level control system that enables the tracking of trajectories and, in particular, velocity commands. Hence, this architecture is well-suited for utilizing the approach outlined in Section 6. Specifically, the kinematic equations of the robot can be used as a ROM to design safe velocity commands via CBF-based safety filters, which can be tracked by the low-level controller. Ensuring safety at the ROM level via velocity commands – rather than for the full dynamics by filtering the low-level controller – was also motivated by the fact that the details of the low-level controller were proprietary, and could not be modified. At the same time, the industrial low-level controller is well-designed for velocity tracking and capable of keeping the tracking error bounded as in (81). As established by Theorem 14, this enables safe behavior for the full dynamics by the appropriate choice of a ROM-based safety filter.

In particular, the work in Singletary et al. (2022) used the signed distance between the closest point of the robot and its environment as CBF candidate h_0, and implemented the safety filter:

$$\begin{aligned}\mathbf{k}_0(\mathbf{q},t) = \underset{\mathbf{v}\in\mathbb{R}^n}{\operatorname{argmin}} \quad & \|\mathbf{v}-\mathbf{k}_{0,\mathrm{d}}(\mathbf{q},t)\|^2 \\ \text{s.t.} \quad & \mathbf{n}(\mathbf{q})^\top \mathbf{J}(\mathbf{q})\mathbf{v} \geq -\alpha h_0(\mathbf{q}) + 2J_{\max}\dot{q}_{\max},\end{aligned}$$

that minimally modifies a desired velocity $\mathbf{k}_{0,\mathrm{d}}(\mathbf{q},t)$ given by a nominal motion plan to a safe velocity $\mathbf{k}_0(\mathbf{q},t)$. Here, safety is achieved by enforcing a CBF-based inequality constraint analogous to (74). The term on the left-hand side of this constraint is an approximation of the derivative of function h_0 along the kinematic ROM (with the Jacobian $\mathbf{J}$ and a normal vector $\mathbf{n}$), while the last term on the right-hand side is intended to provide robustness against disturbances and approximation errors (with the bounds $J_{\max}$ and $\dot{q}_{\max}$ on Jacobian and velocity norms). The resulting safe velocity was finally tracked by the robot's low-level controller to execute collision-free cooking.

The performance of the manipulator employing this control architecture is illustrated by hardware experiments in Fig. 15. The objective of the robot is to pick up a food basket that has finished cooking and move it from the fryer to a hanger, allowing the oil to drip off the basket before serving. Throughout this motion, the robot needs to operate in a dense workspace, where collision must be avoided with food baskets, fryers, the hood vent over the fryers, and a glass pane separating the

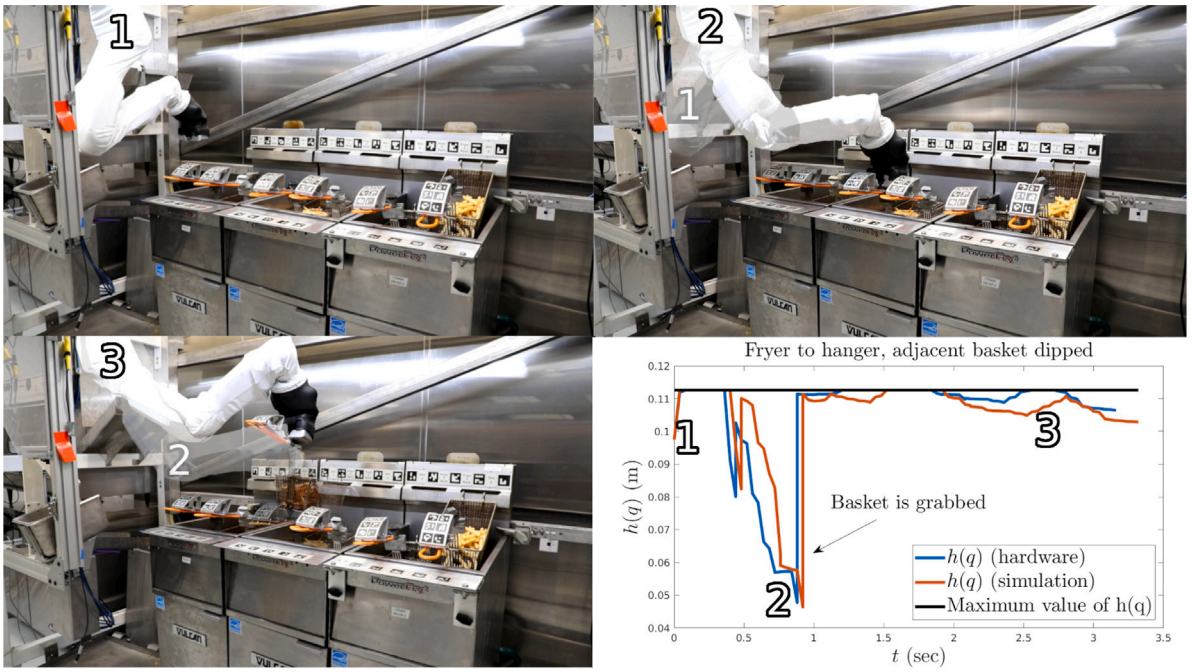

Fig. 15. Collision-free food preparation with a Flippy2 robot, with results from Singletary et al. (2022). Nominal motion plans that manipulate baskets of food are minimally modified using CBFs, in order to avoid collision between the robot and the kitchen equipment. Specifically, the reduced-order kinematics of the robot are used to synthesize a safe velocity using CBFs, which then were tracked by industrial low-level controllers.

manipulator from humans, leading to 36 collision objects in total. Although the manipulation is done in a tight space with a few centimeters of clearance between the robot and the surrounding environment, the manipulator manages to accomplish the task without collision, thanks to the use of a safety filter at the reduced-order kinematics level. This can be confirmed by the value of the underlying CBF candidate h_0, highlighted at the bottom right of Fig. 15, which is positive during the motion while its maximum value is only 11 centimeters. Importantly, the resulting behavior is reproducible: (Singletary et al., 2022) reported that the use of CBFs led to collision-free behavior consistently in 100 subsequent test cases.

7.6. Input-to-state safety on connected automated trucks

Finally, we demonstrate safety-critical control of heavy-duty vehicles as originally reported in Alan et al. (2023). Consider the connected automated truck in Fig. 16 that is controlled longitudinally to follow another vehicle on a straight road. Throughout the motion, the truck must maintain a safe distance to avoid front-end collision, which may be crucial in situations like emergency braking.

The truck is equipped with a low-level control system discussed in He et al. (2020) that regulates gas, brake pressure, and gear shifts to track acceleration commands. Thus, the truck's desired acceleration is viewed as a high-level control input, and double integrator models (or variants thereof, involving resistance terms and other physical effects) can be used as ROMs to control the truck's motion. For example, the following ROM was employed in Alan et al. (2023):

$$\begin{aligned}\dot{D} &= v_{\mathrm{L}} - v,\\ \dot{v} &= u + d,\\ \dot{v}_{\mathrm{L}} &= a_{\mathrm{L}},\end{aligned}$$

where $D \in \mathbb{R}$ is the distance of the vehicles, $v \in \mathbb{R}$ is the speed of the truck, $u \in \mathbb{R}$ is its desired acceleration, $d \in \mathbb{R}$ is a disturbance, $v_{\mathrm{L}} \in \mathbb{R}$ is the speed of the lead vehicle, and $a_{\mathrm{L}} \in \mathbb{R}$ is its acceleration. Furthermore, we have $\mathbf{q} = (D, v, v_{\mathrm{L}})$ and $\xi = u$ with our previous notations. Using the ROM, longitudinal car-following controllers can be designed at the acceleration level by measuring D, v, v_{L} and a_{L} using on-board range sensors like radar, as well as GPS and vehicle-to-vehicle connectivity.

With the estimated states, a desired *connected cruise controller* (Zhang & Orosz, 2016) can be utilized to execute car following:

$$k_{0,\mathrm{d}}(\mathbf{q}) = A(V(D) - v) + B(W(v_{\mathrm{L}}) - v),$$

where $A, B \in \mathbb{R}_{>0}$ are control gains, $V : \mathbb{R} \to \mathbb{R}$ is the range policy that provides a desired velocity based on the distance, and $W : \mathbb{R} \to \mathbb{R}$ is the speed policy that takes the speed limit into account. This desired controller can be incorporated into a CBF-based safety filter, where the CBF of the ROM:

$$h_0(\mathbf{q}) = D - \rho(v, v_{\mathrm{L}})$$

involves a safe distance expression that depends on the speeds as given by $\rho : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$. The corresponding safety filter generates safe acceleration commands, that can ultimately be tracked by the truck in order to maintain a safe distance. If the tracking error is bounded, this leads to safe behavior as highlighted by Theorem 14.

Importantly, accurate tracking of accelerations is challenging on heavy-duty trucks, since they have large inertia and response time, as well as complicated underlying dynamics in the engine, powertrain and brake systems. As a result, significant tracking errors inevitably occur that propagate as disturbance d to the ROM. This necessitates the use of safety-critical controllers that are robust to disturbances. Specifically, (Alan et al., 2023) leveraged the concept of *tunable input-to-state safety* proposed in Alan et al. (2022), and enforced:

$$L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \mathbf{k}_0(\mathbf{q}) \geq -\alpha h_0(\mathbf{q}) + \frac{\|L_{\mathbf{g}_0} h_0(\mathbf{q})\|^2}{\epsilon(h_0(\mathbf{q}))}, \tag{84}$$

as a constraint in QP-based safety filters. This constraint is a *tunable* counterpart of (74), where $\epsilon : \mathbb{R} \to \mathbb{R}_{>0}$ is a tunable function of h_0 to provide robustness near the boundary of the safe set only (while being less robust to disturbances when safety is not in danger of violation). The tunability facilitates reducing the conservativeness of the controller, to allow the truck to keep shorter distances.

The end result is shown in Fig. 16, which presents emergency braking experiments on a Navistar ProStar+ Class-8 truck as reported

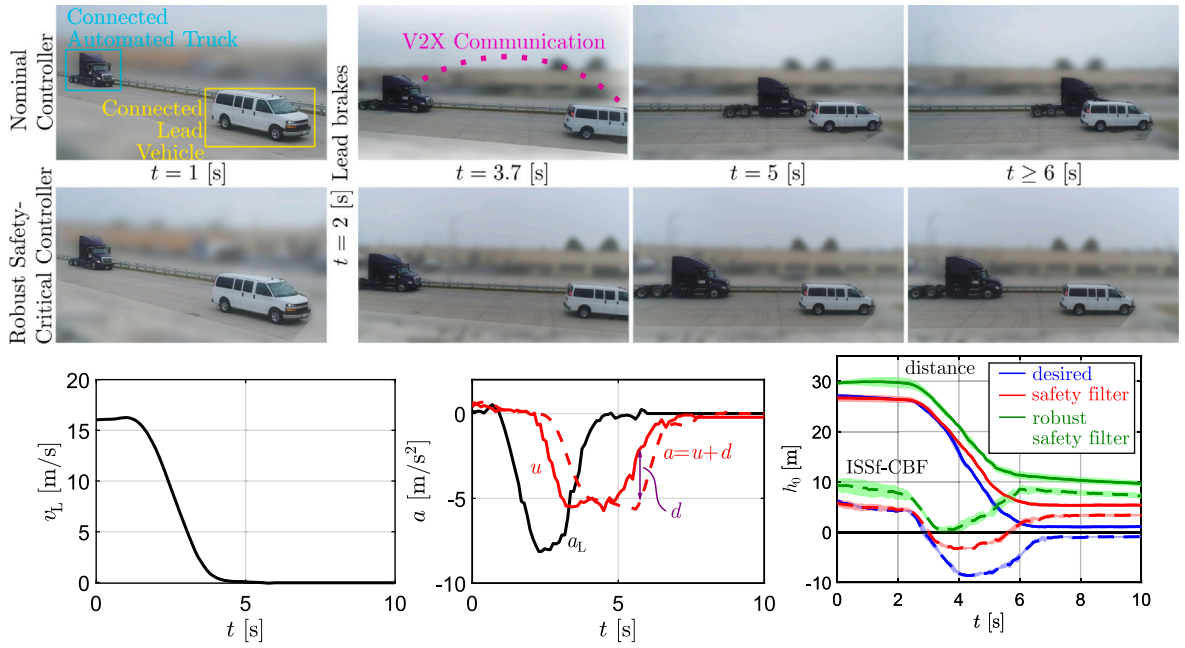

Fig. 16. Input-to-state safety on heavy-duty trucks in emergency braking. A connected automated truck is controlled to track acceleration commands designed in a safety-critical fashion using a double integrator as ROM. The tracking errors act as a significant disturbance, hence robust safety-critical controllers are required to guarantee safe behavior. By utilizing tunable input-to-state CBFs, proposed in Alan et al. (2022), for robust safety-critical control design, the truck safely executes the emergency braking maneuver without maintaining an overly conservative distance. Remarkably, this was not possible by traditional CBFs without added robustness.
Source: These results and figures have been adapted from Alan et al. (2023)

in Alan et al. (2023). The lead vehicle brakes to a full stop (black lines), and the truck responds to this event with various controllers (colored lines). The desired controller is unsafe during such a harsh maneuver (blue lines). Similarly, a safety filter that enforces (38) without a robustness term (i.e., without the term of ϵ), although performing better, still cannot maintain safety (red lines). This is due to the fact that the tracking of acceleration commands is imperfect and a significant disturbance arises (see purple arrow), while the underlying controller is not robust to disturbances. The robust safety-critical controller that enforces (84), on the other hand, successfully guarantees safety. This demonstrates the power of CBFs and ROMs in guaranteeing safe behavior on real-world systems and highlights that robustness against discrepancies between the ROM and the full system is crucial to achieving safety in practice.

8. Discussion and conclusions

Inspired by the success of reduced-order models in robotics, and the need for constructive techniques for CBFs, this paper presented a tutorial on using reduced-order models for safety-critical control. The core idea behind this methodology is to extend a CBF for a relatively simple system to a CBF for a complex system whose behavior, at a high level, is captured by its corresponding reduced-order model. We demonstrated different techniques, such as backstepping and Lyapunov-certified tracking, for constructing CBFs for relevant classes of control systems whose dynamics admit a particular layered structure. These systems include but are not limited to those encountered in robotics such as wheeled, legged, and flying robots. The central ideas of this approach were illustrated through theoretical results, numerical examples, and case studies that demonstrated the successful application of the ideas presented herein across various domains.

Although the methods covered in this tutorial provide a fairly general way to construct CBFs for relevant classes of systems, they also possess several limitations that should be investigated in future research. Perhaps the greatest limitation the approaches presented herein is that CBFs were synthesized under the assumption of unlimited control authority. In reality, any physical system will possess actuator limits and designing CBFs that take into account such limits is of paramount importance. Popular approaches to constructing CBFs that account for actuation limits include backup CBFs (Chen et al., 2021; Gurriet et al., 2018), input-constrained CBFs (Agrawal & Panagou, 2021), and integral CBFs (Ames, Notomista, Wardi, & Egerstedt, 2021), among others. It may be possible to unite the ideas presented herein with such methods to systematically synthesize CBFs for high-dimensional systems with actuation limits. Initial steps towards this unification have been presented in Molnar and Ames (2023b) wherein the methods introduced in Section 6 were combined with backup CBFs to develop safety-critical controllers based on reduced-order models that also account for actuation limits. Alternative approaches to accounting for actuation limits may involve the interplay between planning and control within a multi-rate framework (Csomay-Shanklin, Taylor, Rosolia, & Ames, 2022) in which trajectories of the reduced-order model are designed to be compatible with a lower-level controller with limited actuation authority.

Another question raised by the developments in this tutorial is: how does one choose a suitable reduced-order model? The results in Sections 4 and 5 (with the exception of Section 5.3) effectively require the full-order dynamics to be fully actuated, and demonstrate that, in such a situation, one may simply take the reduced-order model as a single integrator. The procedure in Section 5.3 demonstrates how CBFs may be constructed for underactuated systems under a certain set of assumptions, but falls far short of a complete characterization of synthesizing CBFs for underactuated systems. The challenges presented by underactuated systems are implicitly bypassed in Section 6 by assuming the existence of a low-level controller that tracks commands generated by a reduced-order model. However, the ability to construct such a controller will inevitably depend heavily on both the actuation capability of the system and on the richness of the reduced-order model. Fully characterizing when a reduced-order model is "good" in the sense that its behavior may be roughly replicated by the full-order dynamics is an important open question that deserves a more thorough investigation. We believe classical tools from nonlinear control theory (Isidori, 1995) such as the zero dynamics (Isidori, 2013),

virtual constraints (Hamed & Ames, 2020; Maggiore & Consolini, 2013; Westervelt, Grizzle, Chevallereau, Choi, & Morris, 2007), and output regulation (Di Benedetto & Grizzle, 1994; Grizzle, Di Benedetto, & Lamnabhi-Lagarrigue, 1994; Isidori & Byrnes, 1990) may play an important role in answering such questions.

While there are important theoretical questions that remain unanswered, the case studies presented in Section 7 indicate that the methods outlined in this tutorial tend to perform well in practice (i.e., when deployed on hardware) even when many of our standing assumptions, such as unlimited actuation capability, are violated. Ultimately, we believe developing principled approaches to handle such situations will only further improve the performance of the methods presented herein and facilitate their applications to a broader set of autonomous systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Linked to this paper is a Github repository containing all code used.

Acknowledgments

This work was funded in part by the National Science Foundation under grant number 1932091. We thank Pio Ong for the helpful discussions and Jeeseop Kim and Anil Alan for kindly providing the figures from Kim, Lee et al. (2023) and Alan et al. (2023).

Appendix. Proofs

Proof of Theorem 8. We leverage Lemma 2 to show that h as in (47) is a CBF for the corresponding control affine representation (45) of the mixed relative degree system (44). We begin by computing the gradient of h as:

$$\nabla h(\mathbf{x}) = \begin{bmatrix} \nabla h_0(\mathbf{q}) + \frac{1}{\mu}\frac{\partial \mathbf{k}_0^\xi}{\partial \mathbf{q}}(\mathbf{q})^\top(\xi - \mathbf{k}_0^\xi(\mathbf{q})) \\ -\frac{1}{\mu}(\xi - \mathbf{k}_0^\xi(\mathbf{q})). \end{bmatrix}$$

Thus, the Lie derivative of h along $\mathbf{g}$ as in (45) is:

$$L_{\mathbf{g}}h(\mathbf{x})^\top = \begin{bmatrix} L_{\mathbf{g}_0^\mathbf{u}}h_0(\mathbf{q}) + \frac{1}{\mu}\frac{\partial \mathbf{k}_0^\xi}{\partial \mathbf{q}}(\mathbf{q})^\top(\xi - \mathbf{k}_0^\xi(\mathbf{q}))\mathbf{g}_0^\mathbf{u}(\mathbf{q}) \\ -\frac{1}{\mu}(\xi - \mathbf{k}_0^\xi(\mathbf{q}))\mathbf{g}_1^\mathbf{u}(\mathbf{q},\xi). \end{bmatrix}$$

We now analyze the behavior of $\dot{h}$ when:

$$\begin{bmatrix} L_{\mathbf{g}_0^\mathbf{u}}h_0(\mathbf{q}) + \frac{1}{\mu}\frac{\partial \mathbf{k}_0^\xi}{\partial \mathbf{q}}(\mathbf{q})^\top(\xi - \mathbf{k}_0^\xi(\mathbf{q}))\mathbf{g}_0^\mathbf{u}(\mathbf{q}) \\ -\frac{1}{\mu}(\xi - \mathbf{k}_0^\xi(\mathbf{q}))\mathbf{g}_1^\mathbf{u}(\mathbf{q},\xi) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

It thus follows from the assumption that $\mathbf{g}_1^\mathbf{u}$ is pseudo-invertible and the second equation in the above system that, when $L_\mathbf{g}h(\mathbf{x}) = \mathbf{0}$, we must have $\xi - \mathbf{k}_0^\xi(\mathbf{q}) = \mathbf{0}$. It then follows from the first equation of the above system that, when $L_\mathbf{g}h(\mathbf{x}) = \mathbf{0}$, we must also have $L_{\mathbf{g}_0^\mathbf{u}}h_0(\mathbf{q}) = \mathbf{0}$. Now, computing the Lie derivative of h along $\mathbf{f}$ as in (45) when $L_\mathbf{g}h(\mathbf{x}) = \mathbf{0}$, we have:

$$\begin{aligned} L_\mathbf{f}h(\mathbf{x}) &= \begin{bmatrix} \nabla h_0(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{f}_0(\mathbf{q}) + \mathbf{g}_0^\xi(\mathbf{q})\xi \\ \mathbf{f}_1(\mathbf{q},\xi) \end{bmatrix} \\ &= L_{\mathbf{f}_0}h_0(\mathbf{q}) + L_{\mathbf{g}_0^\xi}h_0(\mathbf{q})\xi \\ &= L_{\mathbf{f}_0}h_0(\mathbf{q}) + L_{\mathbf{g}_0^\xi}h_0(\mathbf{q})\mathbf{k}_0^\xi(\mathbf{q}) \\ &> -\alpha(h_0(\mathbf{q})) - L_{\mathbf{g}_0^\mathbf{u}}h_0(\mathbf{q})\mathbf{k}_0^\mathbf{u}(\mathbf{q}) \\ &= -\alpha(h_0(\mathbf{q})) \\ &= -\alpha(h(\mathbf{x})), \end{aligned}$$

where the third line follows from $\xi = \mathbf{k}_0^\xi(\mathbf{q})$, the fourth from (46), the fifth from $L_{\mathbf{g}_0^\mathbf{u}}h_0(\mathbf{q}) = \mathbf{0}$, and the sixth from $h_0(\mathbf{q}) = h(\mathbf{x})$ (provided $L_\mathbf{g}h(\mathbf{x}) = \mathbf{0}$). It follows from Lemma 2 that h is a CBF for (45) on C as in (40). □

Proof of Theorem 9. We establish this result by showing that the function $h : TQ \to \mathbb{R}$ as defined in (51) satisfies the barrier-like inequality $\dot{h}(\mathbf{q},\dot{\mathbf{q}}) \geq -\alpha(h(\mathbf{q},\dot{\mathbf{q}}))$ for the closed-loop system, allowing one to invoke the comparison lemma (Khalil, 2002, Lemma 3.4) to establish forward invariance of C. To do so, we compute:

$$\dot{h}(\mathbf{q},\dot{\mathbf{q}}) = \dot{h}_0(\mathbf{q},\dot{\mathbf{q}}) - \frac{1}{\mu}\dot{V}(\mathbf{q},\dot{\mathbf{q}}),$$

noting that $\dot{h}_0(\mathbf{q},\dot{\mathbf{q}}) = \nabla h_0(\mathbf{q})\cdot\dot{\mathbf{q}}$ and:

$$\begin{aligned} \dot{V}(\mathbf{q},\dot{\mathbf{q}}) =& (\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\left[\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}}\right] \\ &+ \frac{1}{2}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\dot{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})) \\ =& -(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\Big[\mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\ &+ \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{k}(\mathbf{q},\dot{\mathbf{q}})\Big] \\ &+ \frac{1}{2}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\dot{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}})(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q})) \\ =& -(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\Big[\mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) \\ &+ \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{k}(\mathbf{q},\dot{\mathbf{q}})\Big], \end{aligned}$$

where the second equality follows from substituting in the dynamics (33) and the third from Property 1. Hence, $\dot{h}$ may be expressed as:

$$\begin{aligned} \dot{h}(\mathbf{q},\dot{\mathbf{q}}) =& \nabla h_0(\mathbf{q})\cdot\dot{\mathbf{q}} + \frac{1}{\mu}(\dot{\mathbf{q}} - \mathbf{k}_0(\mathbf{q}))^\top\Big[\mathbf{D}(\mathbf{q})\frac{\partial \mathbf{k}_0}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} \\ &+ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\mathbf{k}_0(\mathbf{q}) + \mathbf{G}(\mathbf{q}) - \mathbf{B}\mathbf{k}(\mathbf{q},\dot{\mathbf{q}})\Big] \\ \geq& -\alpha(h(\mathbf{q},\dot{\mathbf{q}})), \end{aligned}$$

where the inequality follows from (55). It then follows from the comparison lemma that $h(\mathbf{q}(t),\dot{\mathbf{q}}(t)) \geq h(\mathbf{q}_0,\dot{\mathbf{q}}_0)$ for all $t \in I(\mathbf{q}_0,\dot{\mathbf{q}}_0)$ so that if the system's initial condition satisfies $(\mathbf{q}_0,\dot{\mathbf{q}}_0) \in C$, then $h(\mathbf{q}(t),\dot{\mathbf{q}}(t)) \geq 0$ for all $t \in I(\mathbf{q}_0,\dot{\mathbf{q}}_0)$, implying the forward invariance of C. □

Proof of Theorem 10. We use an argument similar to Lemma 2 to show that h as defined in (66) is a CBF. We begin by computing the time derivative of h to obtain:

$$\begin{aligned} \dot{h}(\mathbf{x},\mathbf{u}) =& \nabla h_{0,1}(\mathbf{q}_1)\cdot\dot{\mathbf{q}}_1 + \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{D}}_1(\mathbf{q})\frac{\partial \mathbf{k}_{0,1}}{\partial \mathbf{q}_1}\dot{\mathbf{q}}_1 \\ &- \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{D}}_1(\mathbf{q})\ddot{\mathbf{q}}_1 \\ &- \frac{1}{2\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\dot{\bar{\mathbf{D}}}_1(\mathbf{q},\dot{\mathbf{q}})(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1)) \\ =& \nabla h_{0,1}(\mathbf{q}_1)\cdot\dot{\mathbf{q}}_1 + \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{D}}_1(\mathbf{q})\frac{\partial \mathbf{k}_{0,1}}{\partial \mathbf{q}_1}\dot{\mathbf{q}}_1 \\ &- \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\mathbf{B}_1\mathbf{u} \\ &+ \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{H}}_1(\mathbf{q},\dot{\mathbf{q}}) \\ &- \frac{1}{2\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\dot{\bar{\mathbf{D}}}_1(\mathbf{q},\dot{\mathbf{q}})(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1)). \end{aligned}$$

Collecting various terms in the above, we see that:

$$\begin{aligned} L_\mathbf{f}h(\mathbf{x}) =& \nabla h_{0,1}(\mathbf{q}_1)\cdot\dot{\mathbf{q}}_1 + \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{H}}_1(\mathbf{q},\dot{\mathbf{q}}) \\ &+ \frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\bar{\mathbf{D}}_1(\mathbf{q})\frac{\partial \mathbf{k}_{0,1}}{\partial \mathbf{q}_1}\dot{\mathbf{q}}_1 \\ &- \frac{1}{2\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\dot{\bar{\mathbf{D}}}_1(\mathbf{q},\dot{\mathbf{q}})(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1)) \\ L_\mathbf{g}h(\mathbf{x}) =& -\frac{1}{\mu}(\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top\mathbf{B}_1, \end{aligned}$$

where $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{f}$ and $\mathbf{g}$ are as in (34). Now, since $\mathbf{B}_1$ is pseudo-invertible, we have:

$$\begin{aligned} L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0} &\iff (\dot{\mathbf{q}}_1 - \mathbf{k}_{0,1}(\mathbf{q}_1))^\top \mathbf{B}_1 = \mathbf{0} \\ &\iff \dot{\mathbf{q}}_1 = \mathbf{k}_{0,1}(\mathbf{q}_1). \end{aligned}$$

Hence, when $L_{\mathbf{g}} h(\mathbf{x}) = \mathbf{0}$, we have:

$$\begin{aligned} L_{\mathbf{f}} h(\mathbf{x}) =& \nabla h_{0,1}(\mathbf{q}_1) \cdot \mathbf{k}_{0,1}(\mathbf{q}_1) \\ >& -\alpha(h_{0,1}(\mathbf{q}_1)) \\ =& -\alpha(h(\mathbf{q}, \dot{\mathbf{q}})), \end{aligned}$$

which implies that h is a CBF for (34). □

Proof of Theorem 12. Computing the time derivative of h yields:

$$\begin{aligned} \dot{h}(\mathbf{q}, \xi) =& \dot{h}_0(\mathbf{q}, \xi) - \frac{1}{\mu\gamma_1} \dot{V}(\mathbf{q}, \xi) \\ =& L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \xi - \frac{1}{\mu\gamma_1} \dot{V}(\mathbf{q}, \xi) \\ =& L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q})(\mathbf{k}_0(\mathbf{q}) + \mathbf{d}) - \frac{1}{\mu\gamma_1} \dot{V}(\mathbf{q}, \xi) \\ \geq& L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q})(\mathbf{k}_0(\mathbf{q}) + \mathbf{d}) + \frac{\gamma}{\mu\gamma_1} V(\mathbf{q}, \xi) \\ \geq& -\alpha h_0(\mathbf{q}) + \frac{1}{\varepsilon} \| L_{\mathbf{g}_0} h_0(\mathbf{q}) \|^2 \\ & - \| L_{\mathbf{g}_0} h_0(\mathbf{q}) \| \| \mathbf{d} \| + \frac{\gamma}{\mu\gamma_1} V(\mathbf{q}, \xi), \end{aligned}$$

where the first inequality follows from (71b) and the second from (74). After completing squares and further bounding $\dot{h}$, we have:

$$\begin{aligned} \dot{h}(\mathbf{q}, \xi) \geq& -\alpha h_0(\mathbf{q}) - \frac{\varepsilon}{4} \| \mathbf{d} \|^2 + \frac{\gamma}{\mu\gamma_1} V(\mathbf{q}, \xi) \\ \geq& -\alpha h_0(\mathbf{q}) - \frac{\varepsilon}{4\gamma_1} V(\mathbf{q}, \xi) + \frac{\gamma}{\mu\gamma_1} V(\mathbf{q}, \xi) \\ =& -\alpha h(\mathbf{q}, \xi) + \frac{1}{\mu\gamma_1} \left(\gamma - \alpha - \frac{\varepsilon\mu}{4} \right) V(\mathbf{q}, \xi), \end{aligned}$$

where the second inequality follows from (71a) and the final equality from (75). Hence, provided (77) holds, then:

$$\dot{h}(\mathbf{q}, \xi) \geq -\alpha h(\mathbf{q}, \xi),$$

implying h is a barrier function for (31) with $\mathbf{u} = \mathbf{k}(\mathbf{x})$ on C as in (76), which implies that C is forward invariant for the closed-loop system by Theorem 2. □

Proof of Theorem 14. Taking the time derivative of h from (82) yields:

$$\begin{aligned} \dot{h}(\mathbf{q}, \xi, t) =& \dot{h}_0(\mathbf{q}, \xi) + \frac{\gamma M}{\mu} e^{-\gamma t} \\ =& L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q}) \xi + \frac{\gamma M}{\mu} e^{-\gamma t} \\ =& L_{\mathbf{f}_0} h_0(\mathbf{q}) + L_{\mathbf{g}_0} h_0(\mathbf{q})(\mathbf{k}_0(\mathbf{q}) + \mathbf{d}) + \frac{\gamma M}{\mu} e^{-\gamma t}. \end{aligned}$$

Lower bounding the above using (74) yields:

$$\begin{aligned} \dot{h}(\mathbf{q}, \xi, t) \geq& -\alpha h_0(\mathbf{q}) + \frac{1}{\varepsilon} \| L_{\mathbf{g}_0} h_0(\mathbf{q}) \|^2 - \| L_{\mathbf{g}_0} h_0(\mathbf{q}) \| \| \mathbf{d} \| \\ & + \frac{\gamma M}{\mu} e^{-\gamma t}, \end{aligned}$$

which, after completing squares, may be further bounded as:

$$\dot{h}(\mathbf{q}, \xi, t) \geq -\alpha h_0(\mathbf{q}) - \frac{\varepsilon}{4} \| \mathbf{d} \|^2 + \frac{\gamma M}{\mu} e^{-\gamma t}.$$

It then follows from the above and the bound on $\mathbf{d}$ from (81) that:

$$\begin{aligned} \dot{h}(\mathbf{q}, \xi, t) \geq& -\alpha h_0(\mathbf{q}) - \frac{\varepsilon M}{4} e^{-\gamma t} - \frac{\varepsilon\delta}{4} + \frac{\gamma M}{\mu} e^{-\gamma t} \\ =& -\alpha h_0(\mathbf{q}) + \frac{M}{\mu} \left(\gamma - \frac{\varepsilon\mu}{4} \right) e^{-\gamma t} - \frac{\varepsilon\delta}{4}. \end{aligned}$$

Using the definition of h from (82), we then have:

$$\dot{h}(\mathbf{q}, \xi, t) \geq -\alpha h(\mathbf{q}, \xi, t) + \frac{M}{\mu} \left(\gamma - \alpha - \frac{\varepsilon\mu}{4} \right) e^{-\gamma t}.$$

Thus, provided (77) holds, then:

$$\dot{h}(\mathbf{q}, \xi, t) \geq -\alpha h(\mathbf{q}, \xi, t).$$

It then follows from the comparison lemma that $h(\mathbf{q}(t), \xi(t), t) \geq h(\mathbf{q}_0, \xi_0, 0)$ for all $t \in I(\mathbf{q}_0, \xi_0)$ so that if the system's initial condition satisfies $(\mathbf{q}_0, \xi_0) \in C(0)$, then $h(\mathbf{q}(t), \xi(t), t) \geq 0$ for all $t \in I(\mathbf{q}_0, \xi_0)$, implying the forward invariance of $C(t)$. □

References

Abel, I., Steeves, D., Krstić, M., & Janković, M. (2023). Prescribed-time safety design for strict-feedback nonlinear systems. *IEEE Transactions on Automatic Control*.

Abraham, R., Marsden, J. E., & Ratiu, T. (1983). *Manifolds, tensor analysis, and applications*. Addison-Wesley.

Agrawal, D., & Panagou, D. (2021). Safe control synthesis via input constrained control barrier functions. In *Proc. conf. decis. control* (pp. 6113–6118).

Agrawal, D. R., & Panagou, D. (2023). Safe and robust observer-controller synthesis using control barrier functions. *IEEE Control Systems Letters*, *7*, 127–132.

Alan, A., Taylor, A. J., He, C. R., Ames, A. D., & Orosz, G. (2023). Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Transactions on Control Systems Technology*, *31*(6), 2744–2759.

Alan, A., Taylor, A. J., He, C. R., Orosz, G., & Ames, A. D. (2022). Safe controller synthesis with tunable input-to-state safe control barrier functions. *IEEE Control Systems Letters*, *6*, 908–913.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control barrier functions: theory and applications. In *Proc. eur. control conf.* (pp. 3420–3431).

Ames, A. D., Grizzle, J. W., & Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. conf. decis. control* (pp. 6271–6278).

Ames, A. D., Notomista, G., Wardi, Y., & Egerstedt, M. (2021). Integral control barrier functions for dynamically defined control laws. *IEEE Control Systems Letters*, *5*(3), 887–892.

Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, *62*(8), 3861–3876.

Bansal, S., Chen, M., Herbert, S., & Tomlin, C. J. (2017). Hamilton-Jacobi reachability: A brief overview and recent advances. In *Proc. conf. decis. control* (pp. 2242–2253).

Belta, C., Yordanov, B., & Gol, E. A. (2017). *Formal methods for discrete-time dynamical systems*. Springer.

Blanchini, F., & Miani, S. (2008). *Set-theoretic methods in control*. Springer.

Bony, J. M. (1969). Principe du maximum, inègalite de harnack et unicité du probléme de Cauchy pour les opérateurs elliptiques dégénérés. *Annales de l'Institut Fourier, Grenoble*, *19*, 277–304.

Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press.

Bouligand, G. (1932). *Introducion a la geometrie infinitesimale directe*. Paris: Gauthiers-Villars.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Breeden, J., Garg, K., & Panagou, D. (2021). Control barrier functions in sampled-data systems. *IEEE Control Systems Letters*, *6*, 367–372.

Breeden, J., & Panagou, D. (2022). Predictive control barrier functions for online safety critical control. In *Proc. conf. decis. control* (pp. 924–931).

Breeden, J., & Panagou, D. (2023). Robust control barrier functions under high relative degree and input constraints for satellite trajectories. *Automatica*, *155*, Article 111109.

Brezis, H. (1970). On a characterization of flow-invariant sets. *Communications on Pure and Applied Mathematics*, *23*, 261–263.

Brunke, L., Zhou, S., & Schoellig, A. P. (2022). Barrier Bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems. In *Proc. conf. learning for dyn. and control* (pp. 881–892).

Chen, Y., Jankovic, M., Santillo, M., & Ames, A. D. (2021). Backup control barrier functions: Formulation and comparative study. In *Proc. conf. decis. control* (pp. 6835–6841).

Choi, J. J., Lee, D., Li, B., How, J. P., Sreenath, K., Herbert, S. L., et al. (2023). A forward reachability perspective on robust control invariance and discount factors in reachability analysis. arXiv preprint arXiv:2310.17180.

Choi, J. J., Lee, D., Sreenath, K., Tomlin, C. J., & Herbert, S. L. (2021). Robust control barrier–Value functions for safety-critical control. In *Proc. conf. decis. control* (pp. 6814–6821).

Clark, A. (2021). Verification and synthesis of control barrier functions. In *Proc. conf. decis. control* (pp. 6105–6112).

Clark, A. (2022). A semi-algebraic framework for verification and synthesis of control barrier functions. arXiv preprint arXiv:2209.00081.

Cohen, M. H., & Belta, C. (2023). *Adaptive and learning-based control of safety-critical systems*. Springer Nature.

Cohen, M. H., Ong, P., Bahati, G., & Ames, A. D. (2023). Characterizing smooth safety filters via the implicit function theorem. *IEEE Control Systems Letters*, *7*, 3890–3895.

Cohen, M. H., Serlin, Z., Leahy, K., & Belta, C. (2023). Temporal logic guided safe model-based reinforcement learning: a hybrid systems approach. *Nonlinear Analysis. Hybrid Systems*, *47*, Article 101295.

Cortez, W. S., & Dimarogonas, D. V. (2022). Safe-by-design control for Euler–Lagrange systems. *Automatica*, *146*, Article 110620.

Cortez, W. S., Oetomo, D., Manzie, C., & Choong, P. (2021). Control barrier functions for mechanical systems: Theory and application to robotic grasping. *IEEE Transactions on Control Systems Technology*, *29*(2), 530–545.

Cortez, W. S., Verginis, C. K., & Dimarogonas, D. V. (2021). Safe, passive control for mechanical systems with application to physical human-robot interactions. In *Proc. int. conf. robot. and autom.* (pp. 3836–3842).

Cosner, R. K., Culbertson, P., Taylor, A. J., & Ames, A. D. (2023). Robust safety under stochastic uncertainty with discrete-time control barrier functions. In *Robotics: science and syst.*.

Csomay-Shanklin, N., Taylor, A. J., Rosolia, U., & Ames, A. D. (2022). Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control Lyapunov functions. In *Proc. conf. decis. control* (pp. 3732–3739).

Dai, H., & Permenter, F. (2023). Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints. In *Proc. amer. control conf.* (pp. 4116–4123).

Dawson, C., Gao, S., & Fan, C. (2023). Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, *39*(3), 1749–1767.

Dawson, C., Qin, Z., Gao, S., & Fan, C. (2022). Safe nonlinear control using robust neural Lyapunov-barrier functions. In *Proc. conf. robot learn.*.

Dean, S., Taylor, A. J., Cosner, R. K., Recht, B., & Ames, A. D. (2020). Guaranteed safety of learned perception modules via measurement-robust control barrier functions. In *Proc. conf. robot learn.*.

Dhiman, V., Khojasteh, M. J., Franceschetti, M., & Atanasov, N. (2023). Control barriers in Bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, *68*(1), 214–229.

Di Benedetto, M. D., & Grizzle, J. W. (1994). Asymptotic model matching for nonlinear systems. *IEEE Transactions on Automatic Control*, *39*(8), 1539–1550.

Freeman, R. A., & Kokotović, P. V. (1992). Backstepping design of robust controllers for a class of nonlinear systems. *IFAC Proceedings Volumes*, *25*(13), 431–436. http://dx.doi.org/10.1016/S1474-6670(17)52320-4.

Garg, K., & Panagou, D. (2021). Robust control barrier and control Lyapunov functions with fixed-time convergence guarantees. In *Proc. amer. control conf.* (pp. 2292–2297).

Ghaffari, A., Abel, I., Ricketts, D., Lerner, S., & Krstić, M. (2018). Safety verification using barrier certificates with application to double integrator with input saturation and zero-order hold. In *Proc. amer. control conf.* (pp. 4664–4669).

Glotfelter, P., Cortés, J., & Egerstedt, M. (2017). Nonsmooth barrier functions with applications to multi-robot systems. *IEEE Control Systems Letters*, *1*(2), 310–315.

Glotfelter, P., Cortés, J., & Egerstedt, M. (2020). A nonsmooth approach to controller synthesis for Boolean specifications. *IEEE Transactions on Automatic Control*, *66*(11), 5160–5174.

Grizzle, J. W., Di Benedetto, M. D., & Lamnabhi-Lagarrigue, F. (1994). Necessary conditions for asymptotic tracking in nonlinear systems. *IEEE Transactions on Automatic Control*, *39*(9), 1782–1794.

Gurriet, T., Mote, M., Singletary, A., Nilsson, P., Feron, E., & Ames, A. D. (2020). A scalable safety critical control framework for nonlinear systems. *IEEE Access*, *8*, 187249–187275.

Gurriet, T., Singletary, A., Reher, J., Ciarletta, L., Feron, E., & Ames, A. (2018). Towards a framework for realizable safety critical control through active set invariance. In *Proc. ACM/IEEE int. conf. cyber-physical syst.* (pp. 98–106).

Hamed, K. A., & Ames, A. D. (2020). Nonholonomic hybrid zero dynamics for the stabilization of periodic orbits: Application to underactuated robotic walking. *IEEE Transactions on Control Systems Technology*, *28*(6), 2689–2696.

Hamed, K. A., Kim, J., & Pandala, A. (2020). Quadrupedal locomotion via event-based predictive control and QP-based virtual constraints. *IEEE Robotics and Automation Letters*, *5*(3), 4463–4470.

He, C. R., Alan, A., Molnár, T. G., Avedisov, S. S., Bell, A. H., Zukouski, R., et al. (2020). Improving fuel economy of heavy-duty vehicles in daily driving. In *Proc. amer. control conf.* (pp. 2306–2311). http://dx.doi.org/10.23919/ACC45564.2020.9147970.

Hewing, L., Wabersich, K. P., Menner, M., & Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*, 269–296.

Isidori, A. (1995). *Nonlinear control systems* (3rd e.d). Springer.

Isidori, A. (2013). The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story. *European Journal of Control*, *19*, 369–378.

Isidori, A., & Byrnes, C. (1990). Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control*, *35*(2), 131–140.

Jankovic, M. (2018). Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, *96*, 359–367.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., & Hirukawa, H. (2002). A realtime pattern generator for biped walking. In *Proc. int. conf. robot. and autom.*.

Khalil, H. (2002). *Nonlinear systems* (3rd ed.). Prentice Hall.

Kim, J., Fawcett, R. T., Ramidi, V. R., Ames, A. D., & Hamed, K. A. (2023). Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches. *IEEE Transactions on Robotics*, *39*(6), 4728–4748.

Kim, J., Lee, J., & Ames, A. D. (2023). Safety-critical coordination for cooperative legged locomotion via control barrier functions. In *Proc. IEEE/RSJ int. conf. int. robot. and syst.* (pp. 2368–2375).

Kolathaya, S., & Ames, A. D. (2019). Input-to-state safety with control barrier functions. *IEEE Control Systems Letters*, *3*(1), 108–113.

Konda, R., Ames, A. D., & Coogan, S. (2021). Characterizing safety: Minimal control barrier functions from scalar comparison systems. *IEEE Control Systems Letters*, *5*(2), 523–528.

Krstić, M., & Bement, M. (2006). Nonovershooting control of strict-feedback nonlinear systems. *IEEE Transactions on Automatic Control*, *51*(12), 1938–1943.

Krstić, M., Kanellakopoulus, I., & Kokotović, P. (1995). *Nonlinear and adaptive control design*. Wiley.

Lindemann, L., & Dimarogonas, D. V. (2019). Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, *3*(1), 96–101.

Lindemann, L., Hu, H., Robey, A., Zhang, H., Dimorogonas, D. V., Tu, S., et al. (2020). Learning hybrid control barrier functions from data. In *Proc. conf. robot learn.*.

Long, L., & Wang, J. (2022). Safety-critical dynamic event-triggered control of nonlinear systems. *Systems & Control Letters*, *162*, Article 105176.

Lopez, B. T., Slotine, J. J., & How, J. P. (2021). Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, *5*(3), 1031–1036.

Luca, A. D., Oriolo, G., & Vendittelli, M. (2001). *Lecture notes in control and information sciences*. Springer.

Maggiore, M., & Consolini, L. (2013). Virtual holonomic constraints for Euler–Lagrange systems. *IEEE Transactions on Automatic Control*, *58*(4), 1001–1008.

Mitchell, I. M., Bayen, A. M., & Tomlin, C. J. (2005). A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, *50*(7), 947–957.

Molnar, T. G., & Ames, A. D. (2023a). Composing control barrier functions for complex safety specifications. *IEEE Control Systems Letters*, *7*, 3615–3620.

Molnar, T. G., & Ames, A. D. (2023b). Safety-critical control with bounded inputs via reduced order models. In *Proc. amer. control conf.* (pp. 1414–1421).

Molnar, T. G., Cosner, R. K., Singletary, A. W., Ubellacker, W., & Ames, A. D. (2022). Model-free safety-critical control for robotic systems. *IEEE Robotics and Automation Letters*, *7*(2), 944–951.

Molnar, T. G., Kannan, S. K., Cunningham, J., Dunlap, K., Hobbs, K. L., & Ames, A. D. (2024). Collision avoidance and geofencing for fixed-wing aircraft with control barrier functions. arXiv preprint arXiv:2403.02508.

Nagumo, M. (1942). Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, *24*, 551–559.

Nguyen, Q., & Sreenath, K. (2016). Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *Proc. amer. control conf.* (pp. 322–328).

Ong, P., & Cortes, J. (2019). Universal formula for smooth safe stabilization. In *Proc. conf. decis. control* (pp. 2373–2378).

Polyakov, A., & Krstic, M. (2023). Finite-and fixed-time nonovershooting stabilizers and safety filters by homogeneous feedback. *IEEE Transactions on Automatic Control*, *68*(11), 6434–6449.

Raibert, M. (1986). *Legged robots that balance*. MIT Press.

Redheffer, R. M. (1972). The theorems of Bony and Brezis on flow-invariant sets. *American Mathematical Monthly*, *79*(7), 740–747.

Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimorogonas, D. V., Tu, S., et al. (2020). Learning control barrier functions from expert demonstrations. In *Proc. conf. decis. control* (pp. 3717–3724).

Santoyo, C., Dutreix, M., & Coogan, S. (2021). A barrier function approach to finite-time stochastic system verification and control. *Automatica*, *125*, Article 109439.

Sepulchre, R., Janković, M., & Kokotović, P. (1997). *Constructive nonlinear control*. Springer.

Singletary, A. W., Guffey, W., Molnar, T., Sinnet, R., & Ames, A. D. (2022). Safety-critical manipulation for collision-free food preparation. *IEEE Robotics and Automation Letters*, *7*(4), 10954–10961.

Singletary, A. W., Klingebiel, K., Bourne, J., Browning, A., Tokumaru, P., & Ames, A. D. (2021). Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *Proc. IEEE/RSJ int. conf. int. robot. and syst.* (pp. 8129–8136).

Singletary, A., Kolathaya, S., & Ames, A. D. (2022). Safety-critical kinematic control of robotic systems. *IEEE Control Systems Letters*, *6*, 139–144.

Singletary, A., Swann, A., Chen, Y., & Ames, A. D. (2022). Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions. *IEEE Robotics and Automation Letters*, *7*(2), 2897–2904.

So, O., Serlin, Z., Mann, M., Gonzales, J., Rutledge, K., Roy, N., et al. (2023). How to train your neural control barrier function: Learning safety filters for complex input-constrained systems. arXiv preprint arXiv:2310.15478.

Sontag, E. D. (1989). A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, *13*(2), 117–123.

Spong, M. (1994). Partial feedback linearization of underactuated mechanical systems. In *Proc. IEEE/RSJ int. conf. int. robot. and syst.*.

Srinivasan, M., & Coogan, S. (2021). Control of mobile robots using barrier functions under temporal logic specifications. *IEEE Transactions on Robotics*, *37*(2), 363–374.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Spring Science & Business Media.

Tan, X., Cortez, W. S., & Dimarogonas, D. V. (2022). High-order barrier functions: robustness, safety and performance-critical control. *IEEE Transactions on Automatic Control*, *67*(6), 3021–3028.

Taylor, A. J., & Ames, A. D. (2020). Adaptive safety with control barrier functions. In *Proc. amer. control conf.* (pp. 1399–1405).

Taylor, A. J., Dorobantu, V. D., Cosner, R. K., Yue, Y., & Ames, A. D. (2022). Safety of sampled-data systems with control barrier functions via approximate discrete time models. In *Proc. conf. decis. control* (pp. 7127–7134).

Taylor, A. J., Ong, P., Cortés, J., & Ames, A. D. (2021). Safety-critical event triggered control via input-to-state safe barrier functions. *IEEE Control Systems Letters*, *5*(3), 749–754.

Taylor, A. J., Ong, P., Molnar, T. G., & Ames, A. D. (2022). Safe backstepping with control barrier functions. In *Proc. conf. decis. control* (pp. 5775–5782).

Taylor, A. J., Singletary, A., Yue, Y., & Ames, A. (2020). Learning for safety-critical control with control barrier functions. In *Proceedings of machine learning research*: *Vol. 120*, *Proc. conf. learning for dyn. and control* (pp. 708–717).

Tedrake, R. (2023). *Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation*. course notes for MIT 6.832, https://underactuated.csail.mit.edu.

Tonkens, S., & Herbert, S. (2022). Refining control barrier functions through Hamilton-Jacobi reachability. In *Proc. IEEE/RSJ int. conf. int. robot. and syst.* (pp. 13355–13362).

Ubellacker, W., Csomay-Shanklin, N., Molnar, T. G., & Ames, A. D. (2021). Verifying safe transitions between dynamic motion primitives on legged robots. In *Proc. IEEE/RSJ int. conf. int. robot. and syst.* (pp. 8477–8484). http://dx.doi.org/10.1109/IROS51168.2021.9636537.

Usevitch, J., Garg, K., & Panagou, D. (2020). Strong invariance using control barrier functions: A Clarke tangent cone approach. In *Proc. conf. decis. control* (pp. 2044–2049).

Wabersich, K. P., Taylor, A. J., Choi, J. J., Sreenath, K., Tomlin, C. J., Ames, A. D., et al. (2023). Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, *43*(5), 137–177.

Wabersich, K., & Zeilinger, M. (2022). Predictive control barrier functions: Enhanced safety mechanisms for learning-based control. *IEEE Transactions on Automatic Control*, *68*(5).

Wang, Y., & Xu, X. (2022). Observer-based control barrier functions for safety critical systems. In *Proc. amer. control conf.* (pp. 709–714).

Westervelt, E. R., Grizzle, J. W., Chevallereau, C., Choi, J., & Morris, B. (2007). *Feedback control of dynamic bipedal robot locomotion*. CRC Press.

Xiao, W., & Belta, C. (2019). Control barrier functions for systems with high relative degree. In *Proc. conf. decis. control* (pp. 474–479).

Xiao, W., & Belta, C. (2022). High order control barrier functions. *IEEE Transactions on Automatic Control*, *67*(7), 3655–3662.

Xiao, W., Belta, C., & Cassandras, C. G. (2023). Event-triggered control for safety-critical systems with unknown dynamics. *IEEE Transactions on Automatic Control*, *68*(7), 4143–4158.

Xiao, W., Cassandras, C. G., & Belta, C. (2023). *Safe autonomy with control barrier functions: Theory and applications*. Springer Nature.

Xiong, X., & Ames, A. D. (2022). 3-d underactuated bipedal walking via H-LIP based gait synthesis and stepping stabilization. *IEEE Transactions on Robotics*, *38*(4), 2405–2425.

Xu, X., Tabuada, P., Grizzle, J. W., & Ames, A. D. (2015). Robustness of control barrier functions for safety critical control. In *Proc. IFAC conf. on analysis and design of hybrid syst.* (pp. 54–61).

Yang, G., Belta, C., & Tron, R. (2019). Self-triggered control for safety critical systems using control barrier functions. In *Proc. amer. control conf.* (pp. 4454–4459).

Zhang, L., & Orosz, G. (2016). Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays. *IEEE Transactions on Intelligent Transportation Systems*, *17*(6), 1638–1651. http://dx.doi.org/10.1109/TITS.2015.2509782.

Zhao, P., Ghabcheloo, R., Cheng, Y., Abdi, H., & Hovakimyan, N. (2023). Convex synthesis of control barrier functions under input constraints. *IEEE Control Systems Letters*, *7*, 3102–3107.

Zhao, S., & Sun, Z. (2017). Defend the practicality of single-integrator models in multi-robot coordination control. In *Proc. int. conf control autom.* (pp. 666–671).