

Safety-Critical Control with Bounded Inputs via Reduced Order Models

Tamas G. Molnar and Aaron D. Ames

Abstract—Guaranteeing safe behavior on complex autonomous systems—from cars to walking robots—is challenging due to the inherently high dimensional nature of these systems and the corresponding complex models that may be difficult to determine in practice. With this as motivation, this paper presents a safety-critical control framework that leverages reduced order models to ensure safety on the full order dynamics—even when these models are subject to disturbances and bounded inputs (e.g., actuation limits). To handle input constraints, the backup set method is reformulated in the context of reduced order models, and conditions for the provably safe behavior of the full order system are derived. Then, the input-to-state safe backup set method is introduced to provide robustness against discrepancies between the reduced order model and the actual system. Finally, the proposed framework is demonstrated in high-fidelity simulation, where a quadrupedal robot is safely navigated around an obstacle with legged locomotion by the help of the unicycle model.

I. INTRODUCTION

Real-life engineering systems often exhibit complicated, nonlinear and high-dimensional dynamic behavior. This is especially true of autonomous (robotic) systems, where dynamics play a key role in achieving desired behaviors. This makes them challenging to control, and to attain formal guarantees of stable or safe evolution for the closed control loop. To tackle such complex control problems, simplified, reduced order models (ROMs) of the dynamic behavior are often utilized during controller synthesis with great practical success [1], [2]. Yet there is often a theoretic gap between behaviors certifiable on the ROM and the resulting behaviors observed on the full order system (FOS).

In this paper, we focus on the role of ROMs in safety-critical control. Given an accurate dynamical model, there exist tools to synthesize controllers that provide formal guarantees of safety. For example, *control barrier functions* (CBFs) [3] have been proposed to achieve this goal, and they have been proven to be successful in a wide variety of applications from multi-robot systems [4] to spacecraft docking [5]. In many applications, a significant challenge is maintaining safety with limited actuation: most physical systems have finite actuation capability, which manifests itself in the underlying models as constraints on the control input. Several methods have been proposed for input constrained safety-critical control, including the backup set method [6], input constrained CBFs [7], and neural CBFs [8]. While these

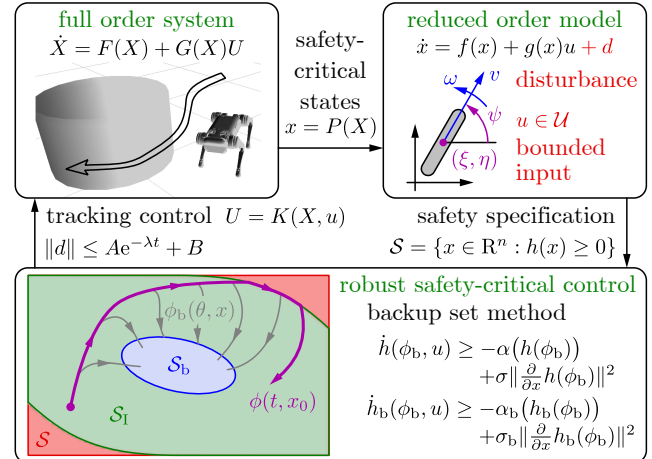


Fig. 1. Overview of the proposed safety-critical control framework.

approaches have shown success in various domains, a general approach remains elusive.

Another important challenge in safety-critical control is that no ROM is ever fully accurate: there is always discrepancy between the ROM and the actual FOS. Consequently, robustness is of key importance, and one needs to ensure that safety is preserved even under such discrepancies—and with limited actuation. There exist CBF formulations that provide robustness against disturbances, such as the approaches of input-to-state safety [9], [10] and robust CBFs [11]. These formulations, however, have not yet accommodated input constraints. Meanwhile, the above approaches that address input constraints have not yet been endowed with robustness. On the other hand, there exist robust reachability approaches that handle both input constraints and disturbances [12], [13], [14], [15], but these methods typically suffer from the curse of dimensionality and become intractable for higher dimensional ROMs.

This paper presents a robust safety-critical control framework, illustrated in Fig. 1, wherein input constrained ROMs and CBFs are leveraged to achieve formal safety guarantees on systems with complex full order dynamics. To this end, we make the following three key contributions. First, the backup set method is reformulated in the context of ROMs, and conditions for provably safe behavior are given that account for the discrepancy between the ROM and the FOS that tracks it. Second, the *input-to-state safe backup set method* is introduced to provide robustness against the discrepancy with less restrictive conditions. Third, the method is implemented in the context of an obstacle avoidance problem, wherein safe walking on a quadrupedal robot using the unicycle ROM is demonstrated in high-fidelity simulation.

*This research is supported in part by the National Science Foundation, CPS Award #1932091, Dow (#227027AT) and Aerovironment.

The Authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, USA. {tmolnar, ames}@caltech.edu

II. SAFETY UNDER INPUT CONSTRAINTS

Consider the control-affine system:

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

with state $x \in \mathbb{R}^n$, input $u \in \mathcal{U}$, convex admissible input set $\mathcal{U} \subseteq \mathbb{R}^m$, and locally Lipschitz continuous functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$. Consider a controller $k : \mathbb{R}^n \rightarrow \mathcal{U}$, $u = k(x)$, that yields the closed control loop:

$$\dot{x} = f(x) + g(x)k(x), \quad (2)$$

associated with the initial condition $x(0) = x_0 \in \mathbb{R}^n$. If k is locally Lipschitz continuous, the closed-loop system has a unique solution $\phi(t, x_0)$ over an interval of existence. For simplicity, we assume that the solution exists for all $t \geq 0$.

Our goal is to design the controller k such that the closed-loop system is safe. Specifically, we consider the system to be safe if its state x is located within a safe set $\mathcal{S} \subset \mathbb{R}^n$. For the safe evolution of the closed control loop, we require the forward invariance of the safe set \mathcal{S} along (2).

Definition 1. Given $k : \mathbb{R}^n \rightarrow \mathcal{U}$, set $\mathcal{S} \subset \mathbb{R}^n$ is *forward invariant* along (2) if $x_0 \in \mathcal{S} \implies \phi(t, x_0) \in \mathcal{S}, \forall t \geq 0$.

This requirement can be met only if \mathcal{S} is control invariant.

Definition 2. Set $\mathcal{S} \subset \mathbb{R}^n$ is *control invariant* if there exists $k : \mathbb{R}^n \rightarrow \mathcal{U}$ such that \mathcal{S} is forward invariant along (2).

A. Control Barrier Functions

Control barrier functions [3] provide a powerful tool for safe control design, hence we briefly revisit this method. Throughout the paper, we consider safe sets defined as the 0-superlevel set of a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\mathcal{S} = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \quad (3)$$

such that h is continuously differentiable and zero is a regular value of h , i.e., $h(x) = 0 \implies \nabla h(x) \neq 0$.

Definition 3. Function h is a *control barrier function (CBF)* for (1) on \mathcal{S} if there exists $\alpha \in \mathcal{K}_\infty$ such that¹:

$$\sup_{u \in \mathcal{U}} \dot{h}(x, u) > -\alpha(h(x)) \quad (4)$$

holds $\forall x \in \mathcal{S}$, where:

$$\dot{h}(x, u) = \nabla h(x)(f(x) + g(x)u). \quad (5)$$

Given a CBF, [3] established the following safety result.

Theorem 1 ([3]). *If h is a CBF for (1) on \mathcal{S} , then any locally Lipschitz continuous controller $k : \mathbb{R}^n \rightarrow \mathcal{U}$ satisfying:*

$$\dot{h}(x, k(x)) \geq -\alpha(h(x)) \quad (6)$$

$\forall x \in \mathcal{S}$ renders \mathcal{S} forward invariant along (2).

Condition (6) can be used as constraint when synthesizing safe controllers. For example, given a desired controller

¹Function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class- \mathcal{K}_∞ ($\alpha \in \mathcal{K}_\infty$) if it is continuous, $\alpha(0) = 0$ and $\lim_{r \rightarrow \infty} \alpha(r) = \infty$. Note that extended class- \mathcal{K}_∞ functions defined over \mathbb{R} are also used to ensure the attractivity of the safe set.

$k_d : \mathbb{R}^n \rightarrow \mathcal{U}$, the following quadratic program-based controller can be used for safety-critical control:

$$k(x) = \underset{u \in \mathcal{U}}{\operatorname{argmin}} \|u - k_d(x)\|_\Gamma^2 \quad (7)$$

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

where $\|u\|_\Gamma^2 = u^\top \Gamma u$ and $\Gamma \in \mathbb{R}^{m \times m}$ is a positive definite weight matrix that can be tuned.

B. Backup Set Method

While CBFs provide safe behavior, it is nontrivial to verify that a certain choice of h is indeed a CBF satisfying (4), especially with bounded inputs ($\mathcal{U} \subset \mathbb{R}^m$). An arbitrary h may not have control invariant 0-superlevel set, it may not be a CBF, and safe inputs satisfying (6) may not exist. Consequently, optimization problems like (7) may be infeasible with input bounds. The backup set method [6] was proposed to solve this problem, by synthesizing control invariant sets and corresponding safe controllers via the CBF framework.

The backup set method is described as follows; while examples are given below and in [6], [16]. First, one must specify a control invariant subset of \mathcal{S} , called the *backup set*:

$$\mathcal{S}_b = \{x \in \mathbb{R}^n : h_b(x) \geq 0\}, \quad (8)$$

such that $h_b : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable, zero is a regular value of h_b , i.e., $h_b(x) = 0 \implies \nabla h_b(x) \neq 0$, and $\mathcal{S}_b \subseteq \mathcal{S}$. Furthermore, one must define a *backup controller* $k_b : \mathbb{R}^n \rightarrow \mathcal{U}$ that renders the backup set forward invariant along the closed-loop system:

$$\dot{x} = f(x) + g(x)k_b(x) \triangleq f_b(x). \quad (9)$$

We denote the solution of (9) with $x(0) = x_0 \in \mathbb{R}^n$ by $\phi_b(t, x_0)$. To summarize, the choice of backup set and backup controller must satisfy the following assumption.

Assumption 1. The backup set $\mathcal{S}_b \subseteq \mathcal{S}$ is control invariant, and the backup controller k_b renders \mathcal{S}_b forward invariant along (9) while satisfying the input constraints:

$$x \in \mathcal{S}_b \implies \phi_b(\theta, x) \in \mathcal{S}_b \subseteq \mathcal{S}, \forall \theta \geq 0, \quad (10)$$

and $k_b(x) \in \mathcal{U}, \forall x \in \mathcal{S}$.

Finding a control invariant subset \mathcal{S}_b is considerably less difficult than verifying that a given \mathcal{S} is control invariant. With this, by construction, we have a control invariant set \mathcal{S}_b and a safe controller k_b at our disposal. However, methods for constructing \mathcal{S}_b (see examples in [6], [16]) may result in a very small set, hence operating the system directly within \mathcal{S}_b may make the behavior overly conservative. To reduce this conservatism, we enlarge \mathcal{S}_b to the set $\mathcal{S}_I \subseteq \mathcal{S}$:

$$\mathcal{S}_I = \left\{ x \in \mathbb{R}^n : \begin{array}{l} \phi_b(\theta, x) \in \mathcal{S}, \forall \theta \in [0, T], \\ \phi_b(T, x) \in \mathcal{S}_b \end{array} \right\}, \quad (11)$$

with $T \geq 0$; cf. Fig. 1. Note that T is a design parameter, the size of \mathcal{S}_I increases with T , and $T = 0$ yields $\mathcal{S}_I = \mathcal{S}_b$.

Lemma 1 ([6]). *The set \mathcal{S}_I is control invariant, and the backup controller k_b renders \mathcal{S}_I forward invariant along (9):*

$$x \in \mathcal{S}_I \implies \phi_b(\vartheta, x) \in \mathcal{S}_I, \forall \vartheta \geq 0. \quad (12)$$

For the proofs of Lemmas 1 and 2, see the Appendix.

Thus, (11) yields a larger, practically more useful control invariant set \mathcal{S}_I than the backup set \mathcal{S}_b ; see [16] for an analysis about the size of \mathcal{S}_I . We use \mathcal{S}_I to provide safety, based on the framework of CBFs. We rely on the derivatives:

$$\dot{h}(\phi_b(\theta, x), u) = \frac{\partial h(\phi_b(\theta, x))}{\partial x} (f(x) + g(x)u), \quad (13)$$

$$\dot{h}_b(\phi_b(T, x), u) = \frac{\partial h_b(\phi_b(T, x))}{\partial x} (f(x) + g(x)u).$$

Then, we can state that the backup controller k_b satisfies safety conditions analogous to (6).

Lemma 2 ([6]). *There exist $\alpha, \alpha_b \in \mathcal{K}_\infty$ such that $\forall x \in \mathcal{S}_I$:*

$$\dot{h}(\phi_b(\theta, x), k_b(x)) \geq -\alpha(h(\phi_b(\theta, x))), \quad \forall \theta \in [0, T], \quad (14)$$

$$\dot{h}_b(\phi_b(T, x), k_b(x)) \geq -\alpha_b(h_b(\phi_b(T, x))).$$

This leads to the main result of the backup set method.

Theorem 2 ([6]). *Consider system (1), set \mathcal{S} in (3), set \mathcal{S}_b in (8), Assumption 1, and set \mathcal{S}_I in (11). Then, there exist $\alpha, \alpha_b \in \mathcal{K}_\infty$ such that a controller $k : \mathbb{R}^n \rightarrow \mathcal{U}$ satisfying:*

$$\dot{h}(\phi_b(\theta, x), k(x)) \geq -\alpha(h(\phi_b(\theta, x))), \quad \forall \theta \in [0, T], \quad (15)$$

$$\dot{h}_b(\phi_b(T, x), k(x)) \geq -\alpha_b(h_b(\phi_b(T, x))).$$

$\forall x \in \mathcal{S}_I$ is guaranteed to exist. Moreover, any locally Lipschitz continuous controller $k : \mathbb{R}^n \rightarrow \mathcal{U}$ that satisfies (15) $\forall x \in \mathcal{S}_I$ renders $\mathcal{S}_I \subseteq \mathcal{S}$ forward invariant along (2).

Proof. The existence of a controller k satisfying (15) follows from Lemma 2, since k_b is such a controller. The forward invariance of \mathcal{S}_I is the consequence of Theorem 1. \square

C. Implementation in Optimization Problems

Theorem 2 can be directly used for controller synthesis, for example, by using (15) in optimization problems like (7):

$$k(x) = \operatorname{argmin}_{u \in \mathcal{U}} \|u - k_d(x)\|_\Gamma^2$$

$$\text{s.t. } \dot{h}(\phi_b(\theta, x), u) \geq -\alpha(h(\phi_b(\theta, x))), \quad \forall \theta \in [0, T],$$

$$\dot{h}_b(\phi_b(T, x), u) \geq -\alpha_b(h_b(\phi_b(T, x))). \quad (16)$$

Note that the constraints are affine in u , cf. (13), hence the optimization problem is convex, and it leads to a quadratic program if $u \in \mathcal{U}$ is also described by affine constraints. Moreover, unlike (7), the optimization problem (16) is guaranteed to be feasible even if h is not verified to be a CBF.

The constraints of (16) contain the terms in (13), where:

$$\frac{\partial h(\phi_b(\theta, x))}{\partial x} = \nabla h(\phi_b(\theta, x)) \frac{\partial \phi_b(\theta, x)}{\partial x}, \quad (17)$$

$$\frac{\partial h_b(\phi_b(T, x))}{\partial x} = \nabla h_b(\phi_b(T, x)) \frac{\partial \phi_b(T, x)}{\partial x}.$$

Here, $Q(\theta, x) \triangleq \partial \phi_b(\theta, x) / \partial x$ is the sensitivity of the flow $\phi_b(\theta, x)$ to its initial condition x . $\phi_b(\theta, x)$ and $Q(\theta, x)$ can be calculated together by solving the initial value problem:

$$\begin{aligned} \phi_b'(0, x) &= f_b(\phi_b(0, x)), & \phi_b(0, x) &= x, \\ Q'(0, x) &= \frac{\partial f_b}{\partial x}(\phi_b(0, x))Q(0, x), & Q(0, x) &= I, \end{aligned} \quad (18)$$

where prime denotes derivative with respect to θ , f_b is as in (9), and I is the $n \times n$ identity matrix.

The optimization problem (16) contains infinitely many constraints parameterized by $\theta \in [0, T]$. For computational tractability, they are usually discretized into finitely many, N_c constraints at $\theta_i = iT/N_c$, $i \in \mathcal{I} = \{0, 1, \dots, N_c\}$, yielding:

$$\begin{aligned} k(x) &= \operatorname{argmin}_{u \in \mathcal{U}} \|u - k_d(x)\|_\Gamma^2 \\ \text{s.t. } \dot{\bar{h}}_i(x, u) &\geq -\alpha(\bar{h}_i(x)), \quad \forall i \in \mathcal{I}, \\ \dot{\bar{h}}_b(x, u) &\geq -\alpha_b(\bar{h}_b(x)). \end{aligned} \quad (19)$$

Here, the shorthand notations $\bar{h}_i(x) = h(\phi_b(\theta_i, x))$ and $\bar{h}_b(x) = h_b(\phi_b(T, x))$ are used. In what follows, we implement controller (19) in an example.

Example 1 (Unicycle model). Consider the unicycle model:

$$\begin{aligned} \dot{\xi} &= v \cos \psi, \\ \dot{\eta} &= v \sin \psi, \\ \dot{\psi} &= \omega, \end{aligned} \quad (20)$$

where the planar position $p = [\xi \ \eta]^\top \in \mathbb{R}^2$ and yaw angle $\psi \in \mathbb{R}$ constitute the state $x = [p^\top \ \psi]^\top$, while the speed $v \in [v_{\min}, v_{\max}] \subset \mathbb{R}$ and yaw rate $\omega \in [-\omega_{\max}, \omega_{\max}] \subset \mathbb{R}$ form the control input $u = [v \ \omega]^\top$. We seek to drive the unicycle in the ξ direction at a goal position η_g with a speed v_g , while avoiding a circular obstacle of radius $R_O > 0$ at position p_O . First we consider a stationary obstacle, then a moving obstacle with velocity \dot{p}_O and acceleration \ddot{p}_O . Note that this latter problem is well-studied [17], and closed-form expressions of control invariant sets exist [18].

We realize the target motion by the desired controller:

$$k_d(x) = \begin{bmatrix} v_g \\ K_\eta(\eta_g - \eta) - K_\psi \sin \psi \end{bmatrix}, \quad (21)$$

that is to be modified to obtain a safe controller $k(x)$. To characterize safety, we first introduce the Euclidian distance D from the obstacle center, the normal vector n pointing away from the obstacle, and a related projection matrix P :

$$D = \|p - p_O\|, \quad n = \frac{p - p_O}{\|p - p_O\|}, \quad P = I - nn^\top. \quad (22)$$

Notice that $\partial D / \partial p = n^\top$ and $\partial n / \partial p = P/D$ hold. Furthermore, let us describe the heading direction by:

$$q = \begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix}, \quad r = \begin{bmatrix} -\sin \psi \\ \cos \psi \end{bmatrix}. \quad (23)$$

With these preliminaries, we introduce the following function from [19] to characterize safety:

$$h(x, t) = D - R_O + \delta n^\top q, \quad (24)$$

where a tunable parameter $\delta \geq 0$ penalizes heading towards the obstacle. The corresponding derivatives read:

$$\begin{aligned} \nabla h(x, t) &= [n^\top + \delta q^\top P/D \quad \delta n^\top r], \\ \frac{\partial h}{\partial t}(x, t) &= -n^\top \dot{p}_O - \delta q^\top P \dot{p}_O / D. \end{aligned} \quad (25)$$

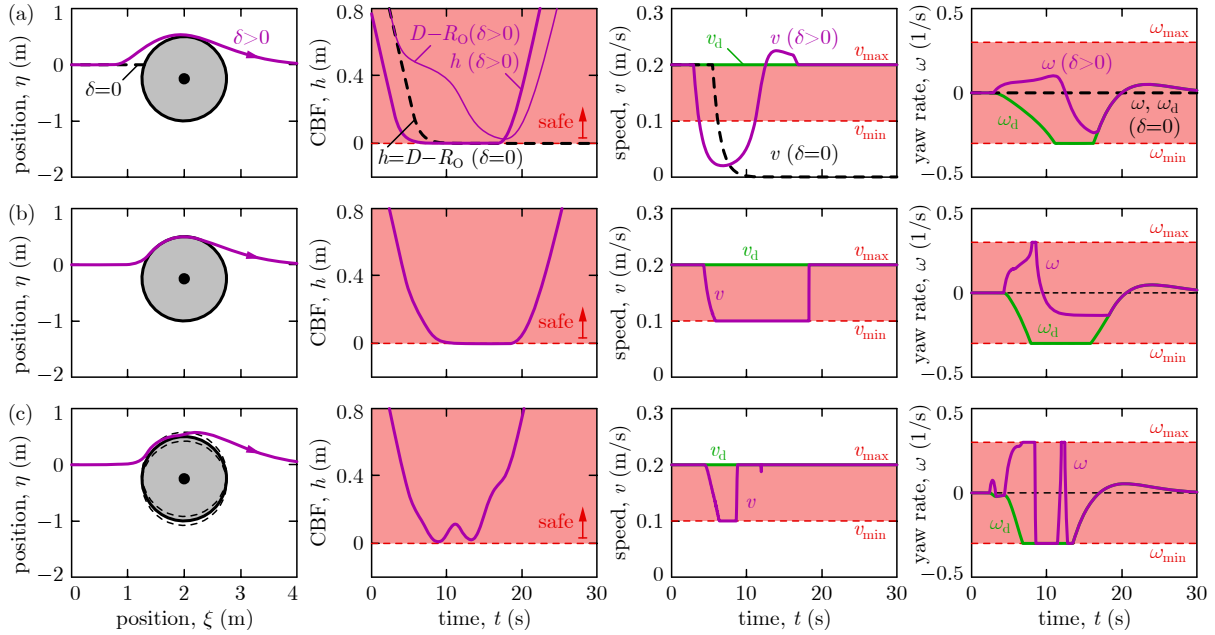


Fig. 2. Safety-critical control of the unicycle model for obstacle avoidance. (a) The CBF-based controller (7) maintains safety, without limits on the inputs (speed and yaw rate). (b) The backup set method-based controller (19) enforces safety with input constraints. (c) Controller (19) handles moving obstacle.

Note that h explicitly depends on time through p_O if the obstacle is moving, and one must include $\partial h/\partial t$ in \dot{h} . For stationary obstacle, this dependence on t can be omitted.

Without input bounds, h could be used as CBF and controller (7) would ensure safe behavior. The result of executing (7) while excluding the input bounds (i.e., taking $\mathcal{U} = \mathbb{R}^2$) is illustrated in Fig. 2(a) for the parameters in Table I and $x_0 = 0$. For $\delta = 0$, i.e., when the heading direction is not penalized by the CBF, the unicycle stops in front of the obstacle (dashed line), which is safe but overly conservative. For $\delta > 0$, the unicycle safely executes the task (solid line). However, since the input bounds are not incorporated into the optimization problem, the lower and upper speed limits are violated. On the other hand, h is not necessarily a valid CBF in the presence of input bounds.

To address input bounds, we rely on the backup controller:

$$k_b(x, t) = \begin{bmatrix} v_{\max} \\ \omega_{\max} \tanh(n^\top r/\varepsilon) \end{bmatrix}, \quad (26)$$

that seeks to turn the unicycle away from the obstacle as fast as possible and drive away with maximum speed. Parameter ε tunes the aggressiveness of turning, and the yaw rate $\pm\omega_{\max}$ is achieved as $\varepsilon \rightarrow 0$. Hence, this controller allows us to keep safety against obstacles that move slower than v_{\max} and turn slower than ω_{\max} .

The backup controller is associated with:

$$h_b(x, t) = n^\top (qv_{\max} - \dot{p}_O), \quad (27)$$

whose derivatives are:

$$\begin{aligned} \nabla h_b(x, t) &= [(qv_{\max} - \dot{p}_O)^\top P/D \quad n^\top r v_{\max}], \\ \frac{\partial h_b}{\partial t}(x, t) &= -(qv_{\max} - \dot{p}_O)^\top P \dot{p}_O/D - n^\top \ddot{p}_O. \end{aligned} \quad (28)$$

TABLE I
PARAMETER VALUES FOR THE NUMERICAL EXAMPLES

Parameter	Value	Unit	Parameter	Value	Unit
v_{\max}	0.2	m/s	δ (Ex. 1)	0, 0.5	m
v_{\min}	0.1	m/s	δ (Ex. 2)	0	m
ω_{\max}	0.3	rad/s	ε	0.01	1
η_g	0	m	γ	1	1/s
v_g	0.2	m/s	γ_b	1	1/s
K_η	0.5	1/(ms)	Γ	diag{1, 0.25}	{1, m ² }
K_ψ	0.5	1/s	T	4	s
R_O	0.75	m	N_c (Ex. 1)	80	1
ξ_O	2	m	N_c (Ex. 2)	400	1
$\bar{\eta}_O$	-0.25	m	σ	0.1	m/s
A_η	0.1	m/s	σ_b	0.1	m
Ω	$2\pi/5$	rad/s	p	10^{18}	1
			p_b	10^{18}	s ²

We remark that the backup set that is kept invariant by k_b is in fact given by both $h_b(x, t) \geq 0$ and $h(x, t) \geq 0$, (rather than just $h_b(x, t) \geq 0$), but both of these functions are involved in (15).

The efficacy of the backup set method with controller (19) is shown in Fig. 2(b) for parameters in Table I and $\delta = 0$. The controller maintains safety while satisfying the input bounds, and note that even $\delta = 0$ yields desired behavior. The same controller is tested for the case of a moving obstacle in Fig. 2(c). The obstacle moves in the η direction sinusoidally, with $p_O(t) = [\xi_O \quad \eta_O(t)]^\top$, $\eta_O(t) = \bar{\eta}_O - A_\eta \sin(\Omega t)/\Omega$. The end result is still safety with bounded inputs.

III. REDUCED ORDER MODELS

Model (1) is often a simplified representation of a real control system. The actual dynamics may be more complicated, higher dimensional, involving unmodeled phenomena.

Hence, we call (1) as *reduced order model (ROM)*. The backup set method is able to control the ROM with formal safety guarantees while respecting input constraints. Yet, the safety of the actual *full order system (FOS)* is not necessarily ensured. Next, we investigate the effect of unmodeled dynamics on safety, and derive conditions for the safety of the FOS by following our previous work [19]. Then, we propose a robustified backup set method. We consider the ROM to be given, while approaches to construct ROMs are out of scope of this paper. Finally, we demonstrate our framework on an example, in which the locomotion of a quadruped (FOS) is controlled to follow the unicycle model (ROM).

Consider a FOS given by state $X \in \mathbb{R}^N$, input $U \in \mathbb{R}^M$, locally Lipschitz continuous functions $F: \mathbb{R}^N \rightarrow \mathbb{R}^N$ and $G: \mathbb{R}^N \rightarrow \mathbb{R}^{N \times M}$, and dynamics:

$$\dot{X} = F(X) + G(X)U. \quad (29)$$

Furthermore, let a *reduced order state* $x \in \mathbb{R}^n$ be defined by a continuously differentiable map $P: \mathbb{R}^N \rightarrow \mathbb{R}^n$:

$$x = P(X). \quad (30)$$

The reduced order state is selected such that it describes safety-critical behavior. Specifically, consider the safe set:

$$\mathcal{C} = \{X \in \mathbb{R}^N : h(P(X)) \geq 0\} \quad (31)$$

for the FOS with $h: \mathbb{R}^n \rightarrow \mathbb{R}$ given as before.

To achieve safe FOS behavior, one may construct a ROM like (1), design a safety-critical ROM controller $u = k(x)$, and utilize a tracking controller $K: \mathbb{R}^N \times \mathbb{R}^m \rightarrow \mathbb{R}^M$, $U = K(X, u)$ so that the closed-loop FOS:

$$\dot{X} = F(X) + G(X)K(X, u) \quad (32)$$

tracks the ROM. With appropriate ROM and tracking controller, the true dynamics of the reduced order state x track the ROM accurately. The true reduced order dynamics are:

$$\dot{x} = f(x) + g(x)u + d, \quad (33)$$

where $d \in \mathbb{R}^n$ is the deviation from the ROM, given by:

$$d = \nabla P(X)(F(X) + G(X)K(X, u)) - f(P(X)) - g(P(X))u. \quad (34)$$

Note that while d acts as disturbance on the ROM, it can be viewed as tracking error that the FOS seeks to eliminate.

If the discrepancy d is zero, the ROM captures the safety-critical behavior of the FOS accurately, and the backup set method can be used directly with the control invariant set:

$$\mathcal{C}_1 = \left\{ X \in \mathbb{R}^N : \begin{array}{l} h(\phi_b(\theta, P(X))) \geq 0, \quad \forall \theta \in [0, T], \\ h_b(\phi_b(T, P(X))) \geq 0 \end{array} \right\}, \quad (35)$$

for which $X \in \mathcal{C}_1 \iff x \in \mathcal{S}_1$. Then, per Theorem 2, there exists a controller k that satisfies (15) and renders $\mathcal{C}_1 \subseteq \mathcal{C}$ forward invariant along (29). However, nonzero discrepancy d may lead to safety violations. Below we discuss conditions under which safety is preserved, and we investigate how to provide robustness against d . During robustification, (33) is considered while the discrepancy d is viewed as an unknown but bounded term (see assumptions below) that represents modeling errors and disturbances associated with the ROM.

A. Safety with Ideal Tracking

If the ROM and the tracking controller are well-designed, the true reduced order dynamics converges to the ROM and the discrepancy d vanishes. First, we consider this ideal scenario as reflected by the following assumption.

Assumption 2. The tracking controller $U = K(x, u)$ drives the discrepancy between the true reduced order dynamics and the ROM to zero exponentially. That is, there exist $A \geq 0$ and $\lambda > 0$ such that $\forall t \geq 0$:

$$\|d\| \leq Ae^{-\lambda t}. \quad (36)$$

For simplicity, we assume exponential convergence, although one could also consider asymptotic stability with a class- \mathcal{KL} function on the right-hand side. Similarly, to simplify our discussion, we choose linear class- \mathcal{K}_∞ functions: $\alpha(r) = \gamma r$, $\alpha_b(r) = \gamma_b r$, with $\gamma, \gamma_b > 0$. Furthermore, we assume that the gradients of $h(\phi_b(\theta, x))$ and $h_b(\phi_b(\theta, x))$ are bounded, i.e., there exist $D, D_b \geq 0$ such that $\|\partial h(\phi_b(\theta, x))/\partial x\| \leq D$, $\forall \theta \in [0, T]$ and $\|\partial h_b(\phi_b(\theta, x))/\partial x\| \leq D_b$ hold $\forall x \in \mathcal{S}_1$ with the Euclidean norm $\|\cdot\|$. These assumptions are relaxed in the next section.

Under these assumptions, we show that a time-varying subset of \mathcal{C}_1 is control invariant. We define this set $\mathcal{C}_d(t)$ by:

$$\mathcal{C}_d(t) = \left\{ X \in \mathbb{R}^N : \begin{array}{l} H(\theta, X, t) \geq 0, \quad \forall \theta \in [0, T], \\ H_b(T, X, t) \geq 0 \end{array} \right\}, \quad (37)$$

with:

$$\begin{aligned} H(\theta, X, t) &= h(\phi_b(\theta, P(X))) - \frac{DAe^{-\lambda t}}{\lambda - \gamma}, \\ H_b(T, X, t) &= h_b(\phi_b(T, P(X))) - \frac{D_b Ae^{-\lambda t}}{\lambda - \gamma_b}. \end{aligned} \quad (38)$$

Theorem 3. Consider the ROM (1), set \mathcal{S}_1 in (11) and a locally Lipschitz continuous controller $k: \mathbb{R}^n \rightarrow \mathcal{U}$ that satisfies (15) with $\alpha(r) = \gamma r$, $\alpha_b(r) = \gamma_b r$, $\forall x \in \mathcal{S}_1$. Furthermore, consider the FOS (29), reduced order state (30), set \mathcal{C} in (31), set $\mathcal{C}_d(t)$ in (37)-(38), and Assumption 2. If $\gamma, \gamma_b < \lambda$, then $\mathcal{C}_d(t) \subseteq \mathcal{C}$ is forward invariant along (32).

Proof. For $\gamma, \gamma_b < \lambda$, the terms $DA/(\lambda - \gamma)e^{-\lambda t}$ and $D_b A/(\lambda - \gamma_b)e^{-\lambda t}$ in (38) are nonnegative $\forall t \geq 0$. Hence, $\mathcal{C}_d(t) \subseteq \mathcal{C}_1$, $\forall t \geq 0$, and $X \in \mathcal{C}_d(t)$ implies $x \in \mathcal{S}_1$. Then, Theorem 2 can be applied, and a controller k satisfying (15) is guaranteed to exist for all $X \in \mathcal{C}_d(t)$ since $x \in \mathcal{S}_1$. Given (38) and (15), the derivative of H along (32) satisfies:

$$\begin{aligned} & \dot{H}(\theta, X, t, k(P(X)), d) \\ &= \dot{h}(\phi_b(\theta, x), k(x)) + \frac{\partial h(\phi_b(\theta, x))}{\partial x} d + \frac{\lambda DAe^{-\lambda t}}{\lambda - \gamma} \\ &\geq -\gamma h(\phi_b(\theta, x)) - \left\| \frac{\partial h(\phi_b(\theta, x))}{\partial x} \right\| \|d\| + \frac{\lambda DAe^{-\lambda t}}{\lambda - \gamma} \\ &\geq -\gamma h(\phi_b(\theta, x)) - DAe^{-\lambda t} + \frac{\lambda DAe^{-\lambda t}}{\lambda - \gamma} \\ &\geq -\gamma H(\theta, X, t). \end{aligned} \quad (39)$$

Similarly, $\dot{H}_b(T, X, t, k(P(X)), d) \geq -\gamma_b H_b(T, X, t)$ can be proven. Thus, by Theorem 1 we can conclude the forward invariance of $\mathcal{C}_d(t) \subseteq \mathcal{C}_I$, that implies a safe FOS. \square

Remark 1. Theorem 3 states that with fast enough tracking of the ROM, the FOS stays safe and evolves in a region where backup set method-based controllers are guaranteed to exist. However, this result is conditioned on ideal exponential tracking (and the technical assumption about the bounded gradients of h and h_b). Next, we relax these restrictions.

B. Input-to-State Safe Backup Set Method

Let us use the following weaker assumption on tracking.

Assumption 3. The tracking controller $U = K(x, u)$ drives the discrepancy between the true reduced order dynamics and the ROM to a neighborhood of zero exponentially. That is, there exist $A, B \geq 0$ and $\lambda > 0$ such that $\forall t \geq 0$:

$$\|d\|^2 \leq Ae^{-\lambda t} + B. \quad (40)$$

Note that this assumption includes the case $A = 0$, i.e., when the discrepancy does not necessarily decay but stays bounded below B . We also remark that the square after the norm of d is introduced for algebraic convenience only. The assumption is shown to hold for the quadruped example below.

When the discrepancy does not decay to zero ($B \neq 0$), safety can no longer be formally guaranteed by (15). To remedy this, some CBF approaches add extra robustifying terms to their safety constraints [9], [11]. For example, the approach of *input-to-state safe* CBFs modifies (6) to $\dot{h}(x, k(x)) \geq -\alpha(h(x)) + \sigma \|\nabla h(x)\|^2$ with $\sigma > 0$ (where $\nabla h(x)$ could be replaced with $\nabla h(x)g(x)$ in case of matched disturbances) [10]. We propose to extend this approach to the *input-to-state safe backup set method*, by modifying (15) to:

$$\begin{aligned} \dot{h}(\phi_b(\theta, x), k(x)) &\geq -\alpha(h(\phi_b(\theta, x))) \\ &+ \sigma \left\| \frac{\partial h(\phi_b(\theta, x))}{\partial x} \right\|^2, \quad \forall \theta \in [0, T], \\ \dot{h}_b(\phi_b(T, x), k(x)) &\geq -\alpha_b(h_b(\phi_b(T, x))) \\ &+ \sigma_b \left\| \frac{\partial h_b(\phi_b(T, x))}{\partial x} \right\|^2, \end{aligned} \quad (41)$$

with tunable parameters $\sigma, \sigma_b > 0$.

The approach of input-to-state safe CBFs is able to keep a neighborhood of the safe set invariant even with disturbances, and this neighborhood can be tuned as small as desired by parameter σ . We seek to achieve the same results with input constraints using the backup set method. Accordingly, we consider a neighborhood \mathcal{S}_d of the control invariant set \mathcal{S}_I :

$$\mathcal{S}_d = \left\{ x \in \mathbb{R}^n: \begin{array}{l} h(\phi_b(\theta, x)) \geq -B/(4\sigma\gamma), \quad \forall \theta \in [0, T], \\ h_b(\phi_b(T, x)) \geq -B/(4\sigma_b\gamma_b) \end{array} \right\}, \quad (42)$$

determined by σ, σ_b , and we redefine set $\mathcal{C}_d(t)$ in (37) with:

$$\begin{aligned} H(\theta, X, t) &= h(\phi_b(\theta, P(X))) - \frac{Ae^{-\lambda t}}{4\sigma(\lambda - \gamma)} + \frac{B}{4\sigma\gamma}, \\ H_b(T, X, t) &= h_b(\phi_b(T, P(X))) - \frac{Ae^{-\lambda t}}{4\sigma_b(\lambda - \gamma_b)} + \frac{B}{4\sigma_b\gamma_b}. \end{aligned} \quad (43)$$

Then, we state the invariance of set $\mathcal{C}_d(t)$ that can be made arbitrarily close to the safe set \mathcal{C} by increasing σ, σ_b .

Theorem 4. Consider the ROM (1), set \mathcal{S}_d in (42) and a locally Lipschitz continuous controller $k: \mathbb{R}^n \rightarrow \mathcal{U}$ that satisfies (41) with $\alpha(r) = \gamma r$, $\alpha_b(r) = \gamma_b r$, $\forall x \in \mathcal{S}_d$. Furthermore, consider the FOS (29), reduced order state (30), set $\mathcal{C}_d(t)$ in (37)-(43), and Assumption 3. If $\gamma, \gamma_b < \lambda$, then $\mathcal{C}_d(t)$ is forward invariant along (32).

Proof. First, we show that the following inequality holds:

$$\begin{aligned} &\sigma \left\| \frac{\partial h(\phi_b(\theta, x))}{\partial x} \right\|^2 - \left\| \frac{\partial h(\phi_b(\theta, x))}{\partial x} \right\| \|d\| \\ &\geq \left(\sqrt{\sigma} \left\| \frac{\partial h(\phi_b(\theta, x))}{\partial x} \right\| - \frac{\|d\|}{2\sqrt{\sigma}} \right)^2 - \frac{\|d\|^2}{4\sigma} \\ &\geq -\frac{Ae^{-\lambda t} + B}{4\sigma}. \end{aligned} \quad (44)$$

Then, the rest of the proof follows that of Theorem 3:

$$\begin{aligned} &\dot{H}(\theta, X, t, k(P(X)), d) \\ &= \dot{h}(\phi_b(\theta, x), k(x)) + \frac{\partial h(\phi_b(\theta, x))}{\partial x} d + \frac{\lambda Ae^{-\lambda t}}{4\sigma(\lambda - \gamma)} \\ &\geq -\gamma h(\phi_b(\theta, x)) - \frac{Ae^{-\lambda t} + B}{4\sigma} + \frac{\lambda Ae^{-\lambda t}}{4\sigma(\lambda - \gamma)} \\ &\geq -\gamma H(\theta, X, t), \end{aligned} \quad (45)$$

cf. (39), where in the second step we used the Cauchy-Schwartz inequality and substituted (44). \square

Remark 2. Theorem 4 states that input-to-state stable tracking of the ROM (i.e., when the discrepancy d decays to or is within a neighborhood of zero) makes the FOS stay in the set $\mathcal{C}_d(t)$. This set can be tuned to be as close to the safe set \mathcal{C} as desired using σ and σ_b , cf. (43), and it no longer depends on the bounds of the gradients of h and h_b . However, we cannot claim the existence of a controller k satisfying (41) $\forall x \in \mathcal{S}_d$ anymore. Hence, instead of (19), one may implement a relaxed optimization problem:

$$\begin{aligned} k(x) &= \underset{\substack{u \in \mathcal{U} \\ \delta_i, \delta_b \geq 0}}{\operatorname{argmin}} \|u - k_d(x)\|_{\Gamma}^2 + \sum_{i \in \mathcal{I}} p_i \delta_i^2 + p_b \delta_b^2 \\ \text{s.t. } \dot{\bar{h}}_i(x, u) &\geq -\alpha(\bar{h}_i(x)) + \sigma \left\| \frac{\partial \bar{h}_i(x)}{\partial x} \right\|^2 - \delta_i, \\ &\quad \forall i \in \mathcal{I}, \\ \dot{\bar{h}}_b(x, u) &\geq -\alpha_b(\bar{h}_b(x)) + \sigma_b \left\| \frac{\partial \bar{h}_b(x)}{\partial x} \right\|^2 - \delta_b, \end{aligned} \quad (46)$$

with slack variables $\delta_i, \delta_b \geq 0$ and penalties $p_i, p_b \gg 1$, $i \in \mathcal{I}$. Formulating provably safe and feasible controllers without this relaxation is subject to future research.

Example 2 (Quadrupedal locomotion). Consider the Unitree A1 quadrupedal robot shown in Fig. 3. We seek to execute legged locomotion and accomplish the obstacle avoidance task of Example 1. We consider the quadruped as FOS, and we rely on an existing walking controller for locomotion with

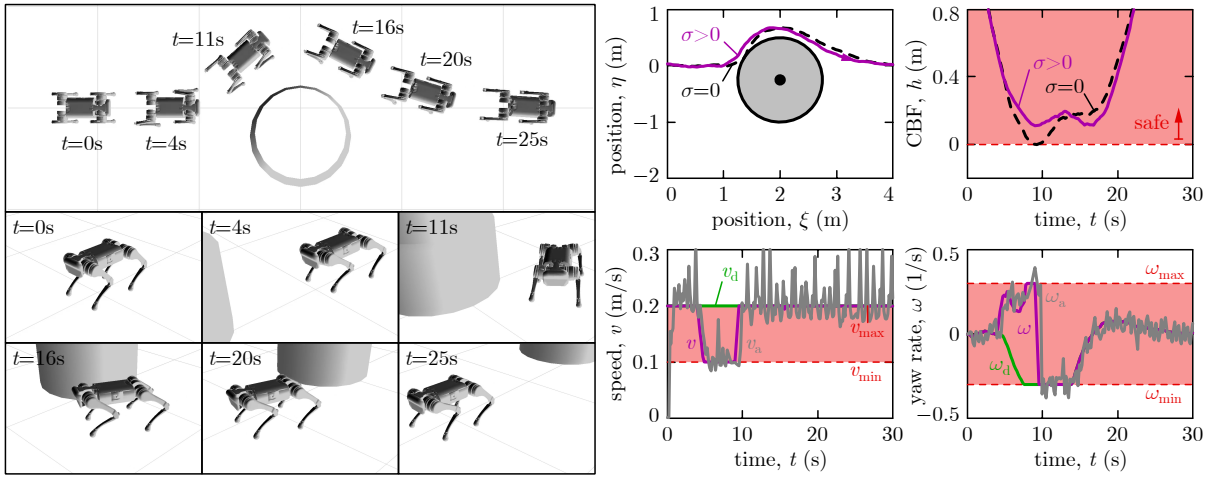


Fig. 3. Application of the proposed safety-critical control framework with input constrained reduced order model in quadrupedal locomotion. The quadruped safely navigates by tracking the speed and yaw rate synthesized with the unicycle model and the input-to-state safe backup set method.

given speed and yaw rate. As such, the walking tracks the unicycle model in Example 1, which serves as ROM.

The quadruped has 18 degrees of freedom and 12 actuated joints. Its motion is described by the configuration $q \in \mathbb{R}^{18}$, velocities $\dot{q} \in \mathbb{R}^{18}$, inputs $U \in \mathbb{R}^{12}$ and holonomic constraints $c(q) \equiv 0 \in \mathbb{R}^{n_c}$ at the n_c number of contacts between the feet and the ground. The dynamics are governed by the Euler-Lagrange equations:

$$\begin{aligned} D(q)\ddot{q} + H(q, \dot{q}) &= BU + J(q)^\top \lambda, \\ J(q)\dot{q} + \dot{J}(q, \dot{q})\dot{q} &= 0, \end{aligned} \quad (47)$$

with mass matrix $D(q) \in \mathbb{R}^{18 \times 18}$, Coriolis and gravity terms $H(q, \dot{q}) \in \mathbb{R}^{18}$, Jacobian $J(q) = \partial c(q) / \partial q \in \mathbb{R}^{n_c \times 18}$, and constraint wrench $\lambda \in \mathbb{R}^{n_c}$. This yields the FOS (29) with the state $X = [q^\top \ \dot{q}^\top]^\top \in \mathbb{R}^{36}$ and expressions:

$$F(X) = \begin{bmatrix} \dot{q} \\ -D(q)^{-1}(H(q, \dot{q}) - J(q)^\top \lambda) \end{bmatrix}, \quad G(X) = \begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}. \quad (48)$$

During obstacle avoidance, safety is determined by the planar body position ξ and η and the yaw angle ψ , leading to the reduced order state $x \in \mathbb{R}^3$ of Example 1. These states are elements of the full state X . The corresponding equations in the FOS (47) reduce to the unicycle model (20) if roll and pitch are neglected, thus the unicycle is chosen as ROM. For legged locomotion, we use the inverse dynamics quadratic program based walking controller, $U = K(X, u)$, specified in [20]. This controller is able to track speed and yaw rate commands in the reduced order input $u \in \mathbb{R}^2$ as long as they are below v_{\max} and between $\pm\omega_{\max}$, respectively. We also prescribe the minimum speed v_{\min} so that the quadruped is not allowed to stop. We use the input-to-state safe backup set method, with details in Example 1, to find safe speed and yaw rate commands within these bounds.

Fig. 3 shows high-fidelity simulations of the quadrupedal locomotion². The speed and yaw rate are commanded using

the proposed controller (46), the formulas in Example 1, the parameters in Table I, and the CVXOPT solver [21]. The radius R_O , that the robot's center should stay outside of, consists of the radius of the obstacle (0.45 m) and the size of the quadruped (0.3 m). With the proposed controller, the quadruped successfully navigates around the obstacle as shown by the motion tiles. Observe that safety is maintained with respect to the specification h . Meanwhile, speed and yaw rate commands stay within desired bounds (while their actual value may exceed the bounds). The figure also indicates the tracking performance of the walking controller, by comparing the actual speed v_a and yaw rate ω_a (extracted from X) to the commands v and ω . Indeed, the discrepancy d between the commanded velocities $\dot{x} = [v \cos \psi \ v \sin \psi \ \omega]^\top$ and the corresponding actual values decays and stays bounded, as in (40) in Assumption 3. Finally, the trajectory with the standard backup set method, (i.e., controller (19) and $\sigma = 0, \sigma_b = 0$) is shown by dashed lines. This case gets closer to safety violations due to lack of robustness to the discrepancy between the ROM and FOS.

This example demonstrates the efficacy of the proposed safety-critical control approach, in which an input constrained ROM is combined with the backup set method and a reliable tracking controller. The results show safe behavior on a complex quadrupedal robot during obstacle avoidance.

IV. CONCLUSIONS

This paper addressed safety-critical control using reduced order models that have bounded inputs. To formally guarantee safety while respecting input bounds, the backup set method was used. Robustness with respect to the discrepancy between the reduced order model and the full order system was analyzed. Conditions were derived for the safety of the full system, and the input-to-state safe backup set method was proposed to robustify against the above mentioned discrepancy. The efficacy of the proposed control framework was demonstrated by controlling a quadruped for obstacle

²See video at: <https://youtu.be/h8-x7-4eqWs>.

avoidance while relying on the unicycle model. Future work includes studying the feasibility of the robustified controller.

APPENDIX

Proof of Lemma 1. By definition (11) of \mathcal{S}_I and Assumption 1, we have:

$$x \in \mathcal{S}_I \implies \phi_b(\theta, x) \in \mathcal{S}_b \subseteq \mathcal{S}, \quad \forall \theta \geq T. \quad (49)$$

From this, and the fact that:

$$\phi_b(\theta + \vartheta, x_0) = \phi_b(\theta, \phi_b(\vartheta, x_0)), \quad (50)$$

holds for any arbitrary $\theta, \vartheta \geq 0$ and $x_0 \in \mathbb{R}^n$, we obtain:

$$x \in \mathcal{S}_I \implies \phi_b(T, \phi_b(\vartheta, x)) \in \mathcal{S}_b, \quad \forall \vartheta \geq 0. \quad (51)$$

Furthermore, the definition (11) of \mathcal{S}_I and (49) give:

$$x \in \mathcal{S}_I \implies \phi_b(\theta, x) \in \mathcal{S}, \quad \forall \theta \geq 0. \quad (52)$$

Using the property (50) again, we obtain:

$$x \in \mathcal{S}_I \implies \phi_b(\theta, \phi_b(\vartheta, x)) \in \mathcal{S}, \quad \forall \theta \in [0, T], \quad \forall \vartheta \geq 0. \quad (53)$$

Thus, (51), (53) and the definition (11) of \mathcal{S}_I yield (12). \square

Proof of Lemma 2. The definition (11) of \mathcal{S}_I can be rewritten as:

$$\mathcal{S}_I = \left\{ x \in \mathbb{R}^n : \begin{array}{l} h(\phi_b(\theta, x)) \geq 0, \quad \forall \theta \in [0, T], \\ h_b(\phi_b(T, x)) \geq 0 \end{array} \right\}. \quad (54)$$

\mathcal{S}_I is rendered forward invariant by the backup controller k_b per Lemma 1. Therefore, Nagumo's theorem [22] states:

$$\begin{aligned} h(\phi_b(\theta, x)) = 0 &\implies \dot{h}(\phi_b(\theta, x), k_b(x)) \geq 0, \\ h_b(\phi_b(T, x)) = 0 &\implies \dot{h}_b(\phi_b(T, x), k_b(x)) \geq 0. \end{aligned} \quad (55)$$

Consider the second condition and let:

$$\tilde{\mathcal{S}}(x) = \{\tilde{x} \in \mathbb{R}^n : h_b(\phi_b(T, x)) \geq h_b(\phi_b(T, \tilde{x})) \geq 0\}. \quad (56)$$

Note that $\forall x \in \mathcal{S}_I$, $\tilde{\mathcal{S}}(x)$ is nonempty and $x \in \tilde{\mathcal{S}}(x)$, thus:

$$\dot{h}_b(\phi_b(T, x), k_b(x)) \geq \inf_{\tilde{x} \in \tilde{\mathcal{S}}(x)} \dot{h}_b(\phi_b(T, \tilde{x}), k_b(\tilde{x})). \quad (57)$$

Now let us define the set $\tilde{\mathcal{S}}_r$ for $r \geq 0$ and $\tilde{\alpha}_b : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$:

$$\tilde{\mathcal{S}}_r = \{\tilde{x} \in \mathbb{R}^n : r \geq h_b(\phi_b(T, \tilde{x})) \geq 0\}. \quad (58)$$

$$\tilde{\alpha}_b(r) = - \inf_{\tilde{x} \in \tilde{\mathcal{S}}_r} \dot{h}_b(\phi_b(T, \tilde{x}), k_b(\tilde{x})). \quad (59)$$

Then, (57) is equivalent to:

$$\dot{h}_b(\phi_b(T, x), k_b(x)) \geq -\tilde{\alpha}_b(h_b(\phi_b(T, x))). \quad (60)$$

Note that $\tilde{\alpha}_b$ is monotonically increasing with respect to r since the inf is taken over a larger set $\tilde{\mathcal{S}}_r$ as r grows. Furthermore, $\tilde{\alpha}_b$ satisfies $\tilde{\alpha}_b(0) \leq 0$ based on (55). Therefore, there exists $\alpha_b \in \mathcal{K}_\infty$ such that $\alpha_b(r) \geq \tilde{\alpha}_b(r)$, $\forall r \geq 0$. This, together with (60), leads to the second statement in (14). The first statement can be proven the same way: showing the existence of $\alpha_\theta \in \mathcal{K}_\infty$ for each $\theta \in [0, T]$ and defining $\alpha \in \mathcal{K}_\infty$ such that $\alpha(r) = \max_{\theta \in [0, T]} \alpha_\theta(r)$. \square

Acknowledgment. We thank Albert Li and Andrew Taylor for discussions about safety with reduced order models, and Wyatt Ubellacker for his invaluable help in synthesizing low-level controllers for the quadruped.

REFERENCES

- [1] R. T. Fawcett, K. Afsari, A. D. Ames, and K. A. Hamed, "Toward a data-driven template model for quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7636–7643, 2022.
- [2] X. Xiong and A. Ames, "3-D underactuated bipedal walking via H-LIP based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022.
- [3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [4] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [5] K. Dunlap, M. Hibbard, M. Mote, and K. Hobbs, "Comparing run time assurance approaches for safe spacecraft docking," *IEEE Control Systems Letters*, vol. 6, pp. 1849–1854, 2022.
- [6] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.
- [7] D. R. Agrawal and D. Panagou, "Safe control synthesis via input constrained control barrier functions," in *60th IEEE Conference on Decision and Control*, 2021, pp. 6113–6118.
- [8] S. Liu, J. Dolan, and C. Liu, "Safe control under input saturation with neural control barrier functions," in *6th Annual Conference on Robot Learning*, 2022.
- [9] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019.
- [10] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, "Safe controller synthesis with tunable input-to-state safe control barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 908–913, 2022.
- [11] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.
- [12] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2160–2165.
- [13] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *56th IEEE Conference on Decision and Control*, 2017, pp. 2242–2253.
- [14] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.
- [15] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *60th IEEE Conference on Decision and Control*, 2021, pp. 6814–6821.
- [16] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *60th IEEE Conference on Decision and Control*, 2021, pp. 6835–6841.
- [17] I. Exarchos, P. Tsiotras, and M. Pachter, "On the suicidal pedestrian differential game," *Dynamic Games and Applications*, vol. 5, no. 3, pp. 297–317, 2015.
- [18] Y. Shoukry, P. Tabuada, S. Tsuei, M. B. Milam, J. W. Grizzle, and A. D. Ames, "Closed-form controlled invariant sets for pedestrian avoidance," in *American Control Conference*, 2017, pp. 1622–1628.
- [19] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022.
- [20] W. Ubellacker, N. Csomay-Shanklin, T. G. Molnar, and A. D. Ames, "Verifying safe transitions between dynamic motion primitives on legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 8477–8484.
- [21] M. S. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A Python package for convex optimization," 2012, available: <https://cvxopt.org/>.
- [22] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.