Model-Free Safety-Critical Control for Robotic Systems

Tamas G. Molnar, Ryan K. Cosner, Andrew W. Singletary, Wyatt Ubellacker, and Aaron D. Ames

***Abstract*—This paper presents a framework for the safety-critical control of robotic systems, when safety is defined on safe regions in the configuration space. To maintain safety, we synthesize a safe velocity based on control barrier function theory without relying on a – potentially complicated – high-fidelity dynamical model of the robot. Then, we track the safe velocity with a tracking controller. This culminates in *model-free safety critical control*. We prove theoretical safety guarantees for the proposed method. Finally, we demonstrate that this approach is application-agnostic. We execute an obstacle avoidance task with a Segway in high-fidelity simulation, as well as with a Drone and a Quadruped in hardware experiments.**

***Index Terms*—Dynamics, Motion Control, Robot Safety**

I. INTRODUCTION

SAFETY is a fundamental requirement in the control of many robotic systems, including legged [1], flying [2] and wheeled robots [3]. Provable safety guarantees and safety-critical control for robotics have therefore attracted significant attention. Synthesizing safety-critical controllers, however, typically relies on high-fidelity dynamical models describing the robots, which are often complicated and high-dimensional. The underlying control laws, therefore, are non-trivial to synthesize and implement [4], [5]. For example, control barrier functions (CBFs) [6] are a popular tool to achieve provable safety guarantees, although designing CBFs and calculating the corresponding safe control inputs may be nontrivial if the dynamics are complicated.

To tackle this, [7] proposed model-free barrier functions by a data-driven approach, while [8], [9] used robust CBFs to overcome the effects of unmodeled dynamics. Furthermore, many works rely on reduced-order models for planning and control [10]. These include single integrator models for multi-robot applications [11], [12] or unicycle models for wheeled robots [13], [14], which have proven to be extremely useful models despite being overly simplistic. Here we draw inspiration from these models and approaches.

In this paper, we rely on CBFs to synthesize safe controllers for robotic systems in which safe regions are defined in the configuration space. We treat the safety-critical aspect of this problem in a model-free fashion, without relying on the full-order dynamics of the robot. We follow the approach of [15], [16], where a safe velocity was designed based on reduced-order kinematics – i.e., without the full dynamical model – and this safe velocity was tracked by a velocity tracking controller. This approach is agnostic to the application domain, although the underlying tracking controllers depend on the system and their synthesis or tuning may require knowledge about the full model. Velocity tracking, however, is well-established in robotics [17] and controllers executing stable tracking are available for many robots. Once velocity tracking is established, enforcing safety does not require further consideration of the high-fidelity model — we refer to this as *model-free safety-critical control*.

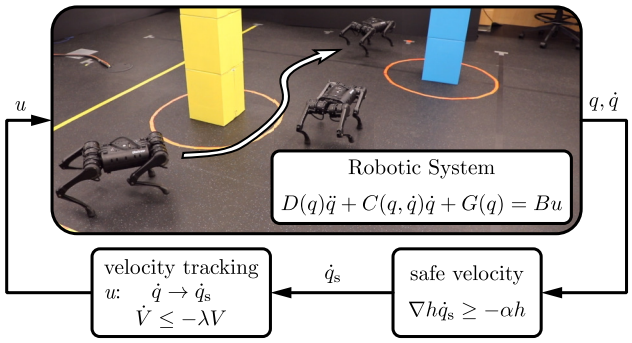

Fig. 1. The proposed control method and its execution on hardware. While the safety-critical controller does not rely on the full dynamical model of the robot, it controls the motion in a provably safe manner.

While the idea behind this control method was established in [15], the present paper formalizes and generalizes this approach via two main contributions. First, we provide a theoretical proof of the safe behavior for robotic systems executing the proposed control approach. Second, we demonstrate the applicability of this method on wheeled, flying and legged robots: a Segway (in simulation), a Drone and a Quadruped (in hardware experiments). This justifies that the method is agnostic to the application domain.

The paper is organized as follows. Section II revisits control Lyapunov and control barrier functions to achieve stability and safety. Section III outlines the proposed control method, states and proves the safety guarantees thereof. Section IV discusses robotic applications through simulations and hardware experiments. Section V concludes the paper.

II. PRELIMINARIES

Our approach relies on stable tracking of a safe velocity to achieve safety for robotic systems. Thus, first we introduce

Manuscript received: September 9, 2021; Revised: December 4, 2021; Accepted: December 7, 2021.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

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

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

Digital Object Identifier (DOI): see top of this page.

the notions of stability and safety, and the guarantees thereof provided by control Lyapunov functions (CLFs) and control barrier functions (CBFs). CLFs and CBFs are illustrated in Fig. 2 together with a stable and a safe trajectory.

Consider control-affine systems with state space $X \subseteq \mathbb{R}^n$, state $x \in X$, set of admissible inputs $U \subseteq \mathbb{R}^m$, and control input $u \in U$:

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

Let $f : X \to \mathbb{R}^n$ and $g : X \to \mathbb{R}^{n \times m}$ be Lipschitz continuous. For an initial condition $x(0) = x_0 \in X$ and a Lipschitz continuous controller $k : X \to U$, $u = k(x)$, the system has a unique solution $x(t)$ which we assume to exist for all $t \geq 0$. We also assume that $x(t) \equiv 0$ is an equilibrium of (1) if $u(t) \equiv 0$ (i.e., $f(0) = 0$) and X is an open and connected neighborhood of $x = 0$.

Throughout the paper we use the following notation. $\|\cdot\|$ is Euclidean norm and $\|\cdot\|_\infty$ is maximum norm. We say that a continuous function $\gamma : [0, b) \to \mathbb{R}_{\geq 0}$, $b \in \mathbb{R}_{>0}$ is of *class-$\mathcal{K}$* (or $\gamma : (-a, b) \to \mathbb{R}$, $a, b \in \mathbb{R}_{>0}$ is of *extended class-$\mathcal{K}$*) if γ is strictly monotonically increasing and $\gamma(0) = 0$.

A. Stability and Control Lyapunov Functions

Hereinafter, we rely on the notion of exponential stability.

Definition 1. The equilibrium $x = 0$ of system (1) is *exponentially stable* if there exist $a, M, \beta \in \mathbb{R}_{>0}$ such that $\|x_0\| < a \Rightarrow \|x(t)\| \leq M e^{-\beta t}\|x_0\|, \forall t \geq 0$.

An efficient technique to achieve exponential stability is control synthesis via control Lyapunov functions (CLFs) [18], as stated formally below.

Definition 2. A continuously differentiable function $V : X \to \mathbb{R}_{\geq 0}$ is a *control Lyapunov function (CLF)* for (1) if there exists $c, k_1, k_2, \lambda \in \mathbb{R}_{>0}$ such that $\forall x \in X$:

$$\begin{aligned} k_1\|x\|^c \leq V(x) \leq k_2\|x\|^c \\ \inf_{u \in U} \dot{V}(x, u) \leq -\lambda V(x), \end{aligned} \tag{2}$$

where

$$\dot{V}(x, u) = \nabla V(x)(f(x) + g(x)u) \tag{3}$$

is the derivative of V along system (1).

Theorem 1 ([18]). *If V is a CLF for (1), then any locally Lipschitz continuous controller $u = k(x)$ satisfying*

$$\dot{V}(x, k(x)) \leq -\lambda V(x), \tag{4}$$

$\forall x \in X$ renders $x = 0$ exponentially stable.

Theorem 1 establishes that synthesizing a control input u while enforcing condition (4) achieves exponential stability.

B. Safety and Control Barrier Functions

We consider system (1) safe if its state $x(t)$ is contained in a *safe set* $S \subset X$ for all time, as stated below.

Definition 3. System (1) is *safe* w.r.t. S if S is forward invariant under (1), that is, $x_0 \in S \Rightarrow x(t) \in S, \forall t \geq 0$.

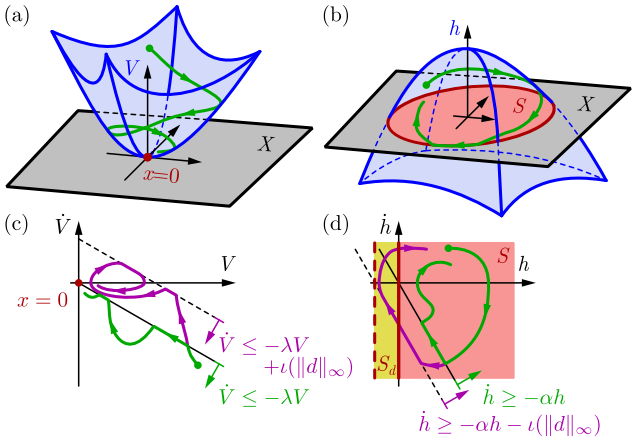

Fig. 2. (a) A CLF and a stable trajectory. (b) A CBF and a safe trajectory. While V is nonnegative, h may take any real value. (c) The stability condition (4) and a stable trajectory (green), the ISS condition (10) and an input-to-state stable trajectory (purple). For ISS the trajectory converges to a neighborhood of $x = 0$. (d) The safety condition (8) and a safe trajectory (green), the ISSf condition (12) and an input-to-state safe trajectory (purple). For ISSf a superset S_d of S is forward invariant.

The choice of the safe set is application-driven, e.g., it may represent positions where a robot does not collide with obstacles. Here, we define the safe set S as the 0-superlevel set of a continuously differentiable function $h : X \to \mathbb{R}$:

$$S = \{x \in X : h(x) \geq 0\}. \tag{5}$$

Then, control barrier functions (CBFs) can be used as tools to synthesize provably safe controllers in a similar fashion to how CLFs achieve stability.

Definition 4. A continuously differentiable function $h : X \to \mathbb{R}$ is a *control barrier function (CBF)* for (1) if there exists $\alpha \in \mathbb{R}_{>0}$ such that $\forall x \in S$: [1]

$$\sup_{u \in U} \dot{h}(x, u) \geq -\alpha h(x), \tag{6}$$

where

$$\dot{h}(x, u) = \nabla h(x)(f(x) + g(x)u) \tag{7}$$

is the derivative of h along system (1).

Theorem 2 ([6]). *If h is a CBF for (1), then any locally Lipschitz continuous controller $u = k(x)$ satisfying*

$$\dot{h}(x, k(x)) \geq -\alpha h(x), \tag{8}$$

$\forall x \in S$ renders (1) safe w.r.t. S.

Theorem 2 establishes safety-critical controller synthesis by condition (8). For example, a desired but not necessarily safe controller $k_d(x)$ can be modified in a minimally invasive way to a safe controller by solving the quadratic program:

$$\begin{aligned} k(x) = \operatorname*{argmin}_{u \in U} \; & (u - k_d(x))^\top (u - k_d(x)) \\ \text{s.t.} \; & \dot{h}(x, u) \geq -\alpha h(x). \end{aligned} \tag{9}$$

The Lipschitz continuity of this controller is discussed in [6].

[1] In general, α can be chosen as an extended class-$\mathcal{K}$ function, while here we use a constant for simplicity.

C. Effect of Disturbances

In practice, robotic systems are often subject to unknown disturbances that may compromise stability or safety. For example, a bounded disturbance $d \in \mathbb{R}^m$ added to the input u leads to the system $\dot{x} = f(x) + g(x)(u + d)$.

To address disturbances, the notion of exponential stability can be extended to *exponential input-to-state stability (ISS)* by modifying Definition 1. Namely, we require that there exists a class-$\mathcal{K}$ function μ such that $\|x_0\| < a \Rightarrow \|x(t)\| \leq Me^{-\beta t}\|x_0\| + \mu(\|d\|_1), \forall t \geq 0$. That is, solutions converge to a neighborhood of the origin which depends on the size of the disturbance. [19], [20] showed that exponential ISS is achieved by strengthening (4) in Theorem 1 to:

$$\dot{V}(x, u, d) \leq -\lambda V(x) + \iota(\|d\|_1), \tag{10}$$

for some class-$\mathcal{K}$ function ι.

Similarly, safety can be extended to *input-to-state safety (ISSf)* by requiring that the system stays within a neighborhood $S_d \supset S$ of the safe set S which depends on the size of the disturbance: $x_0 \in S_d \Rightarrow x(t) \in S_d, \forall t \geq 0$. We define this neighborhood as a 0-superlevel set:

$$S_d = \{x \in X : h(x) + \gamma(\|d\|_1) \geq 0\}, \tag{11}$$

with some class-$\mathcal{K}$ function γ. It was established in [21] that ISSf is guaranteed by replacing (8) in Theorem 2 with:

$$\dot{h}(x, u, d) \geq -\alpha h(x) - \iota(\|d\|_1), \tag{12}$$

for some class-$\mathcal{K}$ function ι.

III. MODEL-FREE SAFETY-CRITICAL CONTROL

Now consider robotic systems with configuration space $Q \subset \mathbb{R}^n$, configuration coordinates $q \in Q$, set of admissible inputs $U \subset \mathbb{R}^m$, control input $u \in U$, and dynamics:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu, \tag{13}$$

where $D(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ contains centrifugal and Coriolis forces, $G(q) \in \mathbb{R}^n$ involves gravity terms and $B \in \mathbb{R}^{n \times m}$ is the input matrix. $D(q)$ is symmetric, positive definite, $\dot{D}(q, \dot{q}) - 2C(q, \dot{q})$ is skew-symmetric. We consider control laws $k : Q \subset \mathbb{R}^n \to \mathbb{R}^m$, $u = k(q, \dot{q})$, initial conditions $q(0) = q_0$, $\dot{q}(0) = \dot{q}_0$, and assume that a unique solution $q(t)$ exists for all $t \geq 0$.

We consider the robotic system *safe* if its configuration q lies within a *safe set* S for all time: $q(t) \in S$, $t \geq 0$.

Assumption 1. The safe set is defined as the 0-superlevel set of a continuously differentiable function $h : Q \to \mathbb{R}$:

$$S = \{q \in Q : h(q) \geq 0\}, \tag{14}$$

where the gradient of h is finite: $\exists C_h \in \mathbb{R}_{>0}$ such that $\|\nabla h(q)\| \leq C_h, \forall q \in S$. That is, safety depends on the configuration q only and h is independent of $\dot{q}$.

Problem Statement. *For the robotic system (13), design a controller $u = k(q, \dot{q})$ that achieves safety with respect to set S in (14), i.e., $q(t) \in S, \forall t \geq 0$ given certain initial conditions $q_0 \in Q$ and $\dot{q}_0 \in \mathbb{R}^n$.*

A. Control Method

Following [15], [16], we seek to maintain safety by synthesizing and tracking a safe velocity. This reduces the complexity of safety-critical control significantly, while velocity tracking controllers are widely used [17]. The approach allows safety-critical control in a model-free fashion.

We synthesize the *safe velocity* $\dot{q}_s \in \mathbb{R}^n$ so that it satisfies:

$$\nabla h(q)\dot{q}_s \geq -\alpha h(q), \tag{15}$$

cf. (8), for some $\alpha \in \mathbb{R}_{>0}$ to be selected. The safe velocity $\dot{q}_s$ depends on the configuration q. Note that (15) is a kinematic condition that does not depend on the full dynamics (13).

To track the safe velocity, we define the tracking error:

$$\dot{e} = \dot{q} - \dot{q}_s. \tag{16}$$

and use a velocity tracking controller $u = k(q, \dot{q})$. First, we consider the scenario that u is able to drive the error $\dot{e}$ to zero exponentially, then we address the effect of disturbances.

Assumption 2. The velocity tracking controller $u = k(q, \dot{q})$ achieves exponentially stable tracking: $\|\dot{e}(t)\| \leq M\|\dot{e}_0\|e^{-\lambda t}$ for some $M, \lambda \in \mathbb{R}_{>0}$. That is, if $\dot{e}$ is differentiable ($\ddot{e}$, $\ddot{q}_s$ exist), there exists a continuously differentiable Lyapunov function $V : Q - \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that $\forall (q, \dot{e}) \in Q - \mathbb{R}^n$:

$$k_1\|\dot{e}\| \leq V(q, \dot{e}) \leq k_2\|\dot{e}\|, \tag{17}$$

for some $k_1, k_2 \in \mathbb{R}_{>0}$, and there exists $\lambda \in \mathbb{R}_{>0}$ such that $\forall (q, \dot{e}, \dot{q}, \ddot{q}_s) \in Q - \mathbb{R}^n - \mathbb{R}^n - \mathbb{R}^n$ u satisfies:

$$\dot{V}(q, \dot{e}, \dot{q}, \ddot{q}_s, u) \leq -\lambda V(q, \dot{e}), \tag{18}$$

cf. (4). For exposition's sake, below we assume $\ddot{q}_s$ exists and we use (18). This assumption is relaxed later in Remark 4.

Before discussing its safety guarantees, we demonstrate the applicability of this method on a motivating example.

Example 1 (Double integrator system)**.** Here we revisit the example in [15]. As the simplest instantiation of (13), consider a double integrator system in two dimensions:

$$\ddot{q} = u, \tag{19}$$

where $q \in \mathbb{R}^2$ is the planar position of the robot and $u \in \mathbb{R}^2$. Our goal is to navigate the system from a start position q_0 to a goal q_g while avoiding obstacles. A simple solution is to realize the desired velocity $\dot{q}_d = -K_P(q - q_g)$ that is based on a proportional controller with gain $K_P \in \mathbb{R}_{>0}$.

We can avoid an obstacle of radius r centered at q_o by the help of the distance $d = \|q - q_o\|$ and the CBF:

$$h(q) = d - r, \tag{20}$$

with gradient $\nabla h(q) = (q - q_o)^\top / \|q - q_o\| = n_o^\top$ equal to the unit vector n_o pointing from the obstacle to the robot. Then, the safe velocity can be found by using condition (15). Specifically, we modify the desired velocity $\dot{q}_d$ in a minimally invasive fashion by solving the quadratic program:

$$\begin{aligned} \underset{\dot{q}_s \in \mathbb{R}^2}{\arg\min}\ & (\dot{q}_s - \dot{q}_d)^\top (\dot{q}_s - \dot{q}_d) \\ \text{s.t.}\ & n_o^\top \dot{q}_s \geq -\alpha(d - r), \end{aligned} \tag{21}$$

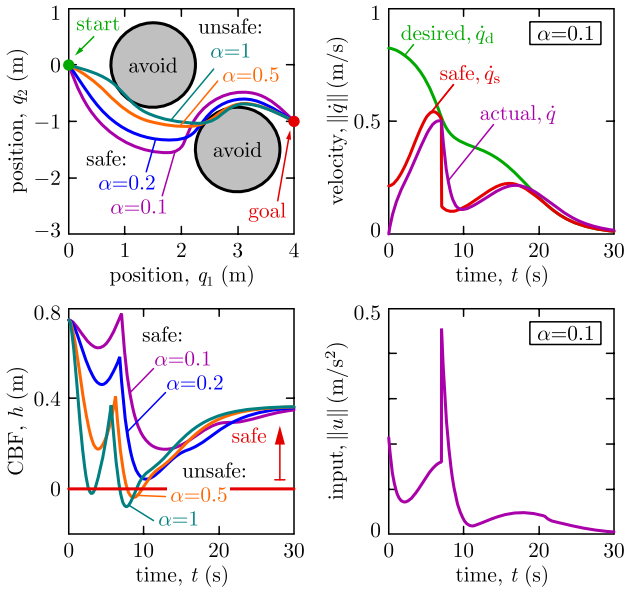

Fig. 3. Numerical simulation of the double integrator system (19) tracking the safe velocity (22). The controller is able to keep the system safe if parameter α is selected to be small enough.

cf. (9). Based on the KKT conditions [22], it has the solution:

$$\dot{q}_s = \dot{q}_d + \max\{0, -n_o^\top \dot{q}_d - \alpha(d - r)\}\, n_o. \tag{22}$$

The safe velocity can be tracked for example by the controller $u = -K_D(\dot{q} - \dot{q}_s)$ with gain $K_D \in \mathbb{R}_{>0}$.

Fig. 3 shows four simulation results for avoiding two obstacles with parameters $K_P = 0.2\,\mathrm{s}^{-1}$, $K_D = 1\,\mathrm{s}^{-1}$ and $\alpha = 0.1, 0.2, 0.5$ and $1\,\mathrm{s}^{-1}$, respectively. With the proposed approach, the double integrator system avoids the obstacles, although the second-order dynamics was not directly taken into account during the CBF and control design. The condition for safety, however, is picking a small enough α value (e.g. 0.1 or 0.2), while safety is violated for larger α (e.g. 0.5 or 1). We remark that for multiple obstacles we considered the closest one at each time. This results in a nonsmooth CBF which has been analyzed in [23]. Accordingly, the safe velocity $\dot{q}_s$ is only piecewise differentiable; for simplicity, our constructions are restricted to the differentiable segments.

B. Main Result

In what follows, our main result proves that tracking the safe velocity achieves safety for the full dynamics if parameter α is selected to be small enough. Specifically, for tracking controllers satisfying Assumption 2 stability translates into safety for the full system (13) if $\lambda > \alpha$. As this result is agnostic to the application domain, this culminates in *model-free safety-critical control*. Realizing velocity tracking controllers, however, depends on the application. Later we give examples for such controllers and corresponding CLFs.

The following theorem summarizes the safety guarantees provided by tracking the safe velocity.

Theorem 3. *Consider system (13), safe set (14), safe velocity satisfying (15), and velocity tracking controller satisfying (18). If $\lambda > \alpha$, safety is achieved such that $(q_0, \dot{e}_0) \in S_V \Rightarrow q(t) \in S, \forall t \geq 0$, where:*

$$\begin{aligned} S_V &= \{(q,\dot{e}) \in Q \times \mathbb{R}^n : h_V(q,\dot{e}) \geq 0\}, \\ h_V(q,\dot{e}) &= -V(q,\dot{e}) + \alpha_e h(q), \end{aligned} \tag{23}$$

with $\alpha_e = (\lambda - \alpha)k_1/C_h > 0$ and C_h, k_1 defined at (14, 17).

Proof. Since $V(q,\dot{e}) \geq 0$, the implication $h_V(q,\dot{e}) \geq 0 \Rightarrow h(q) \geq 0$ holds. Thus, $h_V(q(t),\dot{e}(t)) \geq 0, \forall t \geq 0$ is sufficient to prove. We prove this by noticing that the initial conditions satisfy $h_V(q_0,\dot{e}_0) \geq 0$ and we also have:

$$\begin{aligned} \dot{h}_V(q,\dot{e},\dot{q},\ddot{q}_s,u) &= -\dot{V}(q,\dot{e},\dot{q},\ddot{q}_s,u) + \alpha_e \nabla h(q)\dot{q} \\ &\geq \lambda V(q,\dot{e}) + \alpha_e \nabla h(q)\dot{q}_s + \alpha_e \nabla h(q)\dot{e} \\ &\geq \lambda V(q,\dot{e}) - \alpha_e \alpha h(q) + \alpha_e \nabla h(q)\dot{e} \\ &\geq (\lambda - \alpha)V(q,\dot{e}) - \alpha_e k \nabla h(q) k k\dot{e} k - \alpha h_V(q,\dot{e}) \\ &\geq (\lambda - \alpha)k_1 k\dot{e} k - \alpha_e C_h k\dot{e} k - \alpha h_V(q,\dot{e}) \\ &\geq -\alpha h_V(q,\dot{e}). \end{aligned} \tag{24}$$

Here we used the following properties in the 6 lines of the inequality: (i) definition (23) of h_V, (ii) stability condition (18) and definition (16) of $\dot{e}$, (iii) condition (15) on the safe velocity, (iv) definition (23) of h_V and the Cauchy-Schwartz inequality, (v) lower bound of V in (17) and upper bound C_h of $k\nabla h(q)k$, (vi) definition of α_e. This guarantees $h_V(q(t),\dot{e}(t)) \geq 0$, $\forall t \geq 0$ by Theorem 2. □

Remark 1. Condition $\lambda > \alpha$ means the controller tracks the safe velocity fast enough (characterized by λ) compared to how fast the boundary of the safe set may be approached (characterized by α). In practice, one can pick a small enough α for a given velocity tracking controller, for example, by gradually increasing α from 0. The existence of such α is guaranteed by the Theorem. Note that there is a trade-off: for smaller α the system may become more conservative, evolving farther from the boundary of the safe set.

Remark 2. Condition (15) is equivalent to designing a safe control input $\dot{q}_s$ for the single integrator system $\dot{q} = \dot{q}_s$. Thus, this approach is a manifestation of control based on reduced-order models. While h is a CBF for the reduced-order model, h_V is a CBF for the full system (13) as a dynamic extension of h, similar to the energy-based extension in [16]. Other reduced-order models of the form $\dot{q} = A(q)\mu_s$ with control input $\mu_s \in \mathbb{R}^k$ and transformation $A(q) \in \mathbb{R}^{n \times k}$ can also be used. This, for example, includes the unicycle model for wheeled robots with $q = (x, y, \psi) \in \mathbb{R}^3$ containing Cartesian positions and yaw angle and $\mu_s = (v_s, \omega_s) \in \mathbb{R}^2$ containing forward velocity and yaw rate:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_s \\ \omega_s \end{bmatrix}. \tag{25}$$

The safe velocity μ_s is given by $\nabla h(q)A(q)\mu_s \geq -\alpha h(q)$ based on (15), and the proof of Theorem 3 holds with substitution $\dot{q}_s = A(q)\mu_s$. The tracking controller u, however, must provide property (18) with respect to $\dot{e} = \dot{q} - A(q)\mu_s$.

Remark 3. Theorem 3 requires initial conditions to satisfy $(q_0, \dot{e}_0) \in S_V \Leftrightarrow h(q_0) \geq V(q_0,\dot{e}_0)/\alpha_e$. This is a stricter

condition than $q_0 \in S$ ( ) $h(q_0) \geq 0$ that is usually required in safety-critical control (cf. Definition 3). The additional conservatism is reduced when the initial tracking error $\dot{e}_0$ is smaller (since $V(q_0, \dot{e}_0)$ is smaller) and when the tracking is faster, i.e., $\lambda \uparrow \alpha$ is larger (since α_e is larger).

Remark 4. The error $\dot{e}$ is assumed to be differentiable in Assumption 2 only for exposition's sake. Theorem 3 can be extended to non-differentiable signals satisfying $\|\dot{e}(t)\| \leq M\|\dot{e}_0\|e^{-\lambda t}$. The proof relies on the fact that $\dot{h}(q, \dot{q}) \geq -\alpha h(q) - C_h M\|\dot{e}_0\|e^{-\lambda t}$ holds, and by the comparison lemma with $\dot{y}(t) = -\alpha y(t) - C_h M\|\dot{e}_0\|e^{-\lambda t}$, $y(0) = h(q_0)$ one can show that $h(q(t)) \geq y(t) \geq 0$.

C. Effect of Disturbances

Now consider that ideal exponential tracking of the safe velocity is not possible. This can be captured via a bounded input disturbance d, that represents the effect of imperfect tracking controllers, time delays or modeling errors. Then, instead of safety, one shall guarantee input-to-state safety (ISSf), i.e., the invariance of the larger set $S_d \supset S$:

$$\begin{aligned} S_d &= \{q \in Q : h_d(q) \geq 0\}, \\ h_d(q) &= h(q) + \gamma(\|d\|_1), \end{aligned} \tag{26}$$

where γ is a class-$\mathcal{K}$ function to be specified. We also introduce the dynamic extension $S_{Vd} \supset S_V$ of set S_d:

$$\begin{aligned} S_{Vd} &= \{(q, \dot{e}) \in Q \times \mathbb{R}^n : h_{Vd}(q, \dot{e}) \geq 0\}, \\ h_{Vd}(q, \dot{e}) &= h_V(q, \dot{e}) + \gamma(\|d\|_1). \end{aligned} \tag{27}$$

We show that ISSf is guaranteed by input-to-state stable (ISS) tracking: $\|\dot{e}(t)\| \leq M\|\dot{e}_0\|e^{-\lambda t} + \mu(\|d\|_1)$. Note that exponential ISS is our strongest assumption. When the tracking is poor, $\mu(\|d\|_1)$ dominates this bound. If the error does not decay ($M = 0$), the bound reduces to $\|\dot{e}(t)\| \leq \|\dot{e}\|_1$ and we recover the traditional ISSf guarantees in [21]. For ISS, instead of (18) the tracking controller shall satisfy:

$$\dot{V}(q, \dot{e}, \dot{q}, \ddot{q}_s, u, d) \leq -\lambda V(q, \dot{e}) + \iota(\|d\|_1), \tag{28}$$

for some class-$\mathcal{K}$ function ι. The connection between ISS and ISSf is summarized in the following Corollary of Theorem 3.

Corollary 1. *Consider system (13), sets S_d and S_{Vd} in (26) and (27), safe velocity satisfying (15), and velocity tracking controller satisfying (28). If $\lambda > \alpha$, input-to-state safety is achieved such that $(q_0, \dot{e}_0) \in S_{Vd}$) $q(t) \in S_d$, $\forall t \geq 0$, where α_e is given in Theorem 3 and $\gamma(\|d\|_1) = \iota(\|d\|_1)/\alpha$.*

The proof follows the same steps as those in the proof of Theorem 3, by replacing h and h_V with h_d and h_{Vd}. Corollary 1 concludes that input-to-state stable tracking of a safe velocity implies input-to-state safety for the full system, i.e., robust velocity tracking implies robust safety guarantees.

D. Velocity Tracking Controllers

Finally, we consider examples of velocity tracking controllers that provide stability by (18) or ISS by (28). As the simplest choice, we consider a model-free D controller:

$$u = -K_D \dot{e}, \tag{29}$$

where $K_D \in \mathbb{R}^{m \times n}$ is selected so that $K = BK_D$ is positive definite. Furthermore, when model-dependent terms are well-known, they can also be included in the control law. If $n = m$ and B is invertible (i.e., the system is fully actuated), one may use a D controller with gravity compensation:

$$u = B^{-1}(G(q) - K\dot{e}), \tag{30}$$

with a positive definite gain $K \in \mathbb{R}^{n \times n}$. Moreover, one can also use a heavily model-dependent extension:

$$u = B^{-1}(D(q)\ddot{q}_s + C(q, \dot{q})\dot{q}_s + G(q) - K\dot{e}). \tag{31}$$

While this controller may achieve better tracking, it requires $D(q)$ and $C(q, \dot{q})$ which may have complicated expressions and may be expensive to compute in practice.

We characterize these controllers by the constant $\lambda \in \mathbb{R}_{>0}$:

$$\lambda = \frac{\sigma_{\min}(K)}{\sup_{q \in Q} \sigma_{\max}(D(q))}, \tag{32}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest eigenvalue. The eigenvalues are positive real numbers due to the positive definiteness of $D(q)$ and K. Accordingly, λ represents the smallest gain divided by the largest inertia, hence characterizes how fast controllers may track. We associate the controllers with the Lyapunov function candidate:

$$V(q, \dot{e}) = \sqrt{\frac{1}{2}\dot{e}^\top D(q)\dot{e}}, \tag{33}$$

that has the bound (17) with $k_1 = \inf_{q \in Q} \sqrt{\sigma_{\min}(D(q))/2}$ and $k_2 = \sup_{q \in Q} \sqrt{\sigma_{\max}(D(q))/2}$. We also define the linear class-$\mathcal{K}$ function $\iota(\|d\|_1) = \|d\|_1/(2k_1)$.

With the above controllers, the parameters to be selected during control design are α and K_D or K. Now we state that these controllers satisfy the required stability properties.

Proposition 1. *Consider system (13), Lyapunov function V defined by (33), constant λ given by (32) and $\dot{e} \neq 0$.*

(i) *Controller (29) satisfies the ISS condition (28) with respect to $d = -D(q)\ddot{q}_s - C(q, \dot{q})\dot{q}_s - G(q)$.*
(ii) *Controller (30) satisfies the ISS condition (28) with respect to $d = -D(q)\ddot{q}_s - C(q, \dot{q})\dot{q}_s$ when $\dot{q}_s \neq 0$ and the stability condition (18) when $\dot{q}_s = 0$.*
(iii) *Controller (31) satisfies the stability condition (18).*

Proof. The proof follows that in Section 8.2 of [17]. Here we prove case (i) only. The proof of case (ii) is the same when $\dot{q}_s \neq 0$, whereas the proofs of case (ii) when $\dot{q}_s = 0$ and case (iii) can be obtained by substituting $d = 0$.

We differentiate V given by (33):

$$\dot{V}(q, \dot{e}, \dot{q}, \ddot{q}_s, u, d) = \frac{1}{2V(q, \dot{e})}\left(\frac{1}{2}\dot{e}^\top \dot{D}(q, \dot{q})\dot{e} + \dot{e}^\top D(q)\ddot{e}\right), \tag{34}$$

and substitute the error dynamics corresponding to (13, 16):

$$D(q)\ddot{e} = -C(q, \dot{q})\dot{e} - D(q)\ddot{q}_s - C(q, \dot{q})\dot{q}_s - G(q) + Bu. \tag{35}$$

For controller (29) this leads to:

$$\dot{V}(q, \dot{e}, \dot{q}, \ddot{q}_s, u, d) = \frac{-\dot{e}^\top K\dot{e} + \dot{e}^\top d}{2V(q, \dot{e})}, \tag{36}$$

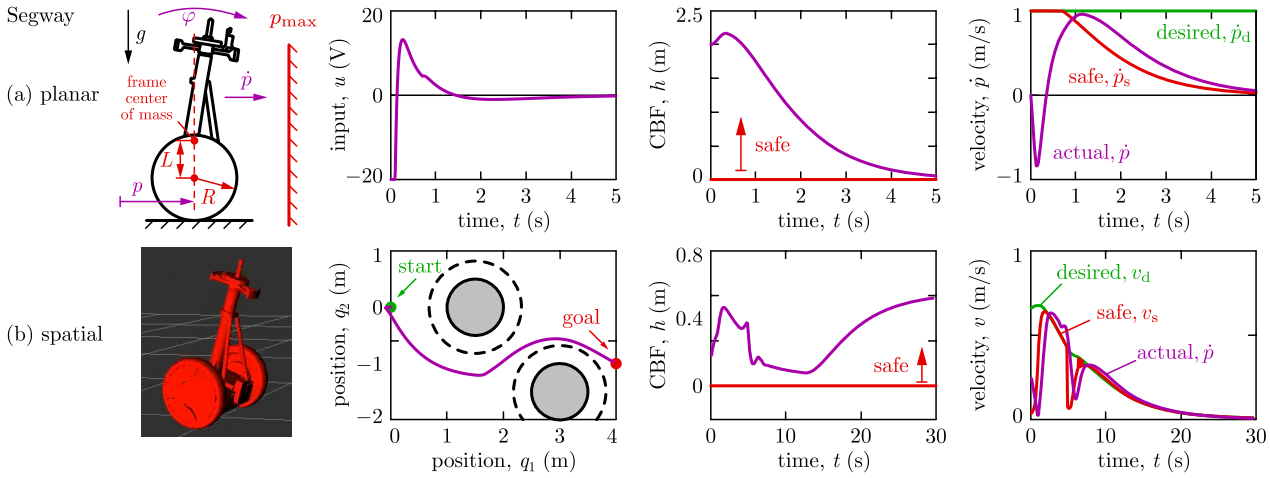

Fig. 4. High-fidelity simulation of a Ninebot E+ Segway platform. (a) Planar dynamical model (13, 39) with the model-free safety-critical controller (41, 42). (b) Spatial dynamical model with the model-free controller (44, 45). The controllers keeps the system safe (the CBF h is positive for all time).

where the term $\dot{e}^{\top}(\dot{D}(q,\dot{q}) \quad 2C(q,\dot{q}))\dot{e}$ dropped since $\dot{D}(q,\dot{q}) \quad 2C(q,\dot{q})$ is skew-symmetric.

Based on (36), now we show (28) holds. Since (32) implies $\dot{e}^{\top}K\dot{e} \quad \lambda\dot{e}^{\top}D(q)\dot{e} \quad 0$, the definition (33) of V leads to:

$$\frac{\dot{e}^{\top}K\dot{e}}{2V(q,\dot{e})} \qquad \lambda V(q,\dot{e}), \tag{37}$$

$8q \; 2 \; Q, \dot{e} \; 2 \; \mathsf{R}^n$. Furthermore, the Cauchy-Schwartz inequality, the bound (17) on V and the definition of ι yield:

$$\frac{\dot{e}^{\top}d}{2V(q,\dot{e})} \qquad \frac{k\dot{e}kkdk_1}{2k_1k\dot{e}k} = \iota(kdk_1), \tag{38}$$

where $k\dot{e}k$ drops, making the right-hand side independent of time. Substituting (37, 38) into (36) yields (28). □

IV. APPLICATIONS TO WHEELED, FLYING AND LEGGED ROBOTS

Now we apply the proposed control method to robotic platforms, including high-fidelity simulations of a Segway and hardware experiments on a Drone and a Quadruped.

A. Numerical Simulation of Segway

We consider a Ninebot E+ Segway platform with its planar and spatial high-fidelity dynamical models described in [25].

Example 2 (Segway in plane)**.** Consider the two-degrees of freedom planar Segway model in Fig. 4(a) with configuration $q = [p, \varphi]^{\top} \; 2 \; Q = \mathsf{R} \quad [0, 2\pi]$ including the position p and pitch angle φ. The dynamics are in form (13), where:

$$D(q) = \begin{bmatrix} m_0 & mL\cos\varphi \\ mL\cos\varphi & J_0 \end{bmatrix}, \quad G(q) = \begin{bmatrix} 0 \\ mgL\sin\varphi \end{bmatrix},$$
$$C(q,\dot{q}) = \begin{bmatrix} b_t/R & b_t & mL\dot{\varphi}\sin\varphi \\ b_t & & b_tR \end{bmatrix}, \quad B = \begin{bmatrix} K_m/R \\ K_m \end{bmatrix}, \tag{39}$$

with parameters given in Table I and $u \; 2 \; U = [\quad 20, 20]\,\mathrm{V}$.

TABLE I
PARAMETERS OF THE SEGWAY MODEL

Description	Parameter	Value	Unit
gravitational acceleration	g	9.81	m/s^2
radius of wheels	R	0.195	m
mass of wheels	M	2 2.485	kg
mass moment of inertia of wheels	J_C	2 0.0559	kgm^2
distance of wheel center to frame CoM	L	0.169	m
mass of frame	m	44.798	kg
mass moment of inertia of frame	J_G	3.836	kgm^2
lumped mass $m_0 = m + M + J_C/R^2$	m_0	52.710	kg
lumped inertia $J_0 = mL^2 + J_G$	J_0	5.108	kgm^2
torque constant of motors	K_m	2 1.262	Nm/V
damping constant of motors	b_t	2 1.225	Ns

Our goal is to realize a desired forward velocity $\dot{p}_d$ until reaching a wall at position p_{max}, then stop automatically and safely in front of the wall. This is captured by the CBF:

$$h(q) = p_{max} \quad p, \tag{40}$$

which, by condition (15), leads to the safe forward velocity:

$$\dot{p}_s = \min f\dot{p}_d, \alpha(p_{max} \quad p)g, \tag{41}$$

similar to (22). This safe velocity is tracked by the controller:

$$u = K_p(\dot{p} \quad \dot{p}_s) + K_\varphi\varphi + K_{\dot{\varphi}}\dot{\varphi} \tag{42}$$

with $K_p = 50\,\mathrm{Vs/m}$, $K_\varphi = 150\,\mathrm{V/rad}$, $K_{\dot{\varphi}} = 40\,\mathrm{Vs/rad}$, which also stabilizes the Segway to the upright position.

Fig. 4(a) shows simulation results where the Segway executes the task starting from $p_0 = 0$, $\varphi_0 = \quad 0.138\,\mathrm{rad}$ (where its frame is vertical), $\dot{p}_0 = 0$, $\dot{\varphi}_0 = 0$, for $\dot{p}_d = 1\,\mathrm{m/s}$, $p_{max} = 2\,\mathrm{m}$ and $\alpha = 0.5\,\mathrm{s}$ [1]. Notice that controller (41, 42) is model-free, it does not rely on the full dynamics (13, 39). The gains K_p, K_φ and $K_{\dot{\varphi}}$, however, are tuned so that the full dynamics achieves stable velocity tracking. These gains were tuned based on linearization and LQR in [25] and they determine the tracking performance with the associated λ.

Example 3 (Segway in space)**.** Consider the spatial model of the Segway in Fig. 4(b) with 7-dimensional state space and

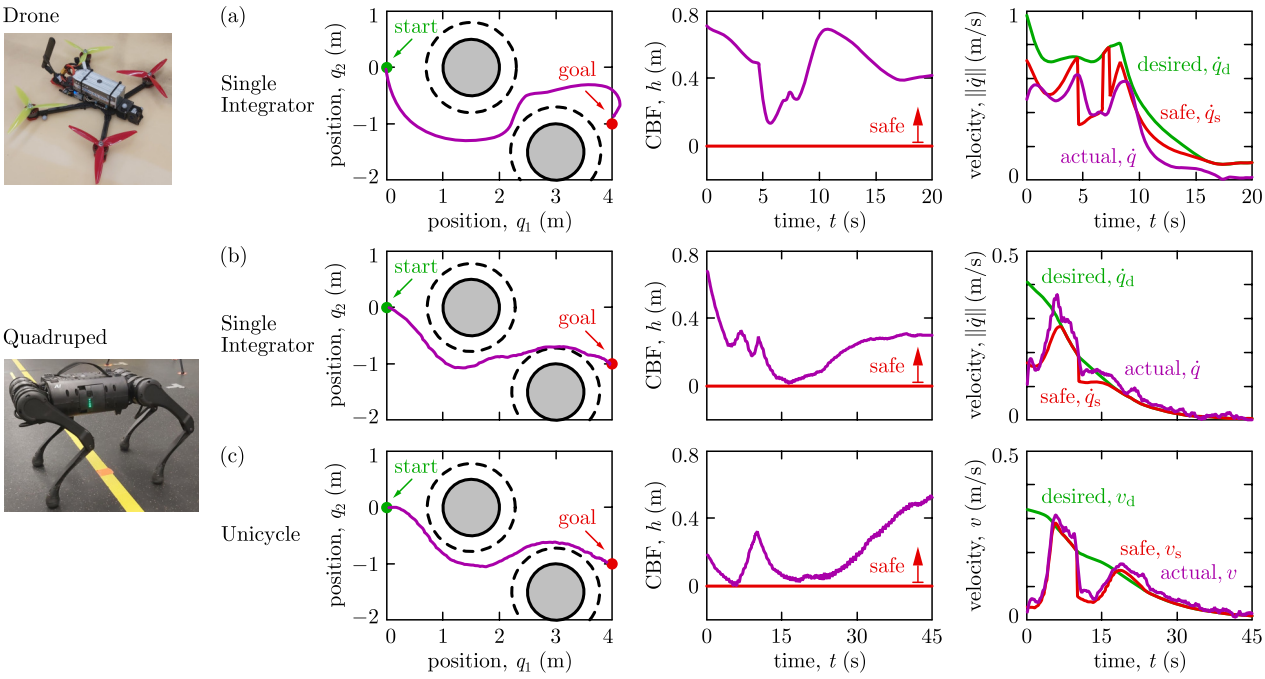

Fig. 5. Hardware experiments using the proposed model-free safety-critical control method. An obstacle avoidance task is accomplished by two fundamentally different robots: a custom-made racing Drone (top) and a Unitree A1 Quadruped (bottom). (a) The Drone is tracking a safe velocity determined based on single integrator model. (b) The Quadruped is tracking a safe velocity based on single integrator model via side-stepping and (c) based on unicycle model via turning. Both robots executed the task with guaranteed safety. A video of the experiments can be found at [24].

2 control inputs. The task is to navigate it from a start point to a goal (left panel) while avoiding obstacles of radius 0.5 m (solid black), similar to Example 1. The obstacle radius is buffered by the size of the Segway (dashed black) and the Segway's center must be kept outside this zone.

This task is accomplished by tracking a safe velocity obtained for the unicycle model (25); cf. Remark 2. We set the desired forward velocity and yaw rate $\mu_d = (v_d, \omega_d)$ based on the distance $d_g = \|(x_g \quad x, y_g \quad y)\|$ to the goal as $v_d = K_v d_g$ and $\omega_d = \quad K_\omega(\sin\psi \quad (y_g \quad y)/d_g)$. To avoid obstacles, we use a CBF that includes the heading direction:

$$h(q) = d \quad r \quad \delta\cos(\psi \quad \theta), \tag{43}$$

where $d = \|(x_o \quad x, y_o \quad y)\|$ is the distance from the obstacle, $\theta = \arctan((y_o \quad y)/(x_o \quad x))$ is the angle towards the obstacle, and $\delta \in \mathbb{R}_{>0}$ is a tunable parameter.

This CBF is incorporated into the quadratic program:

$$\begin{aligned} \underset{\mu_s \in \mathbb{R}^2}{\operatorname{argmin}} \ & (\mu_s \quad \mu_d)^\top \Gamma(\mu_s \quad \mu_d) \\ \text{s.t.} \ & r\, h(q) A(q) \mu_s \quad \alpha h(q), \end{aligned} \tag{44}$$

cf. (21), where $\Gamma = \operatorname{diag}\{1, R^2\}$ is a weight between forward velocity and yaw rate with parameter $R \in \mathbb{R}_{>0}$. The resulting safe velocity $\mu_s = (v_s, \omega_s)$ is tracked by the controller:

$$u_{1,2} = K_p(\dot{p} \quad v_s) + K_\varphi \varphi + K_{\dot\varphi}\dot\varphi \quad K_\psi(\dot\psi \quad \omega_s) \tag{45}$$

used at the two wheels with the same gains as in Example 2 and a gain $K_\psi = 10\,\mathrm{Vs/rad}$ on the yaw rate $\dot\psi$.

With this approach, the Segway is able to move to the goal safely, while its controller (44, 45) is model-free. Fig. 4(b) shows the safe motion for $K_v = 0.16\,\mathrm{s}^{\ 1}$, $K_\omega = 0.8\,\mathrm{s}^{\ 1}$, $\alpha = 0.2\,\mathrm{s}^{\ 1}$, $\delta = 0.5\,\mathrm{m}$ and $R = 0.25\,\mathrm{m}$.

B. Hardware Experiments on Drone and Quadruped

We executed the obstacle avoidance task of Example 3 on two fundamentally different hardware platforms: a Drone and a Quadruped; see Fig. 5. The obstacle locations were known to the robots, sensory information was used to determine the robots' position only. We performed two classes of experiments: by synthesizing safe velocities based on the single integrator and unicycle models, respectively; cf. Remark 2. A video of the experiments can be found at [24].

First, we considered the single integrator model, and we tracked the associated safe velocity with the Drone and the Quadruped by platform-specific tracking controllers. We used CBF (20) and safe velocity (22). The desired velocity was $\dot{q}_d = \quad K_P(q \quad q_g)$ with saturation; cf. Example 1.

The Drone was a custom-built racing drone [26], shown in Fig. 5(a). It has 6 degrees of freedom and 4 actuators. The state of the Drone (position, orientation and corresponding velocities) were measured by IMU and an OptiTrack motion capture system. State estimation and control action computation ran at 400 Hz. The safe velocity was commanded to the drone wireless from a desktop computer, while velocity tracking was done using an on-board betaflight flight controller. The safe velocity was calculated with $K_P = 0.7\,\mathrm{s}^{\ 1}$ and $\alpha = 0.2\,\mathrm{s}^{\ 1}$. Fig. 5(a) shows the Drone reaching the goal safely, as guaranteed by Theorem 3 since α was selected small enough for the available tracking performance. The value of α was chosen based on the simulated response of the single integrator. α was not tuned for optimal performance, and it could potentially be increased for less conservatism.

The Quadruped was a Unitree A1 quadrupedal robot, shown in Fig. 5(b), which has 18 degrees of freedom and 12 actuators.

Its position was measured based on odometry assuming the feet do not slip, while joint states were available via built-in encoders. An ID-QP walking controller was realized at 1 kHz loop rate on this robot to track a stable walking gait with prescribed forward and lateral velocities and yaw rate, designed using the concepts in [27]. Individual commands were tracked via a motion primitive framework described in [28]. In the single integrator experiments, the yaw rate was set to zero, while the safe velocity (22) with $K_{\mathsf{P}} = 0.1\,\mathrm{s}^{\ 1}$ and $\alpha = 0.2\,\mathrm{s}^{\ 1}$ was tracked by forward- and side-stepping. The Quadruped executed the task safely similar to the Drone (see Fig. 5(b)), although it has fundamentally different dynamic behavior. This indicates the application-agnostic nature of our model-free approach.

Finally, we used the unicycle model (25) and CBF (43) to achieve safety on the Quadruped. The safe forward velocity and yaw rate in (44) were tracked by the same ID-QP walking controller. Fig. 5(c) shows the Quadruped traversing the obstacle course with $K_v = 0.08\,\mathrm{s}^{\ 1}$, $K_\omega = 0.4\,\mathrm{s}^{\ 1}$, $\alpha = 0.2\,\mathrm{s}^{\ 1}$, $\delta = 0.5\,\mathrm{m}$ and $R = 0.5\,\mathrm{m}$. While safety is maintained, the Quadruped performs the task with different behavior than in the previous experiment: it walks forward and turns instead of forward- and side-stepping. Still, safety is provably guaranteed — and in a model-free fashion.

V. CONCLUSIONS

We considered safety-critical control for robotic systems in a model-free fashion following [15]. Our control method relies on a synthesizing a safe velocity using control barrier functions and tracking this velocity. We stated and proved theoretical guarantees for the safety of our method. Namely, safety is achieved when the safe velocity is tracked faster than how fast the corresponding safe motion may approach the boundary of the safe set. Due to its model-free nature, our approach is application-agnostic. By simulation and hardware experiments we demonstrated that it works for various robots such as a Segway, a Drone and a Quadruped.

While our method does not rely on the full dynamical model of the robot to achieve safety, it relies on kinematic models such as the single integrator or unicycle models. Our future work includes further exploration of safety-critical control based on reduced-order models beyond simple kinematic ones. We also plan to study how to relax the assumption on the performance of the velocity tracking controller.

REFERENCES

[1] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," *arXiv preprint*, no. arXiv:2103.14252, 2021.

[2] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 1934–1940.

[3] 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.

[4] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

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

[6] 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.

[7] E. Squires, R. Konda, S. Coogan, and M. Egerstedt, "Model free barrier functions via implicit evading maneuvers," *arXiv preprint*, no. arXiv:2107.12871, 2021.

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

[9] P. Seiler, M. Jankovic, and E. Hellstrom, "Control barrier functions with unmodeled dynamics using integral quadratic constraints," *arXiv preprint*, no. arXiv:2108.10491, 2021.

[10] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust tracking with model mismatch for fast and safe planning: An SOS optimization approach," in *International Workshop on the Algorithmic Foundations of Robotics*, M. Morales, L. Tapia, G. Sánchez-Ante, and S. Hutchinson, Eds., 2020, pp. 545–564.

[11] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013.

[12] S. Zhao and Z. Sun, "Defend the practicality of single-integrator models in multi-robot coordination control," in *IEEE International Conference on Control Automation*, 2017, pp. 666–671.

[13] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *Lecture Notes in Control and Information Sciences*, S. Nicosia, S. B., A. Bicchi, and P. Valigi, Eds. Berlin: Springer, 2001, vol. 270, pp. 181–226.

[14] D. Koung, I. Fantoni, O. Kermorgant, and L. Belouaer, "Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system," in *International Conference on Control, Automation, Robotics and Vision*, 2020, pp. 92–97.

[15] A. Singletary, K. Klingebiel, J. R. Bourne, N. A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[16] A. Singletary, S. Kolathaya, and A. D. Ames, "Safety-critical kinematic control of robotic systems," *IEEE Control Systems Letters*, vol. 6, pp. 139–144, 2022.

[17] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: John Wiley and Sons, 2005.

[18] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River: Prentice Hall, 2002.

[19] E. D. Sontag and Y. Wang, "On characterizations of input-to-state stability with respect to compact sets," in *Nonlinear Control Systems Design*. Elsevier, 1995, pp. 203–208.

[20] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*. Springer, 2008, pp. 163–220.

[21] 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.

[22] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[23] P. Glotfelter, J. Cortes, and M. Egerstedt, "A nonsmooth approach to controller synthesis for Boolean specifications," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[24] Supplementary video: https://youtu.be/vNcc5vgswx0.

[25] 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.

[26] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robotics and Automation Letters*, 2021, submitted.

[27] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 814–820.

[28] W. Ubellacker, N. Csomay-Shanklin, T. G. Molnar, and A. D. Ames, "Verifying safe transitions between dynamic motion primitives on legged robots," *arXiv preprint*, no. arXiv:2106.10310, 2021.