Control Barrier Functions for Cyber-Physical Systems and Applications to NMPC

Jan Schilliger, Thomas Lew, Spencer M. Richards, Severin Hänggi, Marco Pavone, and Christopher Onder

***Abstract*—Tractable safety-ensuring algorithms for cyber-physical systems are important in critical applications. Approaches based on Control Barrier Functions assume continuous enforcement, which is not possible in an online fashion. This letter presents two tractable algorithms to ensure forward invariance of discrete-time controlled cyber-physical systems. Both approaches are based on Control Barrier Functions to provide strict mathematical safety guarantees. The first algorithm exploits Lipschitz continuity and formulates the safety condition as a robust program which is subsequently relaxed to a set of affine conditions. The second algorithm is inspired by tube-NMPC and uses an affine Control Barrier Function formulation in conjunction with an auxiliary controller to guarantee safety of the system. We combine an approximate NMPC controller with the second algorithm to guarantee strict safety despite approximated constraints and show its effectiveness experimentally on a mini-Segway.**

***Index Terms*—Optimization and optimal control, robot safety, control barrier functions, nonlinear model predictive control.**

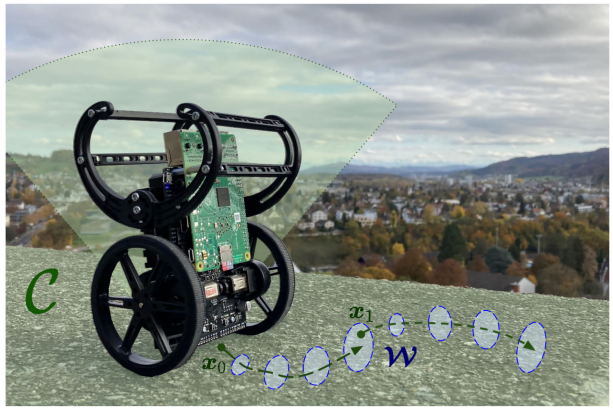

Fig. 1. Safety-critical systems require fast and tractable control algorithms. These should stabilize the system, while always remaining safe. For instance, the mini-Segway shown in the figure must follow a reference position without falling down, i.e., satisfy safety bounds on the pitch angle at all times.

I. Introduction

TWO cornerstones of safety-critical control for robotic systems are "stability" (i.e., convergence towards desired behavior) and "safety" (i.e., remaining within a designated set of safe states). A key challenge to deployment of cyber-physical systems is the design of fast, tractable control algorithms with *safety guarantees* that hold in the face of real-world challenges such as discretization error. For instance, as shown in Fig. 1, a mini-Segway must maintain a bounded pitch angle at all times to avoid falling over.

Nonlinear Model Predictive Control (NMPC) is a promising method for safety-critical control. In NMPC, a control input is obtained as the first input of an optimal trajectory computed by a constrained optimization at each time step. Indeed, the NMPC problem can encode safety with state and input constraints, but these are generally only enforced at discrete time steps. One approach to obtain continuous-time constraint enforcement is to use tube NMPC, where an auxiliary control law keeps arising disturbances and discretization errors in an invariant tube [1]. However, NMPC is computationally demanding and thus remains an open problem for systems with pronounced nonlinearities, high dimensionality, or fast response times [2]. To address this, an NMPC problem can be replaced with an approximation that is easier to solve [3]–[5]. For example, the Real-Time Iteration (RTI) scheme approximates an NMPC problem with a Quadratic Program (QP) at each time-step [6]. While RTI has been shown to be stabilizing, strict constraint satisfaction, recursive feasibility, and hence safety are no longer guaranteed, due to linearly approximated constraints [5], [7].

A popular tactic in ensuring both safety and stability is to de-couple the two tasks [8]–[11]. To this end, a stabilizing controller can be applied jointly with some form of safety certificate function, such as a Control Barrier Function (CBF). Specifically, for nonlinear control-affine systems, CBFs can be used to reformulate nonlinear, nonconvex constraints as affine constraints point-wise in time. CBFs have been applied to stochastic [12], [13] and hybrid systems [14], as a part of control Lyapunov function-based controllers [14]–[16], and in dedicated safety filters [11]. However, the notion of safety in CBFs relies on the assumption that one has continuous access to the system states and has the ability to continuously modify the states. However, these assumptions do not hold in systems with discrete-time control schemes [17]. An extension of CBFs to such systems is

Manuscript received April 28, 2021; accepted August 20, 2021. Date of publication September 13, 2021; date of current version September 30, 2021. The work of Jan Schilliger was supported by the Master's Thesis Grant of the Zeno Karl Schindler Foundation for his work at the Autonomous Systems Lab, Stanford University. This letter was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor, and Reviewers' comments.

Jan Schilliger, Severin Hänggi, and Christopher Onder are with the Institute for Dynamic Systems and Control (IDSC), ETH Zürich, Zürich 8092, Switzerland (e-mail: janschi@student.ethz.ch; shaenggi@idsc.mavt.ethz.ch; onder@idsc.mavt.ethz.ch).

Thomas Lew, Spencer M. Richards, and Marco Pavone are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: thomas.lew@stanford.edu; spenrich@stanford.edu; pavone@stanford.edu).

Digital Object Identifier 10.1109/LRA.2021.3111010

presented in [18], where the point-wise CBF constraints must be robustly satisfied over a set. This leads to robust optimization problems, which may be difficult to solve. For purely discrete-time systems there is a distinct CBF formulation [19]. However, said formulation also leads to non-convex optimization problems and only enforces constraints at discrete-time steps.

Contributions. We propose a new tractable approach to safely control cyber-physical systems with *continuous-time* constraint satisfaction using *discrete-time* controllers. Specifically

- We extend CBFs to account for discretization error to provide a sufficient affine condition to guarantee safety for nonlinear control-affine systems also with *discrete-time* nominal controllers.
- We combine CBFs and approximate NMPC into an efficient algorithm with continuous-time safety guarantees and recursive feasibility despite the approximate nature.
- We validate our proposed approach in hardware experiments on a mini-Segway, and demonstrate the need to account for discretization errors to guarantee safety at all times given limited computational resources.

Notation: For a vector $v \in \mathbb{R}^n$ or vector-valued function $v : \mathbb{R}^n \to \mathbb{R}^m$, we use v_i to denote its i-th component. We denote the Euclidean norm as $\|v\| = \sqrt{v^\mathsf{T} v}$. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous on a set $\mathcal{X} \subseteq \mathbb{R}^n$ if there exists $L_f \in \mathbb{R}$ such that $\|f(x) - f(y)\| \leq L_f \|x - y\|$ for all $x, y \in \mathcal{X}$. If f is also differentiable, then $\|\nabla f(x)\| \leq L_f$ for all $x \in \mathcal{X}$, so we use $L_f := \max_{x \in \mathcal{X}} \|\nabla f(x)\|$ when $\mathcal{X}$ is compact. We denote the Minkowski sum of two sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^n$ as $\mathcal{X} \oplus \mathcal{Y}$, and similarly the Pontryagin difference as $\mathcal{X} \ominus \mathcal{Y}$. We denote the interior and boundary of a set $\mathcal{S}$ as $\mathrm{int}(\mathcal{S})$ and $\partial \mathcal{S}$, respectively.

II. Continuous Time Control Barrier Functions

Consider the nonlinear control-affine dynamical system

$$\dot{x}(t) = f(x(t)) + B(x(t))u(t), \tag{1}$$

with state $x(t) \in \mathbb{R}^n$ and control input $u(t) \in \mathbb{R}^m$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $B : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are Lipschitz continuous. In this section, we largely follow prior work [15], [20], [21] to formalize "safety" as controlling the state of (1) to remain within a designated safe set $\mathcal{X} \subset \mathbb{R}^n$ using only control inputs from an admissible set $\mathcal{U} \subset \mathbb{R}^m$. To this end, we search for a subset $\mathcal{C} \subseteq \mathcal{X}$ which is *controlled invariant*, i.e., such that for each $x(0) \in \mathcal{C}$ there exists an admissible input trajectory $u(t) \in \mathcal{U}$ such that $x(t) \in \mathcal{C}$ for all $t \geq 0$. Our objective is to render this $\mathcal{C}$ *forward invariant*, i.e. design a controller, such that $x(t) \in \mathcal{C}$ for all $t \geq 0$. Specifically, we restrict ourselves to the case where, given a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ such that $\nabla h(x) \neq 0$ whenever $h(x) = 0$, we define $\mathcal{C}$ as the super-level set

$$\mathcal{C} = \{x \in \mathcal{X} \mid h(x) \geq 0\}, \tag{2}$$

and assume $0 \in \mathcal{C}$ without loss of generality. The description in (2) conveniently ensures that $\mathcal{C}$ is controlled invariant if and only if for each $x \in \partial C$, there exists an input $u \in \mathcal{U}$ such that

$$\dot{h}(x, u) = \nabla h(x)^\mathsf{T}(f(x) + B(x)u) \geq 0. \tag{3}$$

We use property (3) to define CBFs.

Definition 1 (Control Barrier Function): Let $h : \mathcal{X} \to \mathbb{R}$ be continuously differentiable and satisfy $\nabla h(x) \neq 0$ whenever $h(x) = 0$. Then h is a Control Barrier Function (CBF) for the dynamical system (1) if there exists an extended class-$\mathcal{K}$ function[1] $\alpha : \mathbb{R} \to \mathbb{R}$ such that

$$\sup_{u \in \mathcal{U}} \nabla h(x)^\mathsf{T}(f(x) + B(x)u) \geq -\alpha(h(x)), \tag{4}$$

for all x satisfying $h(x) \geq 0$.

Finding a CBF for the system (1) is sufficient to guarantee that (3) holds and hence that the system is safe. We slightly rearrange (4) into

$$\mathrm{CBF}(x, u) := \nabla h(x)^\mathsf{T}(f(x) + B(x)u) + \alpha(h(x)) \geq 0, \tag{5}$$

which we term the *affine CBF condition* to highlight that it is indeed affine in the control input u. Thus, (5) can be embedded as a simple affine constraint to design safe optimization-based controllers [15] and safety filters [11].

However, the affine CBF condition (5) assumes the control signal $u(t)$ is applied in continuous-time, while in practice, controllers operate at discrete time instants. Thus, we generally lose the safety guarantees provided by CBFs. Moreover, prior work on CBFs implicitly assumes the control input u chosen to satisfy the affine CBF condition (3) lies in the admissible set $\mathcal{U}$, while in general this may not hold.

III. Problem Definition

The objective of this work is to design a control law using only admissible inputs from $\mathcal{U}$ which steer the system (1) to a desired state $x_d \in \mathcal{X}$ *safely*, i.e., we require $x(t) \in \mathcal{X}$ for all $t \geq 0$. Specifically, we assume $\mathcal{X} \subset \mathbb{R}^n$ is a compact set encoding any safety constraints, while $\mathcal{U} \subset \mathbb{R}^m$ is a convex polytopic set encoding control input constraints. Furthermore, we consider discrete-time controllers of the form

$$u(\tau) = u_k, \ \forall \tau \in [t_k, t_{k+1}). \tag{6}$$

The input $u_k \in \mathcal{U}$ is computed at the time instant $t_k \geq 0$ and applied over the interval $[t_k, t_{k+1})$, where $t_{k+1} - t_k = T$ for all $k \in \mathbb{N}_0$ and some sampling time $T > 0$, which corresponds to a zero-order hold.

As stated in Section II, controllers derived using CBFs rely on the continuous enforcement of the affine CBF condition (5), while in practice this condition can only be enforced at each sample time t_k. This discretization contradicts prior continuous-time analyses, thus safety constraints may not hold at all times $\tau \in [t_k, t_{k+1})$. Alternatively, NMPC controllers only consider constraints enforcement at a finite number of discretization nodes $\{t_k\}_{k=0}^N$. Although both approaches may lead to controllers satisfying constraints at all times given *fast enough* update rates, computational limitations motivate a finer analysis to explicitly account for this type of error and guarantee constraint satisfaction at all times.

[1] A continuous function $\beta : (-b, a) \to (-\infty, \infty)$ for some $a, b > 0$ is said to belong to extended class-$\mathcal{K}$ if β is strictly increasing and $\beta(0) = 0$.

IV. Discrete-Time Control Barrier Functions With Continuous Constraint Satisfaction

In this section, we derive two controllers of the form (6) which guarantee constraint satisfaction at all times. To this end, we first characterize the maximum variation of the CBF condition (5) over a time interval, and use it to derive a discrete-time CBF condition which results in a general and intuitive controller which explicitly accounts for the discretization error. For the second controller we propose, we exploit additional problem structure in the form of an auxiliary controller. We then characterize the difference between an ideal but intractable continuous-time controller which leverages the affine CBF condition (5), and a tractable discrete-time approximation which follows (6). With this analysis, we provide a tube-based CBF controller with continuous-time safety guarantees.

A. Discrete Barrier Condition

Consider the affine CBF condition (5) for some safe set $\mathcal{C}$ and a corresponding CBF h. For a discrete-time controller (6), we want to bound the maximum change in the CBF condition over some time to construct a guarantee that holds over an entire time interval. To this end, consider some differentiable, Lipschitz continuous function $\phi : \mathbb{R}^n \to \mathbb{R}^d$. To bound the change in ϕ along the trajectory $(x(t), u(t))$, we seek a constant $\widetilde{L}_\phi$ such that

$$\|\phi(x(t+T)) - \phi(x(t))\| \leq \widetilde{L}_\phi T. \tag{7}$$

By the chain rule, $\dot{\phi}(t) = \nabla\phi(x(t))^\mathsf{T}(f(x(t)) + B(x(t))u(t))$, and thus we can use

$$\widetilde{L}_\phi := L_\phi \max_{x\in\mathcal{X}, u\in\mathcal{U}} [-1]. \tag{8}$$

This augments the Lipschitz constant L_ϕ for ϕ into a Lipschitz-like constant $\widetilde{L}_\phi$ with respect to time along *any* state-input trajectory of the dynamical system. Then, over the time interval $\tau \in [t_k, t_k + T)$, we can bound the evolution of ϕ along $(x(t), u(t))$ according to

$$\phi(x(\tau)) \in \phi(x(t_k)) \oplus \mathcal{W}_\phi,\ \forall \tau \in [t_k, t_k + T), \tag{9}$$

where we define the truncation error hypercube

$$\mathcal{W}_\phi := \{w \in \mathbb{R}^d \mid \|w\|_\infty \leq \widetilde{L}_\phi T\}. \tag{10}$$

The affine CBF condition (5) depends on f, B, h, and ∇h, so we can treat

$$\mathcal{W} := \mathcal{W}_f \times \mathcal{W}_B \times \mathcal{W}_h \times \mathcal{W}_{\nabla h} \tag{11}$$

as a set of possible disturbances $w = (w_f, w_B, w_h, w_{\nabla h})$ which should be accounted for to guarantee safety at all times $\tau \in [t_k, t_{k+1})$. From this observation, we introduce the *Discrete-time Barrier Condition (DBC)*

$$\begin{aligned}&\mathrm{DBC}(x, u, w)\\ &:= (\nabla h(x) + w_{\nabla h})^\mathsf{T}[-1]\\ &\quad + \alpha(h(x) + w_h).\end{aligned} \tag{12}$$

At time t_k, this discrete barrier condition captures all possible system evolutions, thus guaranteeing that the affine CBF condition (5) holds for an entire time interval $\tau \in [t_k, t_{k+1})$. Further, if a discrete time controller u_k is synthesized such that it enforces this condition at all discrete times t_k, then the system is safe for all times under this controller. These two statements are formalized in the following theorem.

Theorem 1 (Controlled Invariance Using a DBC): Consider the dynamical system in (1) and a safe set $\mathcal{C}$ with a corresponding CBF h. Assume that the control law $u(\tau) = u_k \in \mathcal{U}$, $\tau \in [t_k, t_{k+1})$ satisfies

$$\mathrm{DBC}(x(t_k), U.K.w) \geq 0, \tag{13}$$

for all $w \in \mathcal{W}$, where $\mathcal{W}$ is defined as in (11). Then $\mathcal{C}$ is forward invariant for (1) at all times $\tau \in [t_k, t_{k+1})$. Furthermore, if $x(0) \in \mathcal{C}$ and the discrete-time controller with $u_k \in \mathcal{U}$ satisfies (13) at every time step t_k for $k \in \mathbb{N}_0$, then the system is safe for all $t \geq 0$.

Proof: Denote $x_t := x(t)$ for conciseness. We start with the proof of the first claim, and show that (13) holding at time t_k implies that (5) holds for all times $\tau \in [t_k, t_{k+1})$. According to (9), let $w \in \mathcal{W}$ be such that $f(x_\tau) = f(x_{t_k}) + w_f$, $B(x_\tau) = B(x_{t_k}) + w_B, h(x_\tau) = h(x_{t_k}) + w_h$, and $\nabla h(x_\tau) = \nabla h(x_{t_k}) + w_{\nabla h}$. Substituting these into the affine CBF condition (5) along with the given discrete-time control input yields

$$\begin{aligned}&\mathrm{CBF}(x_\tau, u_k)\\ &= \nabla h(x_\tau)^\mathsf{T}(f(x_\tau) + B(x_\tau)u_k) + \alpha(h(x_\tau))\\ &= (\nabla h(x_{t_k}) + w_{\nabla h})^\mathsf{T}(f(x_{t_k}) + w_f + (B(x_{t_k}) + w_B)u_k)\\ &\quad + \alpha(h(x_{t_k}) + w_h)\\ &= \mathrm{DBC}(x_{t_k}, u_k, w)\end{aligned}$$

If $\mathrm{DBC}(x_{t_k}, u_k, w) \geq 0$ for all $w \in \mathcal{W}$, then this holds for the particular w above, so (13) ensures $\mathrm{CBF}(x_\tau, u_k) \geq 0$ for all $\tau \in [t_k, t_{k+1})$. This, and the fact that h is a CBF, guarantee $x(\tau) \in \mathcal{C}$ for all $\tau \in [t_k, t_{k+1})$ as long as $x(t_k) \in \mathcal{C}$.

The second statement follows; by assumption, $x(t_k) \in \mathcal{C}$ and there exists a $u_k \in \mathcal{U}$ such that (13) holds. By induction with the previous result, $\mathcal{C}$ is forward invariant under this discrete-time controller. ■

Remark 1: We can reduce conservatism of our derivations in two ways. First, we can define the Lipschitz constant component-wise as $\|f(x) - f(x_0)\| \leq \sum_{i=1}^n L_{f,i}|x_i - x_{0,i}|$, which is equivalent to normalizing each dimension of the dynamics. Second, local upper bounds can reduce conservatism of the Lipschitz-like constants.

Since (13) is nonconvex, it can be challenging to enforce for general dynamical systems. Inspired by work on CBF-based robust controllers [11], we relax (13) to a set of affine conditions, which can then be used to construct safety filters and safe controllers. To this end, we write (13) as

$$\mathrm{DBC}(x, u, w) = -a(x, w)^\mathsf{T}u - b(x, w), \tag{14}$$

where

$$a(x,w) := -(B(x)+w_B)^\mathsf{T}(\nabla h(x)+w_{\nabla h})$$
$$b(x,w) := -(\nabla h(x)+w_{\nabla h})(f(x)+w_f)-\alpha(h(x)+w_h).$$

Then $\mathrm{DBC}(x,u,w)\geq 0$ for all $w\in\mathcal{W}$ if and only if

$$\max_{w\in\mathcal{W}}[-1]\leq 0. \tag{15}$$

For fixed u, the left-hand side of (15) is nonconvex in w. A sufficient relaxed condition for (15) to hold is

$$\max_{\tilde{a}_j,\tilde{b}}\ \left(\tilde{a}^\mathsf{T}u+\tilde{b}\right)\leq 0,$$
$$\text{s.t. } \tilde{a}_j\in\mathcal{A}^j_\mathcal{W}=\{a_j(x,w)|w\in\mathcal{W}\},\ \tilde{b}\in\{b(x,w)|w\in\mathcal{W}\}, \tag{16}$$

where we omit the dependency on x for conciseness, and $j=1,\ldots,m$. Note that $\mathcal{A}^j_\mathcal{W}$ are generally non-convex sets. Compared to (15), this last condition is conservative, since a different disturbance w can be chosen for each j-th dimension of a. As $\mathcal{W}$ and a can be nonconvex, (16) is nonconvex. This robust constraint can be relaxed as a set of affine constraints.

However, they are compact and thus can be outerbounded conservatively as polytopic sets. In this work for tractability, we consider conservative hyperrectangular sets of the form

$$\mathcal{D}^j_\mathcal{W}=\{\tilde{a}_j\mid \tilde{a}_j\leq\bar{a}_j,\ -\tilde{a}_j\leq-\underline{a}_j\},\quad j=1,\ldots,m,$$
$$\mathcal{B}_\mathcal{W}=\{\tilde{b}\mid \tilde{b}\leq\bar{b},\ -\tilde{b}\leq-\underline{b}\},$$

where $\underline{a}_j=\min_{\tilde{a}_j\in\mathcal{A}^j_\mathcal{W}}\tilde{a}_j$, and $\bar{a}_j=\max_{\tilde{a}_j\in\mathcal{A}^j_\mathcal{W}}\tilde{a}_j$. With these rectangular sets, we can rewrite the condition in (16) as affine constraints of the form $D[\tilde{a},\tilde{b}]^\mathsf{T}\leq d(x)$, where

$$D^\mathsf{T}=\begin{bmatrix}1&-1&\ldots&0&0\\0&0&\ldots&0&0\\0&0&\ldots&1&-1\end{bmatrix},$$
$$d(x)^\mathsf{T}=\begin{bmatrix}\bar{a}_1&-\underline{a}_1&\ldots&\bar{a}_m&-\underline{a}_m&\bar{b}&-\underline{b}\end{bmatrix}. \tag{17}$$

Then, a sufficient condition for (16) is given as

$$\max_{\tilde{a},\tilde{b}}\ \left(\tilde{a}^\mathsf{T}u+\tilde{b}\right)\leq 0,\quad \text{s.t. } D[\tilde{a},\tilde{b}]^\mathsf{T}\leq d(x). \tag{18}$$

(18) is a linear program in $\tilde{a}$ and $\tilde{b}$ (guaranteed feasible by a proper choice of $\mathcal{C}$). Denoting $\tilde{u}\triangleq[u,1]^\mathsf{T}\in\mathbb{R}^{m+1}$ for the concatenation of u with the scalar 1, we can express its dual as

$$\max_{\tilde{\lambda}}\ \left(d(x)^\mathsf{T}\tilde{\lambda}\right)\leq 0,\quad \text{s.t. } D^\mathsf{T}\tilde{\lambda}=\tilde{u},\quad \tilde{\lambda}\geq 0, \tag{19}$$

where $\tilde{\lambda}\in\mathbb{R}^{2(m+1)}$ are the dual variables. Since (18) is convex, its dual problem (19) is equivalent by strong duality. Any feasible solution of this problem will guarantee safety. These solutions must necessarily satisfy the following set of affine conditions

$$d(x)^\mathsf{T}\tilde{\lambda}\leq 0,\quad D^\mathsf{T}\tilde{\lambda}=\tilde{u},\quad \tilde{\lambda}\geq 0. \tag{20}$$

Using these affine constraints, we propose an optimization-based controller which guarantees safety at all times. Specifically, given a nominal control input u_k, it consists of solving, at each time t_k, the following quadratic program (QP):

$$\min_{u,\tilde{\lambda}}\ \|u-u_k\|^2,\quad \text{s.t. } (20). \tag{21}$$

To summarize, the affine conditions (20) allow to continuously guarantee safety for our system (1), if satisfied at discrete times t_k only. The resulting formulation can be used as a safety filter or to synthesize controllers in combination with optimization-based control approaches, such as Control Lyapunov Functions-based controllers or MPC. Compared to the CBF, the DBC has a restricted feasible region due to the added conservatism, up to a potential total collapse. We discuss the advantages and limitations in Section VI.

B. Tube-Cbf

We present our second algorithm, Tube-CBF, which in contrast to the DBC, uses the unaltered affine CBF condition from section II.

First, consider a nominal case where we have an *ideal* continuous-time controller fulfilling the affine CBF condition at all times and denote the nominal trajectory as $\bar{x}(t)$. As we use a discrete-time controller (6), naturally, discretization leads to a difference between the resulting trajectory $x(t)$ and $\bar{x}(t)$. As the affine CBF condition only guarantees safety for the nominal trajectory $\bar{x}(t)$, the true trajectory $x(t)$ is potentially outside of the safe set and may be unsafe. Inspired by tube MPC, we propose the use of a discrete-time auxiliary controller κ to regulate the error $z(t)=x(t)-\bar{x}(t)$ to zero and ideally bound it in an invariant tube. Knowledge of such an auxiliary controller and a corresponding invariant tube together with the affine CBF condition, allows us to design a discrete-time controller (6) with continuous-time safety guarantees. To this end, we first define a robust invariant set.

Definition 2 (Robust Invariant Set with Discrete Feedback): A set $\Omega\subset\mathcal{C}\subset\mathbb{R}^n$ is a robust control invariant set for the error $x-\bar{x}$ if there exists an auxiliary feedback control law $\kappa:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^m$ of the form (6), such that $\kappa(x,\bar{x})\in\mathcal{U}$ for all $x,\bar{x}\in\mathcal{C}$, and under this controller, if $x(0)-\bar{x}(0)\in\Omega$, then $x(t)-\bar{x}(t)\in\Omega$, for all $t\geq 0$.

Note that finding Ω and κ is in general not straightforward. A rather coarse over-approximation of Ω with a corresponding κ can be found in [22]. This leads to the following assumption, which is strong but common in tube NMPC literature [22]:

Assumption 1: Suppose we have a control law κ of the form (6), and a set Ω, such that Ω is robust control invariant, i.e., (κ,Ω) satisfy definition 2.

We use the auxiliary controller to compensate discretization errors in addition to the nominal controller. In the presence of control input constraints, we need to account for the contribution to the control input from the auxiliary controller. We capture this in the following set:

$$G:=\mathrm{Poly}\left(\{\kappa(x,\bar{x})\mid\bar{x}\in\mathcal{C}\ominus\Omega,\ \text{and } x-\bar{x}\in\Omega\}\right)\subset\mathbb{R}^m,$$

where $\mathrm{Poly}()$ denotes the smallest polytopic hull. To account for the error $z\in\Omega$, we define a reduced compact safe set $\mathcal{C}'$ as summarized in the following definition:

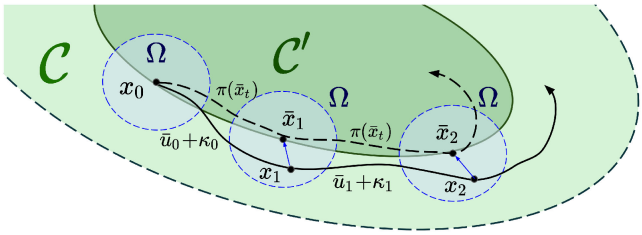

Fig. 2. The Tube-CBF algorithm consists of applying the affine CBF condition to a reduced set $\mathcal{C}'$. Then, the discretization error arising between the trajectories under an ideal continuous time controller (denoted as $\bar{x}$), and a discrete counterpart (denoted as x) is kept in the invariant tube Ω with an auxiliary controller κ.

Definition 3 (Reduced Safe Set): A reduced safe set $\mathcal{C}'$ to a set $\mathcal{C}$ is a compact set such that $\mathcal{C}' \subseteq \mathcal{C} \ominus \Omega \subset \mathbb{R}^n$, where $\mathcal{C}'$ is characterized by a CBF h under $\mathcal{U}' := \mathcal{U} \ominus G$.

Assumption 2: We have access to a reduced safe set $\mathcal{C}'$ according to definition 3.

Building on assumptions 1 and 2, we propose the Tube-CBF algorithm.

Tube-CBF algorithm:

1) At time t_0, set $\bar{x}(t_0) = x(t_0) \in \mathcal{C}'$.
2) At time t_k, find a nominal input $\bar{u}(t_k) \in \mathcal{U}'$, such that the affine CBF condition for $\mathcal{C}'$ holds at $\bar{x}(t_k)$.
3) Apply the control input $u(t_k) = \bar{u}(t_k) + \kappa(x(t_k), \bar{x}(t_k))$ to the system during the time interval $\tau \in [t_k, t_{k+1})$.
4) At time t_{k+1}, measure the state $x(t_{k+1})$ and find a state $\bar{x}(t_{k+1}) \in \mathcal{C}'$ such that $x(t_{k+1}) \in \bar{x}(t_{k+1}) \oplus \Omega$ and go to step 2.

Fig. 2 depicts multiple time steps of the algorithm. Furthermore, we formalize the safety guarantees in the following theorem.

Theorem 2 (Controlled Invariance Using Tube-CBF): Consider the dynamical system in (1) and a safe set $\mathcal{C}$. Suppose assumptions 1 and 2 hold. Then, $\mathcal{C}$ is forward invariant under the Tube-CBF algorithm for (1) under (6) at all times $t \geq 0$ if $x(0) \in \mathcal{C}'$.

Proof: We proceed by induction and consider first the induction step. At time t_k, we assume we have $\bar{x}(t_k) \in \mathcal{C}'$ and $x(t_k) \in \bar{x}(t_k) \oplus \Omega$. Then, according to Step 2 of the Tube-CBF algorithm, we compute an input $\bar{u}(t_k)$ that satisfies the CBF condition for $\mathcal{C}'$ at $\bar{x}(t_k)$. A feasible control input exists by the definition of $\mathcal{C}'$. Next we apply the control input $u(t_k) = \bar{u}(t_k) + \kappa(x(t_k), \bar{x}(t_k))$ to the system over a time interval $t \in [t_k, t_{k+1})$ according to Step 3. By definition of (κ, Ω) and $\mathcal{C}'$, $u(t_k) \in \mathcal{U}$. Further, by definition of (κ, Ω), this guarantees that $x(\tau) - \bar{x}(\tau) \in \Omega$ for $\tau \in [t_k, t_{k+1})$. Also, $\bar{x}$ follows an ideal controller, such that $\bar{x}(\tau) \in \mathcal{C}'$ for $\tau \in [t_k, t_{k+1})$. Therefore, at Step 4, there exists a $\bar{x}(t_{k+1})$ such that $x(t_{k+1}) \in \bar{x}(t_{k+1}) \oplus \Omega$. We continue with $\bar{x}(t_{k+1})$ at step 2.

The initial condition $x(t_0) = \bar{x}(t_0) \in \mathcal{C}'$ completes the proof. ■

V. Continuous Safety Guarantees With NMPC

In this section, we design an optimal controller of the form (6). We consider a direct NMPC approach to reduce the optimal control problem to a nonlinear program (NLP). We discretize the system dynamics (1)

$$\begin{aligned} x(t_{k+1}) &= f_k^d(x_k, u_k) \\ &= x(t_k) + \int_{t_k}^{t_{k+1}} [f(x(\tau)) + B(x(\tau))u_k] \mathrm{d}\tau, \end{aligned} \tag{22}$$

where the integral in (22) can be approximated using a numerical integration scheme of choice. In NMPC, we solve the NLP at each time step $t_k \geq 0$ to obtain the controller (6). The NLP formulation we consider is

$$\begin{aligned} \min_{X,U,S} \; & l_N(x_N) + \zeta_N(s_{2N-1}) + \sum_{i=0}^{N-1} l_k(x_i, u_i) + \zeta_i(s_{2i}, s_{2i+1}) \\ \text{s.t.} \; & x_{i+1} - f_i^d(x_i, u_i) = s_{2i}, \quad i = 0, \dots, N-1 \\ & g_i(x_i, u_i) \leq s_{2i+1}, \; i = 0, \dots, N-1 \\ & x_0 - \hat{x} = 0, \quad g_N(x_N) \leq s_{2N-1}, \end{aligned} \tag{23}$$

with $X = [x_0, \dots, x_N]^\mathsf{T}$, $U = [u_0, \dots, u_{N-1}]^\mathsf{T}$, and $S = [s_0, \dots, s_{2N-1}]^\mathsf{T}$. At time t_k, the NLP is initialized with $\hat{x}$. Along the prediction horizon $i = [0, \dots, N]$, the functions l_k and l_N denote the stage and terminal cost, while the functions g_k and g_N denote (in)-equality constraints. To guarantee feasibility of the NLP, we soften the constraints and introduce the slack variables S, which we penalize with a function of the form $\zeta_i(s_i) = \|s_i\| + s_i^2$. To facilitate reading, we collect all decision variables in $v = [X^\mathsf{T}, U^\mathsf{T}, S^\mathsf{T}]^\mathsf{T}$ and split the constraints into equality constraints I and inequality constraints G to arrive at a general NLP formulation with the cost function J,

$$\min_v J(v) \quad \text{s.t. } I(v) = 0, \; G(v) \leq 0. \tag{24}$$

We tackle (24) using Sequential Quadratic Programming (SQP) [23]. In SQP, we solve a series of approximated subproblems to iteratively arrive at a solution to (24). Specifically, SQP methods minimize a second-order Taylor expansion of the Lagrangian of (24),

$$\mathcal{L}(v, \lambda, \mu) = J(v) + \lambda^\mathsf{T} I(v) + \mu^\mathsf{T} G(v), \tag{25}$$

where λ and $\mu \geq 0$ are the Lagrange multipliers. We start out with an initial guess v^0 and incrementally update it with computed update steps δw^i according to $v^{i+1} = v^i + \delta v^i$. Minimizing the second-order Taylor expansion $\mathcal{L}_{SQP}(\delta v^i, \delta\lambda^i, \delta\mu^i) = T_{\mathcal{L}(v^i, \lambda^i, \mu^i)}(\delta v^i, \delta\lambda^i, \delta\mu^i)$ is equivalent to solving the following (potentially nonconvex) QP

$$\begin{aligned} \min_{\delta v^i} \quad & \nabla_v J(v^i)^\mathsf{T} \delta v^i + \frac{1}{2} \delta v^{i\mathsf{T}} H(v^i, \lambda^i, \mu^i) \delta v^i \\ \text{s.t.} \quad & I(v^i) + \nabla_v I(v^i)^\mathsf{T} \delta v^i = 0, \\ & G(v^i) + \nabla_v G(v^i)^\mathsf{T} \delta v^i \leq 0, \end{aligned} \tag{26}$$

where $H = \nabla_v^2 \mathcal{L}(v^i, \lambda^i, \mu^i)$ denotes the Hessian. The optimization variables are then updated according to

$$w^{i+1} = w^i + \delta w^i, \quad \lambda^{i+1} = \lambda_{QP}^i, \quad \mu^{i+1} = \mu_{QP}^i.$$

Algorithm 1: RTI with Tube-CBF.

initialize: $\bar{x}(t_0) = x(t_0)$ with $x(t_0) \in \mathcal{C}'$
while IsRunning() **do**
  $\bar{u}_k \leftarrow$ obtain by solving (27) at time t_k
  $u(t_k) \leftarrow \bar{u}(t_k) + \kappa(x(t_k), \bar{x}(t_k))$
  apply $u(t_k)$ to plant
  wait()
  measure $x(t_{k+1})$
  find $\bar{x}(t_{k+1})$ s.t. $x(t_{k+1}) \in \bar{x}(t_{k+1}) \oplus \Omega$
end while

Solving a sequence of QPs until convergence of the underlying NLP is still computationally intense. Thus, we opt for an approximate NMPC scheme, also known as the real-time iteration scheme (RTI). The RTI is designed for NMPC problems with a quadratic cost function and uses the Gauss-Newton approximation for the Hessian. Furthermore, in RTI, we only solve a single QP at each time step t_k, which leads to a non-converged and generally suboptimal solution. This non-converged solution poses a few challenges. First, the linear approximation of the constraints in (26) no longer guarantees strict constraint satisfaction of the original problem (24). Second, recursive feasibility is lost and (24) might become infeasible at some point, which must be avoided at all cost in safety critical systems.

Now, we use Tube-CBF introduced in IV to combine the enhanced performance of RTI with safety and recursive feasibility. To this end, we return again to the RTI formulation in (26) and complement it with two hard constraints. First, we add the constraint $\bar{u}_0 \in \tilde{\mathcal{U}}$. To facilitate notation, we use an affine operator Ξ_{u_0} with $u_0 \equiv \Xi_{u_0} v$ to extract u_0 from our optimization variables v. Second, we add the affine CBF condition to our CBF for $\mathcal{C}'$, where we use $x_0 \equiv \Xi_{x_0} v$. This leads to

$$\begin{aligned} \min_{\delta v^i} \quad & \nabla_v J(v^i)^\mathsf{T} \delta v^i + \frac{1}{2} \delta {v^i}^\mathsf{T} H(v^i, \lambda^i, \mu^i) \delta v^i \\ \text{s.t.} \quad & I(v^i) + \nabla_v I(v^i)^\mathsf{T} \delta v^i = 0, \\ & \tilde{G}(v^i) + \nabla_v \tilde{G}(v^i)^\mathsf{T} \delta v^i \leq 0, \\ & - \text{CBF}\left(\Xi_{x_0} v^i, \Xi_{u_0}(v^i + \delta v^i)\right) \leq 0, \\ & \Xi_{u_0}(v^i + \delta v^i) \in \mathcal{U}'. \end{aligned} \tag{27}$$

The formulation (27) provides point-wise constraint satisfaction, but does not yet guarantee recursive feasibility. To obtain recursive feasibility, we additionally perform the Tube-CBF algorithm from section IV-B. The combination of RTI and Tube-CBF is summarized in algorithm 1 and the following theorem.

Theorem 3 (Safe RTI Using Tube-CBF): Consider the dynamical system in (1) under the RTI controller (27) and a reduced safe set $\mathcal{C}'$ with a corresponding CBF h according to definition 3. We assume that $x(t_0) \in \mathcal{C}'$. Then $\mathcal{C}$ is forward invariant for (1) under (27) at all times $t \geq 0$ and hence the constraints are strictly satisfied under the Tube-CBF algorithm. Furthermore, since the constraints are satisfied at all times, the RTI is also recursively feasible.

Proof: We leverage theorem 2 for the proof. Theorem 2 states that $\mathcal{C}$ is forward invariant for all times $t \geq 0$ if $x(0) \in \mathcal{C}'$ and there is a discrete-time controller that provides a nominal input $\bar{u}(t_k) \in \mathcal{U}'$ such that $\text{CBF}(\bar{x}(t_k), \bar{u}(t_k)) \geq 0$ for all $k \geq 0$.

First, note that since $\mathcal{U}$ and G are convex polytopes, $\mathcal{U}'$ is a convex polytope. Thus, $u \in \mathcal{U}' \Leftrightarrow Au \leq b$ for some A and b, i.e., we can encode $\bar{u} \in \mathcal{U}'$ as a set of affine conditions. From the definition of CBFs we have that $\mathcal{U}' \cap \{u \mid \text{CBF}(\bar{x}, u) \geq 0\} \neq \varnothing$ for all $\bar{x} \in \mathcal{C}'$. Thus, there exists a $\bar{u}(t_k)$ such that $A\bar{u}(t_k) \leq b$ and $\text{CBF}(\bar{x}, \bar{u}) \geq 0$ and hence (27) is feasible for all $\bar{x} \in \mathcal{C}'$, as all other constraints are soft constraints. By assumption $x(0) \in \mathcal{C}'$ and thus we can use theorem 2 to guarantee forward invariance of $\mathcal{C}$ and correspondingly safety for (1). And since the constraints are affine and thus convex, the feasible $\bar{u}(t_k)$ can be found with the QP.

Recursive feasibility can be seen directly from step 4 in the Tube-CBF algorithm. Since the nominal state $\bar{x}(t_k) \in \mathcal{C}'$ for all $k \geq 0$ and since (27) is feasible for all $\bar{x} \in \mathcal{C}'$, we see that (27) is recursively feasible for all time steps t_k with $k \geq 0$. ■

VI. Results

To demonstrate the performance of the proposed algorithms DBC and Tube-CBF, we performed simulations and hardware experiments. We simulated the DBC as a safety filter in conjunction with a nominal LQR controller, and we implemented the Tube-CBF algorithm with a nominal RTI on a mini-Segway and compared it to several baseline controllers. Fig. 1 shows the mini-Segway, which is equipped with an Arduino capable ATmega32U4 MCU, wheel encoders, low-level motor controllers, and an LSM6DS33 IMU. The mini-Segway connects to a Raspberry Pi model 3B+ through I2C which runs Ubuntu 18.04 and performs the controller computations. We only run the system in a planar mode, i.e., apply the same input to both wheels. The dimensions of the segway are $118 \times 112 \times 80\,\text{mm}$ with a total weight of $450\,\text{g}$. The paper [24] presents the dynamics of the mini-Segway.

A. Discrete Barrier Condition

Although the DBC provides a very general formulation for safety guarantees, the approach is too conservative for the particularly fast dynamics of our hardware. Specifically, the resulting Lipschitz constants are large and consequently the feasible region has collapsed.

However, the DBC is applicable for dynamics with smaller Lipschitz constants. To show the feasibility of the DBC we performed simulations for two different mini-Segways. One has the same dynamics as our hardware, whereas the other has slower dynamics and consequently smaller Lipschitz constants. We use an LQR controller in conjunction with the DBC safety filter (21) to ensure safety. We simulate a regulation task on the position and apply the safety filter at various rates. Without the filter, the system is unsafe for both the fast and slow dynamics. For sufficiently large filter rates and slow dynamics, the DBC is viable and successfully keeps the system safe. Fig. 3 shows the simulation results.

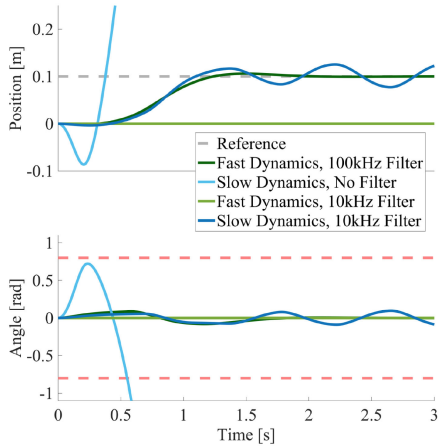

Fig. 3. The figure shows the simulation of two mini-Segways with different dynamics. We use an LQR controller in conjunction with a DBC safety filter (21) to keep the system safe. We simulate a regulation task with the safety filter applied at various rates. Without the filter, the systems are unsafe (only slow unsafe dynamics are shown). For sufficiently large filter rates and slow dynamics, the DBC is viable and successfully keeps the system safe.

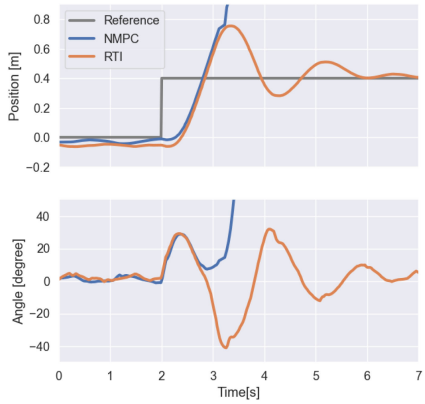

Fig. 4. The figure shows a full NMPC controller with $N = 15$ and a plain RTI controller with $N = 50$, both applied at 33 Hz. At $t = 2\,\mathrm{s}$ the reference position changes from $0\,\mathrm{m}$ to $0.4\,\mathrm{m}$. While the full NMPC controller violates the safe set, the RTI remains safe due to its longer prediction horizon.

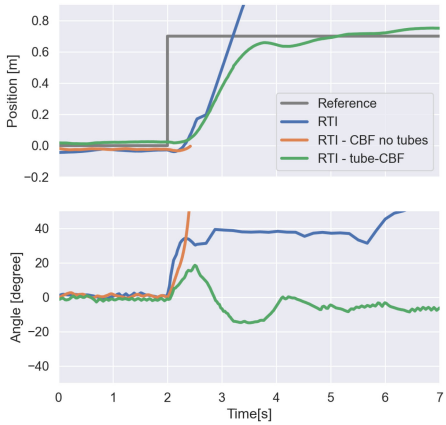

Fig. 5. The figure shows three RTI controllers, one with $N = 50$, applied at 33 Hz and two with $N = 15$ applied at 100 Hz, where one uses Tube-CBF and the other uses the affine CBF condition without the Tube-CBF algorithm. At $t = 2\,\mathrm{s}$ the reference position changes from $0\,\mathrm{m}$ to $0.7\,\mathrm{m}$. Only the RTI controller with Tube-CBF remains safe.

B. Tube-CBF Implementation on Hardware

We implement the proposed combination of RTI with Tube-CBF on the mini-Segway and compare its performance to three baseline controllers:

- Full NMPC: We tackle the NLP (23) with an interior-point solver
- Plain RTI: RTI without CBFs
- RTI with CBF: We solve the QP (27) with the affine CBF condition, but do not use the Tube-CBF algorithm

C. Derivation of Tube-CBF

The only hard constraint present in our system is the motor input voltage of $\pm 5.4\,\mathrm{V}$. First, we design an auxiliary controller. We linearize the mini-Segway's dynamics around the origin and compute a state feedback controller using LQR,

$$K_{aux} = \begin{bmatrix} -3.7304 & -3.4806 & -0.7343 \end{bmatrix}.$$

We need the invariant tube Ω for K_{aux} to perform the constraint tightening to arrive at a reduced safe set $\mathcal{C}'$ and the corresponding input constraints $\mathcal{U}'$. While [22] presents and approach to compute Ω using Lipschitz continuity, our system's closed-loop Lipschitz constants are large and the approach in [22] is too conservative. Thus, we opted for a *not exact* constraint tightening of 33 % or $\pm 1.8\,\mathrm{V}$, which we found to work well in practice.

Now that we have an auxiliary controller and the corresponding constraint tightening, we compute a CBF for $\mathcal{C}'$. We perform a numerical Hamilton-Jacobi reachability-analysis [25] on a $99 \times 99 \times 99$ sampling grid of our state space $x = [\dot{s}, \theta, \dot{\theta}]^\mathsf{T}$ using our tightened motor voltage constraints of $\pm 3.6\,\mathrm{V}$. From the sampled data we obtained an analytical representation of a CBF using polynomial regression.

D. Controller Design

All RTI and NMPC controllers are designed with the same dynamics and a sampling time of 70 ms using FORCES PRO [26], with a quadratic cost of the form

$$l(x, u) = \|\Delta u\|_{R_1} + \|u\|_{R_2} + \|x\|_Q.$$

When it comes to computational performance, there is a trade-off between prediction horizon length and update frequency. A longer horizon increases computational intensity, as does a higher frequency. We found the minimal frequency required for stability to be around 33 Hz. The following configurations exploit all of the available computational resources.

E. Control Task and Results

The experimental task is to track a reference position. The results of the experiments are captured in Figs. 4 and 5. Fig. 4 shows the performance of the full NMPC and the RTI. While the full NMPC cannot handle a $0.4\,\mathrm{m}$ step in the reference position,

the RTI controller remains safe due to its longer prediction horizon. However, increasing the step in the reference position from $0.4\,\mathrm{m}$ to $0.7\,\mathrm{m}$ renders also the RTI controller unsafe, as shown in fig. 5. Extending the RTI controller with our Tube-CBF algorithm successfully keeps the system safe. It's noteworthy that just extending the RTI formulation with the affine CBF condition but without applying the Tube-CBF algorithm, does not lead to safety. In fact, the affine CBF condition becomes infeasible shortly after the step in the reference position.

Thus, this shows that extending MPC with CBFs enhances safety. When using the affine CBF condition, compensating the discretization error is necessary, e.g., with Tube-CBF.

VII. Conclusion

In this work, we extend CBFs to account for discretization error and guarantee constraints satisfaction for nonlinear control-affine systems under discrete-time nominal controllers. We presented two algorithms that result in enforcing (a set of) affine conditions at a finite rate to guarantee safety. The DBC relies on Lipschitz continuity to capture the discretization error. The Tube-CBF algorithm relies on an auxiliary controller that spans an invariant tube to compensate for arising discretization errors. We combine Tube-CBF and approximate NMPC to obtain an efficient algorithm with continuous-time safety guarantees despite the approximate nature of the controller. We validate our proposed approach in hardware experiments on a mini-Segway, and demonstrate the need to account for discretization errors to guarantee safety at all times given limited computational resources.

Future work will focus on making auxiliary controllers and robust control invariant sets available for a larger number of systems. Furthermore, we will extend our formulation to include more real-world challenges, like external disturbances to provide more realistic implementations.

References

[1] M. Kögel and R. Findeisen, "Discrete-time robust model predictive control for continuous-time nonlinear systems," in *Proc. Amer. Control Conf.*, 2015, pp. 924–930.

[2] D. Liao-McPherson, M. Nicotra, and I. Kolmanovsky, "Time distributed sequential quadratic programming for model predictive control: Stability and robustness," 2019, *arXiv:1903.02605*.

[3] V. R. Desaraju and N. Michael, "Fast nonlinear model predictive control via partial enumeration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1243–1248.

[4] M. Neunert *et al.*, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1398–1404.

[5] A. Zanelli, R. Quirynen, and M. Diehl, "Efficient zero-order NMPC with feasibility and stability guarantees," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 2769–2775.

[6] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, 2005.

[7] A. Zanelli, R. Quirynen, and M. Diehl, "An efficient inexact nmpc scheme with stability and feasibility guarantees," *IFAC-PapersOnLine*, vol. 49, no. 18, 2016.

[8] S. Magdici and M. Althoff, "Adaptive cruise control with safety guarantees for autonomous vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5774–5781, 2017.

[9] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, 2021, Art. no. 109597.

[10] S. Bak, D. K. Chivukula, O. Adekunle, M. Sun, M. Caccamo, and L. Sha, "The system-level simplex architecture for improved real-time embedded system safety," in *Proc. 15th IEEE Real-Time Embedded Technol. Appl. Symp.*, 2009, pp. 99–107.

[11] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *Proc. 9th ACM/IEEE Int. Conf. Cyber- Phys. Syst.*, 2018, pp. 98–106.

[12] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Safe policy synthesis in multi-agent pomdps via discrete-time barrier functions," 2019, *arXiv:1903.07823*.

[13] R. Takano, H. Oyama, and M. Yamakita, "Application of robust control barrier function with stochastic disturbance model for discrete time systems," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 46–51, 2018.

[14] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3D dynamic walking on stepping stones with control barrier functions," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 827–834.

[15] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[16] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[17] T. Gurriet, P. Nilsson, A. Singletary, and A. D. Ames, "Realizable set invariance conditions for cyber-physical systems," in *Proc. Amer. Control Conf.*, 2019, pp. 3642–3649.

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

[19] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Robot.: Sci. Syst.*, 2017.

[20] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proc. Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[21] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," 2019, *arXiv:1903.11199*.

[22] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Syst. Control Lett.*, vol. 62, no. 2, pp. 194–200, 2013.

[23] J. Nocedal and S. J. Wright, "Sequential quadratic programming," *Numer. Optim.*, pp. 529–562, 2006.

[24] S. Kim and S. Kwon, "Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot," *Int. J. Control, Automat. Syst.*, vol. 13, no. 4, pp. 926–933, 2015.

[25] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[26] A. Domahidi and J. Jerez, "Forces professional," Embotech AG, 2014–2021 [Online]. Available: https://embotech.com/FORCES-Pro.