Episodic Learning for Safe Bipedal Locomotion with Control Barrier Functions and Projection-to-State Safety

Noel Csomay-Shanklin[*]   NOELCS@CALTECH.EDU
Ryan K. Cosner[*]   RKCOSNER@CALTECH.EDU
Min Dai[*]   MDAI@CALTECH.EDU
Andrew J. Taylor   AJTAYLOR@CALTECH.EDU
Aaron D. Ames   AMES@CALTECH.EDU

Abstract

This paper combines episodic learning and control barrier functions in the setting of bipedal locomotion. The safety guarantees that control barrier functions provide are only valid with perfect model knowledge; however, this assumption cannot be met on hardware platforms. To address this, we utilize the notion of projection-to-state safety paired with a machine learning framework in an attempt to learn the model uncertainty as it affects the barrier functions. The proposed approach is demonstrated both in simulation and on hardware for the AMBER-3M bipedal robot in the context of the stepping-stone problem, which requires precise foot placement while walking dynamically.

Keywords: bipedal locomotion, supervised learning, safety, control barrier functions

1. Introduction

Safety is of significant importance in many modern control applications, with complex systems demanding the rigorous encoding of safety properties in the controller design process. One particularly challenging example is the safe locomotion of bipedal robots in difficult terrain; a concrete example of this is the "stepping-stone problem" where the feet must be placed in precise locations. In practice, the models used to design safety-critical controllers are imperfect—especially in the context of highly dynamic systems like walking robots—with model uncertainty arising due to parametric error and unmodeled dynamics. This uncertainty can lead to unsafe or dangerous behavior, thus requiring that safety-critical control be understood in the presence of model uncertainty.

In this work we consider a machine learning based approach for attenuating the impact of model uncertainty on the safe behavior of a system. The application of learning to safety-critical control of systems with uncertain models has recently exhibited significant success (Taylor et al., 2019b; Cheng et al., 2019; Choi et al., 2020; Lee et al., 2020; Castañeda et al., 2020). We look to achieve safety defined in terms of *set invariance* (Blanchini, 1999), which is an area of active research at the intersection of data-driven methods and nonlinear control theory (Berkenkamp et al., 2016; Fisac et al., 2018; Wang et al., 2018; Fan et al., 2019; Dean et al., 2020).

We will leverage control barrier functions (CBFs) for synthesizing safety controllers (Ames et al., 2014a, 2017) that achieve set invariance. CBFs have become a popular method for ensuring

---

safety (Nguyen and Sreenath, 2016; Wang et al., 2018), but they require an accurate model of the system dynamics and model uncertainty can result in a loss of safety guarantees (Kolathaya and Ames, 2018). Efforts have been made to make the safety guarantees of CBFs robust to the effects of model uncertainty, but in doing so these methods can cause the system's behavior to be overly conservative (Gurriet et al., 2018; Xu et al., 2015; Taylor and Ames, 2020).

One challenge in incorporating learning methods with nonlinear control is the need for diverse data capturing system behavior, and in particular the input-to-state relationship. While in simulated environments it is possible to collect high coverage data sets that accurately determine how inputs affect the evolution of the system, collecting such data on real physical systems may be prohibitively costly or damaging to the system. The lack of this data can lead to challenges with under-determination in supervised learning problems that seek to preserve the underlying structure of dynamic systems (Taylor et al., 2019a,b). Often this leads to model fits with low training loss that result in poor control performance when integrated in closed-loop controllers.

One setting for which collecting diverse data is infeasible is bipedal locomotion, as consistent walking requires well defined input profiles to ensure stability (Grizzle et al., 2001). The stepping-stone task is a historical benchmark for evaluating the safety-critical control of biped platforms (Nguyen et al., 2016, 2018, 2020). As compared to the statically stable motion which has been successfully employed to accomplish this task on a variety of legged robotic platforms (Griffin et al., 2019; Fankhauser et al., 2018), the underactuated nature of the bipedal robot under consideration in this work requires dynamic gaits to successfully traverse a set of stepping-stones. During such dynamic motion, satisfaction of safety constraints is predicated on having accurate model information, a requirement that cannot be met on hardware platforms, thus suggesting the need for model uncertainty reduction via learning.

Our Contribution: The main contributions of this work are two-fold. First, we propose an episodic learning framework for iteratively reducing the impact of disturbances on the safety-critical behavior of a system. Compared to previous work, our method resolves challenges with input data diversity by recursively defining disturbance estimators and controllers that are agnostic to the underlying control-affine structure. We successfully deploy this method in the context of the classical bipedal robotic stepping-stone problem. We validate that this method is able to effectively reduce model uncertainty and improve the safe behavior of a complex system both in simulation and experiment. The second contribution is the first experimental demonstration of CBFs for safety-critical control on a bipedal robot. This challenge has been the focus of significant prior work and we demonstrate that it can be achieved through the introduction of learning.

2. Background

In this section we provide a review of CBFs and projection-to-state safety (PSSf), which will be utilized in Section 3 to formulate a learning approach for mitigating model uncertainty.

2.1. Safety via Control Barrier Functions

Consider the nonlinear control affine system given by:

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

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous on $\mathbb{R}^n$. Given a Lipschitz continuous state-feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$, the closed-loop

system dynamics are:

$$\dot{\mathbf{x}} = \mathbf{f}_{cl}(\mathbf{x}) \triangleq \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}). \tag{2}$$

The assumption of local Lipschitz continuity of $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{k}$ implies that $\mathbf{f}_{cl}$ is locally Lipschitz continuous. Thus, for any initial condition $\mathbf{x}_0 := \mathbf{x}(0) \in \mathbb{R}^n$ there exists a maximum time interval $I(\mathbf{x}_0) = [0, t_{max})$ such that $\mathbf{x}(t)$ is the unique solution to (2) on $I(\mathbf{x}_0)$ (Perko, 2013). In the case that $\mathbf{f}_{cl}$ is forward complete, $t_{max} = \infty$.

The notion of safety that we consider in this paper is formalized by specifying a *safe set* in the state space that the system must remain in to be considered safe. In particular, consider a set $\mathcal{C} \subset \mathbb{R}^n$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, yielding:

$$\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\}, \tag{3}$$

with $\partial \mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) = 0\}$ and $\text{Int}(\mathcal{C}) \triangleq \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) > 0\}$. We assume that $\mathcal{C}$ is nonempty and has no isolated points, that is, $\text{Int}(\mathcal{C}) \neq \emptyset$ and $\overline{\text{Int}(\mathcal{C})} = \mathcal{C}$. We refer to $\mathcal{C}$ as the *safe set*. This construction motivates the following definitions of forward invariance and safety:

Definition 1 (*Forward Invariance & Safety*) *A set $\mathcal{C} \subset \mathbb{R}^n$ is forward invariant if for every $\mathbf{x}_0 \in \mathcal{C}$, the solution $\mathbf{x}(t)$ to (2) satisfies $\mathbf{x}(t) \in \mathcal{C}$ for all $t \in I(\mathbf{x}_0)$. The system (2) is safe on the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant.*

Before defining CBFs, we denote a continuous function $\alpha : [0, a) \to \mathbb{R}_+$, with $a > 0$, as *class $\mathcal{K}$* ($\alpha \in \mathcal{K}$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a = \infty$ and $\lim_{r \to \infty} \alpha(r) = \infty$, then α is *class $\mathcal{K}_\infty$* ($\alpha \in \mathcal{K}_\infty$). A continuous function $\alpha : (-b, a) \to \mathbb{R}$, with $a, b > 0$, is said to belong to *extended class $\mathcal{K}$* ($\alpha \in \mathcal{K}_e$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a, b = \infty$, $\lim_{r \to \infty} \alpha(r) = \infty$, and $\lim_{r \to -\infty} \alpha(r) = -\infty$, then α is said to belong to *extended class $\mathcal{K}_\infty$* ($\alpha \in \mathcal{K}_{\infty,e}$).

Certifying the safety of the closed-loop system (2) with respect to a set $\mathcal{C}$ may be impossible if the controller $\mathbf{k}$ was not chosen to enforce the safety of $\mathcal{C}$. CBFs can serve as a synthesis tool for attaining the forward invariance, and thus the safety of a set:

Definition 2 (*Control Barrier Function (CBF)* (Ames et al., 2017)) *Let $\mathcal{C} \subset \mathbb{R}^n$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ with 0 a regular value. The function h is a control barrier function (CBF) for (1) on $\mathcal{C}$ if there exists $\alpha \in \mathcal{K}_{\infty,e}$ such that for all $\mathbf{x} \in \mathbb{R}^n$:*

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \dot{h}(\mathbf{x}, \mathbf{u}) \triangleq \nabla h(\mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) = L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \geq -\alpha(h(\mathbf{x})), \tag{4}$$

where $L_{\mathbf{f}}h(\mathbf{x})$ and $L_{\mathbf{g}}h(\mathbf{x})$ are Lie derivatives.

Given a CBF h for (1) and a corresponding $\alpha \in \mathcal{K}_{\infty,e}$, we define the point-wise set of all control values that satisfy (4):

$$K_{cbf}(\mathbf{x}) \triangleq \left\{ \mathbf{u} \in \mathbb{R}^m \mid \dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})) \right\}.$$

We have the following result relating controllers taking values in $K_{cbf}(\mathbf{x})$ to the safety of (1) on $\mathcal{C}$:

Theorem 1 (Ames et al. (2014a)) *Given a set $\mathcal{C} \subset \mathbb{R}^n$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, if h is a CBF for (1) on $\mathcal{C}$, then any Lipschitz continuous controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$, such that $\mathbf{k}(\mathbf{x}) \in K_{cbf}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, renders the system (1) safe with respect to the set $\mathcal{C}$.*

Given a nominal (but not necessarily safe) locally Lipschitz continuous controller $\mathbf{k}_d : \mathbb{R}^n \to \mathbb{R}^m$, a controller taking values in the set $K_{\text{cbf}}(\mathbf{x})$ is the safety-critical CBF-QP:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2} \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \qquad \text{(CBF-QP)}$$

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

2.2. Model Uncertainty and Projection-to-State Safety

In practice, the system dynamics (1) are not known during control design due to parametric error and unmodeled dynamics. Instead, a nominal model of the system is utilized:

$$\widehat{\dot{\mathbf{x}}} = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u}, \qquad (5)$$

where $\widehat{\mathbf{f}} : \mathbb{R}^n \to \mathbb{R}^n$ and $\widehat{\mathbf{g}} : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be Lipschitz continuous on $\mathbb{R}^n$. By adding and subtracting the right hand side of (5) to (1), the dynamics of the system are:

$$\dot{\mathbf{x}} = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u} + \overbrace{\underbrace{\mathbf{f}(\mathbf{x}) - \widehat{\mathbf{f}}(\mathbf{x})}_{\mathbf{b}(\mathbf{x})} + \underbrace{(\mathbf{g}(\mathbf{x}) - \widehat{\mathbf{g}}(\mathbf{x}))}_{\mathbf{A}(\mathbf{x})} \mathbf{u}}^{\mathbf{d}(\mathbf{x},\mathbf{u})}, \qquad (6)$$

where the unknown disturbance $\mathbf{d}(\mathbf{x}, \mathbf{u}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u}$ is assumed to be time invariant, but depends on the state and input to the system. If the function $h : \mathbb{R}^n \to \mathbb{R}$ is a CBF for the nominal model (5) on $\mathcal{C}$, the uncertainty in the dynamics directly manifests in the time derivative of h:

$$\dot{h}(\mathbf{x}, \mathbf{u}) = \underbrace{\nabla h(\mathbf{x})(\widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{u})}_{\widehat{\dot{h}}(\mathbf{x},\mathbf{u})} + \underbrace{\nabla h(\mathbf{x})\mathbf{b}(\mathbf{x})}_{b(\mathbf{x})} + \underbrace{\nabla h(\mathbf{x})\mathbf{A}(\mathbf{x})}_{\mathbf{a}(\mathbf{x})^\top} \mathbf{u}. \qquad (7)$$

Given that h is a CBF for (5) on $\mathcal{C}$, let $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ be a Lipschitz continuous state-feedback controller such that $\widehat{\dot{h}}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \geq -\alpha(h(\mathbf{x}))$. Defining the *projected disturbance* as:

$$\delta(\mathbf{x}) \triangleq \dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) - \widehat{\dot{h}}(\mathbf{x}, \mathbf{k}(\mathbf{x})) = b(\mathbf{x}) + \mathbf{a}(\mathbf{x})^\top \mathbf{k}(\mathbf{x}), \qquad (8)$$

yields:

$$\dot{h}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \geq -\alpha(h(\mathbf{x})) - \delta(\mathbf{x}) \qquad (9)$$

Assuming that δ is essentially bounded in time (there exists $M \in \mathbb{R}$, $M > 0$, such that $|\delta|_\infty \triangleq \operatorname{ess\,sup}_{t \geq 0} |\delta(\mathbf{x}(t))| < M$), we may make use of the following definition:

Definition 3 (*Projection-to-State Safety (PSSf)* (Taylor et al. (2021))) *Given a feedback controller $\mathbf{k}$, the closed-loop system (2), $\dot{\mathbf{x}} = \mathbf{f}_{cl}(\mathbf{x}) = \widehat{\mathbf{f}}(\mathbf{x}) + \widehat{\mathbf{g}}(\mathbf{x})\mathbf{k}(\mathbf{x}) + \mathbf{d}(\mathbf{x})$ with $\mathbf{d}(\mathbf{x}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{k}(\mathbf{x})$, is* projection-to-state safe *(PSSf) on $\mathcal{C}$ with respect to the function h and projected disturbances $\delta : \mathbb{R}^n \to \mathbb{R}$ if there exists $\overline{\delta} > 0$ and $\gamma \in \mathcal{K}_\infty$ such that the set $\mathcal{C}_\delta \supset \mathcal{C}$,*

$$\mathcal{C}_\delta \triangleq \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) + \gamma(|\delta|_\infty) \geq 0\}, \qquad (10)$$

is forward invariant for all δ satisfying $|\delta|_\infty \leq \overline{\delta}$.

PSSf captures the fact that in the presence of model uncertainty, satisfying the CBF condition (4) for the estimated time derivative $\widehat{\dot{h}}$ is not sufficient for safety, as the projected disturbance δ appears in the lower bound on true time derivative of h as in (9). This results in a larger forward invariant set, given by $\mathcal{C}_\delta$, that grows with the magnitude of the projected disturbance.
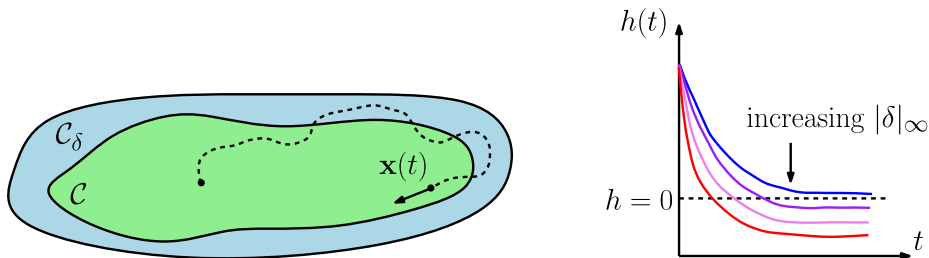
Figure 1. Geometric visualization of Projection-to-State Safety. The system is able to leave the safe set $\mathcal{C}$, but remains within the larger set $\mathcal{C}_\delta$. Maximum possible deviation from the safe set grows larger with $|\delta|_\infty$.

3. Learning Projected Disturbances

In this section we explore how an estimate of the projected disturbance δ can be learned episodically from data and incorporated into control synthesis to improve PSSf behavior.

As seen in Section 2, the projected disturbance δ appears in the time derivative of the barrier function $\dot{h}$, and potentially leads to unsafe behavior since it compromises the CBF condition as in (9). If an upper bound $\overline{\delta}$ on $|\delta|_\infty$ is known (or determined heuristically), it could be directly incorporated into the inequality enforced in the controller:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \qquad\qquad (\overline{\delta}\text{-CBF-QP})$$

$$\text{s.t.} \qquad \widehat{\dot{h}}(\mathbf{x}, \mathbf{u}) - \overline{\delta} \geq -\alpha(h(\mathbf{x})).$$

While this will enforce safety of the original set $\mathcal{C}$, it can be exceedingly conservative if $\overline{\delta}$ is larger than the actual projected disturbance. Furthermore, as the projected disturbance is a function of the state, its magnitude (and possibly sign) may change along a trajectory, leading to additional conservativeness in this approach.

Instead, we consider a learning approach to resolve the impact of δ. To motivate such an approach, consider the following setting: in an experiment, the system is allowed to evolve forward in time from a particular initial condition and under a given state-feedback controller. During this experiment, data is collected which provides a discrete-time history of the CBF, h. This time history is smoothed and numerically differentiated to compute an approximate time history of the true value of the time derivative of the CBF, $\dot{h}$. This yields a collection of input-output pairs:

$$D_i = ((\mathbf{x}_i, \mathbf{k}(\mathbf{x}_i)), \dot{h}_i) \in (\mathbb{R}^n \times \mathbb{R}^m) \times \mathbb{R} \qquad\qquad (11)$$

whereby a dataset $\mathfrak{D} = \{D_i\}_{i=1}^N$ can be constructed. Given a nonlinear function class $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}$ and a loss function $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, a learning problem can be specified as finding a function $\widehat{\delta} \in \mathcal{H}$ to estimate δ via empirical risk minimization:

$$\inf_{\widehat{\delta}\in\mathcal{H}} \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left(\widehat{\dot{h}}(\mathbf{x}_i, \mathbf{k}(\mathbf{x}_i)) + \widehat{\delta}(\mathbf{x}_i), \dot{h}_i\right). \qquad\qquad (\text{ERM})$$

A controller can be synthesized which incorporates $\widehat{\delta}$ as follows:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \qquad\qquad (\widehat{\delta}\text{-CBF-QP})$$

$$\text{s.t.} \qquad \widehat{\dot{h}}(\mathbf{x}, \mathbf{u}) + \widehat{\delta}(\mathbf{x}) \geq -\alpha(h(\mathbf{x})).$$

Note that compared with CBF-QP, the extended safe set with $\widehat{\delta}$-CBF-QP shrinks from (10) to $\mathcal{C}_\delta = \left\{ \mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) + \gamma(|\delta - \hat{\delta}|_\infty) \geq 0 \right\}$.

We directly build upon the episodic learning framework from (Taylor et al., 2019a,b) by seeking to learn δ. Our approach is outlined in Algorithm 1. In each episode, the algorithm runs the current controller to collect data, learns a new $\widehat{\delta}$ using the newly collected data, and synthesizes a new controller. In this prior work, which was applied to less complex dynamical systems, the collected data was rich enough to determine a control affine structure. In many contexts, such as bipedal robots, such a degree of diversity is infeasible without damaging the system. We instead directly learn δ as a function of the previous controller $\mathbf{k}$ via a recursive relationship, as updating the estimator leads to the definition of a new projected disturbance $\delta' = b(\mathbf{x}) + \mathbf{a}(\mathbf{x})^\top \mathbf{k}'(\mathbf{x})$. This yields a projected disturbance δ learned iteratively by modifying $\widehat{\delta}$ over the course of multiple episodes. This episodic approach to safety-critical control is captured in Algorithm 1.

Algorithm 1 Projected Disturbance Learning (PDL)

---

input : CBF h, CBF derivative estimate $\widehat{h}$, model class $\mathcal{H}$, loss function $\mathcal{L}$, nominal state-feedback controller $\mathbf{k}_0$, number of episodes T, initial condition $\mathbf{x}_0$

output: Augmented Controller $\mathbf{k}_T$

for $j = 1, \ldots, T$ **do**
    $\mathfrak{D}_j \leftarrow$ experiment$(\mathbf{x}_0, \mathbf{k}_{j-1})$ // Execute experiment
    $\widehat{\delta} \leftarrow$ ERM$(\mathcal{H}, \mathcal{L}, \mathfrak{D}_j, \widehat{h}_0)$ // Fit estimator
    $\widehat{h}_j \leftarrow \widehat{h}_0 + \widehat{\delta}$ // Update derivative estimator
    $\mathbf{k}_j \leftarrow \widehat{\delta}$-CBF-QP$(\widehat{h}_j)$ // Synthesize new controller
end

---

4. Bipedal Robotics

In this section we specify the notion of learning projected disturbances to the setting of bipedal locomotion. We briefly introduce the theory of bipedal locomotion and then describe the barrier function formulations which allow us to achieve safe bipedal locomotion across stepping-stones. A deeper exploration of this material may be found in (Westervelt et al., 2003).

4.1. Bipedal Robotic Dynamics

The bipedal robotic system we consider is the AMBER-3M robotic platform seen in Figure 2, modeled as an underactuated, planar five-link robot with point feet (Ambrose et al., 2017) whose physical parameters are reported in (Ma et al., 2014, Table 1). The configuration coordinates $\mathbf{q} \in \mathcal{Q} \subset \mathbb{R}^5$ are given by $\mathbf{q} = [q_{sf}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}]^\top$, with stance foot angle q_{sf}, stance knee angle q_{sk}, stance hip angle q_{sh}, non-stance hip angle q_{nsh} and non-stance knee angle q_{nsk}. The continuous-time equations of motion, derived from the Euler-Lagrange equations, are given by:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u}, \tag{12}$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{S}^5_{++}$ is the positive definite mass-inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{5 \times 5}$ contains the centrifugal and Coriolis forces, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^5$ contains the gravitational forces, $\mathbf{B} \in \mathbb{R}^{5 \times 4}$ is the actuation

matrix, and $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^4$ is the input. Note that this is the pinned model of the robot dynamics; for the unpinned model, refer to (Hereid et al., 2018). For AMBER-3M, the number of inputs is one fewer than the degrees of freedom, meaning the system has one degree of underactuation.

Taking $p^v : \mathcal{Q} \to \mathbb{R}$ to represent the vertical position (height) of the swing foot, the admissible states are given by the *domain* $\mathcal{D} = \{(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \mid p^v(\mathbf{q}) \geq 0\}$. The switching surface on which the impact events occur, also known as the *guard*, is defined by:

$$\mathcal{S} = \{(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q} \mid p^v(\mathbf{q}) = 0, \dot{p}^v(\mathbf{q}, \dot{\mathbf{q}}) < 0\} \subset \mathcal{D}. \tag{13}$$

With the full system state given by $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q}$, the impact dynamics (Westervelt et al., 2007) are defined by a *reset map* $\mathbf{\Delta} : \mathcal{S} \to \mathcal{D}$ relating pre-impact states $\mathbf{x}^-(t) \triangleq \lim_{\tau \nearrow t} \mathbf{x}(\tau)$ and post-impact states $\mathbf{x}^+(t) = \lim_{\tau \searrow t} \mathbf{x}(\tau)$ via $\mathbf{x}^+(t) = \mathbf{\Delta}(\mathbf{x}^-(t))$. Combining these concepts and rearranging (12) into control affine form yields the following *hybrid control system*:

$$\mathcal{HC} = \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} & \mathbf{x}^- \in \mathcal{D} \setminus \mathcal{S}, \\ \mathbf{x}^+ = \mathbf{\Delta}(\mathbf{x}^-) & \mathbf{x}^- \in \mathcal{S}. \end{cases} \tag{14}$$

4.2. Bipedal Robotic Control

Control of bipedal robotic systems centers around a *phasing variable*, $\tau : \mathcal{Q} \to [0, 1]$, given by:

$$\tau(\mathbf{q}) = \frac{\delta_{hip}(\mathbf{q}) - \delta_{hip}^+}{\delta_{hip}^- - \delta_{hip}^+}, \tag{15}$$

where $\delta_{hip} : \mathbb{R}^5 \to \mathbb{R}$ defined as $\delta_{hip}(\mathbf{q}) = [-l_t - l_f, \ -l_f, \ 0, \ 0, \ 0]\mathbf{q}$ is the linearized hip position with l_t and l_f the length of the robots tibia and femur, respectively. The constants δ_{hip}^+ and δ_{hip}^- are the linearized hip positions at the beginning and the end of a step, ensuring that $\tau(\mathbf{q})$ increases monotonically in time within a step. Desired trajectories resulting in walking gaits for the robot can be rapidly synthesized via a hybrid zero dynamics framework (Westervelt et al. (2003); Ames et al. (2014b)). We are now well equipped to define the relative degree 2 (Sastry (1999)) outputs $\mathbf{y} : \mathcal{Q} \to \mathbb{R}^4$ as the difference between the actual output $\mathbf{y}_a$ and the desired output trajectory $\mathbf{y}_d$:

$$\mathbf{y}(\mathbf{q}, \boldsymbol{\alpha}) \triangleq \mathbf{y}_a(\mathbf{q}) - \mathbf{y}_d(\tau(\mathbf{q}), \boldsymbol{\alpha}), \tag{16}$$

with $\boldsymbol{\alpha}$ being the coefficients of a Bézier polynomial coming from the trajectory generation step. The actual output is given by the actuated coordinates: $\mathbf{y}_a(\mathbf{q}) = \begin{bmatrix} \mathbf{0}_{4 \times 1} & \mathbf{I}_{4 \times 4} \end{bmatrix} \mathbf{q}$. The nominal controller for this system is then given by the proportional-derivative controller $\mathbf{k}_d(\mathbf{x}) = \mathbf{k}_{PD}(\mathbf{x}) \triangleq -\mathbf{K}_P \mathbf{y}(\mathbf{q}) - \mathbf{K}_D \dot{\mathbf{y}}(\mathbf{q})$ with proportional gain $\mathbf{K}_P \in \mathbb{S}_{++}^4$ and derivative gain $\mathbf{K}_D \in \mathbb{S}_{++}^4$.

4.3. Control Barrier Functions for Stepping-Stones

The stepping-stone problem is captured through the use of *virtual* stepping-stones, which shrink over the course of a step to confine foot placement to a safe region defined on a targeted stone (Grandia et al., 2020). The CBFs used to specify these foot position constraints are given by:

$$h_1(\mathbf{q}) = R(\tau(\mathbf{q})) - (O_x - F_x(\mathbf{q})), \tag{17}$$
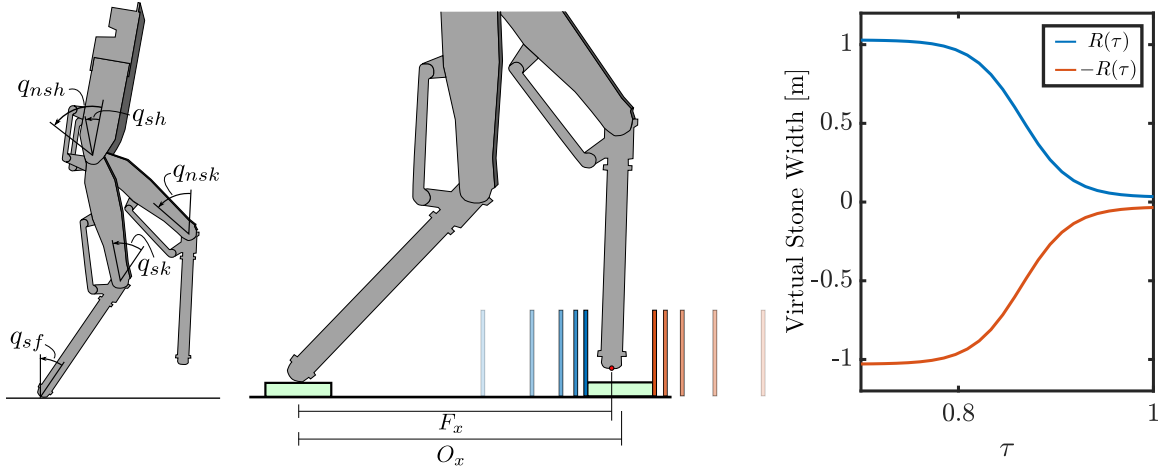$$h_2(\mathbf{q}) = R(\tau(\mathbf{q})) + (O_x - F_x(\mathbf{q})), \tag{18}$$

Figure 2. (Left): Schematic diagram of the AMBER-3M robot with position coordinates. **(Center):** Schematic of the foot placement in the stepping-stone problem. The boundaries of virtual stepping-stones are captured via the blue and orange vertical lines. **(Right):** Virtual stepping stone width as function of the phase variable $\tau(\mathbf{q})$.

where $F_x(\mathbf{q})$ is the horizontal position of the swing foot and $O_x > 0$ is the horizontal position of the center of stepping-stone. The virtual stone width is given by the function $R : \mathbb{R} \to \mathbb{R}$:

$$R(\tau(\mathbf{q})) = \frac{ar - 1}{1 + ar(e^{-m(\tau(\mathbf{q})-1)} - 1)} + 1 + r \tag{19}$$

where $m > 0$ determines the decay rate of the barrier function, $(1 + a)r$ is half of the targeted stone width, and $1 + r$ defines the half the width of the virtual stepping-stone when $\tau = 0$. These functions are visualized in Figure 2. The safety constraints can be interpreted as keeping the swing foot horizontal position in an interval centered at the middle of the stepping-stone, where the interval shrinks as τ increases. As this formulation of CBFs is position-based and therefore relative degree two, we employ the exponential control barrier function (ECBF) extension technique (Nguyen and Sreenath, 2016) to both CBFs to attain the relative degree 1 CBFs: $h_{e,i}(\mathbf{x}) \triangleq L_{\mathbf{f}} h_i(\mathbf{x}) + \alpha_e h_i(\mathbf{q})$.

Combining the results in Section 3 and 4, with $\hat{\mathbf{f}}$ and $\hat{\mathbf{g}}$ the nominal model in (14), the final Stepping Stone QP (SS-QP) controller combines the robustifying term of ($\hat{\delta}$-CBF-QP) with the stepping-stone ECBF extensions of (17) and (18):

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \frac{1}{2} \|\mathbf{u} - \mathbf{k}_{PD}(\mathbf{x})\|_2^2 \tag{SS-QP}$$

$$\text{s.t.} \quad L_{\hat{\mathbf{f}}}^2 h_1(\mathbf{x}) + L_{\hat{\mathbf{g}}} L_{\hat{\mathbf{f}}} h_1(\mathbf{x}) \mathbf{u} + \alpha_e L_{\hat{\mathbf{f}}} h_1(\mathbf{x}) + \widehat{\delta}_1(\mathbf{x}) \geq -\alpha(h_{e,1}(\mathbf{x})) \tag{20}$$

$$L_{\hat{\mathbf{f}}}^2 h_2(\mathbf{x}) + L_{\hat{\mathbf{g}}} L_{\hat{\mathbf{f}}} h_2(\mathbf{x}) \mathbf{u} + \alpha_e L_{\hat{\mathbf{f}}} h_2(\mathbf{x}) + \widehat{\delta}_2(\mathbf{x}) \geq -\alpha(h_{e,2}(\mathbf{x})). \tag{21}$$

We assume that this (SS-QP) is feasible and we encountered no infeasibilities in simulation or experimentation.

5. Simulation and Experimental Validation

In this section we apply our episodic learning framework (Algorithm 1) to the AMBER-3M platform in both simulation with injected model uncertainty and on hardware with the model error inherent

to real-world systems. In each instance the estimator $\widehat{\delta}$ was implemented as a neural network with two hidden layers of 50 hidden units using the ReLU activation function. The network was trained minimizing mean absolute error using mini-batch gradient descent. Mean absolute error was chosen over other loss functions for its robustness to outliers. The same controller (SS-QP) was deployed in the RaiSim (Hwangbo et al., 2018) simulation environment and on the AMBER-3M hardware platform, as seen in the supplementary video (vid). The complete learning code used in simulation and experiments can be found at (git).

5.1. Simulation

The controllers and learning algorithm were first validated in simulation. Model error was introduced by increasing the inertia of all limbs on the true model by a factor of ten while maintaining constant mass. Due to the underactuated nature of the robot and the relationship between step length and zero dynamics stability, not every set of stepping stones is navigable, even if safety is perfectly enforced with respect to the CBFs. Therefore, a feasible stepping stone configuration was first generated for the robot to traverse with stones of 4 cm in width. Without knowledge of the modified model $(\widehat{\delta}_1(\mathbf{x}) = \widehat{\delta}_2(\mathbf{x}) = 0)$, the controller did not satisfy the CBF constraints (20-21), resulting in a maximum violation at foot placement of 2.0 cm, causing the robot to miss the stepping stone and fall over. Three episodes of the PDL algorithm were run, after which the maximum violation was reduced to be 0.3 cm, only 15% of the original violation. Additionally, the $\bar{\delta}$-CBF-QP controller was implemented, which ensured safety but resulted in extremely conservative behavior, resulting in poor qualitative walking, i.e. harsh foot strikes and an over-bending torso. A comparison of the barrier functions h_1 and h_2 over the steps with these controllers can be seen in Figure 3.
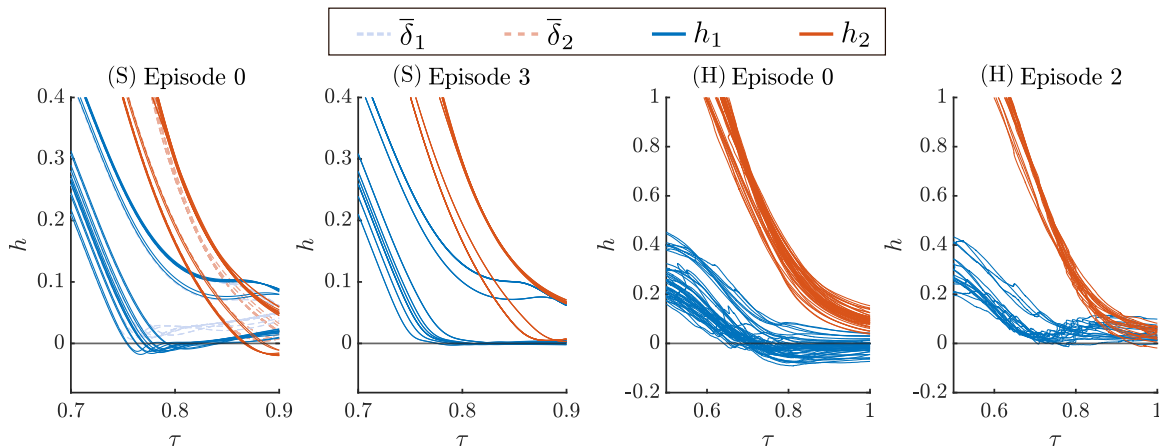

Figure 3. Simulation (S) and Hardware (H) data where model mismatch causes violations. **(Far-Left)**: Simulation where the barrier functions h_1 (solid blue) and h_2 (solid orange) are enforced via a CBF-QP. The $\bar{\delta}$-CBF-QP is also shown for $\bar{\delta}_1$ (dashed blue) and $\bar{\delta}_2$ (dashed orange), which results in more conservative behavior over many steps. **(Mid-Left)**: After three episodes of learning the SS-QP in simulation, the maximum barrier violation decreases from 2.0 to 0.3 cm. **(Mid-Right)**: Hardware where the barrier functions h_1 (blue) and h_2 (orange) enforced via a CBF-QP. **(Far-Right)**: After two episodes of learning on hardware, the maximum barrier violation decreases from 9.2 to 1.9 cm via the SS-QP.
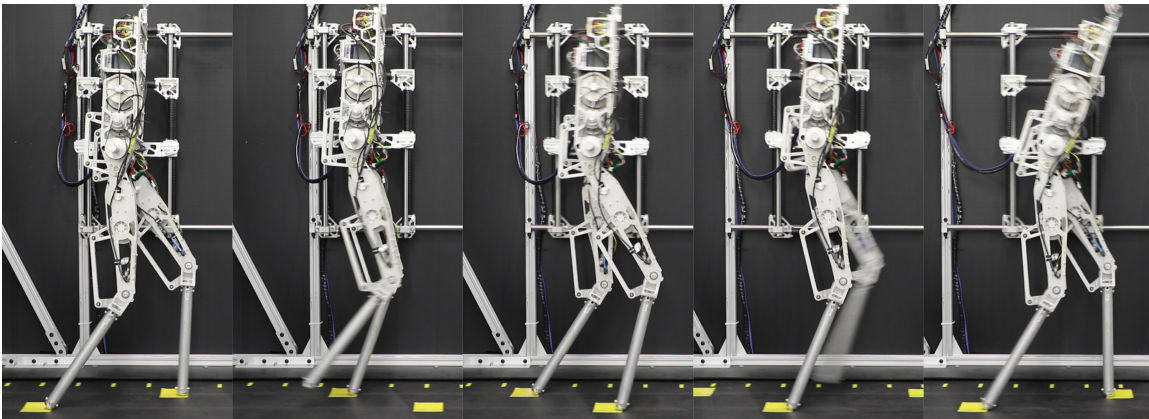
Figure 4. Gait tiles for Episode 2 of learning showing the AMBER-3M robot safely traversing a set of stepping stones. Notice the change in step width and added lean of the torso induced by the barrier functions.

5.2. Hardware

The same nominal model for the robot was used in the hardware experiments as in simulation, with model uncertainty presenting itself as significant friction in the joints, as well as imperfect mass and inertia measurements. The PDL algorithm was implemented on the AMBER-3M robot across a sequence of two episodes. The controllers ran on an off-board i7-6700HQ CPU @ 2.6GHz with 16 GB RAM, which computed desired torques and communicated them with the ELMO motor drivers on the 137 cm tall, 22 kg robot. The motor driver communication ran at 2kHz, and the SS-QP ran at 1kHz. The stepping stone configuration was specified to the controller with stones of 8 cm in width. As with simulation, the CBF-QP resulted in a maximum violation of the barriers of 9.2 cm due to model error. After running the PDL algorithm for two episodes, the maximum violation of the barriers was 1.9 cm, only 21% of the original violation, as depicted in Figure 3. Although learning improves our estimate of the safe set size, as discussed in Section 3, there is still uncertainty in the stone size. This was accommodated by utilizing physical stones which were 10 cm in width. The 7.3 cm reduction in stone size mismatch captures the change from $|\delta|_\infty$ to $|\delta - \widehat{\delta}|_\infty$. Gait tiles for this improved traversal of the stepping stones are shown in Figure 4.

6. Conclusion

In this paper we presented an episodic learning approach for reducing the impact of model uncertainty on safety-critical control using control barrier functions. Our method is able to learn a projected disturbance and incorporate the learned model information into an optimization-based controller, as demonstrated in both a high-fidelity simulation and on the AMBER-3M robot hardware platform. Future work includes theoretical analysis of the convergence properties of Algorithm 1, modification of the controller to ensure trajectories remain close to previously-seen data points to reduce the effects of generalization error, and an extension to three dimensional bipedal walking with vision sensing for stepping-stone identification.

References

Learning Code, https://github.com/noelc-s/amber_learning.

Supplementary video, https://vimeo.com/481809664.

W.-L. Ambrose, E.and Ma, C. Hubicki, and A. D. Ames. Toward benchmarking locomotion economy across design configurations on the modular robot: Amber-3m. In *Conference on Control Technology and Applications (CCTA)*, pages 1270–1276. IEEE, 2017.

A. Ames, J. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Conference on Decision & Control (CDC)*, pages 6271–6278. IEEE, 2014a.

A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *Transactions on Automatic Control*, 59(4):876–891, 2014b.

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

F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *Conference on Decision & Control (CDC)*, pages 4661–4666. IEEE, 2016.

F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath. Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects. *arXiv preprint arXiv:2011.07183*, 2020.

C. Cheng, X. Yan, E. Theodorou, and B. Boots. Accelerating imitation learning with predictive models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath. Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions. In *Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020. XVI.088.

S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. *arXiv preprint arXiv:2010.16001*, 2020.

D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A.-a. Agha-mohammadi, and E. A. Theodorou. Bayesian learning-based adaptive control for safety critical systems. *arXiv preprint arXiv:1910.02325*, 2019.

P. Fankhauser, M. Bjelonic, D. Bellicoso, T. Miki, and M. Hutter. Robust rough-terrain locomotion with a quadrupedal robot. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *Transactions on Automatic Control*, 2018.

R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. *arXiv preprint arXiv:2011.00032*, 2020.

R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt. Footstep planning for autonomous walking over rough terrain. In *International Conference on Humanoid Robots (Humanoids)*, pages 9–16. IEEE-RAS, 2019.

J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on automatic control*, 46(1):51–64, 2001.

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 *International Conference on Cyber-Physical Systems (ICCPS)*, pages 98–106. IEEE Press, 2018.

Ayonga Hereid, Christian M Hubicki, Eric A Cousineau, and Aaron D Ames. Dynamic humanoid locomotion: A scalable formulation for hzd gait optimization. *IEEE Transactions on Robotics*, 34(2):370–387, 2018.

Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

S. Kolathaya and A. D. Ames. Input-to-state safety with control barrier functions. *Control Systems Letters*, 3(1):108–113, 2018.

J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020.

W.-L. Ma, H.-H. Zhao, S. Kolathaya, and A. D. Ames. Human-inspired walking via unified pd and impedance control. In *International Conference on Robotics and Automation (ICRA)*, pages 5088–5094. IEEE, 2014.

Q. Nguyen and K. Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *American Control Conference (ACC)*, pages 322–328. IEEE, 2016.

Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath. 3d dynamic walking on stepping stones with control barrier functions. In *Conference on Decision and Control (CDC)*, pages 827–834. IEEE, 2016.

Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath. Dynamic bipedal locomotion over stochastic discrete terrain. *The International Journal of Robotics Research*, 37(13-14): 1537–1553, 2018. doi: 10.1177/0278364918791718. URL https://doi.org/10.1177/0278364918791718.

Q. Nguyen, X. Da, J.W. Grizzle, and K. Sreenath. Dynamic walking on stepping stones with gait library and control barrier functions. In *Algorithmic Foundations of Robotics XII*, pages 384–399. Springer, 2020.

L. Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.

S. S. Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer, NY, 1999.

A. J. Taylor and A. D. Ames. Adaptive safety with control barrier functions. In *American Control Conference (ACC)*, pages 1399–1405. IEEE, 2020.

A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames. Episodic learning with control lyapunov functions for uncertain robotic systems. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6878–6884. IEEE, 2019a.

A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames. Learning for safety critical control with control barrier functions. *arXiv preprint arXiv:1912.10099*, 2019b.

A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames. A control barrier perspective on episodic learning via projection-to-state safety. *IEEE Control Systems Letters*, 5(3):1019–1024, 2021.

L. Wang, E. A. Theodorou, and M. Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *International Conference on Robotics and Automation (ICRA)*, pages 2460–2465. IEEE, 2018.

E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek. Hybrid zero dynamics of planar biped walkers. *Transactions on Automatic Control*, 48(1):42–56, 2003.

E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC Press, 2007.

X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.