

# Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation

Ayush Agrawal and Koushil Sreenath.

**Abstract**—In this paper, we extend the concept of control barrier functions, developed initially for continuous time systems, to the discrete-time domain. We demonstrate safety-critical control for nonlinear discrete-time systems with applications to 3D bipedal robot navigation. Particularly, we mathematically analyze two different formulations of control barrier functions, based on their continuous-time counterparts, and demonstrate how these can be applied to discrete-time systems. We show that the resulting formulation is a nonlinear program in contrast to the quadratic program for continuous-time systems and under certain conditions, the nonlinear program can be formulated as a quadratically constrained quadratic program. Furthermore, using the developed concept of discrete control barrier functions, we present a novel control method to address the problem of navigation of a high-dimensional bipedal robot through environments with moving obstacles that present time-varying safety-critical constraints.

## I. INTRODUCTION

Barrier function based control techniques have recently gained success in a wide variety continuous-time systems for safety-critical applications such as precise footstep placement of high degree of freedom bipedal robots [23, 22, 24, 26, 27], adaptive cruise control systems [2], multi-agent systems [30, 4], and quadrotor systems [33, 34, 32]. In these, the problem of stabilization with guaranteed safety is posed as a constrained convex optimization problem, that combines control Lyapunov and control barrier functions, and solves for an optimal control input that maintains the states of the system with a predefined safety set. In [28], control barrier and control Lyapunov functions are combined through an analytical framework.

In this paper, we extend the concept of Control Barrier Functions (CBFs) to discrete-time dynamical systems. Particularly, we analyze the formulation of CBFs presented in [2, 25] and show that these can be applied directly to discrete-time systems. Interestingly, however, unlike their continuous-time counterparts, we find that the resulting optimization problem is not necessarily convex and is in fact nonlinear. For the formulation of exponential barrier functions presented in [25], we show that under certain conditions, the optimization problem is a Quadratically Constrained Quadratic Program (QCQP).

This work is supported by NSF grants IIS-1526515 and CMMI-1538869. A. Agrawal and K. Sreenath are with the Dept. of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, email: {ayushagrawal, koushils}@cmu.edu.

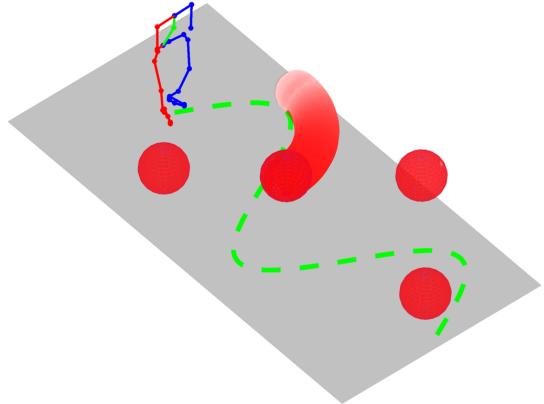


Fig. 1: An example scenario of a humanoid robot in a dynamically changing environment. We present a navigation scheme that can handle static/moving obstacles (represented by red spheres), while trying to follow a desired path (represented in green).

We then apply these concepts to address the challenge of bipedal robot navigation in cluttered and dynamically changing environments such as in indoor urban spaces with moving obstacles, stairs and narrow passages. An example of such a scenario is presented in Fig. 1. Motion planning for humanoid robots has been commonly addressed through planning of footstep placements in constrained environments to handle obstacles, while operating within kinematic limits of the robot. In [20], a vision-based foot placement method is proposed, and implemented on the humanoid robot ASIMO, that searches over a discrete set of actions to avoid obstacles while reaching the goal. A real-time planning algorithm is presented in [3] that utilizes RRTs to search over dense, pre-computed swept volumes to avoid collision with moving 3D obstacles. Other work related to footstep placement planning include [19, 6, 5, 18]. In [8], a Mixed Integer Convex Optimization Program is used to plan footstep placements for bipedal robot walking on uneven terrains with obstacles. A comprehensive review of motion planning for humanoid robots, including whole body motion planning can be found in [15].

In the context of limit-cycle walkers, footstep placement strategies based on barrier functions to traverse terrains with discrete footholds is presented in [22, 24, 26] for 2D robots and in [23] for high degree of freedom 3D robots. Navigation

planning of 3D dynamic walkers has been studied in [11] and more recently in [21] by composing asymptotically stable motion primitives. Additionally, [21] provides analytical stability guarantees.

In our work, obstacle-free regions are treated as *safe sets* in the task space. Discrete-time versions of control barrier and control Lyapunov functions, developed in this paper, are used as high-level planners that issue an optimal heading angle at every step to follow a desired path (obtained through path-planning algorithms like  $A^*$ ) while simultaneously keeping the robot in the safe set. A low-level Hybrid Zero Dynamics (HZD) based controller then executes this plan.

The contributions of this paper with respect to prior work are as follows:

- We introduce the concept of control barrier functions [2, 25] for nonlinear discrete-time systems and mathematically show the forward invariance of sets associated with these barrier functions (known as *Safety Sets*).
- Using the concept of discrete-time control barrier functions developed in this paper, we design feedback controllers for high degree-of-freedom bipedal robots to follow a desired path in the task space while avoiding time-varying unsafe regions such as moving obstacles.
- We evaluate our proposed controller through numerical simulations for different desired paths and for static and moving obstacles in the task space.

The rest of the paper is organized as follows. In Section II, we introduce the concept of Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs) for discrete-time dynamical systems and show the invariance of sets associated with CBFs. We apply the concept of discrete-time CLFs and CBFs to linear systems and show results from numerical simulations. In Section III, we apply the concept of *Exponential CBFs* developed in [25] to discrete-time systems. We apply this concept to both linear and nonlinear systems and show results from numerical simulations. In Section IV, we develop an event-based controller for a high dimensional bipedal robot for the purpose of following a desired path while simultaneously avoiding obstacles in the task space. Finally, in Section V, we conclude by highlighting certain drawbacks of our controller design and indicating directions for future work.

## II. LYAPUNOV AND BARRIER FUNCTIONS FOR DISCRETE-TIME SYSTEMS

In this section, we take the concept of barrier functions developed for continuous-time systems, particularly in [2, 25] and translate it to discrete-time systems that are given by:

$$x(k+1) = f(x(k)), \quad (1)$$

where  $x(k) \in \mathcal{D} \subset \mathbb{R}^n$  represents the state of the system at time step  $k \in \mathbb{Z}^+$  and  $f : \mathcal{D} \rightarrow \mathcal{D}$  is a continuous function.

Note: Throughout the rest of this paper, we will represent the state of the system  $x(k)$  at time step  $k$  as  $x_k$ .

We show that the same formulation of barrier functions developed for continuous-time systems can be applied to

discrete-time systems as well. An important observation for discrete-time systems that we will see later is that, unlike continuous-time systems, the resulting Lyapunov and barrier conditions are not necessarily affine in the control input. Rather, they depend on the nature of the system as well as the nature of the chosen Lyapunov and barrier functions themselves. Implementing Control Lyapunov and control barrier functions through optimization may result in a potentially non-convex nonlinear program. For the formulation of CBFs in [25], we show that the resulting optimization problem is convex for a class of nonlinear systems and CLFs/CBFs.

### A. Lyapunov Functions for Discrete-Time Systems

Consider the nonlinear discrete-time control system,

$$x_{k+1} = f(x_k, u_k), \quad (2)$$

where  $u_k \in \mathbb{U} \subset \mathbb{R}^m$  is the control input at time step  $k$  and  $\mathbb{U}$  is the set of admissible control inputs.

**Definition 1.** (Discrete-Time Exponentially stabilizing Control Lyapunov Function): A map  $V : \mathcal{D} \rightarrow \mathbb{R}$  is an Exponential Control Lyapunov Function for the discrete-time control system (2) if there exists:

- 1) positive constants  $c_1$  and  $c_2$  such that

$$c_1 \|x_k\|^2 \leq V(x_k) \leq c_2 \|x_k\|^2, \quad (3)$$

and

- 2) a control input  $u_k : \mathcal{D} \rightarrow \mathbb{U}$ ,  $\forall x_k \in \mathcal{D}$  and  $c_3 > 0$  such that

$$\Delta V(x_k, u_k) + c_3 \|x_k\|^2 \leq 0, \quad (4)$$

where,  $\Delta V(x_k, u_k) := V(x_{k+1}) - V(x_k) = \Delta V_k$ .

*Remark 1.* The control input  $u_k$  renders the origin exponentially stable. See [17] for a comprehensive stability analysis for discrete-time systems.

The concept of discrete-time control lyapunov functions has been previously studied in [14] and applied to an automotive engine control problem.

Similar to [9] for continuous-time systems, the CLF condition (4) can be enforced through a constrained optimization program:

$$\begin{aligned} u_k^* &= \underset{u_k \in \mathcal{D}}{\operatorname{argmin}} u_k^T u_k \\ \text{s.t. } &\Delta V(x_k, u_k) + c_3 \|x_k\|^2 \leq 0. \end{aligned} \quad (5)$$

### B. Barrier Functions for Discrete-Time Systems

We next consider a *Safety Set*  $\mathcal{S}$  and establish its forward invariance. The safety set and its boundary are respectively defined as,

$$\mathcal{S} := \{x(k) \in \mathcal{D} \mid h(x(k)) \geq 0\}, \quad (6)$$

$$\partial \mathcal{S} := \{x(k) \in \mathcal{D} \mid h(x(k)) = 0\}, \quad (7)$$

for a smooth function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  associated with a Barrier function defined similar to [2] as below.

**Definition 2.** (Discrete-Time Barrier Function): A function  $B : \mathcal{S} \rightarrow \mathbb{R}$  is a barrier function for the set  $\mathcal{S}$  if there exists locally Lipschitz class  $\mathcal{K}$  functions  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  such that

$$\frac{1}{\alpha_1(h(x_k))} \leq B(x_k) \leq \frac{1}{\alpha_2(h(x_k))}, \quad (8)$$

$$\Delta B(x_{k+1}, x_k) := B(x_{k+1}) - B(x_k) \leq \alpha_3(h(x_k)). \quad (9)$$

To prove the invariance of the set  $\mathcal{S}$  for discrete-time systems, we will need few other results.

*Proposition 1.* Consider two sequences  $\{u_k\}_{k \geq 0}$  and  $\{v_k\}_{k \geq 0}$  and their first differences  $\{\Delta u_k\}$  and  $\{\Delta v_k\}$ . If  $\Delta u_k \leq \Delta v_k$  and  $u_0 \leq v_0$ , then  $u_k \leq v_k \forall k \geq 0$ .

*Proof:*

$$\begin{aligned} \Delta v_k &= v_{k+1} - v_k \\ \Rightarrow v_{k+1} &= v_k + \Delta v_k \\ &= v_0 + \sum_{i=0}^k \Delta v_i \\ &\leq u_0 + \sum_{i=0}^k \Delta u_i = u_{k+1}. \end{aligned}$$

*Proposition 2.* Consider the difference equation

$$\Delta v_k = \alpha \left( \frac{1}{v_k} \right), v_0 \geq 0,$$

where  $\alpha$  is a class  $\mathcal{K}$  function. Then  $v_k \geq 0, \forall k \in \mathbb{Z}^+$ .

*Proof:* The proof is straightforward and is derived from the fact that  $\alpha(\frac{1}{v_k}) \geq 0, \forall v_k \geq 0$ .

Since  $v_0 \geq 0$ ,

$$\begin{aligned} \Rightarrow \frac{1}{v_0} &\geq 0 \\ \Rightarrow \alpha \left( \frac{1}{v_0} \right) &\geq 0 \\ \Rightarrow v_1 &= v_0 + \alpha \left( \frac{1}{v_0} \right) \geq v_0 \end{aligned}$$

Inductively,  $0 \leq v_0 \leq v_1 \leq \dots \leq v_k, \forall k \in \mathbb{Z}^+$ .

We now prove the invariance of the Safety Set  $\mathcal{S}$ .

*Proposition 3. (Main Result)* If there exists a barrier function  $B(x_k)$  for the discrete-time system (1), then the set  $\mathcal{S}$  is invariant along the trajectories of (1), i.e. if  $x_0 \in \mathcal{S}$  then  $f(x_k) \in \mathcal{S}, \forall k \in \mathbb{Z}^+$ .

*Proof:*

$$\Delta B(x_{k+1}, x_k) \leq \alpha_3 \circ \alpha_2^{-1} \left( \frac{1}{B(x_k)} \right) =: \alpha \left( \frac{1}{B(x_k)} \right).$$

Consider now the difference equation  $\Delta \Gamma(x_k) = \alpha \left( \frac{1}{B(x_k)} \right)$ , with initial condition  $B(x_0) \geq 0$ . Let the solution be  $\Gamma(x_k)$ .

From Proposition 2,  $\Gamma(x_k) \geq 0, \forall k \in \mathbb{Z}^+$  and from Proposition 1,

$$B(x_k) \leq \Gamma(x_k) \quad (10)$$

$$\Rightarrow \frac{1}{B(x_k)} \geq \frac{1}{\Gamma(x_k)} \geq 0. \quad (11)$$

From the left-hand inequality of (8) we have,

$$\alpha_1^{-1} \left( \frac{1}{B(x_k)} \right) \leq h(x_k).$$

We define  $\alpha_4 := \alpha_1^{-1}$ , to be another class  $\mathcal{K}$  function. From the inequality in (11),

$$\begin{aligned} 0 &\leq \alpha_4 \left( \frac{1}{B(x_k)} \right) \leq h(x_k) \\ \Rightarrow B(x_k) &\geq 0 \Rightarrow x_k \in \mathcal{S}, \forall k \in \mathbb{Z}^+. \end{aligned}$$

■

**Definition 3. (Discrete-Time Control Barrier Function):** A mapping  $B : \mathcal{S} \rightarrow \mathbb{R}$  is a control barrier function for the discrete-time control system (2) if there exists:

- 1) class  $\mathcal{K}$  functions  $\alpha_1$  and  $\alpha_2$  such that

$$\frac{1}{\alpha_1(\|x_k\|_{\partial\mathcal{S}})} \leq B(x_k) \leq \frac{1}{\alpha_2(\|x_k\|_{\partial\mathcal{S}})}, \quad (12)$$

and

- 2) a control input  $u_k : \mathcal{S} \rightarrow \mathbb{U}, \forall x_k \in \mathcal{S}$  and  $\gamma > 0$  such that

$$\Delta B(x_k, u_k) - \frac{\gamma}{B(x_k)} \leq 0. \quad (13)$$

This essentially means that the control input  $u_k$  maintains the Barrier Function  $B_k \geq 0, \forall k \in \mathbb{Z}^+$  given that  $B_0 \geq 0$ . In other words,  $u_k$  keeps the trajectory  $x_k$  of the system within the safe set  $\mathcal{S}$ , given the initial state  $x_0$  lies in the  $\mathcal{S}$ .

### C. Combining Control Lyapunov and Control Barrier Functions

We pose the problem of combining CLFs and CBFs as a constrained optimization problem [2, 25]. For  $p \geq 0$ ,

$$\begin{aligned} u_k^* &= \underset{U_k}{\operatorname{argmin}} u_k^T u_k + p \cdot d^2 \\ U_k &= \begin{bmatrix} u_k \\ d \end{bmatrix} \in \mathbb{R}^{m+1} \\ \text{s.t. } \Delta V(x_k, u_k) + c_3 \|x_k\|^2 &\leq d \\ \Delta B(x_k, u_k) - \frac{\gamma}{B(x_k)} &\leq 0, \\ u_{\min} &\leq u_k \leq u_{\max}. \end{aligned} \quad (14)$$

Here  $d \geq 0$  is a relaxation term that allows the Lyapunov Function to grow when the CLF and CBF constraints are conflicting. Note that the CBF condition is always satisfied and the trajectory of the system  $x(k)$  always remains within the safe set  $\mathcal{S}$ . The terms  $u_{\min}$  and  $u_{\max}$  are bounds on the control input.

*Remark 2.* Note that we have kept our presentation of the discrete-time version of the CBF-CLF controller similar to the continuous-time case introduced in [2]. This enables readers to intuitively see the extension of the CBF-CLF formulation to the discrete-time domain.

*Remark 3.* With just the (relaxed) CLF and CBF constraints, the above optimization problem is always feasible. In the presence of additional constraints such as bounded control input, the feasibility is not always guaranteed. If the nominal trajectory satisfies these constraints, then the above optimization problem is locally feasible in a neighborhood of the nominal trajectory.

In the next subsection, we apply this method to find a control input that enforces safety-critical constraints for a discrete-time linear system.

#### D. Application to Linear Systems

Consider the discrete-time linear system  $x_{k+1} = Ax_k + Du_k$  with  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $D \in \mathbb{R}^{n \times m}$  and the safe set  $\mathcal{S}$  defined by the linear function

$$h(x_k) = Hx_k + F, \quad (15)$$

with  $H \in \mathbb{R}^{1 \times n}$  and  $F \in \mathbb{R}$ . We choose a quadratic control Lyapunov function  $V_k = x_k^T Px_k$  with  $P$  symmetric and positive-definite, obtained by solving the discrete-time Lyapunov equation,  $A^T PA - P = -Q$ , for a symmetric and positive definite matrix  $Q$ . The CLF condition (4) is then given by:

$$\begin{aligned} \Delta V_k + c_3 \|x_k\|^2 &= V_{k+1} - V_k + c_3 x_k^T x_k \\ &= x_{k+1}^T Px_{k+1} - x_k^T Px_k + c_3 x_k^T x_k \\ &= u_k^T D^T PDu_k + x_k^T (A^T PA - P)x_k + \\ &\quad 2x_k^T APDu_k + c_3 x_k^T x_k \leq 0. \end{aligned}$$

Similar to the continuous-time domain [2], we chose the following Control Barrier Function,

$$B_k = B(x_k) = \frac{1}{h(x_k)} = (Hx_k + F)^{-1}.$$

The CBF condition (13) then becomes

$$\begin{aligned} \Delta B_k - \frac{\gamma}{B_k} &= B_{k+1} - B_k - \frac{\gamma}{B_k} \\ &= -\frac{H(Ax_k - x_k + Du_k)}{(HAx_k + HDu_k + F)(Hx_k + F)} \\ &\quad - \gamma(Hx_k + F) \\ &\leq 0. \end{aligned}$$

*Remark 4.* Note that, unlike for the continuous-time case, the CLF and CBF conditions are no longer affine in the control input  $u_k$  and depend on the choice of  $V_k$  and  $B_k$ . This implies that for general nonlinear systems with nonlinear Lyapunov and Barrier Functions, the resulting optimization problem (14) is not necessarily quadratic, or even convex. Moreover, as we have seen, this is also true even for linear systems with quadratic Lyapunov functions and with linear

functions defining the safety set. In Section III, we will use a different formulation of the barrier function that results in a convex optimization problem for a class of nonlinear systems.

### III. DISCRETE TIME EXPONENTIAL CBF

In this section, we show the forward invariance of the safety set using the exponential control barrier function formulation, derived for continuous-time systems in [25], for the discrete-time system (1). Moreover, for the linear system with the safety set defined by a linear function, as described in Section II-D, the resulting optimization problem (14) turns into a convex, Quadratically Constrained Quadratic Program (QCQP), unlike a general (and potentially non-convex) nonlinear program with the formulation in (8).

We redefine the safety set,  $\mathcal{S}$  as:

$$\mathcal{S} = \{x_k \in \mathcal{D} \mid B(x_k) \geq 0\}, \quad (16)$$

where  $B : \mathcal{D} \rightarrow \mathbb{R}$  is called the discrete-time Exponential Barrier Function.

*Proposition 4.* The set  $\mathcal{S}$  is invariant along the trajectories of the discrete-time system (1) if there exists a map  $B : \mathcal{S} \rightarrow \mathbb{R}$  such that:

- 1)  $B_0 \geq 0$  and,
- 2)  $\Delta B_k + \gamma B_k \geq 0, \forall k \in \mathbb{Z}, 0 < \gamma \leq 1$

*Proof:* From Proposition 1,  $B_k \geq (1 - \gamma)^k B_0 \geq 0 \forall k \in \mathbb{Z}^+, 0 < \gamma \leq 1$ . This implies that  $x_k \in \mathcal{S}, \forall k \in \mathbb{Z}^+$ . ■

*Remark 5.*  $B_k$  is always greater than  $(1 - \gamma)^k B_0$ , an exponential function in  $k$ , and hence the name Exponential Control Barrier Function.

**Definition 4.** (Discrete-Time Exponential Control Barrier Function) A map  $B : \mathcal{D} \rightarrow \mathbb{R}$  is a Discrete-Time Exponential Control Barrier Function if:

- 1)  $B_0 \geq 0$  and,
- 2) there exists a control input  $u_k \in \mathbb{R}^m$  such that  $\Delta B(x_k, u_k) + \gamma B(x_k) \geq 0, \forall k \in \mathbb{Z}^+, 0 < \gamma \leq 1$ .

#### A. Linear System Revisited

In this section, we apply the Discrete-Time Exponential Barrier Function to the linear system presented in Section II-D. Note that  $B(x_k)$  is equal to  $h(x_k)$  defined in (15). The CBF condition now becomes:

$$\begin{aligned} \Delta B_k + \gamma B_k &= B_{k+1} + (\gamma - 1) B_k, \\ &= H(Ax_k + Du_k) + F + (1 - \gamma)(Hx_k + F) \\ &= H(A + (1 - \gamma)\mathbb{I})x_k + (2 - \gamma)F + HDu_k \\ &\geq 0 \end{aligned}$$

Similar to (14), a control input  $u_k$  that ensures the forward invariance of the set  $\mathcal{S}$  while driving the state  $x_k$  as close to the origin as possible can be obtained through a QCQP as:

$$\begin{aligned}
u_k^* &= \underset{U_k}{\operatorname{argmin}} U_k^T P_0 U_k \\
U_k &= \begin{bmatrix} u_k \\ d \end{bmatrix} \in \mathbb{R}^{m+1} \\
\text{s.t. } & U_k^T P_1 U_k + q_1^T U_k + r_1 \leq 0, \quad (D - \text{CLF}) \\
& U_k^T P_2 U_k + q_2^T U_k + r_2 \leq 0, \quad (D - \text{CBF})
\end{aligned} \tag{17}$$

where,

$$\begin{aligned}
P_0 &= \begin{bmatrix} \mathbb{I}_{m \times m} & p \end{bmatrix} \\
P_1 &= \begin{bmatrix} D^T P D & 0 \end{bmatrix} \\
P_2 &= 0_{m \times m} \\
q_1 &= [2x_k^T A P D \quad -1]^T \\
q_2 &= [-H D \quad 0]^T \\
r_1 &= x_k^T (A^T P A - P + c_3 \mathbb{I}_{n \times n}) x_k \\
r_2 &= -H (A + (1 - \gamma) \mathbb{I}) x_k - (2 - \gamma) F.
\end{aligned}$$

*Remark 6.* Note that the above QCQP is convex since the matrices  $P_0, P_1, P_2$  are all positive semi-definite.

*Remark 7.* As we have seen here, the exponential CBF formulation results in a convex optimization problem (particularly a QCQP) for a linear system with quadratic Lyapunov and Barrier functions. Moreover, this is also true for nonlinear control affine systems with Linear and/or Quadratic Lyapunov and Barrier functions. This can be solved efficiently using MATLAB's fmincon [1] using packages such as CVX [10].

#### B. Example

We now present simple examples of the discrete-time CBF-CLF controllers for both linear and nonlinear control affine systems.

Consider the linear system with  $A, D, H$  and  $F$  matrices given by,

$$\begin{aligned}
A &= \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}, & D &= [1 \quad 2]^T, \\
H &= [1 \quad 0], & F &= -1.5.
\end{aligned} \tag{18}$$

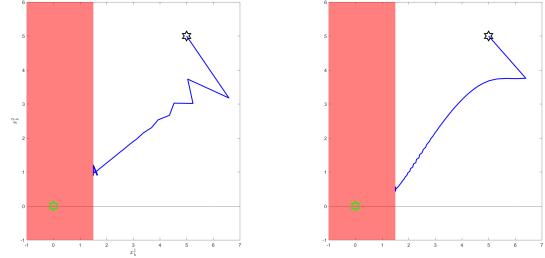
Utilizing the discrete-time CBF-CLF controller in (14), the trajectory of the system always remains within the safe region (see Fig.2a).

Consider now the discrete-time control affine nonlinear system given by,

$$\begin{bmatrix} x_{k+1,1} \\ x_{k+1,2} \end{bmatrix} = \begin{bmatrix} \sin(x_{k,1}) + x_{k,1} + 2x_{k,2} + u_k \\ \sin(x_{k,2}) + 2x_{k,1} + 2x_{k,2} + 2u_k \end{bmatrix}. \tag{19}$$

Note that linearizing the above nonlinear system yields the same linear system as in (18).

On applying the discrete-time CBF-CLF controller to the nonlinear system (19), we see that like in the case of the linear system, the trajectories of the nonlinear system also remain within the safety set (See Fig.2b).



(a) Trajectory of the linear system in (18). (b) Trajectory of the nonlinear system in (19).

Fig. 2: Numerical simulations of the discrete-time CBF-CLF controller applied to linear and nonlinear control affine systems. In both cases, the trajectory (in blue) of the system remains outside the unsafe region (highlighted in red).

#### IV. APPLICATION TO BIPEDAL WALKING

In this section, we use the concept of the discrete-Time control Lyapunov (D-CLF) and barrier functions (D-CBF) to develop a stride-to-stride controller for a 21 degree of freedom bipedal robot model to follow a given path in the task space, while avoiding static and moving obstacles. We begin by presenting a brief overview of the hybrid zero dynamics framework [12, 31], which is used to generate dynamic, stable and periodic gaits and to develop a continuous-time controller for walking along a straight line. Similar to [7, 29], we then present a stride-to-stride controller which can be combined with the continuous-time controller to achieve a desired heading angle in the task space.

Moreover, as pointed out in [21], literature on high-level motion planning for limit-cycle walkers is limited. As we will see, the discrete-time version of CBF-CLF controller allows for the development of a high-level motion planner and provides safety guarantees for the problem of path-following and obstacle avoidance for bipedal robots.

#### A. Hybrid Model of Walking

We consider the particular case of *Flat-footed* walking which consists of alternating phases of a continuous, single-support (swing) phase and an instantaneous, double-support (or impact) phase, with both feet maintained parallel to the ground at all times. we use generalized floating base coordinates [13], where, the configuration of the robot,  $q = (p_b, \phi, q_b) \in \mathcal{Q} = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_b$ , is represented by the Cartesian position,  $p_b$ , and orientation,  $\phi$ , of a body-fixed frame on the robot with respect to the inertial frame and the joint angles,  $q_b \in \mathcal{Q}_b \subset \mathbb{R}^n$ , where  $n = 15$ , is the number of joints. Foot contact with the ground is modeled as a holonomic constraint,  $\eta_c : Q \rightarrow \mathbb{R}$ . These are held constant during each phase i.e.  $\eta_c \equiv \text{constant}$ , and the associated kinematic constraint,  $J(q)\dot{q} = \partial\eta_c/\partial q \cdot \dot{q} = 0$ . In particular, the holonomic constraints here are the stance foot positions and orientations. The switching guard surface  $G$  is defined as the set of states  $x = (q, \dot{q})^T$ , when the swing-foot strikes the

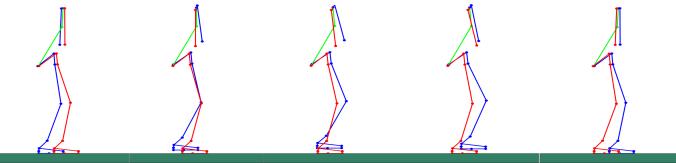


Fig. 3: Snapshots of a single step of walking obtained using the the direct collocation optimization method.

ground, i.e. when the  $z$ - position of the swing foot ( $p_{sw}^z$ ) is equal to zero and the  $z$ - velocity of the swing foot is less than zero:  $G = \{(q, \dot{q}) \in \mathcal{T}\mathcal{Q} \mid p_{sw}^z(q) = 0, \dot{p}_{sw}^z(q) < 0\}$ .

The post impact states,  $(q^+, \dot{q}^+)$  are determined from the impact map  $\Delta(q^-, \dot{q}^-)$ , given the pre-impact states,  $(q^-, \dot{q}^-)$ , assuming a perfectly plastic impact. The hybrid model of the system is then comprised of the continuous time dynamics and the discrete dynamics:

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u, & x \notin G, \\ x^+ = \Delta(x^-), & x \in G. \end{cases} \quad (20)$$

We define the set  $\mathcal{N} := \{(q, \dot{q}) \in \mathcal{T}\mathcal{Q} \mid \eta_c = \eta_c(q^+), J(q)\dot{q} = \mathbf{0}\}$  as the set of states such that the holonomic constraints are satisfied.

### B. Hybrid Zero Dynamics (HZD) Control

a) *Virtual Constraints*: We define a set of outputs (also referred to as virtual constraints),  $y \in \mathbb{R}^m$ , for the control system (20), which consists of velocity regulating and position modulating terms. These are defined as the difference between the actual output,  $y^a(x)$  and desired output,  $y^d(\tau, \alpha)$ ,

$$y := y^a(x) - y^d(\tau, \alpha),$$

where  $\tau(q) \in [0, 1]$  is the gait phasing variable and  $\alpha \in \mathbb{R}^{m \times (b+1)}$  is a set of Bézier Polynomial coefficients of degree  $b$  that parametrize the desired position modulating outputs and is obtained through a constrained nonlinear optimization program [16]. An input-output linearizing controller  $\Gamma_{IO}$  drives  $y \rightarrow 0$  exponentially [31].

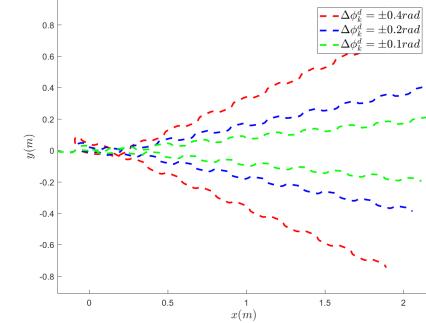
### C. Stride-to-Stride Controller for Turning

The input-output linearizing controller can be combined with a discrete-time controller based on the Poincaré map, similar to [29, 7, 21], to follow a desired heading angle,  $\phi$ . Specifically, we define the Poincaré map  $P : \mathcal{N} \cap \Delta(G) \times \mathcal{B} \mapsto \mathcal{N} \cap \Delta(G) \times \mathcal{B}$  as:

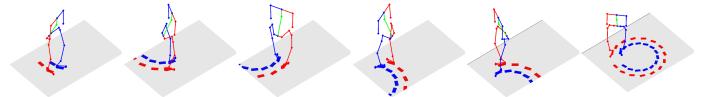
$$x_{k+1} := P(x_k, \beta_k), \quad (21)$$

where  $x_k \in \mathcal{N} \cap \Delta(G) \subset \mathbb{R}^{(2n-n_e+11)}$ , is the reduced set of states on the post impact surface,  $\Delta(G)$ , and which satisfy the holonomic constraints, at the  $k^{\text{th}}$  step.  $\beta_k \in \mathcal{B} \subset \mathbb{R}^m$  is a set of parameters that modifies the outputs,  $y$ , towards the end of a step, as follows,

$$y = y^a(q, \dot{q}) - y^d(\tau, \alpha) - y^b(\tau, \beta_k), \quad (22)$$



(a) Path traversed by the robot for different desired heading angles.



(b) Snapshots of the robot at different time instances for walking on a circular path.

Fig. 4: Path followed by the robot using the event-based controller in (24)

with  $y^b$  defined as,

$$\begin{cases} y^b = 0, & \text{if } \tau = 0, \\ y^b = \beta_k, & \text{if } \tau = 1, \\ \frac{\partial y^b}{\partial \tau} = 0, & \text{if } \tau = \{0, 1\}. \end{cases}$$

The linearized Poincaré map,

$$\delta x_{k+1} = \mathbf{A} \delta x_k + \mathbf{B} \delta \beta_k, \quad (23)$$

can be treated a discrete-time control system, where  $\delta x_k := x_k - x^*$  and  $\delta \beta_k = \beta_k - \beta^*$ , with  $x^*$  a fixed point of (21) and  $\beta^* = \mathbf{0}_{m \times 1}$ . The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are the Jacobians of  $P$  with respect to  $x_k$  and  $\beta_k$  respectively. A feedback controller  $\Gamma_\beta : \mathcal{N} \cap \Delta(G) \rightarrow \mathcal{B}$ ,

$$\delta \beta_k = \Gamma_\beta = -K_\delta \cdot (x_k - x_d(\Delta \phi_k^d)), \quad (24)$$

with  $K_\delta$  obtained using the *DLQR* method such that the eigenvalues of  $(\mathbf{A} - \mathbf{B}K_\delta)$  are within the unit circle. The vector  $x_d(\Delta \phi_k^d)$  is the fixed-point  $x^*$  with the heading angle replaced by the desired change in heading angle  $\Delta \phi_k^d$ . Fig. 4a shows the path traversed by the robot for different desired heading angles while Fig. 4b shows snapshots of the robot for walking on a circular path, both using the event-based controller (24).

### D. Path Following

Consider the desired path to be represented by  $p_y^d = \zeta(p_x)$ , where  $p_y^d$  is the desired position of the robot on the ground plane along the  $y$ -axis,  $p_x$  is the current position of the robot on the ground plane along the  $x$ -axis, and  $\zeta : \mathbb{R} \rightarrow \mathbb{R}$  is a smooth function.

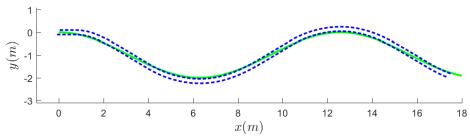


Fig. 5: Foot-step positions of the robot (in blue) for a sinusoidal path (green curve) using the discrete-time CLF controller.

*Remark 8.* A desired path for the robot's position can be computed offline from existing path planning algorithms such as *RRTs*,  $A^*$  search, or any other method.

From (23) and (24), another system can be constructed,

$$p_{k+1}^{st}(x_{k+1}) =: \tilde{P}(p_k^{st}(x_k), \Delta\phi_k^d), \quad (25)$$

where  $p_k^{st} := (p_{x,k}^{st}, p_{y,k}^{st})^T$  is the position of the stance foot at step  $k$ . Again, this can be linearized as,

$$p_{k+1}^{st} = \tilde{\mathbf{A}} p_k^{st} + \tilde{\mathbf{B}} \Delta\phi_k^d, \quad (26)$$

where  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$  are the Jacobian matrices of  $\tilde{P}$  with respect to  $p_k^{st}$  and  $\Delta\phi_k^d$  respectively.

The error between the desired and the vertical position of the stance foot is obtained as,

$$e_k^y := p_{y,k}^{st} - \zeta(p_{x,k}^{st}). \quad (27)$$

A discrete-time CLF controller can then be developed to follow the given path as:

$$\begin{aligned} \Delta\phi_k^* &= \underset{\Delta\phi_k \in \mathbb{R}}{\operatorname{argmin}} \Delta\phi_k^T \Delta\phi_k \\ \text{s.t. } & \Delta V(e_k^y, \Delta\phi_k) + c_3 \|e_k^y\|^2 \leq 0, \\ & -\Delta\phi_k^{\max} \leq \Delta\phi_k \leq \Delta\phi_k^{\max}. \end{aligned} \quad (28)$$

where  $V$  is a control Lyapunov function for the discrete-time control system in (26).

Fig.5 shows the foot-step locations of the robot for a sinusoidal desired path.

*Remark 9.* Note that the above controller saturates the change in the heading angle between  $\pm\Delta\phi_k^{\max}$  to avoid foot slipping and motor torque saturation.

#### E. Obstacle Avoidance

The D-CLF path-following controller in (28) can be augmented with a discrete-time control barrier function (D-CBF) to avoid obstacles in the task space, while staying as close as possible to the desired path. Particularly, we use the exponential barrier function formulation presented in Section III, since this leads to a convex optimization problem (QCQP) for linear barrier and quadratic Lyapunov functions.

In the following examples, we consider the obstacles to be ellipses with semi-major axis  $a$ , semi-minor axis  $b$ , centered at  $(x_0, y_0)$  and the axes of the ellipse aligned with the horizontal

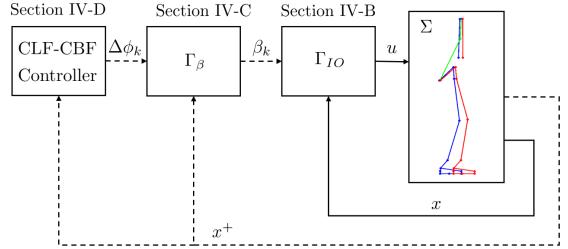


Fig. 6: Overview of the proposed controller. Dashed lines represent signals in discrete-time.

and vertical axes of the inertial frame. The control barrier function is then formulated as:

$$B_k = \frac{(p_{x,k}^{st} - x_0)^2}{a^2} + \frac{(p_{y,k}^{st} - y_0)^2}{b^2} - 1. \quad (29)$$

Note that  $B_k$  is quadratic in  $\Delta\phi_k$ .

*Remark 10.* Although we only consider ellipses here, the obstacles, in general, maybe any convex shape. Shapes that are described by polynomials of degree higher than two may lead the optimization problem in (30) to be non-convex.

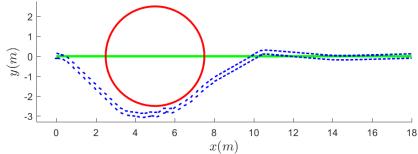
The combined D-CBF-D-CLF controller can be formulated as a QCQP:

$$\begin{aligned} \Delta\phi_k^* &= \underset{U_k}{\operatorname{argmin}} \Delta\phi_k^T \Delta\phi_k + p \cdot d^2 \\ U_k &= \begin{bmatrix} \Delta\phi_k \\ d \end{bmatrix} \in \mathbb{R}^2 \\ \text{s.t. } & \Delta V(e_k^y, \Delta\phi_k) + c_3 \|e_k^y\|^2 \leq d, \\ & \Delta B(p_k^{st}, \Delta\phi_k) + \gamma B(p_k^{st}) \geq 0, \\ & \Delta\phi_k^{\min} \leq \Delta\phi_k \leq \Delta\phi_k^{\max}. \end{aligned} \quad (30)$$

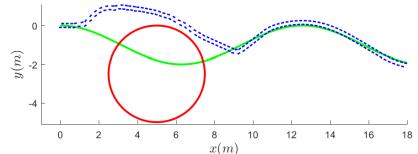
*Remark 11.* In this paper, we look at the discrete-time version of the CBF-CLF based control developed in [2] as the Poincaré map based control problem for path-following and obstacle avoidance for a bipedal robot is inherently discrete-time due to the stride-to-stride event-based control action. Moreover, continuous-time controllers presented in [23, 22] only allow for within-stride control of the foot position and cannot be applied directly to this problem.

*Remark 12.* The discrete-time CBF controller behaves like a step-to-step planner and gives a *minimum* change in heading angle  $\Delta\phi_k^*$  at the beginning of each walking step that is required to keep the robot as close as possible to the desired path and simultaneously avoid obstacles. The required change in heading angle  $\Delta\phi_k^*$  is then tracked by the event-based controller in (24). Fig. 6 shows an overview of the complete control system.

*Remark 13.* We note again that the  $d$  term in (30) is required to relax the discrete-time CLF condition (4) so that the Lyapunov function  $V$  can grow when the the CLF and CBF conditions are conflicting (when the obstacles are on the desired path, our primary goal is to avoid the obstacles rather than to follow the desired path).



(a) Straight-line desired path.



(b) Sinusoidal desired path.

Fig. 7: Foot-step positions of the robot (in blue) for different desired trajectories (green curve) while avoiding static obstacles (red circle) using the discrete-time CBF-CLF controller.

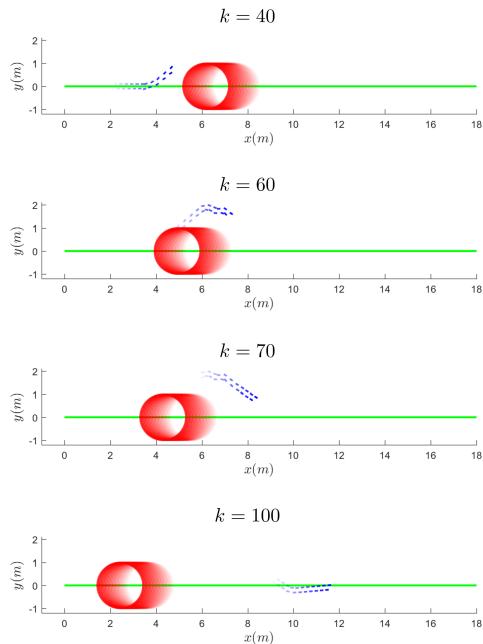


Fig. 8: Snapshots of the robot's stance foot (in blue) at different steps,  $k$ . Red circle represents a moving obstacle, green line represents the desired trajectory.

Fig. 7 shows the results of our proposed controller for following a desired path, while avoiding static obstacles. Fig. 8 shows snapshots of the robot's foot position for the case with moving obstacles.

*Remark 14.* For the case of moving obstacles, we assume that the robot does not strike any obstacles during the swing phase. This is a reasonable assumption since the step times are in the order of 1s.

*Remark 15.* For the case of avoiding moving obstacles, traditional path-planning algorithms such as RRTs could be potentially inefficient.

*Remark 16.* The discrete time CBF-CLF controller (30) can be concatenated with a continuous-time CBF-CLF controller [9, 22, 26] to traverse over more complex terrains such as stepping stones while avoiding obstacles.

## V. CONCLUSION

In this paper, we took tools recently developed for safety-critical applications of continuous-time systems and showed mathematically how they can be extended to discrete-time systems. For discrete-time systems, however, using the definition of barrier functions in (8) and (9) an additional complexity arises in that the resulting optimization problem to solve for the optimal control input is not necessarily convex. We then used the concept of exponential control barrier functions and showed that for nonlinear control-affine discrete-time systems, the CLF and CBF conditions are quadratic for quadratic Lyapunov and linear barrier functions and the resulting optimization problem is a convex Quadratically Constrained Quadratic Program. Using this concept of CLFs and CBFs for discrete-time systems, we then developed a stride-to-stride controller for path-following and obstacle avoidance in the task space for a high-dimensional bipedal robot.

A few shortcomings of our method are detailed next. In our control design we make certain simplifications and assumptions, which include linearized models for the Poincaré map (23) and foot-step placement (26), which results in a slightly inaccurate estimate of footstep position. This can be addressed either by considering the nonlinear discrete-time systems at the expense of not having a QCQP, or by formulating enlarged barriers that are equivalent to the amount of uncertainty in estimating the footstep positions. We also assumed that we have knowledge about the full state of the robot (such as its position in the inertial frame) and the environment (such as position of obstacles with respect to the robot). In future work, we intend to address this by integrating inertial and vision sensors in our control design to estimate the location of the robot and obstacles. Another important assumption we make is the feasibility of the optimization problem in (14) to guarantee invariance of the safe set.

As part of future work, we seek to enable bipedal robots to walk over uneven and discrete terrain while avoiding obstacles. We hope to facilitate this by integrating the discrete-time CBF controller presented here with the continuous-time controllers presented in [22, 25, 24, 26].

## REFERENCES

- [1] “Documentation. Linear or Quadratic Objective with Quadratic constraints - MATLAB & Simulink.” [Online]. Available: <https://www.mathworks.com/help/optim/ug/linear-or-quadratic-problem-with-quadratic-constraints.html>
- [2] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 6271–6278.

- [3] L. Baudouin, N. Perrin, T. Moulard, F. Lamiriaux, O. Stasse, and E. Yoshida, “Real-time replanning using 3D environment for humanoid robot,” in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 584–589.
- [4] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.
- [5] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, “Planning biped navigation strategies in complex environments,” in *IEEE Int. Conf. Hum. Rob., Munich, Germany*, 2003.
- [6] J. Chestnutt, K. Nishiwaki, J. Kuffner, and S. Kagami, “An adaptive action model for legged navigation planning,” in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007, pp. 196–202.
- [7] C. Chevallereau, J. W. Grizzle, and C.-L. Shih, “Steering of a 3D bipedal robot with an underactuated ankle,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1242–1247.
- [8] R. Deits and R. Tedrake, “Footstep planning on uneven terrain with mixed-integer convex optimization,” in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 279–286.
- [9] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, “Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, Apr. 2015.
- [10] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [11] R. D. Gregg, A. K. Tilton, S. Candido, T. Bretl, and M. W. Spong, “Control and planning of 3-d dynamic walking with asymptotically stable gait primitives,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1415–1423, 2012.
- [12] J. W. Grizzle, G. Abba, and F. Plesstan, “Asymptotically stable walking for biped robots: Analysis via systems with impulse effects,” *IEEE Transactions on automatic control*, vol. 46, no. 1, pp. 51–64, 2001.
- [13] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, “3D bipedal robotic walking: models, feedback control, and open problems,” *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 505–532, 2010.
- [14] J. Grizzle and J.-M. Kang, “Discrete-time control design with positive semi-definite lyapunov functions,” *Systems & Control Letters*, vol. 43, no. 4, pp. 287–292, 2001.
- [15] K. Harada, E. Yoshida, and K. Yokoi, *Motion planning for humanoid robots*. Springer Science & Business Media, 2010.
- [16] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, “3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.
- [17] R. Kalman and J. Bertram, “Control system analysis and design via the second method of lyapunov: Iidiscrete-time systems.” *ASME. J. Basic Eng.*, vol. 1, no. 82, pp. 394–400, 1960.
- [18] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, “Online footstep planning for humanoid robots,” in *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 932–937.
- [19] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, “Footstep planning among obstacles for biped robots,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 500–505.
- [20] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, “Vision-guided humanoid footstep planning for dynamic environments,” in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*. IEEE, 2005, pp. 13–18.
- [21] M. S. Motahar, S. Veer, and I. Poulikakos, “Composing limit cycles for motion planning of 3D bipedal walkers,” in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 6368–6374.
- [22] Q. Nguyen, X. Da, J. W. Grizzle, and K. Sreenath, “Dynamic walking on stepping stones with gait library and control barrier,” in *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2016.
- [23] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, “3D dynamic walking on stepping stones with control barrier functions,” in *IEEE International Conference on Decision and Control (CDC)*, Las Vegas, NV, Dec. 2016, pp. 827–834.
- [24] Q. Nguyen and K. Sreenath, “Safety-critical control for dynamical bipedal walking with precise footstep placement,” in *IFAC Analysis and Design of Hybrid Systems (ADHS)*, Atlanta, GA, Oct. 2015.
- [25] ——, “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints,” in *American Control Conference (ACC)*, Boston, MA, Jul. 2016, pp. 322–328.
- [26] ——, “Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion,” in *American Control Conference (ACC)*, Boston, MA, Jul. 2016, pp. 4807–4813.
- [27] ——, “Optimal robust safety-critical control for dynamic robotics,” *International Journal of Robotics Research (IJRR)*, in review, 2016.
- [28] M. Z. Romdlony and B. Jayawardhana, “Uniting control lyapunov and control barrier functions,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2293–2298.
- [29] C.-L. Shih, J. Grizzle, and C. Chevallereau, “From stable walking to steering of a 3D bipedal robot with passive point feet,” *Robotica*, vol. 30, no. 07, pp. 1119–1130, 2012.
- [30] L. Wang, A. Ames, and M. Egerstedt, “Safety barrier

- certificates for heterogeneous multi-robot systems,” in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 5213–5218.
- [31] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007, vol. 28.
- [32] G. Wu and K. Sreenath, “Safety-critical control of a 3D quadrotor with range-limited sensing,” in *ASME Dynamics Systems and Control Conference (DSCC)*, Minneapolis, MN, Oct. 2016.
- [33] ——, “Safety-critical control of a planar quadrotor,” in *American Control Conference (ACC)*, Boston, MA, Jul. 2016, pp. 2252–2258.
- [34] ——, “Safety-critical geometric control for systems on manifolds subject to time-varying constraints,” *IEEE Transactions on Automatic Control (TAC)*, in review, 2016.