2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Multi-Layered Safety for Legged Robots via Control Barrier Functions and Model Predictive Control

Ruben Grandia, Andrew J. Taylor, Aaron D. Ames, Marco Hutter

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561510

Abstract— **The problem of dynamic locomotion over rough terrain requires both accurate foot placement together with an emphasis on dynamic stability. Existing approaches to this problem prioritize immediate safe foot placement over longer term dynamic stability considerations, or relegate the coordination of foot placement and dynamic stability to heuristic methods. We propose a multi-layered locomotion framework that unifies Control Barrier Functions (CBFs) with Model Predictive Control (MPC) to simultaneously achieve safe foot placement and dynamic stability. Our approach incorporates CBF based safety constraints both in a low frequency kino-dynamic MPC formulation and a high frequency inverse dynamics tracking controller. This ensures that safety-critical execution is considered when optimizing locomotion over a longer horizon. We validate the proposed method in a 3D stepping-stone scenario in simulation and experimentally on the ANYmal quadruped platform.**

I. Introduction

A key motivation behind the development of legged robots is their ability to overcome complex terrain. Because legged locomotion only requires discrete footholds, obstacle such as steps, gaps, and stairs can be traversed, making legged robots a compelling alternative to wheeled systems. When a statically stable motion pattern is considered, several mature strategies for rough terrain locomotion have been proposed and successfully demonstrated on hardware for bipedal [2], quadrupedal [3], [4], and hexapedal [5] robots. However, inspired by the fast and dynamic motions seen in nature, the use of dynamic gaits—a gait where individual contact phases are statically unstable—is still an active area of research.

The challenge in dynamic locomotion lies in the fact that foothold locations are not only constrained by the terrain, but also affect the dynamic stability of the resulting contact configuration. Additionally, as the speed of the motions increases, the inertial and nonlinear effects described by the full rigid body dynamics of the system become more relevant. There is therefore a need for methods that can guarantee a safe foot placement while simultaneously considering the future impact on the dynamic stability of the system. A classical locomotion challenge that demands safe foot placement and dynamic stabilization is the “stepping-stones” scenario, see Figure 1, where viable foothold locations are discontinuous and sparsely available. We propose to combine the safety guarantees endowed by Control Barrier Functions (CBFs) with the longer horizon considered in Model Predictive Control (MPC) to guarantee safe foot placement while achieving dynamic locomotion and high tracking performance.

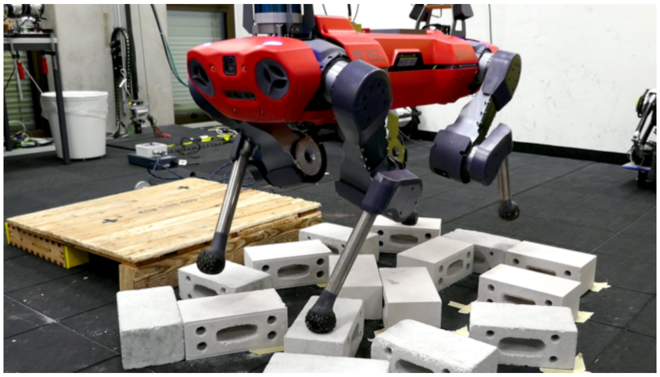

Fig. 1. ANYmal [6] performing a trotting gait on stepping-stones.

A. Related work

Control Barrier Functions [7] are a tool for synthesizing controllers that ensure safety of nonlinear systems [8], [9]. Moreover, CBFs have been used in the stepping-stones problem via a quadratic programming (QP) based tracking controller [10], [11]. An offline optimized walking trajectory, or a library thereof [12], is tracked and locally modified to satisfy CBF safety constraints. While promising in simulation, we are not aware of the successful transfer of a CBF based stepping controller to hardware, despite extensions that add robustness [13], or a learning based model error correction [14]. Indeed, in [15], the stepping-stones problem is demonstrated experimentally by increasing the look-ahead horizon of the gait library and through subsequent gait interpolation rather than a CBF based method. We hypothesize that it is exactly this reasoning over a longer horizon that is missing with the CBF-QP control formulation.

In contrast, Model Predictive Control has become a central method for the online synthesis and control of dynamic systems over a given time horizon [16]. In the context of the stepping-stones problem, a distinction can be made between MPC based approaches where the footholds locations are determined separately from the torso motion optimization [17], [18], [19], and MPC based approaches where the foothold location and torso motions are jointly optimized.

R. Grandia and M. Hutter are supported via the European Union’s Horizon 2020 research and innovation programme under grant agreement No 780883 and by the Swiss National Science Foundation (SNSF) as part of project No.188596. A. Taylor, and A. Ames are supported via DARPA award HR00111890035, and NSF awards 1923239 and 1924526.

R. Grandia and M. Hutter are with the Department of Mechanical and Process Engineering, ETH Zürich, 8092 Zürich, Switzerland {rgrandia,mahutter}@ethz.ch. A. Taylor and A. Ames are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA {ajtaylor,ames}@caltech.edu.

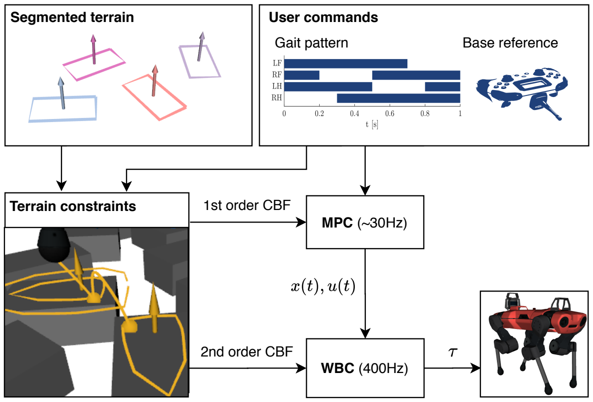

Fig. 2. Overview of the proposed multi-layed control setup showing both the MPC and WBC layer receiving terrain CBF constraints.

The benefit of jointly optimizing torso and leg motions has been demonstrated in the field of trajectory optimization [20], [21]. Following this idea, real-time capable methods have been proposed with the specification of leg motions made at the position [22], velocity [23], or acceleration level [24]. One challenge of this approach is its computational costs, which can be resolved by coupling a low-frequency MPC controller with a high-frequency tracking controller [25].

B. Contribution

In this work, we build upon a kino-dynamic MPC formulation [23] where joint velocities and contact forces are decision variables in a low frequency MPC controller. This allows direct integration of CBF safety constraints into the MPC formulation similar to [26]. By jointly optimizing torso and leg motions our method avoids the heuristic coordination that is needed when foot placement and torso motion are delegated to separate controllers. A higher rate tracking controller is implemented that fuses inverse dynamics with the CBF safety constraints to offer guarantees of safety with the whole-body dynamics in consideration. In the context of collision avoidance, CBFs can be thought of (and have shown to be) a generalization of artificial potential fields used in inverse dynamics methods [27][28]. Finally, we note that the combination of discrete time CBFs with MPC has been considered in [29], but it did not consider a multi-layered approach nor provided experimental results.

The main contributions of this work are two-fold. First, we propose a multi-layered control approach that combines CBFs with MPC (see Figure 2). This framework allows CBF safety constraints on the position coordinates of robotic systems to be incorporated in a low frequency MPC controller determining desired velocities as well as in a high frequency tracking controller that incorporates the dynamics of the system. Compared to standard CBF approaches, this adds a horizon when determining safe control inputs. Compared to MPC approaches, the safety critical constraint is enforced at a higher rate, and incorporates a higher fidelity whole-body dynamics model. The second contribution is, to the best of the author's knowledge, the first successful experimental demonstration of CBFs, not only as an approach to the stepping-stones problem, but on a legged robot.

II. Background

This section provides a review of Control Barrier Functions (CBFs) and Nonlinear Model Predictive Control.

Consider the nonlinear control affine system given by:

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

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$. $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ 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 $\mathbf{k} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^m$, the closed-loop system dynamics are:

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

The assumption on local Lipschitz continuity of $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{k}$ implies that $\mathbf{f}_{\text{cl}}$ 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 $I(\mathbf{x}_0)$ [30].

A. Control Barrier Functions

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. In particular, consider a time-varying set $\mathcal{C}_t \subset \mathbb{R}^n$ defined as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$, yielding:

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

We refer to $\mathcal{C}_t$ as the *safe set*. This construction motivates the following definitions of forward invariant and safety:

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

Certifying the safety of the closed-loop system (2) with respect to a set $\mathcal{C}_t$ may be impossible if the controller $\mathbf{k}$ was not chosen to enforce the safety of $\mathcal{C}_t$. Control Barrier Functions can serve as a synthesis tool for attaining the forward invariance, and thus the safety of a set. Before defining CBFs, we note a continuous function $\alpha : (-\infty, \infty) \to \mathbb{R}$, is said to belong to *extended class* $\mathcal{K}_\infty$ *(*$\alpha \in \mathcal{K}_{\infty,e}$*)* if α is strictly monotonically increasing, $\alpha(0) = 0$, and if $\lim\limits_{r\to\infty} \alpha(r) = \infty$, and $\lim\limits_{r\to-\infty} \alpha(r) = -\infty$.

Definition 2 (*Control Barrier Function (CBF)*, [31])**.** Let $\mathcal{C}_t \subset \mathbb{R}^n$ be the time-varying 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ with 0 a regular value. The function h is a time-varying *Control Barrier Function* (CBF) for (1) on $\mathcal{C}_t$ if there exists $\alpha \in \mathcal{K}_{\infty,e}$ such that for all $\mathbf{x} \in \mathbb{R}^n$ and $t \in \mathbb{R}_+$:

$$\begin{aligned} \sup_{\mathbf{u}\in\mathbb{R}^m} \dot{h}(\mathbf{x}, t, \mathbf{u}) \triangleq & \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t)\left(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}\right) \\ & + \frac{\partial h}{\partial t}(\mathbf{x}, t) \geq -\alpha(h(\mathbf{x}, t)). \end{aligned} \tag{4}$$

Controllers that take inputs satisfying (4) ensure the safety of the closed-loop system (2) [7].

Given a nominal (but not necessarily safe) locally Lipschitz continuous controller $\mathbf{k}_d : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^m$, a possible controller taking values satisfying (4) is the safety-critical CBF-QP:

$$\begin{aligned} \mathbf{k}(\mathbf{x}, t) = \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \mathbf{k}_d(\mathbf{x}, t)\|_2^2 \\ \text{s.t.} \quad & \dot{h}(\mathbf{x}, t, \mathbf{u}) \geq -\alpha(h(\mathbf{x}, t)). \end{aligned} \tag{CBF-QP}$$

B. Nonlinear Model Predictive Control

We consider the following nonlinear optimal control problem with cost functional

$$\min_{u(\cdot)} \Phi(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}(t), \mathbf{u}(t), t)\ \mathrm{d}t, \tag{5}$$

where $\mathbf{x}(t)$ is the state, $\mathbf{u}(t)$ is the input at time t, $L(\cdot)$ is an intermediate cost, and $\Phi(\cdot)$ is the cost at the terminal state $\mathbf{x}(T)$. The goal is to find a continuous control signal $\mathbf{u} : I(\mathbf{x}_0) \to \mathbb{R}^m$ that minimizes this cost subject to the system dynamics, initial condition, and general constraints:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t), \qquad \mathbf{x}(0) = \mathbf{x}_0, \tag{6}$$

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, t) = 0, \qquad \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \geq 0. \tag{7}$$

Various methods exist to solve this problem [16], and a detailed discussion is beyond the scope of this paper. In this work we use the Sequential Linear Quadratic (SLQ) method, which is a Differential Dynamic Programming (DDP) based algorithm for continuous-time systems. In particular, the method in [25] is being used which extends the (SLQ) formulation of [32] for use with inequality constraints.

III. Multi-Layered Control Formulation

In this section we present a multi-layered control formulation that unifies CBFs with MPC to achieve safety and longer horizon optimality for a general robotic system. Consider a robotic system with generalized coordinates $\mathbf{q} \in \mathbb{R}^d$ and coordinate rates $\dot{\mathbf{q}} \in \mathbb{R}^d$ with dynamics given by:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\boldsymbol{\tau}, \tag{8}$$

with inertia matrix $\mathbf{D}$, centrifugal and Coriolis terms $\mathbf{C}$, gravitational forces $\mathbf{G}$, actuation matrix $\mathbf{B}$, and torques $\boldsymbol{\tau} \in \mathbb{R}^m$. Consider a continuously differentiable function $h : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}$ that determines a time-varying safe set for the position coordinates of the robot, with a time derivative given by:

$$\dot{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{\partial h}{\partial \mathbf{q}}(\mathbf{q}, t)\dot{\mathbf{q}} + \frac{\partial h}{\partial t}(\mathbf{q}, t).$$

The torques $\boldsymbol{\tau}$ do not appear in this time derivative, making it impossible to choose inputs that ensure the barrier constraint:

$$\dot{h}(\mathbf{q}, \dot{\mathbf{q}}, t) \geq -\alpha_1(h(\mathbf{q}, t)), \tag{9}$$

is met for some $\alpha_1 \in \mathcal{K}_{\infty,e}$. This challenge is often resolved through the notion of *exponential* CBFs [33], in which an auxiliary function $h_e : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}$ is defined as:

$$h_e(\mathbf{q}, \dot{\mathbf{q}}, t) = \dot{h}(\mathbf{q}, \dot{\mathbf{q}}, t) + \alpha_1(h(\mathbf{q}, t)), \tag{10}$$

$$\dot{h}_e(\mathbf{q}, \dot{\mathbf{q}}, t, \boldsymbol{\tau}) = \frac{\partial h_e}{\partial \mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \frac{\partial h_e}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} + \frac{\partial h_e}{\partial t}(\mathbf{q}, \dot{\mathbf{q}}, t).$$

As $\ddot{\mathbf{q}}$ appears in affine relation to $\boldsymbol{\tau}$ in (8), h_e can serve as a CBF for the set $\mathcal{C}_{t,e} \triangleq \left\{(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2d} : h_e(\mathbf{q}, \dot{\mathbf{q}}, t) \geq 0\right\}$ by enforcing:

$$\dot{h}_e(\mathbf{q}, \dot{\mathbf{q}}, t, \boldsymbol{\tau}) \geq -\alpha_2(h_e(\mathbf{q}, \dot{\mathbf{q}}, t)), \tag{11}$$

for some $\alpha_2 \in \mathcal{K}_{\infty,e}$. Enforcing the forward invariance of this set implies the desired safety constraint (9) is met, implying the forward invariance of the set $\mathcal{C}_t \cap \mathcal{C}_{t,e}$. Thus the constraint on the position coordinates of the robot are met.

Typical approaches using exponential CBFs only enforce the final constraint (11), often in a CBF-QP controller [34]. In practice, when the desired controller $\mathbf{k}_d$ is synthesized without considering safety, this can lead to aggressive behavior when the system approaches the boundary of the safe set. Using MPC in a multi-layered setup allows the safety constraint to be incorporated into the specification of $\mathbf{k}_d$. When the MPC directly operates on the full state and input of (8), the safety constraint in (11) is readily incorporated, as was done in [29]. In contrast, we consider a MPC controller that operates on a reduced order model in which case the barrier constraint in (9) is added to the MPC problem instead. For simplicity of exposition, we present here an MPC layer that operates on a purely kinematic model of the system.

Given a current estimate of the state $(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})$ at time $\hat{t}$, a kinematic MPC solves the following optimization problem:

Low-Frequency Safe Kinematic MPC:

$$\begin{aligned} \min_{\mathbf{q}^d(t), \dot{\mathbf{q}}^d(t)} \quad & \Phi(\mathbf{q}^d(T)) + \int_0^T L(\mathbf{q}^d(t), \dot{\mathbf{q}}^d(t), t)\ \mathrm{d}t, \\ \text{s.t} \quad & \mathbf{q}^d(0) = \hat{\mathbf{q}}, \qquad \frac{\partial \mathbf{q}^d}{\partial t} = \dot{\mathbf{q}}^d, \\ & \dot{h}(\mathbf{q}^d, \dot{\mathbf{q}}^d, t) + \alpha_1(h(\mathbf{q}^d, t)) \geq 0, \end{aligned}$$

where $\mathbf{q}^d(t)$ and $\dot{\mathbf{q}}^d(t)$ are trajectories of generalized coordinates and velocities, forming the safe desired trajectory for the tracking controller. A desired acceleration is obtained through a combination of tracking terms and a forward difference of the desired velocities:

$$\ddot{\mathbf{q}}^d = \frac{\dot{\mathbf{q}}^d(\hat{t} + \delta t) - \dot{\mathbf{q}}^d(\hat{t})}{\delta t} + \mathbf{D}(\dot{\mathbf{q}}^d(\hat{t}) - \dot{\hat{\mathbf{q}}}) + \mathbf{P}(\mathbf{q}^d(\hat{t}) - \hat{\mathbf{q}}).$$

Drawing inspiration from the inverse dynamics approach in [35], the high-frequency controller is given by:

High-Frequency ID-CBF-QP:

$$\begin{aligned} k(\mathbf{q}, \dot{\mathbf{q}}, t) = \underset{\boldsymbol{\tau}, \ddot{\mathbf{q}}}{\operatorname{argmin}} \quad & \frac{1}{2}\|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^d\|_2^2 \\ \text{s.t.} \quad & \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\boldsymbol{\tau}, \\ & \dot{h}_e(\mathbf{q}, \dot{\mathbf{q}}, t, \boldsymbol{\tau}) \geq -\alpha_2(h_e(\mathbf{q}, \dot{\mathbf{q}}, t)). \end{aligned}$$

This controller seeks to track the desired acceleration determined by the low-frequency MPC controller while ensuring that the full dynamics are incorporated into the determination of safe inputs according to (11).

IV. ANYmal Implementation

In this Section we provide an overview of how the multi-layer control formulation discussed in Section III is applied to the ANYmal quadrupedal robotic platform. An overview of the control structure is provided in Figure 2.

A. MPC System Model

We apply our approach to the kino-dynamic model of a quadruped robot, which describes the dynamics of a single free-floating body along with the kinematics for each leg. The state $\mathbf{x} \in \mathbb{R}^{24}$ and input $\mathbf{u} \in \mathbb{R}^{24}$ are defined as:

$$\mathbf{x} = \begin{bmatrix}\boldsymbol{\theta}^T, & \mathbf{p}^T, & \boldsymbol{\omega}^T, & \mathbf{v}^T, & \mathbf{q}^T\end{bmatrix}^T, \quad \mathbf{u} = \begin{bmatrix}\boldsymbol{\lambda}_B^T, & \dot{\mathbf{q}}^{d^T}\end{bmatrix}^T,$$

where $\boldsymbol{\theta} \in \mathbb{R}^3$ is the orientation of the base in Euler angles, $\mathbf{p} \in \mathbb{R}^3$ is the position of the center of mass in the world frame $\mathcal{F}_W$, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular rate of the base, $\mathbf{v} \in \mathbb{R}^3$ is the linear velocity of the center of mass in the body frame $\mathcal{F}_B$, and $\mathbf{q} \in \mathbb{R}^{12}$ is the joint positions. The joint positions for leg i are given by $\mathbf{q}_i \in \mathbb{R}^3$. The inputs of the model are end-effector contact forces $\boldsymbol{\lambda}_B \in \mathbb{R}^{12}$ in the body frame and desired joint velocities $\dot{\mathbf{q}}^d \in \mathbb{R}^{12}$ with equations of motion:

$$\begin{aligned}
&\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta})\boldsymbol{\omega}, \qquad \dot{\mathbf{p}} = {}_W\mathbf{R}_B(\boldsymbol{\theta})\,\mathbf{v},\\
&\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(-\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \textstyle\sum_{i=1}^4 \mathbf{r}_{B_i}(\mathbf{q}_i) \times \boldsymbol{\lambda}_{B_i}\right),\\
&\dot{\mathbf{v}} = \mathbf{g}(\boldsymbol{\theta}) + \tfrac{1}{m}\textstyle\sum_{i=1}^4 \boldsymbol{\lambda}_{B_i}, \qquad \dot{\mathbf{q}} = \dot{\mathbf{q}}^d,
\end{aligned}$$

where ${}_W\mathbf{R}_B : \mathbb{R}^3 \to SO(3)$ is the rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$ and $\mathbf{T} : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ transforms angular velocities to the Euler angles derivatives. Model parameters include the gravitational acceleration in the body frame $\mathbf{g} : \mathbb{R}^3 \to \mathbb{R}^3$, the total mass $m \in \mathbb{R}_+$, and the moment of inertia $\mathbf{I} \in \mathbb{R}^{3\times3}$. The moment of inertia is assumed constant and taken at the upright state of the robot. We denote $\mathbf{r}_{B_i} : \mathbb{R}^3 \to \mathbb{R}^3$ as the position of foot i relative to the center of the mass in the body frame.

B. MPC Constraints

In this subsection we list the constraints that are included in the low-frequency kino-dynamic MPC controller.

1) Mode Constraints: The mode constraints capture the different modes of each leg at any given point in time. We assume that the mode sequence is a predefined function of time. The resulting mode-dependent constraints are

$$\begin{cases} \mathbf{v}_{W_i}(\mathbf{x},\mathbf{u}) = \mathbf{0}, & \text{if } i \text{ is a stance leg,}\\ \mathbf{n}^T\mathbf{v}_{W_i}(\mathbf{x},\mathbf{u}) = c(t), \quad \boldsymbol{\lambda}_{B_i} = \mathbf{0}, & \text{if } i \text{ is a swing leg,}\end{cases}$$

where $\mathbf{v}_{W\,i}$ is the end-effector velocity in world frame. These constraints ensure that stance legs remain on the ground and a swing legs follow a predefined curve $c : \mathbb{R}_+ \to \mathbb{R}$ in the direction of the local surface normal $\mathbf{n} \in \mathbb{R}^3$ to avoid foot scuffing.

2) Friction Cone Constraints: The end-effector forces are constrained to lie in the friction cone, $\boldsymbol{\lambda}_{W_i} \in \mathcal{Q}(\mathbf{n}, \mu_c)$, defined by the surface normal $\mathbf{n}$ and friction coefficient $\mu_c = 0.7$. After resolving the contact forces in the local frame of the surface, given by $\mathbf{F} = [F_x, F_y, F_z]$, a second-order cone constraint is specified, $h_{cone} = \mu_c F_z - (F_x^2 + F_y^2)^{\frac{1}{2}} \geq 0$.

3) State-Only Foot Placement Constraints: When foot-placement is formulated as a state-only constraint (rather than encoded in a CBF), it is specified as the following inequality constraint on stance feet:

$$\mathbf{h}_i^t(\mathbf{x}) = \mathbf{A}_i \cdot \mathbf{p}_{W_i}(\mathbf{x}) + \mathbf{b}_i \geq \mathbf{0}, \tag{12}$$

where $\mathbf{A}_i \in \mathbb{R}^{p_i\times3}$, $\mathbf{b}_i \in \mathbb{R}^{p_i}$, and $\mathbf{p}_{W_i} : \mathbb{R}^{24} \to \mathbb{R}^3$ is the position of foot i in the world frame. The matrix $\mathbf{A}_i$ and $\mathbf{b}_i$ project the position of foot i on to the target terrain and form a set of half-space constraints to ensure the foot lands within a desired target region. Instead of constraining the stance feet, a similar constraint can be placed on the swing feet with a constraint set that shrinks in time and converges to the desired foot placement region:

$$\mathbf{h}_i^w(\mathbf{x}, t) = \mathbf{A}_i \cdot \mathbf{p}_{W_i}(\mathbf{x}) + \mathbf{b}_i + s(t)\cdot\mathbf{1} \geq \mathbf{0}, \tag{13}$$

where $s : \mathbb{R}_+ \to \mathbb{R}_+$ converges to 0 as the t approaches the duration of the swing phase.

4) Barrier Foot Placement Constraints: When posed as a CBF constraint as in the proposed low-frequency Safe Kinematic MPC controller, the foot placement constraints are specified with constant $\gamma \in \mathbb{R}_{++}$ as:

$$\mathbf{h}_{e,i}^w(\mathbf{x}, \dot{\mathbf{q}}, t) = \dot{\mathbf{h}}_i^w(\mathbf{x}, t, \mathbf{u}) + \gamma\mathbf{h}_i^w(\mathbf{x}, t) \geq \mathbf{0}. \tag{14}$$

C. Whole-Body Tracking Control

The control signal $\mathbf{u}$ determined by the low-frequency MPC layer consists of contact forces and desired joint velocities. A high-frequency hierarchical inverse dynamics controller is used to convert the optimized MPC trajectory into torque commands [36]. This whole body control (WBC) approach considers the full nonlinear rigid body dynamics of the system. At each priority, a QP is solved in the null space of all higher priority tasks. Each task is a equality or inequality constraint that is affine in the generalized accelerations, torques, and contact forces. The CBF constraints, which are by design affine in the control torques, are therefore readily integrated into this framework. The full list of tasks is given in Table I.

As described in Section III, the following CBF constraint incorporating the dynamics can be included in the whole-body controller:

$$\dot{\mathbf{h}}_{e,i}^w(\mathbf{x}, \dot{\mathbf{q}}, t, \boldsymbol{\tau}) + \xi\mathbf{h}_{e,i}^w(\mathbf{x}, \dot{\mathbf{q}}, t) \geq \mathbf{0} \tag{15}$$

with $\xi \in \mathbb{R}_{++}$. Finally, the torque derived from the whole body controller, $\boldsymbol{\tau}_{WBC} \in \mathbb{R}^{12}$, is computed. To compensate for model uncertainty for swing legs (on hardware, not in simulation), the integral of joint acceleration error with gain $K \in \mathbb{R}_{++}$ is added to the torque applied to the system:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{WBC} - K\int_{t_0^{sw}}^{t} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{WBC})\,\mathrm{d}t \tag{16}$$

While this modification implies $\boldsymbol{\tau}$ may not satisfy the CBF condition in (15), we note that $\boldsymbol{\tau}_{WBC}$ may not satisfy (15) in the presence of model uncertainty. To achieve safe behavior in practice, it is necessary to balance the choice of safe inputs with model uncertainty.

TABLE I. WHOLE-BODY CONTROL TASK HIERARCHY.

Priority	Type	Task
0	$=$	Floating base equations of motion.
	$\geq$	Torque limits.
	$\geq$	Friction cone constraint.
	$=$	No motion at the contact points.
	$\geq$	**Control barrier constraints.**
1	$=$	Torso linear and angular acceleration.
	$=$	Swing leg motion tracking.
2	$=$	Contact force tracking.

TABLE II. SIMULATION RESULTS

	I	II	III	IV	V
MPC constr.	None	CBF	State	State	CBF
WBC constr.	CBF	None	None	CBF	CBF
num. steps	28	140	140	140	140
num. missteps	5	6	5	0	0
avg. misstep [mm]	1.4	2.5	4.3	-	-
total swing time [s]	11.0	49.0	48.6	48.4	48.6
$\mathbf{h}_i^w < 0$ time [s]	2.4	2.3	15.3	2.6	0.4
$\mathbf{h}_{e,i}^w < 0$ time [s]	3.3	5.4	15.6	3.7	0.8

D. User Commands & Terrain Selection

User commanded twists and a desired gait pattern are provided to the robot via joystick and extrapolated to a state reference signal $\mathbf{x}_{ref}(t)$. The reference input $\mathbf{u}_{ref}(t)$ is constructed by equally distributing the weight over all contact feet. The MPC cost function is a frequency dependent quadratic cost around the reference trajectories to promote smooth optimal inputs [37].

We assume that a segmented terrain model with each segment described by a planar boundary and a surface normal is available. For each contact phase within the MPC horizon, the terrain segment is selected that is closest to the reference end-effector position determined by $\mathbf{x}_{ref}(t)$, evaluated at the middle of the stance phase. A convex polygon is fit to the selected terrain, starting from the reference end-effector position projected onto the segment boundary. This polygon, together with the surface normal, define the half spaces for the constraints in (12) and (13).

V. Results

We evaluate the controller proposed in Section IV in simulation on a classical stepping-stones scenario as shown in Figure 3. The stones are configured with a pattern of $0.5\,\mathrm{m}$ width and $0.35\,\mathrm{m}$ longitudinal spacing, with random displacements up to 10, 15, and $5\,\mathrm{cm}$, in longitudinal, lateral, and vertical direction respectively. The controller is commanded to perform a trotting gait with a forward velocity of $0.25\,\mathrm{m/s}$, and commanded to stop on the final stone. We compare our proposed controller, numbered V, against four alternative formulations and report results in Table II.

As seen in the supplementary video [1], the controller with no foot placement constraints in the MPC controller and a CBF constraint in the high-frequency controller (denoted CBF-QP, and the closest to the related work [11]) is able to enforce safety for a number of steps, but quickly destabilizes. The absence of information on the safety constraint in the MPC layer results in an abrupt and strong correction for safety by the high-frequency CBF. This approach work well only when the stepping-stones are placed close to the nominal gait of the robot, but it fails in this more challenging scenario.

The second and third controllers include foot placement constraints in the MPC controller, but not in the high-frequency controller. In the second controller the constraints are implemented as CBFs through (14) and in the third controller they are implemented as state constraints through (12). Both of these controllers are able to successfully traverse the length of the stepping-stones scenario. We see that the controllers exhibit similar numbers of missteps, but the MPC controller with CBFs has smaller average misstep size.

The fourth and fifth controller enforce the high frequency CBFs (15) and contain either state constraints (12) or CBFs (14) in the MPC formulation. Both controllers complete the scenario without missteps. However, the proposed controller shows the least amount of time violating the barrier conditions. The difference can be explained through the results in Figure 6. Because the MPC with state constraints (top) is not aware of the CBF condition, it plans for a trajectory that violates these constraints during the swing phase. During execution, the high-frequency tracking controller strictly enforces the CBF, resulting in a deviation from the MPC plan. Such abrupt deviations can cause problems, for example when operating close to kinematic limits. Consistently enforcing the CBF condition removes this mismatch (bottom).

The simulation experiments indicate that including CBF

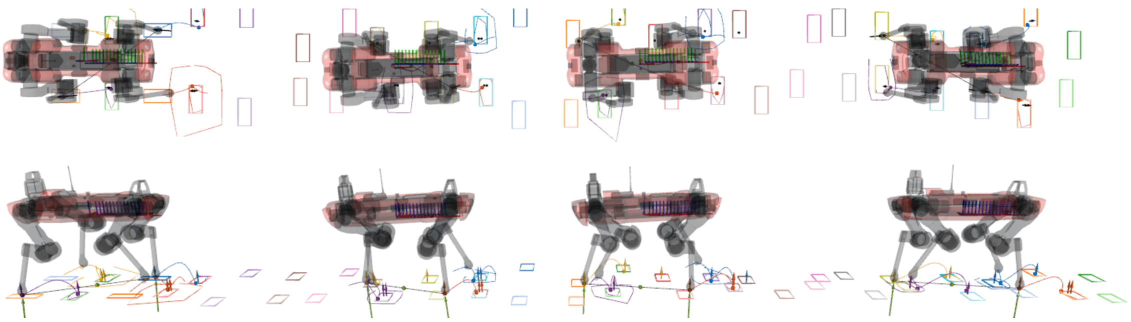

Fig. 3. ANYmal traversing stepping-stones in simulation using the multi-layer CBF-MPC controller. The target foothold regions as well as the contracting barrier constraints are shown at snapshots in the motion. See the video in the supplementary material for the full motion [1].

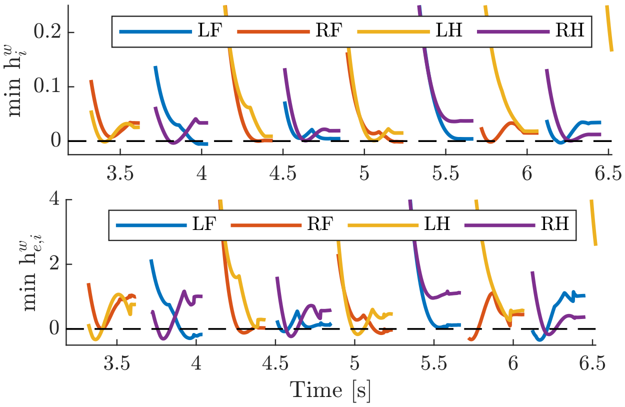

Fig. 4. The minimum values of $\mathbf{h}_i^w$ and $\mathbf{h}_{e,i}^w$ per leg during the stepping-stones simulation with the CBF only at MPC level.

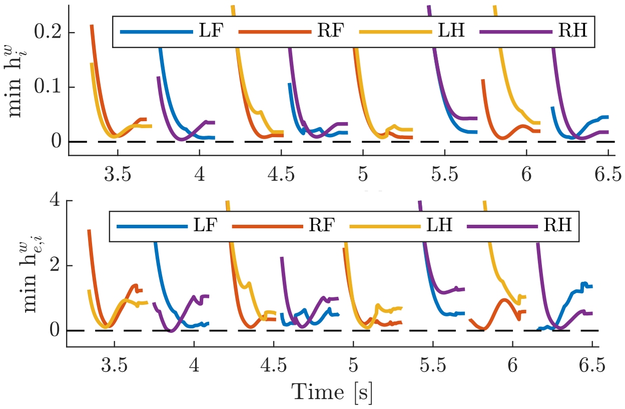

Fig. 5. The minimum values of $\mathbf{h}_i^w$ and $\mathbf{h}_{e,i}^w$ per leg during the stepping-stones simulation with the CBF in both WBC & MPC.

constraints in the high-frequency controller leads to safer behavior, and that including terrain constraints in the MPC controller prevents the high-frequency CBF from destabilizing the gait. Finally, enforcing CBF constraints in both layers of the hierarchy prevents an inconsistency that results in large deviations from the optimal solution determined by the MPC layer. The values of $\mathbf{h}_i^w$ and $\mathbf{h}_{e,i}^w$ for the controller with CBFs only in the MPC and for with the CBFs in both WBC & MPC can be seen in Figures 4 and 5. The controller with CBFs at both levels has smaller violations of constraints (13) and (14).

We evaluate the efficacy of this method experimentally on the ANYmal robotic platform. All computation runs on a single onboard PC (Intel i7-8850H, 2.6 GHz, hexa-core 64-bit) with the MPC solver running asynchronously at 30 Hz and the whole-body QP tracking controller running at 400 Hz.

The robot is initialized on pre-mapped terrain and receives external base twist and gait commands. The size of the segmented regions are decreased by 5 cm with respect to the real boundary to provide a margin for state estimation errors. In the supplementary video [1] we visualize the internal state of the controller. For legs that are in swing, a projection of the barrier constraint in (13) onto the terrain is plotted. This barrier constraint shrinks over time and converges to the selected target foothold region at foot contact. Furthermore, it can be seen how the foothold target is large when stepping onto the wooden pallet. This shows that the proposed method can seamlessly transition between rough and flat terrain, restricting the motion only when necessary for safe foot placement. The values of $\mathbf{h}_i^w$ and $\mathbf{h}_{e,i}^w$ for several steps can be seen in Figure 7. Both constraints are rarely violated, which confirms that the safety constraints are successfully transferred to hardware.

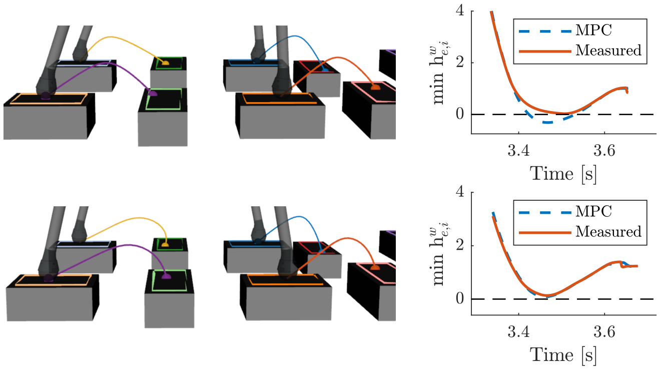

Fig. 6. Visualization of the planned MPC trajectories for different constraint formulations. **Top:** MPC with state constraints on the touchdown location (controller IV). **Bottom:** MPC with CBF constraints (controller V). The plots on the right show the planned and measured values of $\mathbf{h}_{e,i}^w$ for the right front foot, with deviations from the MPC optimal trajectory occurring when CBF constraints are absent from the MPC formulation.

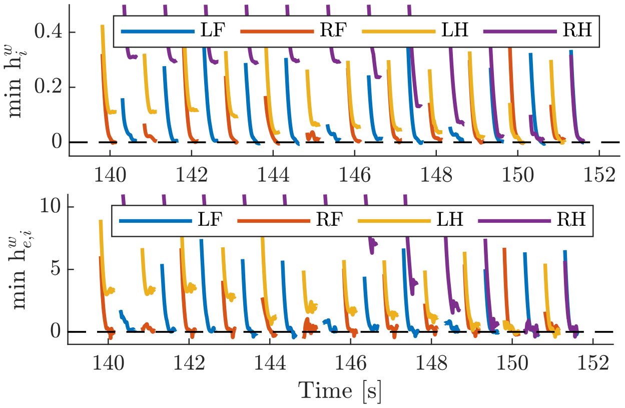

Fig. 7. The minimum values of $\mathbf{h}_i^w$ and $\mathbf{h}_{e,i}^w$ per leg during the stepping-stones hardware experiment for the proposed controller with CBF constraints in WBC & MPC.

VI. Conclusions

We proposed a multi-layered control framework that combines CBFs with MPC. Simulation experiments show that enforcing CBF constraints on both the MPC and QP tracking layer outperforms variants where they are enforced at only one of the layers. Additionally, we validated the viability of the approach on hardware by demonstrating dynamic locomotion on stepping-stones with safety constraints. Future work includes developing a perception pipeline to automatically perform terrain-based segmentation from sensor data and studying the theoretical properties of the proposed controller.

References

[1] Supplementary video: https://youtu.be/TCDIirXfByE.

[2] 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)*. IEEE-RAS, 2019, pp. 9–16.

[3] 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.

[4] C. Mastalli, I. Havoutis, M. Focchi, D. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: coupled planning, terrain mapping and whole-body control," *Prepint on HAL*, 2018.

[5] D. Belter, P. Labecki, and P. Skrzypczynski, "Adaptive motion planning for autonomous rough terrain traversal with a walking robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2016.

[6] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[7] 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)*. IEEE, 2014, pp. 6271–6278.

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

[9] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[10] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs." in *Robotics: Science and Systems (RSS)*, 2015.

[11] 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)*. IEEE, 2016, pp. 827–834.

[12] Q. Nguyen, X. Da, J. Grizzle, and K. Sreenath, "Dynamic walking on stepping stones with gait library and control barrier functions," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 384–399.

[13] Q. Nguyen and K. Sreenath, "Optimal robust time-varying safety-critical control with application to dynamic walking on moving stepping stones," in *Dynamic Systems and Control Conference (DSCC)*, vol. 50701. American Society of Mechanical Engineers, 2016, p. V002T28A005.

[14] 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 (RSS)*, Corvalis, Oregon, USA, July 2020.

[15] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1537–1553, 2018.

[16] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[17] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive locomotion in rough terrain–online foothold optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, 2020.

[18] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "Mpc-based controller with terrain insight for dynamic legged locomotion," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2436–2442.

[19] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *International Conference on Humanoid Robots (Humanoids)*. IEEE-RAS, Nov 2014, pp. 295–302.

[21] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, July 2018.

[22] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, Sept 2017, pp. 4102–4109.

[23] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *International Conference on Humanoid Robotics (Humanoids)*. IEEE-RAS, 2017, pp. 577–584.

[24] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, July 2018.

[25] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2019, pp. 4730–4737.

[26] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. D. Ames, "Nonlinear model predictive control of robotic systems with control lyapunov functions," in *Robotics: Science and Systems (RSS)*, 2020.

[27] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[28] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," *arXiv preprint arXiv:2010.09819*, 2020.

[29] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," *arXiv preprint arXiv:2007.11718*, 2020.

[30] L. Perko, *Differential equations and dynamical systems*. Springer Science & Business Media, 2013, vol. 7.

[31] 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*, vol. 62, no. 8, pp. 3861–3876, 2017.

[32] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 93–100.

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

[34] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *arXiv preprint arXiv:2004.01761*, 2020.

[35] J. Reher, C. Kann, and A. D. Ames, "An inverse dynamics approach to control lyapunov functions," in *American Control Conference (ACC)*. IEEE, 2020, pp. 2444–2451.

[36] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *International Conference on Humanoid Robots (Humanoids)*. IEEE-RAS, Nov 2016, pp. 558–564.

[37] R. Grandia, F. Farshidian, A. Dosovitskiy, R. Ranftl, and M. Hutter, "Frequency-aware model predictive control," *arXiv preprint arXiv:1809.04539*, 2018.