Walking in Narrow Spaces: Safety-critical Locomotion Control for Quadrupedal Robots with Duality-based Optimization

Qiayuan Liao, Zhongyu Li, Akshay Thirugnanam, Jun Zeng, and Koushil Sreenath

***Abstract*— This paper presents a safety-critical locomotion control framework for quadrupedal robots. Our goal is to enable quadrupedal robots to safely navigate in cluttered environments. To tackle this, we introduce exponential Discrete Control Barrier Functions (exponential DCBFs) with duality-based obstacle avoidance constraints into a Nonlinear Model Predictive Control (NMPC) with Whole-Body Control (WBC) framework for quadrupedal locomotion control. This enables us to use polytopes to describe the shapes of the robot and obstacles for collision avoidance while doing locomotion control of quadrupedal robots. Compared to most prior work, especially using CBFs, that utilize spherical and conservative approximation for obstacle avoidance, this work demonstrates a quadrupedal robot autonomously and safely navigating through very tight spaces in the real world. (Our open-source code is available at `https://github.com/HybridRobotics/quadruped_nmpc_dcbf_duality`, and the video is available at `https://youtu.be/p1gSQjwXm1Q`.)**

I. Introduction

Legged robots have high maneuverability compared to wheeled robots, allowing them to traverse rough terrains and challenging environments and realize dynamic motions [1]. However, their highly nonlinear dynamics and hybrid modes make control of legged robots a challenging task. Ensuring safe navigation through an obstacle-filled environment further exacerbates this problem [2]. Some prior work on legged locomotion achieves obstacle avoidance by decoupling the motion planning and the locomotion tasks, like in [3], where the motion planning problem is first solved for the legged robot's shape, and a control loop is used to track the planned path. Since the planning is performed without the robot's dynamics, the trajectory is not necessarily dynamically feasible, which can violate safety constraints. Other works on obstacle avoidance for legged robots combine the planning and control tasks using optimizations such as model predictive control (MPC) [4]. However, most of these works enforce safety online by over-approximating the shape of the robots and obstacles, which results in conservative movements and can lead to deadlock [5]. To tackle this problem, this paper develops a nonlinear model predictive control (NMPC) formulation with discrete-time control barrier functions (DCBF) constraints for robot locomotion. Unlike most of previous CBFs work [6], [7], [4], the safety constraints are enforced by considering the polytopic shapes of the robot and its surrounding obstacles, allowing for tight obstacle avoidance motions in cluttered environments, as depicted in Fig. 1. We validate our safety-critical locomotion algorithm with our proposed autonomy stack to achieve navigation tasks through tight spaces.

All authors are with Hybrid Robotics Group at the Department of Mechanical Engineering, UC Berkeley, USA. liaoqiayun@gmail.com, {zhongyu_li, akshay_t, zengjunsjtu, koushils}@berkeley.edu. Qiayuan Liao is also with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China.

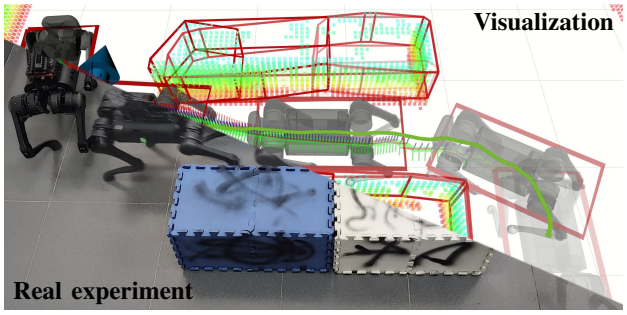

Fig. 1. A quadrupedal robot whose width is 0.32 m autonomously and safely navigates through a tight space that is only 0.5-meter-wide using the proposed NMPC with exponential DCBF Duality framework in experiments.

A. Related Work

Previous work tackling the navigation problem using legged robots can be broadly classified into three categories: 1) considering obstacles only in the motion planning layer without robot's dynamics, 2) including collision avoidance in optimal control but without considering robot's finer shape, and 3) safety-critical control for tight maneuvers.

1) Collision-free Motion Planning: Motion planning for legged robots has been an attractive topic and usually involves planning in the configuration space [8], [9]. Some approaches consider planning footholds for these robots to tackle the problems of stepping over discrete terrains [10], [11]. Other approaches also consider avoiding obstacles in confined spaces, as demonstrated in [12], [13]. Whole-body motion planning for quadrupedal robot with signed distance field (SDF) [14] and elevation map are demonstrated in [9] using a hierarchical motion planning framework. However, most motion planning work for legged robots [15], [16], [9] only considers planning in configuration space without the dynamics of the legged robot, which results in slow and statically stable gaits. Additionally, the entire framework could fail due to collisions since the control layer doesn't consider the safety criteria reported in [9]. This motivates us to consider system dynamics while avoiding obstacles.

2) Obstacle Avoidance in Trajectory Generation & Control: Obstacle avoidance with respect to system dynamics could be considered in both trajectory generation and control problems. Some existing work considers the trajectory gener-

ation problem within an MPC formulation [17], [18], where positivity of the SDF is regarded as an optimization constraint to ensure the robot's safety. Other approaches usually consider obstacle avoidance constraints as enforcing distance functions to be positive in trajectory generation problems. To make the distance functions differentiable, the robot and the obstacles are usually considered circular or spherical controlled regions for 2D or 3D navigation. The optimization with distance function constraints is validated by simulation analysis [19], [20], [21] and experiments [17], [22], [23] on legged robots for trajectory generation. This approach has been extended by using discrete-time control barrier functions [24] in quadruped stepping on discrete terrains [4] and humanoid navigation simulations [6], [5], [25]. However, the main disadvantage of all the existing work above is that they over-approximate either the robot or the obstacles as circular or spherical objects in the trajectory generation or control problems, which could be overly conservative and result in a deadlock maneuver. Although the SDF method tries not to over-approximate the robot by considering it as a combination of spherical objects, obstacles are still over-approximated by hyper-spheres. This motivates us to carry out robot autonomy algorithms to enable tight maneuvers for legged robot systems with less conservative safety criteria than the existing work.

3) Safety-Critical Tight Maneuvers: To achieve non-conservative obstacle avoidance, polytopic obstacle avoidance is usually required, *i.e.*, the robot and obstacles are considered as polytopes, or combinations of them [26]. The main challenge of polytopic avoidance is that the distance function between polytopes is non-smooth. Mixed-integer programming [27] could solve the problem but is only applicable for simple linear systems. Dual analysis [26] can reformulate the non-smooth polytopic constraints into smooth ones in the trajectory optimization problem for quadrupedal robots with its full-order dynamics offline [28] or simplified dynamics online [29]. This dual analysis has also recently been synthesized by discrete-time control barrier functions (DCBFs) [30], [31] to achieve real-time trajectory generation and control [32]. As we will see here, we apply the dual optimization with a DCBF [32] to achieve safety-critical locomotion with less conservative maneuvers.

B. Contributions

In this paper, we propose a robot autonomy stack that enables safety-critical locomotion in tight spaces with duality-based optimization. The primary contribution of this paper is the first introduction of duality-based Control Barrier Functions (CBFs) to legged locomotion control to ensure safety, *e.g.*, obstacle avoidance, in the real world. The duality-based obstacle avoidance constraints allow us to describe the robot and its surrounding obstacles by polytopes, which provides a finer approximation of their shapes than the commonly used spheres. Combining with CBFs and model predictive control for the quadrupedal robot, we are able to realize non-conservative obstacle avoidance online. Further, we develop an end-to-end navigation framework that combines robot perception feedback and the proposed safety-critical locomotion controller. The proposed framework is deployed on a quadrupedal robot in the real world and enables the robot to autonomously and safely navigate through various narrow spaces in experiments.

II. Background

Before diving into the details of the proposed safety-critical locomotion framework, we first make a brief introduction of the background knowledge that is essential for the development of the following sections.

A. Discrete-time Control Barrier Functions

Consider a discrete-time dynamical system with states $\mathbf{x}_k \in \mathcal{X} \subset \mathbb{R}^n$ and inputs $\mathbf{u}_k \in \mathcal{U}$ as

$$\mathbf{x}_{k+1} = f_k^d(\mathbf{x}_k, \mathbf{u}_k), \tag{1}$$

where $\mathcal{U}$ is the admissible input set, which is compact. In this paper, we consider a safe set of states $\mathcal{C}$, defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ by

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\}. \tag{2}$$

Throughout this paper, we refer to $\mathcal{C}$ as the safe set. For robotics applications, the function h can be the minimum distance function between the robot and the obstacle. To guarantee safety, *e.g.*, obstacle avoidance, we want to make $\mathcal{C}$ an invariant set, *i.e.*, if the initial state lies in $\mathcal{C}$, then the entire evolution of the state trajectory should also lie in $\mathcal{C}$.

The function h becomes a discrete-time control barrier function if it satisfies the following relation $\forall \mathbf{x}_k \in \mathcal{X}$,

$$h(\mathbf{x}_{k+1}) \geq \gamma(\mathbf{x}_k) h(\mathbf{x}_k), \quad 0 \leq \gamma(\mathbf{x}_k) \leq 1, \tag{3}$$

where $\mathbf{x}_{k+1} = f_k^d(\mathbf{x}_k, \mathbf{u}_k)$ for some $\mathbf{u}_k \in \mathcal{U}$, and $\gamma(\mathbf{x}_k)$ is the state-dependent decay rate [31]. The DCBF constraint (3) enforces h to decrease at most exponentially with the decay rate $\gamma(\mathbf{x}_k)$.

Given a choice of $\gamma(\mathbf{x}_k)$, we denote $\mathcal{K}(\mathbf{x}_k)$ as

$$\mathcal{K}(\mathbf{x}_k) \coloneqq \{\mathbf{u} \in \mathcal{U} : h(f_k^d(\mathbf{x}_k, \mathbf{u}_k)) \geq \gamma(\mathbf{x}_k) h(\mathbf{x}_k)\}. \tag{4}$$

Then if $\mathbf{x}_0 \in \mathcal{C}$ and $\mathbf{u}_k \in \mathcal{K}(\mathbf{x}_k) \neq \emptyset$, $\forall k \in \mathbb{Z}^+$, then $\mathbf{x}_k \in \mathcal{C}$ for $\forall k \in \mathbb{Z}^+$, *i.e.*, the resulting trajectory, is safe [6].

B. Minimum Distance between Polytopes

1) Polytope representation: We can describe the shape of the dynamic robot and the i-th static obstacle as convex polytopes in a l-dimensional space, which are defined using inequality constraints, respectively, as

$$\mathcal{R}(\mathbf{x}_k) \coloneqq \{\mathbf{y} \in \mathbb{R}^l : A_{\mathcal{R}}(\mathbf{x}_k)\mathbf{y} \leq \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k)\}, \tag{5a}$$

$$\mathcal{O}_i \coloneqq \{\mathbf{y} \in \mathbb{R}^l : A_{\mathcal{O}_i}\mathbf{y} \leq \mathbf{b}_{\mathcal{O}_i}\}, \tag{5b}$$

where $\mathcal{R}$ and $\mathcal{O}_i$ represent the robot and the i-th obstacle respectively, $A_{\mathcal{R}}(\mathbf{x}_k) \in \mathbb{R}^{s^{\mathcal{R}} \times l}, \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k) \in \mathbb{R}^{s^{\mathcal{R}}}$ define the robot, and $A_{\mathcal{O}_i} \in \mathbb{R}^{s^{\mathcal{O}_i} \times l}$ and $\mathbf{b}_{\mathcal{O}_i} \in \mathbb{R}^{s^{\mathcal{O}_i}}$ define the i-th obstacle. The symbols $s^{\mathcal{R}}$ and $s^{\mathcal{O}_i}$ represent the number of facets of the polytopic sets for the robot and i-th obstacle, respectively. Note that the robot polytope $\mathcal{R}$ depends on the system state.

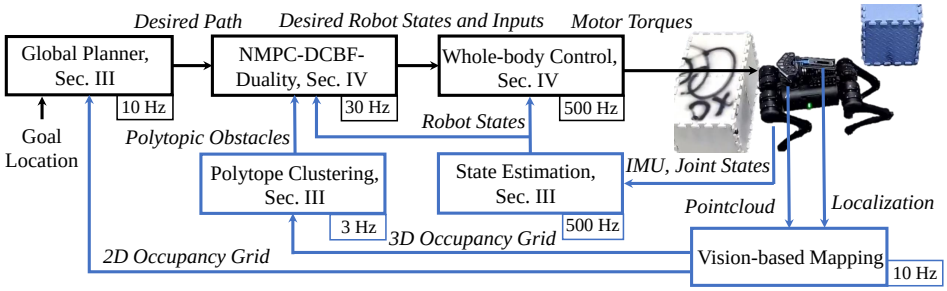

Fig. 2. Overview of the proposed framework for safety-critical locomotion control of quadrupedal robots. The black blocks are related to planning and control, while the blue blocks are for state estimation and vision feedback. The robot uses two onboard depth cameras to perceive the world and leverages an additional tracking camera to localize itself. The sensed pointcloud is filtered and registered into a 3D occupancy grid (using Octomap [33]) and clustered into polytopes. After being given a goal location, the global planner finds the desired path without considering the robot's shape from the robot's current position. The NMPC uses clustered polytopes for obstacle avoidance employing the exponential DCBF duality controller and tracking the desired global path while avoiding obstacles nearby. A higher-frequency WBC then computes the robot's motor torques based on the optimized results of the NMPC. Moreover, a state estimator using a Kalman Filter provides the currently estimated robot states to the NMPC and WBC.

2) Primal Problem: The square of the minimum distance between $\mathcal{R}(\mathbf{x}_k)$ and $\mathcal{O}_i$, denoted by $d_i(\mathbf{x}_k)$, can be computed using a QP as follows:

$$d_i(\mathbf{x}_k) = \min_{\mathbf{y}_k^{\mathcal{R}}, \mathbf{y}_k^{\mathcal{O}_i}} \left\| \mathbf{y}_k^{\mathcal{R}} - \mathbf{y}_k^{\mathcal{O}_i} \right\|^2, \tag{6a}$$

$$\text{s.t.} \quad A_{\mathcal{R}}(\mathbf{x}_k)\mathbf{y}_k^{\mathcal{R}} \leq \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k), \; A_{\mathcal{O}_i}\mathbf{y}_k^{\mathcal{O}_i} \leq \mathbf{b}_{\mathcal{O}_i}. \tag{6b}$$

We directly want to use this as the DCBF function, *i.e.*, $h_i(\mathbf{x}_k) = d_i(\mathbf{x}_k)$, and enforce $h_i(\mathbf{x}_k) \geq 0$. However, the above optimization problem can only be solved numerically and not analytically. This results in the difficulty of adding the minimum distance function in a DCBF constraint to other optimization problems because it is implicitly defined by (6) and non-differentiable.

Moreover, the primal problem is a minimization problem and is not suitable for working with the DCBF constraint (3) [32]. Instead, we convert the minimization problem (6) into a maximization problem using the principle of duality.

3) Duality problem: The dual problem of (6) is given by

$$d_i(\mathbf{x}_k) = \max_{\boldsymbol{\lambda}_k^{\mathcal{R}}, \boldsymbol{\lambda}_k^{\mathcal{O}_i}} -(\boldsymbol{\lambda}_k^{\mathcal{R}})^T \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k) - (\boldsymbol{\lambda}_k^{\mathcal{O}_i})^T \mathbf{b}_{\mathcal{O}_i} \tag{7a}$$

$$\text{s.t.} \quad A_{\mathcal{R}}^T(\mathbf{x}_k)\boldsymbol{\lambda}_k^{\mathcal{R}} + A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} = 0, \tag{7b}$$

$$\left\| A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} \right\|_2 \leq 1, \; \boldsymbol{\lambda}_k^{\mathcal{R}} \geq 0, \; \boldsymbol{\lambda}_k^{\mathcal{O}_i} \geq 0, \tag{7c}$$

where $\boldsymbol{\lambda}_k^{\mathcal{R}}$ and $\boldsymbol{\lambda}_k^{\mathcal{O}_i}$ are the dual variables corresponding to the primal problem.

Since (7) is a SOCP which is slower to compute, we use another dual formulation [32] for the distance computation, which results in a QP, and is given by

$$d_i(\mathbf{x}_k) = \max_{\boldsymbol{\lambda}_k^{\mathcal{R}}, \boldsymbol{\lambda}_k^{\mathcal{O}_i}} -\frac{1}{4}(\boldsymbol{\lambda}_k^{\mathcal{O}_i})^T A_{\mathcal{O}_i}^T A_{\mathcal{O}_i} \boldsymbol{\lambda}_k^{\mathcal{O}_i} - (\boldsymbol{\lambda}_k^{\mathcal{R}})^T \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k) - (\boldsymbol{\lambda}_k^{\mathcal{O}_i})^T \mathbf{b}_{\mathcal{O}_i} \tag{8a}$$

$$\text{s.t.} \quad A_{\mathcal{R}}^T(\mathbf{x}_k)\boldsymbol{\lambda}_k^{\mathcal{R}} + A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} = 0, \tag{8b}$$

$$\boldsymbol{\lambda}_k^{\mathcal{R}} \geq 0, \boldsymbol{\lambda}_k^{\mathcal{O}_i} \geq 0. \tag{8c}$$

The QP (8) results in the same optimal value $d_i(\mathbf{x}_k)$ as (7), but the optimal solution of (8), $\boldsymbol{\lambda}_k^{\mathcal{R}}$ and $\boldsymbol{\lambda}_k^{\mathcal{O}_i}$, is a scaled value of that of (7).

C. Exponential DCBF Duality Constraints

To reduce the complexity of the DCBF constraints when used in MPC, we make use of exponential DCBF constraints, where instead of enforcing (3) we enforce the following constraint,

$$h_i(\mathbf{x}_k) \geq (\Pi_{n=0}^{k} \gamma_n) h_i(\mathbf{x}_0), \tag{9}$$

where k represents the time index in the MPC and $\mathbf{x}_0$ is the current state. Note that (9) is obtained by rolling out time in the DCBF constraint (3). It can be shown that the exponential DCBF constraint (9) is equivalent to the following set of exponential DCBF duality constraints [32],

$$-(\boldsymbol{\lambda}_k^{\mathcal{R}})^T \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k) - (\boldsymbol{\lambda}_k^{\mathcal{O}_i})^T \mathbf{b}_{\mathcal{O}_i} \geq (\prod_{n=0}^{k} \gamma_n) d_i(0) \tag{10a}$$

$$A_{\mathcal{R}}^T(\mathbf{x}_k)\boldsymbol{\lambda}_k^{\mathcal{R}} + A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} = 0, \tag{10b}$$

$$\left\| A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} \right\|_2 \leq 1, \tag{10c}$$

$$\boldsymbol{\lambda}_k^{\mathcal{R}} \geq 0, \boldsymbol{\lambda}_k^{\mathcal{O}_i} \geq 0, \tag{10d}$$

where $d_i(0) = d_i(\mathbf{x}_0)$ is the square of the minimum distance between the robot and the i-th obstacle at the current state $\mathbf{x}_0$, calculated using the distance QP (8). As we will see, the above exponential DCBF duality constraints (10) will be incorporated into an NMPC to guarantee safety for the locomotion control of quadrupedal robots.

III. Framework

Having developed the math background, we next present an overview of the entire framework as illustrated in Fig. 2.

The robot is equipped with depth cameras and a tracking camera. The environment is perceived via the depth camera by a pointcloud, and the odometry of the robot is estimated by the tracking camera through Visual-Inertial Odometry. With estimated robot location, the pointcloud is filtered and registered in a 3D occupancy grid by Octomap [33] at 10Hz. In order to describe the shape of obstacles nearby, the entire registered pointcloud is clustered into multiple small polytopes using Euclidean distance in order to bound the obstacles. In order to be more computationally efficient, we only consider clustering the voxels that are near the robot, *i.e.*, in a local map, and such a polytope clustering runs at 3 Hz.

The clustered 3D polytopes are projected to the ground to find the 2D convex polytope representation of the obstacles. The Octomap also generates a projection to the ground to obtain a 2D occupancy grid map for global planning.

The global planner generates a path at a low frequency using Dijkstra's algorithm [34] without considering the robot's shape. The path of 2D waypoints for the robot is first generated, and yaw angles are obtained using the tangential angles along the path.

The path is then tracked by the following NMPC controller with an exponential DCBF with duality to consider avoidance with the bounding polytopes of obstacles while controlling the robot's locomotion.

The NMPC controller uses a centroidal dynamics model of the quadrupedal robot [35] with the system states as the robot's base pose, base momentum, and joint positions, and the system inputs as the ground contact forces and joint velocities. To avoid collisions with local obstacles, we add the exponential DCBF duality constraints (10) for polytope obstacle avoidance. Combined with other constraints, such as friction cone constraints, to ensure the robot's movement, the NMPC evaluates optimized system states and inputs. It is followed by a Whole-body Controller (WBC) [36] to figure out the robot's optimal generalized coordinate acceleration, contact forces (ground reaction forces), and joint torques according to the optimized states and inputs from the NMPC. The torque computed by WBC is set as a feed-forward term and is sent to the robot's motor controller. This is combined with joint-level PD commands, which could reduce the shock during foot contact and improve tracking performance.

Running in the same loop with WBC, a Kalman Filter estimates the robot's base position and velocity from the onboard measurements of base orientation, base acceleration, and feet positions which are obtained by measured joint positions and the robot's forward kinematics.

IV. Formulation and Implementation

In this section, we briefly describe the formulation of NMPC with the exponential DCBF constraints for obstacle avoidance and then present some implementation details.

A. NMPC for Quadrupedal Locomotion

Consider a Nonlinear Model Predictive Control formulation with horizon N as

$$\min_{\{\mathbf{x}_k,\mathbf{u}_k\}} \quad \sum_{k=0}^{N-1} l_k(\mathbf{x}_k,\mathbf{u}_k), \tag{11a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{x}(0), \tag{11b}$$

$$\mathbf{x}_{k+1} = f_k^d(\mathbf{x}_k,\mathbf{u}_k), \qquad k=0,...,N-1, \tag{11c}$$

$$\mathbf{g}_k(\mathbf{x}_k,\mathbf{u}_k) = \mathbf{0}, \qquad k=0,...,N, \tag{11d}$$

$$\mathbf{h}_k(\mathbf{x}_k,\mathbf{u}_k) \geq \mathbf{0} \qquad k=0,...,N, \tag{11e}$$

where $\mathbf{x}_k$ is the state and $\mathbf{u}_k$ is the input at time k, $\mathbf{x}(0)$ is the current state, l_k is a time-varying stage cost. We want to find the control input that minimizes the total cost subject to the initial state $\mathbf{x}(0)$, system dynamics $f_k^d(\mathbf{x}_k,\mathbf{u}_k)$, and the general equality $\mathbf{g}_k(\mathbf{x}_k,\mathbf{u}_k)$ and inequality $\mathbf{h}_k(\mathbf{x}_k,\mathbf{u}_k)$ constraints.

1) System Dynamics: We use a previous NMPC formulation using centroidal dynamics of a quadruped described in [37], where the system states $\mathbf{x} \in \mathbb{R}^{24}$ and inputs $\mathbf{u} \in \mathbb{R}^{24}$ are defined as:

$$\mathbf{x} = [\mathbf{h}_{com}^T, \mathbf{q}_b^T, \mathbf{q}_j^T]^T, \quad \mathbf{u} = [\mathbf{f}_c^T, \mathbf{v}_j^T]^T \tag{12}$$

where $\mathbf{q} = [\mathbf{q}_b^T, \mathbf{q}_j^T]^T$ is the generalized coordinate, and the ZYX-Euler angle parameterization is assumed to represent the robot's torso's orientation. $\mathbf{h}_{com} = [\mathbf{p}_{com}^T, \mathbf{l}_{com}^T]^T \in \mathbb{R}^6$ is the collection of the normalized centroidal momentum, $\mathbf{f}_c \in \mathbb{R}^{12}$ consists of contact forces at four contact points i.e., four ground reaction force of foot. $\mathbf{q}_j$ and $\mathbf{v}_j$ are the joint positions and velocities. The continuous-time system flow map is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{p}_{com}\\ \mathbf{l}_{com}\\ \mathbf{q}_b\\ \mathbf{q}_j\end{bmatrix} = \begin{bmatrix}\frac{1}{m}\sum_{i=1}^{4}\mathbf{f}_{c_i} + \mathbf{g}\\ \frac{1}{m}\sum_{i=1}^{4}\mathbf{r}_{com,c_i}\times\mathbf{f}_{c_i}\\ M_b^{-1}(\mathbf{h}_{com} - M_j\mathbf{v}_j)\\ \mathbf{v}_j\end{bmatrix} \tag{13}$$

where m is the robot total mass, $\mathbf{r}_{com,\mathrm{c_i}}$ is the position of the i-th foot w.r.t to the center of mass, while $M(q) = [M_b(q)\ M_j(q)] \in \mathbb{R}^{6\times 18}$ is the centroidal momentum matrix which maps generalized velocities to centroidal momentum. Readers can refer to [35] for more details. The dynamics (13) can be discretized and expressed in the discrete-time form (11c).

2) Cost: The cost (11a) is a quadratic tracking cost to follow a given full system state trajectory, including base pose (and/or its twist), normalized momentum, and nominal joint positions.

3) Constraints: The gait (periodic contact sequence) is predefined, so we can formulate the constraints to ensure that stance legs remain on the same footholds and the swing legs follow predefined curves for the feet heights with zero contact force. A friction cone constraint of each stance leg is also added to avoid slipping. For more details about constraints, please see [37].

B. Exponential DCBF Duality Constraint

The NMPC shown in Sec. IV-A has not considered obstacle avoidance. To achieve safety-critical locomotion control with obstacle avoidance, we add exponential DCBF duality constraint (10) to the NMPC (11) by including (10a), (10c), (10d) as inequality constraints $\mathbf{h}_k(\mathbf{x}_k,\mathbf{u}_k)$, and (10b) as equality constraints $\mathbf{g}_k(\mathbf{x}_k,\mathbf{u}_k)$.

1) Signed Distance: In order to let the solver know how "intrusive" the robot is with the obstacles, we use signed distance duality formulation [26] by replacing the inequality norm constraint (10c) with an equality constraint.

$$\left\| A_{\mathcal{O}_i}^T \boldsymbol{\lambda}_k^{\mathcal{O}_i} \right\|_2 = 1. \tag{14}$$

Although the convex constraint (10c) is then turned into a non-convex equality constraint, (14) is an equality constraint

and can be handled through a projection method, and the solving time can be unaffected. The advantage of using such a signed distance formulation is that, even if it collides with the obstacle in the real world, *e.g.*, due to external perturbations, the robot will move away from the obstacle.

2) Margin: The CBFs (10a) are modified with two margins $\alpha, \beta \geq 0$ as:

$$-(\boldsymbol{\lambda}_k^{\mathcal{R}})^T \mathbf{b}_{\mathcal{R}}(\mathbf{x}_k) - (\boldsymbol{\lambda}_k^{\mathcal{O}_i})^T \mathbf{b}_{\mathcal{O}_i} \geq (\prod_{n=0}^{k} \gamma_n)\tilde{d}_i(\mathbf{x}_0) + \alpha,$$

where α is the minimum distance margin between the polytopes of the robot and obstacles. $\tilde{d}_i(\mathbf{x}_0)$ is given by

$$\tilde{d}_i(\mathbf{x}_0) = \max\{d_i(\mathbf{x}_0) - \beta, 0\}, \tag{16}$$

where β is a margin to prevent the robot adopting a conservative strategy, *e.g.*, taking a large detour or stopping in front of the obstacle, at the earlier stage of the NMPC.

C. Optimization Setup

We add the dual variables $\boldsymbol{\lambda}_k^{\mathcal{R}}$ and $\boldsymbol{\lambda}_k^{\mathcal{O}_i}$ into the system inputs of the optimal control problem (11). To solve this, a multiple shooting method is leveraged to transcribe the optimal control problem to a nonlinear program (NLP) problem, and the NLP is solved using Sequential Quadratic Programming (SQP) where the QP subproblem is solved using HPIPM [38]. For more details regarding the solving process and algorithm, we refer the reader to [32]. The numerical optimization is formulated in the OCS2 [39]. In the scenario where four obstacles (each has 15 vertices) are presented, the optimization can be solved in around 25 ms.

D. Whole-Body Control (WBC)

After solving the above-mentioned NMPC with exponential DCBF duality, we can obtain the optimized position and/or velocity profiles of the base, joints, and contact force from the optimizer. These are then utilized in a hierarchical optimization whole body controller [36], which computes the torque for each joint according to the optimized result from the NMPC in a prioritized way. The decision variable of WBC is: $\mathbf{x}_{wbc} = [\ddot{\mathbf{q}}^T, \mathbf{f}_{c_i}^T, \boldsymbol{\tau}^T]^T$, where $\ddot{\mathbf{q}}$ is the generalized acceleration, and $\boldsymbol{\tau}$ is the torque of all actuated joints.

With multiple tasks (equality and inequality constraints) defined, the WBC solves the QP problem in the null space of the higher priority tasks' linear constraints and tries to minimize the slack variables of the inequality constraints. This approach can consider the full nonlinear rigid body dynamics and ensure strict tasks priority [36].

We implement the WBC using the Pinocchio rigid body library [40] and qpOASES QP solver [41]. We also run state estimation in this same control loop.

V. Results

After introducing the entire formulation for the proposed safety-critical locomotion controller for quadrupeds, we now move on to deploy the framework on a quadrupedal robot A1 [42], which has 12 actuators and is $0.3\,\mathrm{m}$ wide and $0.6\,\mathrm{m}$ long. We validate our proposed algorithms in both simulation and experiments, and the results are recorded in the accompanying video (`https://youtu.be/p1gSQjwXm1Q`).

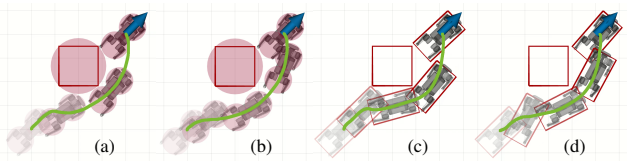

Fig. 3. Snapshots for four different constraint formulations in simulation: (a) Euclidean Distance Constraint, (b) Euclidean Distance Exponential DCBF, (c) Duality Constraint, (d) Exponential DCBF Duality (ours). Lighter snapshots are earlier in time. Overall, the robot paths (green) using CBFs (b)(d) are smoother than the ones without CBF in (a)(c). By considering the obstacle's shape by duality formulation in (c)(d), the robot is able to avoid the obstacle less conservatively compared to the ones using Euclidean distance constraints in (a)(b).

A. Simulation Validation

We first evaluate the proposed method in simulation using the robot dynamics in Gazebo [43]. To better demonstrate the effect of the duality and exponential DCBF constraints, we test four controllers in simulation as shown in Fig. 3. The robot is commanded to a goal while avoiding a square obstacle in the middle between the starting and ending points. There are four controllers which use different formulations for obstacle avoidance: 1) Euclidean Distance Constraint (Fig. 3(a)) where the robot and obstacle are approximated by circles and then their separations (l_2 distances) are constrained to be positive over the entire MPC horizon. 2) Euclidean Distance Exponential DCBF (Fig. 3(b)) where the l_2 distances between robot and obstacles are used as the function h in exponential DCBF defined in (9). 3) Duality Constraint (Fig. 3(c)) where the robot and obstacles are bounded by polytopes which are constrained to be separated by duality-based optimization [26] without CBF, and 4) the proposed Exponential DCBF Duality (Fig. 3(d)) where both polytopes and CBFs are used to avoid collisions.

By comparing Fig. 3(a, b) which use Euclidean distance that can only use circular shape approximation and Fig. 3(c, d) which use duality-based constraint that supports finer shape approximation by polytopes, we find that the robot has less detour in Fig. 3(c, d) than in Fig. 3(a, b). Such a property allows the robot to maneuver in a tighter space. Furthermore, by introducing the exponential DCBF, the robot is able to react to the obstacles earlier, such as Fig. 3(b) compared to Fig. 3(a), and Fig. 3(d) compared to Fig. 3(c). Therefore, the robot shows a smoother trajectory in Fig. 3(b,d) because the robot has more redundancy to avoid the obstacle without having a sudden change of heading only when the robot is close to the obstacle in Fig. 3(a,c). Such a benchmark highlights the advantages of the proposed method, which results in a smooth and safe trajectory for the robot to travel in a tighter space.

B. Experimental Setup

We now deploy the entire pipeline shown in Fig. 2 on the hardware of A1 which is equipped with an onboard computer which has a 4-core CPU. The parameters used in

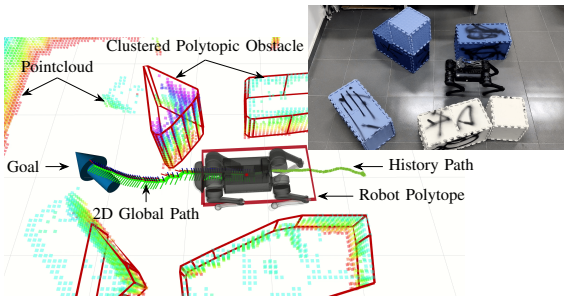

Fig. 4. Visualization of the planning data for a quadruped using the proposed framework, where data is taken from an experiment in the real world. The user-commanded 2D goal is drawn as a blue arrow; the path found by the global planner is shown as a chain of axes; the green curve represents the robot torso's history path; polytopes of the robot and obstacles are marked as the red bounding boxes.

the framework shown in Fig. 2 are explained as follows. The resolution of Octomap is $0.025\,\mathrm{m}$, and it generates a projected grid map with the same resolution. The global planner re-plans with the linear tolerance of $0.1\,\mathrm{m}$. The timespan for NMPC in Sec. IV-D is set to $T = 1.0\,\mathrm{s}$ is used with a nominal time adaptive discretization of $\delta t \approx 0.015\,\mathrm{s}$. We use a trotting gait with a $0.5\,\mathrm{s}$ gait cycle. The polytopic obstacles used for exponential DCBF duality constraints are chosen as the closest 4 obstacles to the robot (within $1\,\mathrm{m}$ box), and the maximum number of vertices of each polytope is set to 15. The robot is represented by a rectangle with the $0.32\,\mathrm{m}$ width and $0.6\,\mathrm{m}$ length. Margins are set to $\alpha = 0.03\,\mathrm{m}$, $\beta = 0.06\,\mathrm{m}$ and decay rate $\gamma(\mathbf{x}) = 1$ is used in the exponential DCBF constraints. The robot's desired linear velocity is set to $0.5\,\mathrm{m/s}$.

C. Navigation through Narrow Environments

We carried out four experiments in the real world as presented in Fig. 5 and consist of navigating in: (1) **Straight corridor** with two obstacles, a $0.5\,\mathrm{m}$ minimum clearance, and $2.5\,\mathrm{m}$-long path, (2) **L-shape corridor** with three obstacles, a $0.6\,\mathrm{m}$ minimum clearance, and a path length of about $2.5\,\mathrm{m}$. (3) **V-shape corridor** with four obstacles and a path length about $2.0\,\mathrm{m}$, and (4) **Random obstacles** with four different obstacles and a path length about $5.5\,\mathrm{m}$.

During experiments, the robot was commanded to go through these narrow corridors respectively and return back; we also commanded the robot randomly in the Random corridor. The experiments are recorded in the accompanying video, and the visualization sequences of environments and states generated by mentioned experiment data are shown in Fig. 5, and the visual representations are detailed in Fig. 4.

As demonstrated in Fig.5, in the straight corridor trial, the robot is able to slow down and turn its heading 90 degrees to fit itself into the corridor. After the robot enters the corridor, it speeds up to the desired speed while avoiding collision with the walls on both sides. In the L-shape and V-shape corridors, the robot decelerates at the corners to turn its body slowly without colliding with the cluttered obstacles nearby. In the random obstacles trial, the robot showcases the capacity to smoothly navigate in the free space formulated by the random obstacles, even fitting into the gap between

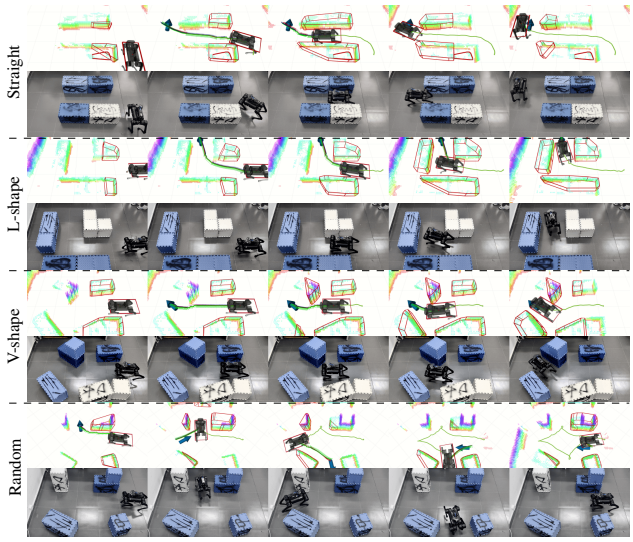

Fig. 5. Planning data visualization from experiments in four scenarios. For each scenario, snapshots of the robot's motion in time are shown. The obstacles are represented by polytopes (a red bounding box) after clustering the pointcloud, and the robot is bounded by a rectangle. Using the proposed method, the robot can safely travel through all of these narrow spaces.

adjacent obstacles. Over the course of these experiments, the minimum distance between the robot and obstacles is typically reached when the robot is turning or trying to squeeze between two obstacles. However, the robot never collides with the obstacle throughout all experiment trials.

However, it is possible that robots get deadlocked and be stuck when the global planner gives a low-quality path. For example, if the global planner does not specify the robot's yaw angle along the global path, the NMPC may find it hard to turn the robot's direction, such as turning the robot 90 degrees to head directly to a gap that is beside the robot. Therefore, we use tangential angles along Dijkstra's path as the robot's reference yaw, but it can be upgraded if the robot's configuration is considered during global planning.

VI. Conclusion and Future Works

In this paper, we proposed a Nonlinear MPC framework based on exponential DCBF duality for safety-critical locomotion control on quadrupedal robots. The proposed framework enabled the quadrupedal robot to safely and smoothly walk in narrow spaces by considering the shapes of the robot and the obstacles as polytopes. We validated our approach on a quadrupedal robot hardware, A1, in various obstacle-laden environments. However, we have only considered the navigation problem in 2D space. Future work could include implementing 3D obstacle avoidance with polytopes in a tighter space.

Acknowledgements

This work is supported in part by NSF Grants CMMI-1931853 and CMMI-1944722. The authors thank Xingxing Wang, Unitree Robotics, and the GDUT DynamicX robot team for their generous help in experiments.

REFERENCES

[1] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *arXiv preprint arXiv:2201.07067*, 2022.

[2] M. Tranzatto, M. Dharmadhikari, L. Bernreiter, M. Camurri, S. Khattak, F. Mascarich, P. Pfreundschuh, D. Wisth, S. Zimmermann, M. Kulkarni *et al.*, "Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned," *arXiv preprint arXiv:2207.04914*, 2022.

[3] J.-K. Huang and J. W. Grizzle, "Efficient anytime clf reactive planning system for a bipedal robot on undulating terrain," *arXiv preprint arXiv:2108.06699*, 2021.

[4] 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," in *International Conference on Robotics and Automation*, 2021, pp. 8352–8358.

[5] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," *arXiv preprint arXiv:2103.14252*, 2021.

[6] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, 2017.

[7] 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.

[8] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[9] R. Buchanan, L. Wellhausen, M. Bjelonic, T. Bandyopadhyay, N. Kottege, and M. Hutter, "Perceptive whole-body planning for multilegged robots in confined spaces," *Journal of Field Robotics*, vol. 38, no. 1, pp. 68–84, 2021.

[10] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *International conference on humanoid robots*, 2014, pp. 279–286.

[11] 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*, 2019, pp. 9–16.

[12] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[13] I. Kumagai, M. Morisawa, S. Nakaoka, and F. Kanehiro, "Efficient locomotion planning for a humanoid robot with whole-body collision avoidance guided by footsteps and centroidal sway motion," in *International Conference on Humanoid Robots*, 2018, pp. 251–256.

[14] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *International Conference on Intelligent Robots and Systems*, 2017, pp. 1366–1373.

[15] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *International Conference on Intelligent Robots and Systems*, 2020, pp. 3664–3671.

[16] 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*, 2020, pp. 2464–2470.

[17] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter, "Collision-free mpc for legged robots in static and dynamic scenes," in *International Conference on Robotics and Automation*, 2021, pp. 8266–8272.

[18] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free mpc for whole-body dynamic locomotion and manipulation," in *International Conference on Robotics and Automation*, 2022, pp. 4686–4693.

[19] M. S. Motahar, S. Veer, and I. Poulakakis, "Composing limit cycles for motion planning of 3d bipedal walkers," in *Conference on decision and control*, 2016, pp. 6368–6374.

[20] S. Veer, M. S. Motahar, and I. Poulakakis, "Almost driftless navigation of 3d limit-cycle walking bipeds," in *International Conference on Intelligent Robots and Systems*, 2017, pp. 5025–5030.

[21] J. Norby, A. Tajbakhsh, Y. Yang, and A. M. Johnson, "Adaptive complexity model predictive control," *arXiv preprint arXiv:2209.02849*, 2022.

[22] Z. Li, J. Zeng, S. Chen, and K. Sreenath, "Vision-aided autonomous navigation of underactuated bipedal robots in height-constrained environments," *arXiv preprint arXiv:2109.05714*, vol. 2, 2021.

[23] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath, "Robotic guide dog: Leading a human with leash-guided hybrid physical interaction," in *International Conference on Robotics and Automation*, 2021, pp. 11 470–11 476.

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

[25] Z. Li, J. Zeng, A. Thirugnanam, and K. Sreenath, "Bridging Model-based Safety and Model-free Reinforcement Learning through System Identification of Low Dimensional Linear Models," in *Robotics: Science and Systems*, 2022.

[26] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.

[27] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and engineering*, vol. 3, no. 3, pp. 227–252, 2002.

[28] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *International Conference on Automation Science and Engineering*, 2021, pp. 2132–2139.

[29] C. Yang, G. N. Sue, Z. Li, L. Yang, H. Shen, Y. Chi, A. Rai, J. Zeng, and K. Sreenath, "Collaborative navigation and manipulation of a cable-towed load by multiple quadrupedal robots," *Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 041–10 048, 2022.

[30] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *American Control Conference*, 2021, pp. 3882–3889.

[31] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *Conference on Decision and Control*, 2021, pp. 6137–6144.

[32] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *International Conference on Robotics and Automation*, 2022, pp. 286–292.

[33] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[34] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[35] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, pp. 161–176, 2013.

[36] C. D. 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*, 2016, pp. 558–564.

[37] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *Robotics and Automation Letters*, pp. 4688–4695, 2021.

[38] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[39] O. A. open source library for optimal control of switched systems", "https://github.com/leggedrobotics/ocs2."

[40] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *International Symposium on System Integration*, 2019, pp. 614–619.

[41] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, pp. 327–363, 2014.

[42] "A1", "https://m.unitree.com/."

[43] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2149–2154.