IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Safety-Critical and Distributed Nonlinear Predictive Controllers for Teams of Quadrupedal Robots

Basit Muhammad Imran[1], Jeeseop Kim[2], Taizoon Chunawala[1], Alexander Leonessa[1], and Kaveh Akbari Hamed[1]

***Abstract*—This paper presents a novel hierarchical, safety-critical control framework that integrates distributed nonlinear model predictive controllers (DNMPCs) with control barrier functions (CBFs) to enable cooperative locomotion of multi-agent quadrupedal robots in complex environments. While NMPC-based methods are widely used to enforce safety constraints and navigate multi-robot systems (MRSs) through complex environments, trajectory optimization frameworks based on invariant sets offer formal safety guarantees for MRSs. CBFs, typically implemented via quadratic programs (QPs) at the planning layer, provide formal safety guarantees. However, their zero-control horizon limits their effectiveness for extended trajectory planning in inherently unstable, underactuated, and nonlinear legged robot models. Furthermore, the integration of CBFs into real-time NMPC for sophisticated MRSs, such as quadrupedal robot teams, remains underexplored. This paper develops computationally efficient, distributed NMPC algorithms that incorporate CBF-based collision safety guarantees within a consensus protocol, enabling longer planning horizons for safe cooperative locomotion under disturbances and rough terrain conditions. The optimal trajectories generated by the DNMPCs are tracked using full-order, nonlinear whole-body controllers at the low level. The proposed approach is validated through extensive numerical simulations with up to four Unitree A1 robots and hardware experiments involving two A1 robots subjected to external pushes, rough terrain, and uncertain obstacle information. Comparative results demonstrate that the proposed CBF-integrated DNMPC achieves a higher success rate than baseline NMPCs employing CBFs at the high or low-level layers.**

***Index Terms*—Legged robots, motion control, multi-contact whole-body motion planning and control**

I. Introduction

MULTI-ROBOT systems (MRSs), especially those consisting of legged robots, play a crucial role in mission-critical tasks due to their exceptional capability to traverse rough terrains. Applications include collaborative firefighting, unmanned rescue, disaster response, and exploration. Recent advancements in legged robotics have significantly enhanced these robots' agility and robustness, enabling them to navigate complex terrains more effectively, see, e.g., [1]–[3]. While multi-robot systems have been extensively studied (see, e.g., [4], [5]), MRSs involving legged robots pose unique challenges due to their high dimensionality, unilateral constraints, underactuation, and hybrid nature.

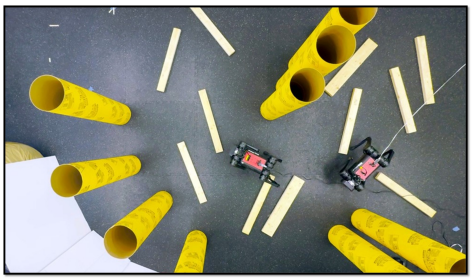

Fig. 1. Top-view snapshot of an experiment demonstrating CBF-based DNMPC algorithms, where two Unitree A1 robots navigate a challenging environment with uncertainties in obstacle positions and rough terrain.

Reduced-order models (i.e., templates) [6] provide low-dimensional representations of complex, nonlinear locomotion systems and can be effectively integrated with model predictive control (MPC) for real-time trajectory planning in legged robots. Common reduced-order models include the linear inverted pendulum (LIP) model [7] and its variants, such as angular momentum LIP [8], spring-loaded inverted pendulum (SLIP) [9], vertical SLIP [10], hybrid LIP [11], variable-height inverted pendulum model [12], centroidal dynamics [13], and the single rigid body (SRB) model [1], [14], [15]. For single-agent robots, MPC algorithms based on linearized templates reduce to convex quadratic programs (QPs), which can be efficiently solved in real-time [1], [16], [17]. However, these QP-based MPCs struggle to handle the nonlinear and non-convex constraints required for collision avoidance in MRSs, necessitating the use of nonlinear MPC (NMPC) [18]. NMPC has also been employed for gait planning in single-agent quadrupedal robots, e.g., [2], [19]–[21]. The previous work [18] explored distributed NMPC (DNMPC)-based planning for multi-agent quadrupedal robots, addressing nonconvex robot-to-robot and robot-to-obstacle collision avoidance using LIP models but without considering robot orientation. While NMPC can accommodate such constraints, a trajectory optimization framework based on control barrier functions (CBFs) [22], [23] provides formal guarantees for the safety of MRSs.

CBF-based controllers for MRSs are formulated as QP-based optimization problems that modify nominal controllers while enforcing barrier constraints [4], [5]. These QPs can be viewed as optimal control problems for input-affine systems with a zero-control horizon. To enable the safe and cooperative locomotion of two quadrupedal robots, [24] proposed a hierarchical framework where a high-level QP-based safety controller with CBFs generates reduced-order reference tra-

Manuscript received: February 27, 2025; Revised May 28, 2025; Accepted June 27, 2025. This paper was recommended for publication by Editor Cosimo Della Santina upon evaluation of the Associate Editor and Reviewers' comments. The work of B. Imran was supported by the National Science Foundation (NSF) under Grant 1924617. The work of K. Akbari Hamed is supported by the NSF under Grants 2024772 and 2423725.

[1]B. Imran, T. Chunawala, A. Leonessa, and K. Akbari Hamed are with the Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061, USA, {basit, taizoonc, aleoness, kavehakbarihamed}@vt.edu

[2]J. Kim is with California Institute of Technology, Pasadena, CA 91125, USA, jeeseop@caltech.edu

Digital Object Identifier (DOI): see the top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

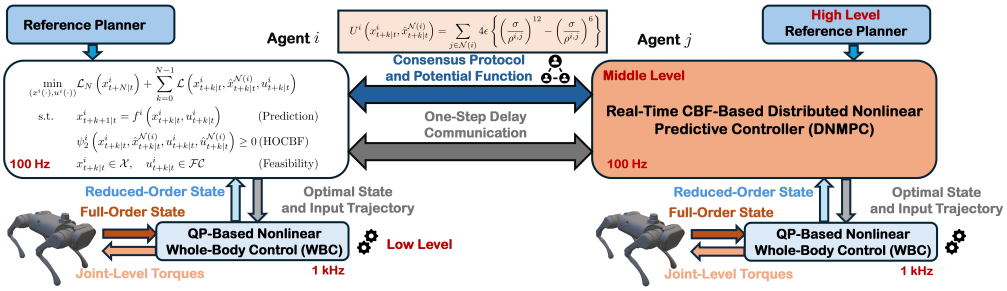

Fig. 2. Overview of the proposed control framework integrating CBF-based distributed NMPCs for safe locomotion of multi-agent quadrupedal robots.

jectories, which are then tracked by a low-level QP-based MPC for gait generation. Notably, CBFs and MPC operate in separate layers rather than within a unified framework. Recent studies [25], [26] have explored integrating CBFs within MPC and NMPC frameworks, combining the benefits of a nonzero prediction horizon with simple integrator dynamics. Additionally, [21] employed CBFs with duality-based obstacle avoidance constraints in an NMPC framework running at 30 Hz for a single-agent quadrupedal robot. Moreover, [20] implemented a low-frequency safe kinematic MPC with first-order CBFs for single-agent quadrupedal robots. However, safety guarantees within a distributed NMPC framework for complex multi-agent legged robots remain unexplored.

The *overarching goal* of this work is to develop computationally efficient, unified distributed NMPC algorithms that integrate CBF-based collision safety guarantees with a consensus protocol, thereby enabling extended planning horizons for the safe locomotion of multi-agent quadrupedal robots in the presence of disturbances and challenging terrain conditions.

A. Related Work

MPC implementations for MRSs are typically categorized as centralized, decentralized, or distributed. Centralized control yields optimal performance but becomes impractical for large-scale or communication-limited systems due to its high computational and communication demands [27]. Decentralized MPC assigns independent controllers to each agent without communication, lowering complexity but limiting performance in tightly coupled systems [28]. Distributed MPC (DMPC) strikes a balance by enabling limited inter-agent communication [29], and is further divided into cooperative and non-cooperative frameworks. Cooperative DMPC optimizes a shared global objective [30], [31], while non-cooperative DMPC focuses on local objectives. Algorithms have been developed for cases involving coupled constraints with decoupled dynamics [32] or requiring iterative information exchange [33]. Non-cooperative DMPC has become popular for collision avoidance with local costs, decoupled dynamics, and coupled constraints, and has been applied to autonomous vehicles [34], differential drive robots [35], and linear systems [36].

While the literature on DMPC is extensive, extending these approaches to distributed nonlinear MPC (DNMPC) for multi-agent quadrupedal robots presents significant *challenges* due to the inherent instability of locomotion models, underactuation, and unilateral constraints. Previous work [18] developed a DNMPC approach for collision-safe navigation in quadrupedal MRSs by directly incorporating Euclidean distance-based constraints into the NMPC framework of LIP models. While effective, this approach faces limitations in computational tractability, with the real-time NMPC loop constrained to just 5.5 Hz. In contrast, [24] proposed collision safety by imposing CBFs on the high-level path planner. However, applying CBFs at this level ensures safety only for the reference path given to the NMPC, not for the real-time NMPC itself.

B. Contributions

This paper aims to develop a unified, computationally efficient, and fast distributed control framework that integrates DNMPC and CBFs to enable the safe locomotion of multi-agent quadrupedal robots. The *contributions* of the paper are as follows. The paper introduces a hierarchical control framework comprising three layers (see Fig. 2). At the middle layer, we propose a real-time distributed NMPC algorithm operating at 100 Hz to address trajectory optimization of nonlinear SRB models of multi-agent quadrupedal robots. The DNMPC incorporates discrete-time CBF constraints to ensure robot-to-robot and robot-to-obstacle safety while leveraging a Lennard-Jones potential function in the cost formulation of NMPCs for consensus protocol. Reference trajectories for the DNMPC are generated offline by a high-level planner using potential fields combined with the $A^\star$ algorithm. The optimal trajectories computed by the middle-layer DNMPC are subsequently provided to low-level nonlinear whole-body controllers (WBCs) operating at 1 kHz, which impose the full-order model of each agent to track the optimal trajectories. The WBCs are developed based on QP and virtual constraints [37].

We validate the proposed formulation through experiments on two Unitree A1 robots and numerical simulations involving a team of four A1 robots navigating uncertain environments with obstacle location uncertainty, rough terrain, and external disturbances, as shown in Fig. 1. The results demonstrate a significant improvement in collision avoidance success rates, increasing from 66% in previous approaches [18] to 94% with our method. To further evaluate the efficacy of the unified CBF-based DNMPC approach with SRB models, we conduct numerical simulations on randomly generated terrains with randomly positioned obstacles, comparing it against DNMPC algorithms that apply CBF conditions separately at the low level or high level via QPs (e.g., [24]), as well as those using LIP models. The results demonstrate a notable improvement in performance with the unified CBF-based DNMPC approach.

The current work differs from prior studies [27], [38], which addressed cooperative locomotion of quadrupedal robots with holonomic constraints for payload transport using convex QP-based MPCs. In contrast, our approach employs NMPC and explicitly incorporates safety, which was not considered in

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

those works. It also differs from [39], which focuses on cable-towed load manipulation using a cascaded planning scheme that combines centralized and decentralized planners at 0.5 and 20 Hz, based on duality methods [40]. In contrast, our method leverages fully distributed CBF-based NMPCs operating at 100 Hz, rather than centralized MPC.

II. Problem Formulation

This section presents the problem formulation by addressing local safety sets for individual agents. We consider a network of n_A quadrupedal agents, indexed by the set $\mathcal{A} := \{1, \cdots, n_A\}$. Variables associated with each agent $i \in \mathcal{A}$ are denoted using the superscript i. The discrete-time nonlinear state equation for each agent $i \in \mathcal{A}$ is given by

$$x^i(t+1) = f^i\left(x^i(t), u^i(t)\right), \tag{1}$$

where $t \in \mathbb{Z}_{\geq 0} := \{0, 1, \cdots\}$ represents the discrete time, $x^i \in \mathcal{X}$ and $u^i \in \mathcal{U}$ denote the local state variables and local control inputs, respectively, and $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ represent the feasible state and the admissible control sets, for some positive integers n and m. The function $f^i : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is continuous. The global state and control input vectors are defined as $x := \text{col}\{x^i \,|\, i \in \mathcal{A}\} \in \mathbb{R}^{nn_A}$ and $u := \text{col}\{u^i \,|\, i \in \mathcal{A}\} \in \mathbb{R}^{mn_A}$, where "col" denotes the column operator.

Collision Safety in the MRS: Each quadrupedal agent navigates a complex environment with static obstacles while ensuring inter-robot and obstacle avoidance. Static obstacles are represented by the Cartesian coordinates of their center points, denoted as $o^\ell \in \mathbb{R}^2$ for $\ell \in \mathcal{O} := \{1, \cdots, n_O\}$, where n_O represents the number of obstacles. To address safety, we assume that the center of mass (COM) coordinates of each agent in the xy-plane are given by $g(x^i)$, where $g : \mathbb{R}^n \to 2$ is a continuous mapping. We next define two global safe sets as super-level sets of the Euclidean distance: $\mathcal{SA}$, capturing safe distances between all agents, and $\mathcal{SO}$, ensuring safe distances between agents and obstacles. More specifically, we define

$$\begin{aligned}\mathcal{SA} &:= \{x \in \mathbb{R}^{nn_A} \,|\, \|g(x^i) - g(x^j)\| \geq d_{\text{th}}, \forall i \neq j \in \mathcal{A}\}\\ \mathcal{SO} &:= \{x \in \mathbb{R}^{nn_A} \,|\, \|g(x^i) - o^\ell\| \geq d_{\text{th}}, \forall i \in \mathcal{A}, \forall \ell \in \mathcal{O}\},\end{aligned} \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean distance, and d_{th} is a threshold distance value. The overall *global safe set* $\mathcal{S}$ is then given by the intersection $\mathcal{S} := \mathcal{SA} \cap \mathcal{SO}$.

Ensuring system-wide safety can be formulated as maintaining the forward invariance of $\mathcal{S}$ [22]. However, designing a centralized NMPC law to enforce this invariance is computationally demanding. To mitigate this, we propose local NMPC laws that ensure the safety of individual agents by defining *local safety sets*. Specifically, for each agent $i \in \mathcal{A}$, two local safe sets can be defined accordingly: $\mathcal{SA}^i$ for capturing safe distances between agent i and any other agent $j \neq i \in \mathcal{A}$, and $\mathcal{SO}^i$ for ensuring safe distances between agent i and obstacles o^ℓ for all $\ell \in \mathcal{O}$, that is,

$$\mathcal{SA}^i := \bigcap_{j \neq i \in \mathcal{A}} \left\{x^i \in \mathcal{X} \,\middle|\, \|g(x^i) - g(x^j)\| \geq d_{\text{th}}\right\} \tag{3}$$

$$\mathcal{SO}^i := \bigcap_{\ell \in \mathcal{O}} \left\{x^i \in \mathcal{X} \,\middle|\, \|g(x^i) - o^\ell\| \geq d_{\text{th}}\right\}. \tag{4}$$

The combined local safe set $\mathcal{S}^i$ is then obtained by the intersection of these two sets as $\mathcal{S}^i := \mathcal{SA}^i \cap \mathcal{SO}^i$.

The local safe set can be reformulated as

$$\mathcal{S}^i = \left\{x^i \in \mathcal{X} \,\middle|\, h^i(x^i, x^{\mathcal{N}(i)}, o) \geq 0\right\}, \tag{5}$$

where $\mathcal{N}(i) := \{j \in \mathcal{A} \,|\, j \neq i\}$ denotes the set of all other agents, $x^{\mathcal{N}(i)} := \text{col}\{x^j \,|\, j \in \mathcal{N}(i)\} \in \mathbb{R}^{n(n_A-1)}$ represents the states of all other agents, and $o := \text{col}\{o^\ell \,|\, \ell \in \mathcal{O}\} \in \mathbb{R}^{2n_O}$ contains the coordinates of all obstacles. In addition, h^i is a continuous local function that can be expressed as follows:

$$h^i(x^i, x^{\mathcal{N}(i)}, o) := \begin{bmatrix}\text{col}\left\{h^{i,j} \,|\, j \in \mathcal{N}(i)\right\}\\ \text{col}\left\{h^{i,\ell} \,|\, \ell \in \mathcal{O}\right\}\end{bmatrix}, \tag{6}$$

where $h^{i,j} := \|g(x^i) - g(x^j)\| - d_{\text{th}}$ for all $j \in \mathcal{N}(i)$ and $h^{i,\ell} := \|g(x^i) - o^\ell\| - d_{\text{th}}$ for all $\ell \in \mathcal{O}$. Since h^i depends on the local state x^i as well as the states of other agents $x^{\mathcal{N}(i)}$, the local safety set $\mathcal{S}^i$ is parameterized by the states of all other agents at every time step t, i.e., $\mathcal{S}^i = \mathcal{S}^i(x^{\mathcal{N}(i)})$.

Problem Statement: We aim to design computationally efficient, distributed NMPC laws that compute the optimal local control law $u^i \in \mathcal{U}$ for each agent with a consensus protocol while (1) ensuring the invariance of its local safe set $\mathcal{S}^i$, and (2) maintaining a consistent "sticking distance" between the agents. To account for the dependence of local safety sets on other agents' states, we will adopt a communication protocol in which agents share their optimal state and input trajectories—computed by their local NMPCs—with a one-step delay.

Before establishing the proposed DNMPC framework, we present definitions that introduce the concept of higher-order, discrete-time CBFs for each agent. For notational simplicity, we will use x_t^i and u_t^i to denote $x^i(t)$ and $u^i(t)$, respectively.

Definition 1 *(Discrete-Time CBF [41])*: The local function h^i is said to be a CBF for (1) if there exists class $\mathcal{K}$ function α satisfying $\alpha(s) < s$ for all $s > 0$ such that

$$\Delta h^i\left(x^i(t), u^i(t)\right) \geq -\alpha\left(h^i(x^i(t))\right), \quad \forall x^i(t) \in \mathcal{X}, \tag{7}$$

where $\Delta h^i(x^i(t), u^i(t)) := h^i(x^i(t+1)) - h^i(x^i(t))$ and s is the argument of α.

For systems with a higher relative degree $r > 1$, the control input does not appear in the first-order difference of the CBF h^i. In this case, a series of functions can be defined based on the original $h^i(x_t^i)$ with the relative degree r as follows:

$$\begin{aligned}\psi_0^i(x_t^i) &:= h^i(x_t^i)\\ \psi_1^i(x_t^i) &:= \Delta\psi_0^i\left(x_t^i, u_t^i\right) + \alpha_1\left(\psi_0^i(x_t^i)\right)\\ &\vdots\\ \psi_r^i(x_t^i, u_t^i) &:= \Delta\psi_{r-1}^i\left(x_t^i, u_t^i\right) + \alpha_r\left(\psi_{r-1}^i(x_t^i)\right),\end{aligned} \tag{8}$$

where $\Delta\psi_a^i(x_t^i, u_t^i) := \psi_a^i(x_{t+1}^i) - \psi_a^i(x_t^i) = \psi_a^i(f^i(x_t^i, u_t^i)) - \psi_a^i(x_t^i)$ for $0 \leq a \leq r-1$ and α_a are class $\mathcal{K}$ functions satisfying $\alpha_a(s) < s$ for all $a = 1, \cdots, r$ and $s > 0$. This series of functions yields a corresponding series of sets, as

$$\mathcal{S}_a^i := \left\{x_t^i \in \mathcal{X} \,\middle|\, \psi_a^i(x_t^i) \geq 0\right\}, \quad 0 \leq a \leq r-1. \tag{9}$$

Definition 2 *(Higher-Order Discrete-Time CBF [41])*: The local function h^i is a higher-order, discrete-time CBF (HOCBF)

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

of relative degree r if there exist functions ψ_a^i for $a \in \{0,1,\dots,r\}$ defined by (8) and corresponding sets $\mathcal{S}_a^i$ for $a \in \{0,1,\dots,r-1\}$ defined by (9) such that

$$\psi_r^i(x_t^i, u_t^i) \geq 0 \tag{10}$$

for all $x_t^i \in \bigcap_{a=0}^{r-1} \mathcal{S}_a^i$ and for some control input $u_t^i \in \mathcal{U}$.

The following theorem formally establishes a sufficient condition on the local control law u^i to ensure that the intersection set $\bigcap_{a=0}^{r-1} \mathcal{S}_a^i$ remains invariant for local dynamics.

Theorem 1: *(HOCBF Condition [41]):* If h^i is a continuous HOCBF of relative degree r defined on $\bigcap_{a=0}^{r-1} \mathcal{S}_a^i$, any control input $u^i(t) \in \mathcal{U}$ satisfying the HOCBF condition (10) will render $\bigcap_{a=0}^{r-1} \mathcal{S}_a^i$ forward invariant for agent $i \in \mathcal{A}$.

III. Distributed NMPCs based on HOCBFs

This section proposes the middle-level distributed NMPC framework based on HOCBFs for real-time trajectory planning to enable the safe locomotion of a team of multi-agent quadrupedal robots in environments with obstacles.

Template Model: The SRB dynamics of each agent are used as the template model, with the state variables defined as the COM position and orientation, along with their time derivatives, that is, $x^i := \mathrm{col}(p^i, \dot{p}^i, \Theta^i, \omega^i) \in \mathbb{R}^n$ with $n = 12$, where $p^i \in \mathbb{R}^3$ represents the Cartesian coordinates of the COM of agent $i \in \mathcal{A}$, $\Theta^i \in \mathbb{R}^3$ denotes the Euler angles (roll, pitch, and yaw) of the body, and $\omega^i \in \mathbb{R}^3$ represents the angular velocities in the body frame. The control inputs $u^i(t)$ are further taken as the ground reaction forces (GRFs) that generate the net force and torque around the COM, denoted by $(f^{i,\text{net}}, \tau^{i,\text{net}})$ and expressed in the world frame. We adopt SRB dynamics over the simpler LIP model to allow quadrupedal agents to dynamically adjust their orientation—especially yaw—when navigating obstacles. Unlike quasi-static gaits with center of pressure constraints, SRB dynamics support dynamic gait generation. Their integration into the DNMPC framework enables real-time optimization of both position and orientation, enhancing feasibility and agility in complex environments. This is demonstrated through numerical studies in Section V-C and Table II. The equations of motion with the SRB model can be described by

$$\Sigma^i : \begin{cases} \ddot{p}^i = \frac{f^{i,\text{net}}}{m_{\text{tot}}} - g_0 \\ \dot{\Theta}^i = A(\Theta^i)\,\omega^i \\ \dot{\omega}^i = I^{-1}\left(R^{i\top}\tau^{i,\text{net}} - \mathbb{S}(\omega^i)\,I\,\omega^i\right), \end{cases} \tag{11}$$

where m_{tot} is the total mass of the agent, $I \in \mathbb{R}^{3\times3}$ represents the moment of the inertia in the body frame, $g_0 \in \mathbb{R}^3$ denotes the gravitational constant, $R^i \in \mathrm{SO}(3)$ represents the orientation matrix of the body frame relative to the world frame, and $\mathbb{S}(\cdot) : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric operator. The net force and torque around the COM are then calculated as a linear combination of the individual GRFs acting at each contacting leg. Specifically, a time-varying matrix $E(t)$ maps the vector of individual GRFs to the net wrench as follows:

$$\begin{bmatrix} f^{i,\text{net}} \\ \tau^{i,\text{net}} \end{bmatrix} = E(t)\,u^i(t). \tag{12}$$

The nonlinear SRB dynamics in (11) can be discretized and expressed in the form of (1) by the Euler method. Furthermore, the model remains valid if the GRFs (i.e., inputs $u^i(t)$) belong to the friction cone, denoted by $\mathcal{U} = \mathcal{FC}$.

Collision Safety for Quadrupedal Robots: From Definition 2, local functions h^i in (6) serve as continuous HOCBF candidates with relative degree $r = 2$ if there exist a series of functions ψ_1^i, ψ_2^i, and corresponding sets $\mathcal{S}_0^i, \mathcal{S}_1^i$ such that

$$\psi_2^i\left(x_t^i, x_t^{\mathcal{N}(i)}, u_t^i, u_t^{\mathcal{N}(i)}\right) \geq 0, \quad \forall x_t^i \in \mathcal{S}_0^i \cap \mathcal{S}_1^i. \tag{13}$$

We remark that ψ_2^i generally depends on the local state and input variables (x_t^i, u_t^i), as well as the variables of all other agents, i.e., $(x_t^{\mathcal{N}(i)}, u_t^{\mathcal{N}(i)})$. Furthermore, from Theorem 1, any feasible local control input (i.e., GRFs) $u^i(t) \in \mathcal{U}$ satisfying the HOCBF condition (13) will render $\mathcal{S}_0^i \cap \mathcal{S}_1^i$ invariant.

HOCBF-based DNMPCs: To address the problem in Section II, we propose the following DNMPC formulation for each agent $i \in \mathcal{A}$:

$$\begin{aligned} \min_{(x^i(\cdot),u^i(\cdot))} \; & \mathcal{L}_N\left(x_{t+N|t}^i\right) + \sum_{k=0}^{N-1} \mathcal{L}\left(x_{t+k|t}^i, \hat{x}_{t+k|t}^{\mathcal{N}(i)}, u_{t+k|t}^i\right) \\ \text{s.t.} \quad & x_{t+k+1|t}^i = f^i\left(x_{t+k|t}^i, u_{t+k|t}^i\right) && \text{(Prediction)} \\ & \psi_2^i\left(x_{t+k|t}^i, \hat{x}_{t+k|t}^{\mathcal{N}(i)}, u_{t+k|t}^i, \hat{u}_{t+k|t}^{\mathcal{N}(i)}\right) \geq 0 && \text{(HOCBF)} \\ & x_{t+k|t}^i \in \mathcal{X}, \quad u_{t+k|t}^i \in \mathcal{FC} && \text{(Feasibility)}, \end{aligned} \tag{14}$$

where N is the control horizon, $x_{t+k|t}^i$ and $u_{t+k|t}^i$ represent the predicted local states and local inputs at time $t+k$, computed at time t using the prediction model with the initial condition of $x_{t|t}^i = x^i(t)$. The terms $\hat{x}_{t+k|t}^{\mathcal{N}(i)}$ and $\hat{u}_{t+k|t}^{\mathcal{N}(i)}$ denote the *estimation* of predicted variables of all other agents, obtained using the *one-step delay communication protocol (OSDCP)*, which will be described later. The inequality constraints in the local NMPC (14) represent the HOCBF condition (13) together with the feasibility conditions for the states and inputs (i.e., GRFs). The local NMPC optimizes the state and input trajectories of agent i, denoted by $(x^i(\cdot), u^i(\cdot))$.

Consensus Protocol: The cost function of the local NMPC consists of the terminal and stage costs. The terminal cost is taken as $\mathcal{L}_N(x_{t+N|t}^i) := \|x_{t+N|t}^i - x_{t+N|t}^{i,\text{ref}}\|_P^2$, where $x^{i,\text{ref}}(\cdot)$ denotes the local reference trajectory generated by the high-level planner (see Section IV), and $P = P^\top$ is a positive definite matrix. The stage cost is defined as

$$\begin{aligned} \mathcal{L}\left(x_{t+k|t}^i, \hat{x}_{t+k|t}^{\mathcal{N}(i)}, u_{t+k|t}^i\right) := & \|x_{t+k|t}^i - x_{t+k|t}^{i,\text{ref}}\|_Q^2 + \|u_{t+k|t}^i\|_R^2 \\ & + w\,U^i\left(x_{t+k|t}^i, \hat{x}_{t+k|t}^{\mathcal{N}(i)}\right), \end{aligned} \tag{15}$$

where $Q = Q^\top$ and $R = R^\top$ are positive definite matrices, $w > 0$ is a weighting factor, and U^i is an artificial potential function for consensus. The potential function U^i, inspired by Lennard-Jones potential model [42], is given by

$$U^i\left(x_{t+k|t}^i, \hat{x}_{t+k|t}^{\mathcal{N}(i)}\right) := \sum_{j\in\mathcal{N}(i)} 4\epsilon\left\{\left(\frac{\sigma}{\rho^{i,j}}\right)^{12} - \left(\frac{\sigma}{\rho^{i,j}}\right)^{6}\right\}, \tag{16}$$

where $\rho^{i,j} := \|p_{\text{x,y}}^i - \hat{p}_{\text{x,y}}^j\|$ denotes the estimated distance between the COMs of two agents i and $j \neq i$, $\hat{p}_{\text{x,y}}^j$ represents the estimate of the COM position of agent j, and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ϵ and σ are tuning parameters. Incorporating the potential function U^i acts as a *consensus protocol*, enabling agents to maintain proximity through van der Waals-like interaction forces. The parameter σ defines the zero-potential distance and is tuned to set the equilibrium at which the interaction force vanishes. Including this term in the local NMPC cost encourages agents to maintain a consistent "sticking distance," effectively emulating soft-holonomic constraints on their COM separation. In the experiments, we show that this consensus protocol enables soft-flocking behavior—referring to non-strict coordination where agents may temporarily deviate from the sticking distance to satisfy hard safety constraints enforced by the HOCBF-DNMPC framework.

OSDCP Protocol: The one-step delay communication protocol (OSDCP) is proposed to address the inherent challenge of solution unavailability from other agents in (14). Specifically, due to the distributed nature of the formulation, the optimal solution $(x^{j,\star}, u^{j,\star})$, of agent j, computed at time t, is not immediately accessible by agent i. Consequently, the condition in (13) cannot be directly applied. To facilitate implementation, the OSDCP leverages the optimal solutions computed by other agents at the preceding time step $(t-1)$. With a sufficiently small time step T_s, agent i approximates the predicted states and inputs of all other agents $j \in \mathcal{N}(i)$ as $\hat{x}^j_{k+t|t} \approx x^{j,\star}_{k+t|t-1}$ and $\hat{u}^j_{k+t|t} \approx u^{j,\star}_{k+t|t-1}$, where $(x^{j,\star}_{k+t|t-1}, u^{j,\star}_{k+t|t-1})$ denotes the optimal predicted variables of agent j from the previous time step $(t-1)$.

IV. High-Level and Low-Level Layers

This section presents a concise overview of the high- and low-level layers of the control scheme, adapted from the previous works [18], [43], with modifications.

High-Level Path Planner: The proposed hierarchical control framework incorporates an offline high-level path planner (see Fig. 2) that generates reference trajectories for the middle-layer DNMPC algorithms. To achieve this, we employ a hybrid path planning approach that combines the $A^\star$ algorithm [44] with the potential fields method [45]. The previous work [18] relied solely on a potential fields-based path planner. However, it is well-known that such planners are prone to local minima issues. Similar to the work in [46], we adopt the $A^\star$ algorithm to generate globally optimal way-points, which are then bridged by locally smooth and dynamics-informed reference paths using potential fields. The reference trajectories are then passed to the middle-layer DNMPC algorithm.

Low-Level Nonlinear WBC: At the low level, we use nonlinear WBCs to impose the full-order dynamical model of each quadrupedal robot for tracking the optimal reduced-order state and GRF trajectories prescribed by the middle-layer DNMPCs. The WBC, adopted from the previous work [43], is formulated as a real-time QP running at 1 kHz. It incorporates the tracking problem within the framework of virtual constraints [37] for the floating-base full-order model while solving for dynamically feasible joint-level torques.

V. Experiments

This section presents the results of our numerical simulations and hardware experiments, thoroughly assessing the proposed distributed control algorithm's effectiveness.

A. Setup and Distributed Controller Synthesis

This work utilizes A1 quadrupedal robots as a test platform for the numerical and experimental validation of the proposed framework. The A1 robots are compact quadrupeds with a mass of 12.45 (kg), height of 0.28 (m), and 18 degrees of freedom (DOFs), 12 of which are actuated through three active joints per leg: hip pitch, hip roll, and knee pitch. The robots are equipped with an NVIDIA Jetson TX2, an RP-LiDAR S1 operating at 15 Hz for point-cloud generation and obstacle detection, and an onboard computer with an Ethernet switch. The distributed algorithms are executed on multi-threads of an offboard desktop PC with an AMD Ryzen 7950X3D CPU and 64 GB of DDR5 RAM. For real-time obstacle detection, we employ the obstacle detector ROS package [47]. To estimate translational motion, each robot uses a kinematic estimator for its COM. Additionally, the position of the other robot is communicated via the OSDCP protocol. Numerical simulations are conducted in the RaiSim physics engine [48], employing four A1 robots in the simulation environment, while two are deployed for hardware evaluations, as shown in Fig. 3. Videos are available in the supplementary file.

The parameters of the CBF-based DNMPCs are set as $Q = \text{block diag}\{Q_p, Q_{\dot{p}}, Q_\Theta, Q_\omega\}$ with $Q_p = \text{diag}\{1e5, 1e5, 8e6\}$, $Q_{\dot{p}} = \text{diag}\{5e5, 5e5, 8e6\}$, $Q_\Theta = \text{diag}\{1e4, 1e4, 1e4\}$, and $Q_\omega = \text{diag}\{1e4, 1e4, 1e4\}$, $P = 100\,Q$ and $R = \mathbb{I}$. The control horizon for the local NMPCs is chosen as $N = 10$ with the sampling time of $T_s = 10$ (ms). Additional hyperparameters include a collision avoidance threshold distance of $d_{\text{th}} = 0.6$ (m), class $\mathcal{K}$ functions $\alpha_1(s) = 0.1\,s$ and $\alpha_2(s) = 0.05\,s$ to satisfy the conditions in Definition 2, the tuning parameters of the Lennard-Jones potential in (16) as $\epsilon = 50$, $\sigma = 0.85$ (m), and a weighting factor of $w = 1e9$ in (15). We choose trotting gaits of quadrupedal robots with a step time of 180 (ms) and the local nonlinear programs (NLPs) in (14) include 240 decision variables. The local real-time DNMPC formulation in (14) is nonconvex and solved in real time without any approximations in the system dynamics or HOCBF constraints. The real-time distributed NMPCs are solved using CasADi [49] framework along with the IPOPT solver with 8 iterations, utilizing the previous solution as the initial guess. We evaluate the framework in a complex environment with up to $n_O = 20$ obstacles. As the number of agents and obstacles increases in experiments, the continuous function h^i in (5) can be reformulated as

$$h^i := \min\{\min_{j \in \mathcal{N}(i)} h^{i,j}, \min_{\ell \in \mathcal{O}} h^{i,\ell}\}, \tag{17}$$

which significantly reduces the number of inequality constraints in the NLPs derived from the HOCBF. Table I summarizes the statistics for the distribution of the NMPC computation times (in milliseconds) for two challenging scenarios involving rough terrains and external disturbances.

B. Hardware Experiments and Numerical Simulations

Modifications of trajectories by DNMPC: In the first experiment, the robots were commanded to follow intersecting trajectories to assess collision avoidance on rough terrain. This setup specifically evaluated the performance of middle-level

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

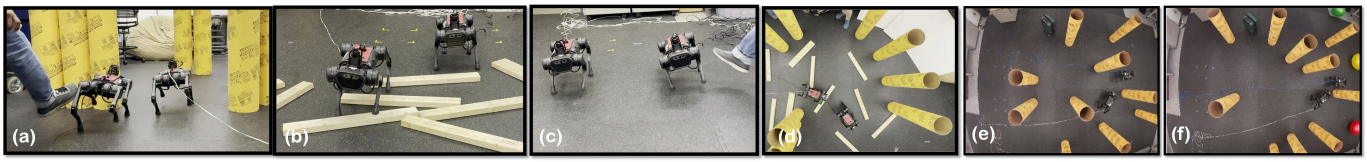

Fig. 3. Snapshots of experiments demonstrating the deployment of CBF-based DNMPCs in a multi-agent A1 setup under various conditions: (a) uncertain obstacles, (b) intentional intersecting reference trajectories on rough terrain, (c) straight-line reference trajectories with one agent physically pushed onto another, (d) cooperative locomotion on rough terrain with uncertain obstacles, and (e) and (f) straight-line reference trajectories with different obstacle configurations.

TABLE I
STATISTICS FOR THE DISTRIBUTION OF DNMPC SOLVE TIMES (MS)

Experiment	min	max	avg	standard deviation
Rough terrain (Fig. 3(d))	4	10	5.27	1.03
External push (Fig. 3(c))	5	9	6.92	0.47

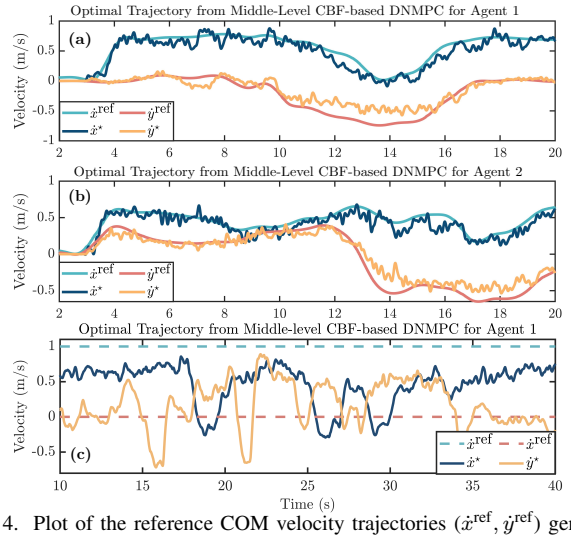

Fig. 4. Plot of the reference COM velocity trajectories ($\dot{x}^{\text{ref}}, \dot{y}^{\text{ref}}$) generated by the high-level planner along with the optimal velocity trajectories ($\dot{x}^{\star}, \dot{y}^{\star}$) adjusted by the middle-level DNMPCs for agents 1 and 2 in (a) and (b) during experiments of Fig. 3(d) and for agent 1 in (c) during experiment of Fig. 3(e).

DNMPCs when handling colliding reference paths. Instead of relying on a high-level planner, both robots were provided with x and y velocity references, leading to an intentional collision scenario. Additionally, wooden blocks were used to simulate rough terrain with a maximum height of 5 (cm) (17.85% of the robots' standing height). Under these conditions, the CBF-based DNMPC algorithm successfully adjusted the reference commands to prevent collisions, as shown in Fig. 3(b).

Safety with disturbances (external push): The second experiment aimed to evaluate the effectiveness of the distributed control framework by applying an external push to one agent and recording its response. Both agents were assigned straight-line forward reference trajectories without lateral deviations. During execution, one agent was physically perturbed to simulate an unexpected collision event. The consensus protocol, combined with the CBFs in the middle-level DNMPC, enabled the unperturbed agent to react promptly by moving away from the collision zone, maintaining a constant separation distance between the agents, as shown in Fig. 3(c).

Robustness against uncertainty and rough terrain: The third scenario was designed to assess the robustness of CBF-based DNMPCs in environments with uncertain obstacle information (see Fig. 3(d)). Initially, the high-level planner operated under the assumption of known obstacle locations. However, during execution, slight perturbations were deliberately introduced to the obstacle positions, creating a mismatch between the assumed and actual environment. The high-level planner first generated the desired trajectories, after which the middle-level DNMPCs were activated to account for the perturbed obstacles. Additionally, wooden blocks were introduced to evaluate the systems robustness under both rough terrain and obstacle uncertainty. Figure 4 illustrates the velocity trajectories prescribed by the high-level planner and the corresponding trajectories generated by DNMPCs for agents 1 and 2.

Different Obstacle Configurations: We conducted eight additional experiments to evaluate the effectiveness of the proposed algorithm across various obstacle configuration scenarios involving more than ten obstacles, in which two agents were commanded to walk forward. The agents navigated the environment safely and robustly in seven out of the eight trials, exhibiting non-conservative behaviors when interacting with obstacles or the other agent. Two representative snapshots from these experiments are shown in Figs. 3(e) and 3(f), and the corresponding velocity trajectories generated by the DNMPC for Agent 1 are presented in Fig. 4.

Consensus protocol: The efficacy of the consensus protocol in (16) and (15) was empirically evaluated through a controlled experiment involving two A1 robots navigating nine uncertain obstacles over wooden blocks. Two experimental configurations were tested: one with the consensus protocol deactivated ($w = 0$) and another with a high-gain activation ($w = 1e9$). Figure 5 demonstrates that activating the protocol improved soft-flocking behavior and enhanced inter-agent cohesion. Figure 5(c) presents the actual cost of the consensus term for both cases, offering insight into the degree of coordination under different protocol settings. Figure 5(d) depicts the inter-agent distance over time, revealing deviations from the nominal 1 (m) separation when CBF constraints are active ($t < 6$ (s)), indicating collision avoidance maneuvers. Notably, when the CBF constraints are inactive ($t > 6$ (s)), the enabled consensus protocol maintains a more stable inter-agent distance.

Network Assumption and Robustness to Latency: We assume a fully connected communication network, where each agent exchanges information with all others via a one-step delay protocol. The distributed algorithms are executed in a parallel and decentralized manner on multiple threads of a desktop PC. To assess robustness against deviations from this assumption, we introduce uniform communication latency in the local mid-level NMPCs, simulating delays caused by control clock synchronization when deploying CBF-based DNMPCs on separate machines. In a four-agent simulation involving wooden blocks and 20 randomly placed obstacles, results show that the algorithm maintains stability and avoids both inter-agent and agent-obstacle collisions with latencies up to 6 (ms). A simulation demonstrating performance under nonuniform communication latency is provided in the supple-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

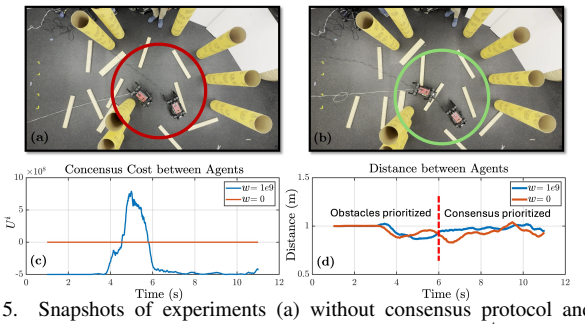

Fig. 5. Snapshots of experiments (a) without consensus protocol and (b) with consensus protocol. (c) Plots for the consensus cost wU^i for $w = 0$ (no consensus) and $w = 10^9$ (with consensus). (d) Distance between the two A1 agents for $w = 0$ and $w = 10^9$.

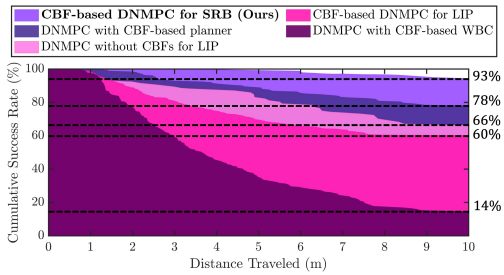

Fig. 6. Success rate comparison of the proposed CBF-based DNMPC with SRB dynamics against four baseline approaches versus the traveled distance.

TABLE II
COMPARISON OF THE PROPOSED APPROACH WITH EXISTING TECHNIQUES.

Approach	Overall Success Rate
CBF-based DNMPC for SRB (Ours)	**93.89%**
DNMPC with CBF-based planner [24]	78%
DNMPC without CBFs for LIP [18]	66%
CBF-based DNMPC for LIP	60%
DNMPC with CBF-based WBC	14.5%

mentary multimedia file. Safety constraints are implemented using the compact formulation in (17), which significantly reduces the number of inequality constraints in the NLPs. This allows solving DNMPC problems with four agents and 20 obstacles in under 10 (ms). As the number of agents and obstacles increases, communication latency may become more significant due to decentralized execution. Future work will investigate the scalability of the proposed framework.

C. Comparison and Quantitative Analysis

We conduct extensive numerical simulations to evaluate the efficacy and robustness of the proposed CBF-based DNMPC algorithms against other distributed predictive control strategies. The evaluation focuses on cooperative and safe locomotion of two A1 quadrupedal agents trotting across 200 randomly generated terrains with 20 randomly positioned obstacles. For comparison, we employ five different mid-level distributed controllers. The first implements the proposed CBF-based DNMPC with SRB models at 100 Hz. The second uses the same algorithm with LIP models, also at 100 Hz. The third applies DNMPC without CBFs, adapted from [18], using LIP dynamics at 5.5 Hz; in this case, safety is enforced via Euclidean distance constraints rather than embedded CBF conditions. The fourth uses DNMPC with SRB dynamics but no CBFs at the NMPC level, enforcing CBF constraints instead at the low-level WBCs, similar to typical single-agent strategies (see, e.g., [2], [3], [50]). These four controllers share the same high-level planning layer. The fifth uses DNMPC with SRB dynamics but without CBFs, where safety is addressed at the high level using a zero-horizon convex QP applied to simple integrator dynamics, as in [24]. Figure 6 shows the success rate of each approach for Agent 1 as a function of traveled distance, where success is defined as both agents reaching 10 (m) without collisions while maintaining gait stability. Table II summarizes the success rates, showing that the proposed CBF-based DNMPC with SRB dynamics significantly outperforms the other approaches.

VI. CONCLUSIONS

This paper presented a unified, computationally efficient, and fast distributed control framework that integrates DNMPC and CBFs to enable safe locomotion for multi-agent quadrupedal robots operating under external disturbances, rough terrain, and uncertain obstacle locations. The proposed DNMPC incorporates discrete-time CBF constraints for nonlinear SRB locomotion models to ensure both robot-to-robot and robot-to-obstacle safety. Additionally, a Lennard-Jones potential function is embedded in the NMPC cost formulation to facilitate consensus among agents. We validated the proposed approach through extensive hardware experiments on two A1 quadrupedal robots and numerical simulations involving a team of four A1 robots navigating uncertain environments with obstacle location uncertainty, rough terrain, and external disturbances. Our numerical results demonstrate a significant improvement in collision avoidance success rates compared to alternative DNMPC algorithms that incorporate CBFs at the high level or the WBC level, or that rely on LIP models.

The proposed DNMPC formulation can be integrated with alternative collision avoidance strategies, such as those based on signed distance fields (SDFs). As part of future work, we plan to incorporate SDFs into the framework—either as soft constraints within the local cost functions or as hard constraints through inequality formulations. The current work utilized the $A^\star$ algorithm and 2D LiDAR at the high level to generate reference trajectories for the network of CBF-based DNMPCs, enabling collision avoidance in the 2D plane. In future work, we will explore the integration of 3D LiDAR to allow multi-agent quadrupedal robots to dynamically adjust their height when encountering complex 3D obstacles.

REFERENCES

[1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–9.

[2] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[3] C. Khazoom, S. Hong, M. Chignoli, E. Stanger-Jones, and S. Kim, "Tailoring solution accuracy for fast whole-body model predictive control of legged robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 074–11 081, 2024.

[4] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[5] M. Cavorsi, L. Sabattini, and S. Gil, "Multirobot adversarial resilience using control barrier functions," *IEEE Transactions on Robotics*, vol. 40, pp. 797–815, 2024.

[6] R. Full and D. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

[7] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1406.

[8] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-adaptive, ALIP-based bipedal locomotion controller via model predictive control and virtual constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6724–6731.

[9] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behavior explains basic dynamics of walking and running," *Proceedings. Biological sciences / The Royal Society*, vol. 273, pp. 2861–7, 08 2006.

[10] Z. Li, J. Zeng, S. Chen, and K. Sreenath, "Autonomous navigation of underactuated bipedal robots in height-constrained environments," *The International Journal of Robotics Research*, vol. 42, no. 8, pp. 565–585, 2023.

[11] X. Xiong and A. Ames, "3-D underactuated bipedal walking via H-LIP based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022.

[12] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: Limitations imposed by unilateral contact," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 8–15.

[13] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[14] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49 785–49 797, 2020.

[15] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[16] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. Akbari Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6622–6629, 2022.

[17] L. Amanzadeh, T. Chunawala, R. T. Fawcett, A. Leonessa, and K. Akbari Hamed, "Predictive control with indirect adaptive laws for payload transportation by quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 359–10 366, 2024.

[18] B. M. Imran, R. T. Fawcett, J. Kim, A. Leonessa, and K. Akbari Hamed, "A Distributed Layered Planning and Control Algorithm for Teams of Quadrupedal Robots: An Obstacle-Aware Nonlinear MPC Approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 147, no. 3, 2025.

[19] S. Hong, J.-H. Kim, and H.-W. Park, "Real-time constrained nonlinear model predictive control on SO(3) for dynamic legged locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3982–3989.

[20] 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 *2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 8352–8358.

[21] Q. Liao, Z. Li, A. Thirugnanam, J. Zeng, and K. Sreenath, "Walking in narrow spaces: Safety-critical locomotion control for quadrupedal robots with duality-based optimization," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2723–2730.

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

[23] A. Thirugnanam, J. Zeng, and K. Sreenath, "Nonsmooth control barrier functions for obstacle avoidance between convex regions," *arXiv:2306.13259*, 2023.

[24] J. Kim, J. Lee, and A. D. Ames, "Safety-critical coordination for cooperative legged locomotion via control barrier functions," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2368–2375.

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

[26] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6137–6144.

[27] J. Kim, R. T. Fawcett, V. R. Kamidi, A. D. Ames, and K. Akbari Hamed, "Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4728–4748, 2023.

[28] D. Siljak, *Decentralized Control of Complex Systems*. Dover Publications, December 2011.

[29] R. Scattolini, "Architectures for distributed and hierarchical model predictive control—A review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[30] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the IEEE Conference on Decision and Control*, 2005, pp. 6680–6685.

[31] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," *Journal of process control*, vol. 18, no. 9, pp. 839–845, 2008.

[32] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of control*, vol. 80, no. 9, pp. 1517–1531, 2007.

[33] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE control systems magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[34] M. Abdelaal and S. Schön, "Distributed nonlinear model predictive control for connected vehicles trajectory tracking and collision avoidance with ellipse geometry," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 2100–2111.

[35] R. Mao and H. Dai, "Distributed non-convex model predictive control for non-cooperative collision avoidance of networked differential drive mobile robots," *IEEE Access*, vol. 10, pp. 52 674–52 685, 2022.

[36] M. Farina and R. Scattolini, "Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," *Automatica*, vol. 48, no. 6, pp. 1088–1096, 2012.

[37] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC, 2007.

[38] R. T. Fawcett, L. Amanzadeh, J. Kim, A. D. Ames, and K. Akbari Hamed, "Distributed data-driven predictive control for multi-agent collaborative legged locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9924–9930.

[39] 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," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 041–10 048, 2022.

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

[41] Y. Xiong, D.-H. Zhai, M. Tavakoli, and Y. Xia, "Discrete-time control barrier function: High-order case and adaptive case," *IEEE Transactions on Cybernetics*, vol. 53, no. 5, pp. 3231–3239, 2023.

[42] J. E. Jones, "On the determination of molecular fields. i. from the variation of the viscosity of a gas with temperature," *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 106, pp. 441–462, 1924.

[43] R. T. Fawcett, A. Pandala, A. D. Ames, and K. Akbari Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via QP-based virtual constraint controllers," *IEEE Control Systems Letters*, pp. 1736–1741, 2021.

[44] O. O. Martins, A. A. Adekunle, O. M. Olaniyan, and B. O. Bolaji, "An improved multi-objective a-star algorithm for path planning in a large workspace: Design, implementation, and evaluation," *Scientific African*, vol. 15, p. e01068, 2022.

[45] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, ser. Wiley select coursepack. Wiley, 2005.

[46] J. Wang, X. Zhu, M. Guo, S. Yao, and Y. Su, "Improved hybrid algorithm of path planning for automated guided vehicle in storage system," in *Proceedings of the International Conference on Control, Automation and Artificial Intelligence*. Atlantis Press, 2017, pp. 332–336.

[47] M. Przybya and A. Milesi, "obstacle_detector," https://github.com/tysik/obstacle_detector, 2021, accessed: April 7, 2023.

[48] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, April 2018.

[49] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[50] 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 Conference on Decision and Control (CDC)*, Dec 2016, pp. 827–834.