

Safety-critical Autonomous Inspection of Distillation Columns using Quadrupedal Robots Equipped with Roller Arms

Jaemin Lee¹, Jeesop Kim², and Aaron D. Ames²

Abstract—This paper proposes a comprehensive framework designed for the autonomous inspection of complex environments, with a specific focus on multi-tiered settings such as distillation column trays. Leveraging quadrupedal robots equipped with roller arms, and through the use of onboard perception, we integrate essential motion components including: locomotion, safe and dynamic transitions between trays, and intermediate motions that bridge a variety of motion primitives. Given the slippery and confined nature of column trays, it is critical to ensure safety of the robot during inspection, therefore we employ a safety filter and footstep re-planning based upon control barrier function representations of the environment. Our framework integrates all system components into a state machine encoding the developed safety-critical planning and control elements to guarantee safety-critical autonomy, enabling autonomous and safe navigation and inspection of distillation columns. Experimental validation in an environment, consisting of industrial-grade chemical distillation trays, highlights the effectiveness of our multi-layered architecture.

I. INTRODUCTION

As robots and their locomotion capabilities continue to advance, they have become integral in various applications such as logistics [1], [2], inspection [3]–[5], and rescue operations [6]. Particularly in industrial settings, there is a growing demand for autonomous robotic systems to undertake hazardous missions, replacing human workers in dangerous, complex, and spatially constrained environments. The challenges posed by such environments necessitate a heightened focus on ensuring the safety of robotic deployments, especially in intricate spaces that may contain unsafe regions, exemplified by distillation column trays [5], [7].

Autonomous robotic systems are crucial in scenarios where human workers or operators cannot be present, and ensuring the safety of these systems becomes paramount. This paper introduces a comprehensive framework designed for the autonomous inspection of legged robots equipped with roller arms in multi-layered trays, as illustrated in Fig. 1. Building upon the results from our previous work [5], [7], we integrate planning and control stacks with a state machine and perception package to enhance the autonomy of the robot. Our proposed architecture enhances the locomotion stack by incorporating considerations for multiple safety conditions within the column tray. Intermediate motions are introduced to bridge the gap between the locomotion stack

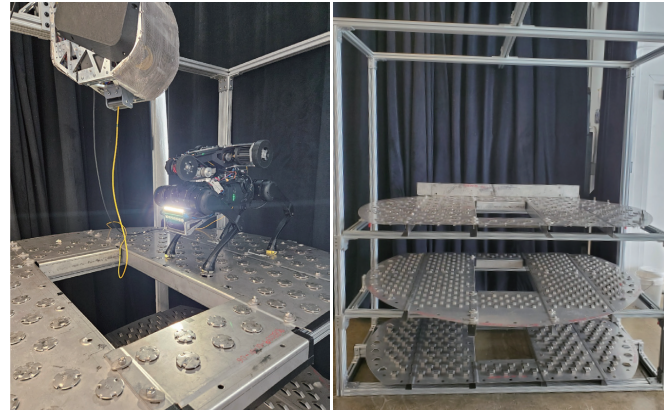


Fig. 1. **Robotic system and industry-grade column tray:** Our robotic system consists of a quadrupedal robot, roller arm, perception package, and tether system, as shown in the left figure.

and transition motion primitives, which is essential for navigating between different column layers. The implemented state machine empowers the robot to autonomously execute inspection missions autonomously, utilizing perceptual information in conjunction with the planning and control stacks. This unified approach ensures seamless planning, control, and perception integration for effective autonomous inspection in complex and confined spaces.

A. Related Works

Safety-critical control plays a crucial role in guiding robotic systems, including quadrupeds [5], [8]–[10], bipedal robots [11], [12], drones [13], and others. Specifically, control barrier functions (CBFs) [14] lead to the notion of safety filters [15], represented as quadratic programs (QPs), to effectively regulate control inputs and ensure safety. Moreover, CBF-based safety filters have been successfully implemented in robotic systems using reduced-order models like the double integrator [8], [10], [16], [17] or single-rigid body systems. This approach enables the implementation of safety filters through low-dimensional state variables, ensuring efficient computational performance. The application of CBF-based approaches extends beyond generating safe velocity commands for quadrupedal robots; it also includes the ability to switch gait types in legged robots, enhancing safety during locomotion [5].

After obtaining body velocity commands through the aforementioned safety filters, the subsequent step typically involves the utilization of either whole-body control (WBC) [18]–[21] or inverse dynamics control (IDC) [22], [23] to compute the actuator commands. To enhance robustness in the face of uncertain disturbances, adaptive approaches

This work is supported by Dow under the project #227027AT.

¹ J. Lee is with the Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, USA.

² J. Kim and A.D. Ames are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA.

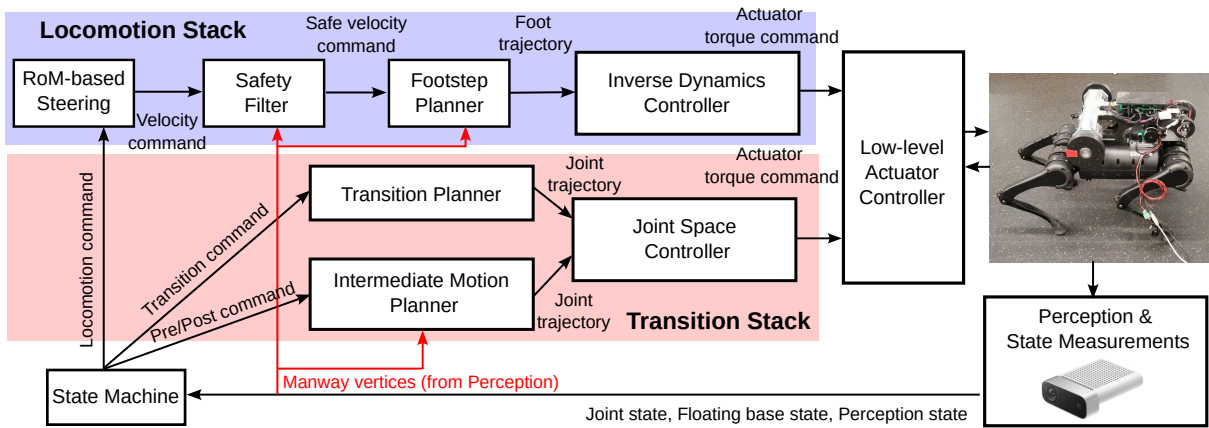


Fig. 2. **Overall structure of the proposed framework:** The proposed framework consists of a locomotion stack, transition stack, and state machine for autonomy. In addition, the perception part provides the manway vertices to ensure the safety in the planning parts.

[24], [25] and task-space decomposition strategies [26] are commonly employed, bolstering the robot's behaviors. While there is a growing interest in exploring the synergy between whole-body control and learning techniques for simpler tasks like walking [27] and push-recovery [28] in legged manipulators, sophisticated and consecutive motions often benefit more from a verifiable model-based whole-body control approach, particularly when augmented with a well-designed state machine.

An additional imperative in safety-critical planning and control involves the integration of a perception stack within a feedback loop. By incorporating the uncertain state data acquired from sensors, advancements have been made in enhancing CBF-based safety filters by utilizing measurement robust CBFs, as demonstrated and validated using Segway in [29]. In the context of bipedal locomotion, point cloud data is transformed into regions comprising convex polygons, effectively delineating safe and unsafe zones in challenging terrains [30], [31]. Quadruped robots similarly leverage perception packages to facilitate more responsive and secure locomotion [32], [33]. As demonstrated in the aforementioned studies, ensuring reliable perception guarantees safety during locomotion in diverse environments, encompassing both the wild outdoors and confined indoor spaces. This becomes particularly critical when controlling a robot within complex and limited spaces, such as our industry-grade column tray.

Autonomy stands as a pivotal factor in the successful execution of inspection missions using robotic systems. Without autonomy, manual control by human operators or direct access to monitor the robots' status becomes necessary. Traditionally, the significance of autonomy has been highlighted in scenarios involving multiple robots or agents [34], [35]. Even in the context of a single agent, autonomy plays a crucial role in the seamless execution of complex missions guided by a predefined logic, often represented by a control graph of activities, as exemplified in [36]. In this context, practical demonstrations with autonomy are facilitated using behavior trees or finite state machines [37], [38]. The key lies in designing the state machine with meticulous consideration of the specific robotic applications and the given missions.

This paper aims to develop a full safety-critical autonomy

stack for the inspection of chemical distillation columns. To this end, we leverage novel hardware consisting of a quadrupedal robot with an attached roller arm [7]. This hardware was designed with a view toward operating in the confined spaces of these column trays and, importantly, transitioning between the trays. Yet utilizing this hardware in an autonomous fashion has yet to be demonstrated.

B. Contributions

The primary innovation presented in this paper lies in introducing a comprehensive framework designed for the autonomous inspection of robotic systems illustrated in Fig. 2. This framework is specifically tailored for deployment in industry-grade trays encompassing multiple columns and consisting of locomotion stack, transition stack, perception part, and state machine, as shown in Fig. 2. Compared with our prior research efforts in [5], the specific contributions achieved in this work are outlined as follows:

- A state machine has been specifically crafted to enhance the autonomy of robotic inspection within the distillation column tray.
- Intermediate motions are meticulously generated to facilitate smooth transitions between the locomotion and transition stacks.
- The incorporation of a perception package enables autonomous navigation, allowing the system to avoid unsafe regions during the execution of safe locomotion intelligently.
- Safeguarding the integrity of the columns is prioritized through the simultaneous consideration of two CBFs in the planning process, leveraging a reduced-order model for enhanced efficiency.
- The efficacy of the proposed architecture is empirically validated through a series of experiments conducted in an industry-graded environment.

The structure of our paper is as follows. In Section II, we provide a concise overview of rigid-body dynamics, reduced-order models for robotic systems, and the fundamental concept of CBFs. Section III details the proposed framework, encompassing safety-critical planning, footstep replanning, full-body control, intermediate motions, state machines, and

perception packages. In Section IV, we validate the effectiveness of the proposed framework through autonomous inspection scenarios involving a quadruped robot equipped with a roller arm in an industry-grade column tray.

II. PRELIMINARIES

This section briefly overviews the system models employed in the proposed framework, including both full-body and reduced-order models. In addition, we introduce the fundamental concept of CBFs.

A. System Models

When considering the full-body dynamics of the legged robots, the equation of motion is expressed as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{F}_c \quad (1)$$

where $\mathbf{q} \in \mathcal{Q}$ and $\boldsymbol{\tau} \in \Gamma$ represent the joint variable and the control torque input, respectively, with a configuration space $\mathcal{Q} \subset \mathbb{R}^n$ and control input space $\Gamma \subset \mathbb{R}^{n-6}$. $\mathbf{D}(\mathbf{q})$, and $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ denote the mass/inertia matrix and the sum of Coriolis/centrifugal and gravitational forces, respectively. $\mathbf{S} \in \mathbb{R}^{(n-6) \times n}$ and $\mathbf{F}_c \in \mathbb{R}^{n_c}$ represent the selection matrix for the actuation joints and the contact forces, where n_c denotes the dimension of the contact forces under a certain supporting phase. $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{n_c \times n}$ is the contact Jacobian associated with the contact force. This full-body dynamics model computes the joint control command given reference trajectories.

Since the above full-body dynamics model is high-dimensional, we have frequently utilized reduced-order models for safety-critical planning, such as the inverted pendulum, lumped mass, double-integrator, etc. Similar to our previous work in [5], we choose the 2-dimensional double integrator system with a state $\boldsymbol{\varphi} = [\varphi_x, \varphi_y] \in \mathbb{R}^2$. We represent the corresponding state-space model as follows:

$$\dot{\boldsymbol{x}}_\varphi := f_\varphi(\boldsymbol{x}_\varphi) + g_\varphi(\boldsymbol{x}_\varphi)\boldsymbol{\nu} \quad (2)$$

where $\boldsymbol{\nu} \in \mathbb{R}^2$ denotes the input of the double-integrator system, which is the velocity command of the robot's base. It is important to figure out the bounds of the velocity commands $\boldsymbol{\nu}_{\max}$ and $\boldsymbol{\nu}_{\min}$ for realizable safety-critical commands.

B. Control Barrier Functions

Let us consider a control affine system described with a state $\mathbf{x} \in \mathcal{X}$ and control input $\mathbf{u} \in \mathcal{U}$ as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \quad (3)$$

where f and g are Lipschitz continuous. With an assumption that a safe set $\mathcal{C} := \{\mathbf{x} \in \mathbb{R}^{n_x} : h(\mathbf{x}) \geq 0\}$ is given where h is a continuous and differentiable function, we can define control barrier functions according to [14].

Definition 1. Given a safe set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_x} : h(\mathbf{x}) \geq 0\}$, $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is a control barrier function (CBF) if there exists a function $\alpha \in \mathcal{K}_\infty^e$ such that for all $\mathbf{x} \in \mathcal{C}$:

$$\sup_{\mathbf{u} \in \mathcal{U}} \dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})).$$

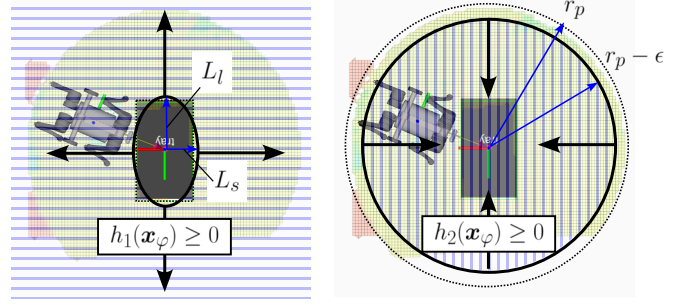


Fig. 3. Two CBFs for safety-critical planning in the tray: h_1 and h_2 are the CBFs to avoid the manway and the edge of the tray, respectively.

If all states in \mathcal{C} satisfied the above inequality for inputs in \mathcal{U} , \mathcal{C} is forward invariant.

In this paper, we employ the defined CBFs to generate safe control velocity of the robot's base based on the perceptual information of the environments by leveraging the double-integrator system.

III. THE PROPOSED FRAMEWORK

In this section, we explain the meticulous update and specifications applied to the safety-critical planning, footstep replanning, and full-body control modules for the confined space within the column tray. Moreover, we introduce novel components, including intermediate motions facilitating the connection between locomotion and transition stacks, a state machine, and the perception modules, contributing to the comprehensive evolution of our framework.

A. Safety-critical Planning

In the industry-graded column tray, the safe space is limited due to the manway. For this reason, we need to define two safety conditions for ensuring safety in the planning process using the reduced-order model. Let's assume that the manway is rectangular and all vertices are given as \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 , and \mathbf{v}_4 where the lengths of long and short sides are L_l and L_s , respectively. In addition, the radius of the column plate is r_p . First, to avoid the manway, which is an unsafe region, we designed the CBF in an ellipsoidal shape as we proposed in [5]:

$$h_1(\boldsymbol{x}_\varphi) = \boldsymbol{\varphi}^\top \mathbf{A} \boldsymbol{\varphi} + \mathbf{B} \boldsymbol{\varphi} + c. \quad (4)$$

The detailed \mathbf{A} , \mathbf{B} , and c are expressed as follows:

$$\mathbf{A} = \begin{bmatrix} \frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{b^2} & \left(\frac{1}{a^2} - \frac{1}{b^2}\right) \cos \theta \sin \theta \\ \left(\frac{1}{a^2} - \frac{1}{b^2}\right) \cos \theta \sin \theta & \frac{\sin^2 \theta}{a^2} + \frac{\cos^2 \theta}{b^2} \end{bmatrix}$$

$$\mathbf{B} = -2\bar{\boldsymbol{\varphi}}^\top \mathbf{A}, \quad c = \bar{\boldsymbol{\varphi}}^\top \mathbf{A} \bar{\boldsymbol{\varphi}} - 1$$

where $a = L_l + \varepsilon_l$, $b = L_s + \varepsilon_s$, and θ are the lengths of the major and minor radii and the orientation of the ellipse. In addition, $\bar{\boldsymbol{\varphi}}$ denotes the center of the manway. Second, the robot's base must be inside the column plates. For simplicity, we can define the CBF for this condition as follows:

$$h_2(\boldsymbol{x}_\varphi) = -(\boldsymbol{\varphi} - \bar{\boldsymbol{\varphi}})^\top (\boldsymbol{\varphi} - \bar{\boldsymbol{\varphi}}) + (r_p - \epsilon)^2 \quad (5)$$

where ϵ denotes the offset for considering the body frame of the robot's base.

Based on the above two CBFs, a QP problem is formulated to ensure the safety of the robotic systems as follows:

$$\begin{aligned} \arg \min_{\nu \in \mathcal{V}} \quad & \| \mathbf{k}^d(\mathbf{x}_\varphi, \boldsymbol{\xi}) - \nu \|^2, \\ \text{s.t.} \quad & \dot{h}_1(\mathbf{x}_\varphi, \nu) \geq -\alpha_1(h_1(\mathbf{x}_\varphi)), \\ & \dot{h}_2(\mathbf{x}_\varphi, \nu) \geq -\alpha_2(h_2(\mathbf{x}_\varphi)) \end{aligned} \quad (6)$$

where $\mathbf{k}^d : \mathcal{X}_\varphi \times \mathbb{R}^4 \rightarrow \mathcal{V}$ is the controller to track the desired position $\boldsymbol{\xi}$. The reference controller can be designed as a simple P controller $\mathbf{k}^d(\mathbf{x}_\varphi) = \mathbf{K}_\varphi(\boldsymbol{\xi} - \mathbf{x}_\varphi)$ where $\mathbf{K}_\varphi = \text{diag}(k_{\varphi_x}, k_{\varphi_y})$ with the proportional gains in the x and y direction, k_{φ_x} and k_{φ_y} . α_1 and α_2 denote the \mathcal{K}_∞^e functions associated with h_1 and h_2 , respectively. In this work, we assume that the goal positions or waypoints of the robot base are reachable. Otherwise, we need to modify L_l and L_s to adjust the CBF h_1 or relax one of the conditions as [10] to make the QP (6) feasible.

B. Footstep Replanning

Since the column tray has really limited space, we implement a quasi-static walking gait, which involves only one leg move, and the rest of the feet maintain solid contact during the swing phase. The walking gait is composed of four phases, w_{FL} , w_{BR} , w_{FR} , and w_{BL} whose the swing foot is front left, back right, front right, and back left. We repeat iterating the phases in a sequence:

$$w_{FL} \rightarrow w_{BR} \rightarrow w_{FR} \rightarrow w_{BL} \rightarrow w_{FL} \rightarrow \dots \quad (7)$$

while executing the quasi-static walking. Based on the current position and orientation of the robot's base, it is possible to compute the goal position of the swing foot \mathbf{p}_s^g with the Raibert heuristics. If the goal position of swing foot \mathbf{p}_s^g is near the manway or inside the manway, the position is modified according to [5].

In addition, we modified the goal position of the swing foot when it is near to the edge of the column tray or outside the tray. The modified goal position \mathbf{p}_s^n is obtained by scaling the position vector with respect to the center of the tray:

$$\mathbf{p}_s^n = \bar{\mathbf{x}}_\varphi + \frac{r_p - \epsilon}{\|\mathbf{p}_s^g - \bar{\mathbf{x}}_\varphi\|} (\mathbf{p}_s^g - \bar{\mathbf{x}}_\varphi). \quad (8)$$

It is important that the support polygon with the replanned foot location is conservative to maintain the CoM inside it while executing the next swing phase. In addition, we tested the modified foot location to be kinematically feasible within ROM. After that, the swing foot trajectories are designed by using cubic interpolation with the swing time interval.

C. Full-body Control

In this paper, we leverage two types of full-body control employed for locomotion control and joint-space control during the transition and intermediate motions. First, the locomotion controller is formulated in a QP problem. After computing joint configuration and velocity, \mathbf{q}^d and $\dot{\mathbf{q}}^d$, based on the swing foot reference trajectories by solving an inverse

kinematics problem, we can formulate the inverse dynamic control QP as follows:

$$\begin{aligned} \min_{\dot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{F}_c} \quad & \|\ddot{\mathbf{q}}^d - \ddot{\mathbf{q}}\|_{\mathbf{W}_{\ddot{\mathbf{q}}}}^2 + \|\boldsymbol{\tau}\|_{\mathbf{W}_\tau}^2 + \|\mathbf{F}_c\|_{\mathbf{W}_c}^2 \\ \text{s.t.} \quad & \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^\top \mathbf{F}_c, \\ & \mathbf{J}_c(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0}, \\ & \mathbf{F}_c \in \mathcal{FC}, \quad \boldsymbol{\tau} \in \Gamma, \\ & \ddot{\mathbf{q}}^d = \mathbf{K}_p(\mathbf{q}^d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) \end{aligned} \quad (9)$$

where \mathbf{K}_p and \mathbf{K}_d are proportional and derivative gains to design the nominal joint acceleration $\ddot{\mathbf{q}}^d$. In addition, $\mathbf{W}_{\ddot{\mathbf{q}}}$, \mathbf{W}_τ , and \mathbf{W}_c denote weighting matrices assigned to the terms associated with joint acceleration error, control torque input, and contact force, respectively. \mathcal{FC} is the contact wrench cone for the given contacts.

By using the optimal solution to the above problem $\ddot{\mathbf{q}}^*$, we can compute the reference joint position and velocity, \mathbf{q}^n and $\dot{\mathbf{q}}^n$, with Euler integration as presented in [5]. Then, the following joint impedance controller is utilized to compute the command torque.

$$\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{\tau}^* + \mathbf{K}_p^{\text{imp}}(\mathbf{q}^n - \mathbf{q}) + \mathbf{K}_d^{\text{imp}}(\dot{\mathbf{q}}^n - \dot{\mathbf{q}}) \quad (10)$$

where $\mathbf{K}_p^{\text{imp}}$ and $\mathbf{K}_d^{\text{imp}}$ represent the proportional and derivative gains for the joint impedance controller. $\boldsymbol{\tau}^*$ denotes the optimal solution to the QP problem in (9). For the transition stack, we make the full-body controller much simpler by replacing \mathbf{q}^n and $\dot{\mathbf{q}}^n$ to \mathbf{q}^d and $\dot{\mathbf{q}}^d$ (10). In addition, the gravity compensation torque $\boldsymbol{\tau}_g$ is plugged into the feedforward torque command instead of $\boldsymbol{\tau}^*$. Since the transition and intermediate motions, such as locomotion, are not dynamic, this simplified version of the full-body controller is fairly effective in tracking the reference trajectories.

D. Intermediate Motions

Since the transition trajectories are designed offline with respect to a specific joint configuration, it is necessary to change the configuration smoothly by changing the contact sequence. Let's suppose that $N = \{1, \dots, \ell\}$ is the discredited horizon and \mathbf{q}_t is the ready configuration for the transition. Then, we can formulate the following mixed-integer quadratic program (MIQP):

$$\begin{aligned} \min_{\mathbf{c}_{[1:\ell]}, \mathbf{q}_{[1:\ell]}} \quad & (\mathbf{q}_t - \mathbf{q}^{(\ell)})^\top \mathbf{Q}_q (\mathbf{q}_t - \mathbf{q}^{(\ell)}) \\ \text{s.t.} \quad & \sum_{i=1}^5 \mathbf{c}_i^{(1)} = 0, \quad \sum_{i=1}^5 \mathbf{c}_i^{(\ell)} = 0, \\ & \sum_{i=1}^5 \mathbf{c}_i^{(k)} = 2, \quad k \in \{2, \dots, \ell - 1\}, \\ & \mathbf{c}_1^{(j)}, \mathbf{c}_4^{(j)}, \mathbf{c}_5^{(j)} \in \{0, 2\}, \quad \mathbf{c}_2^{(j)}, \mathbf{c}_3^{(j)} \in \{0, 1\}, \\ & \mathbf{c}_i^{(j)} \mathbf{S}_i(\mathbf{q}_t - \mathbf{q}^{(j)}) = 0, \quad i \in \{1, \dots, 5\} \end{aligned} \quad (11)$$

where $\mathbf{q}^{(j)}$ denotes the j -th configuration. The decision parameters $\mathbf{c}^{(j)} = [\mathbf{c}_1^{(j)}, \mathbf{c}_2^{(j)}, \mathbf{c}_3^{(j)}, \mathbf{c}_4^{(j)}, \mathbf{c}_5^{(j)}]$ is defined for the j -th step where \mathbf{c}_1 , \mathbf{c}_2 , \mathbf{c}_3 , \mathbf{c}_4 , and \mathbf{c}_5 indicate the

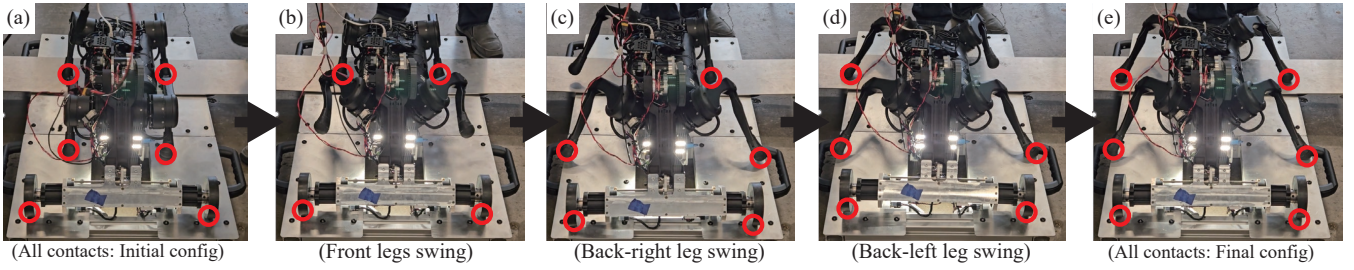


Fig. 4. **Experimental result generating Pre-motion for downward transition:** By strategically planning intermediate motions prior to initiating the downward transition, the sequence of contact points ensures smooth joint configuration adjustments for the transition. CoM is maintained within the support polygon throughout these intermediate motions.

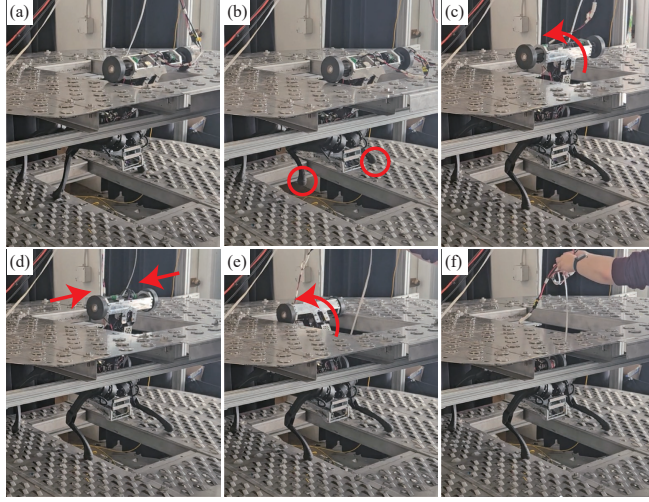


Fig. 5. **Experimental result generating Post-motion for downward transition:** (a) final configuration post downward transition, (b) front legs realigned to adjust the support polygon, (c) arm slightly folded to detach from the upper layer, (d) arm length adjustment, (e) arm fully folded in preparation for subsequent locomotion, (f) configuration for next locomotion.

integers for contacts of wheel, front-left, front-right, back-left, and back-right legs, respectively. In addition, $\mathbf{C}_{[1:\ell]} = \{\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(\ell)}\}$, $\mathbf{q}_{[1:\ell]} = \{\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(\ell)}\}$ and \mathbf{S}_i is the selection matrix for the leg associated with \mathbf{c}_i . The meaningful solution is $\mathbf{C}_{[1:\ell]}$, which is a contact sequence for the intermediate motions.

Once the optimal contact sequence of the intermediate motions is obtained, we generate continuous trajectories to reach the initial continuation of transition or locomotion by avoiding unexpected collisions. The intermediate motion before the transition requires control of the roller arm to make stable contact with the tray before making sequential motions determined by (11). Fig. 4 shows the experimental validation of intermediate motions before making the downward transition behavior. Based on the optimal contact sequence, the CoM remains inside of support polygon while changing configuration to ready the downward transition.

After making the downward transition behavior, we need to consider additional constraints to be ready for next locomotion tasks. The arm must be folded while controlling the CoM inside of the support polygon. As shown in Fig. 5, if needed, we change the foot locations to enlarge the support polygon. These processes are helpful to make smooth swapping between the locomotion and transition stacks.

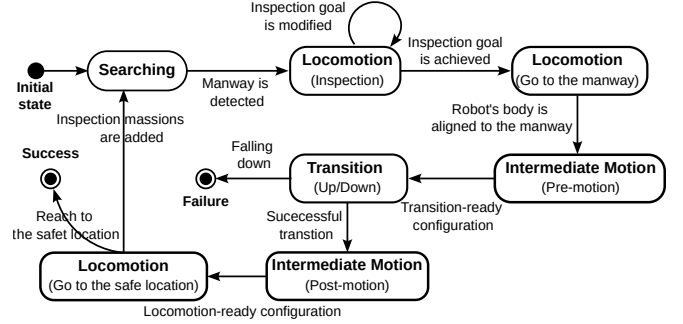


Fig. 6. **State machine for autonomy while performing inspection missions:** From the initial state to the completion of inspection. In the event of failed transitions, the state machine halts the robot's actions.

E. State Machine

To enhance the autonomy of robots within a column tray, we introduce a state machine based on both state and perception measurements. Seven distinct tasks are defined to efficiently execute inspection missions as follows:

- **Searching:** The robot adjusts its body orientation and height at the initial position to obtain the manway vertices.
- **Locomotion (Inspection):** Safely navigating the robot through a single column, ensuring the completion of the inspection mission while avoiding unsafe regions.
- **Locomotion (Go to the manway):** Directing the robot to the manway in preparation for an upward or downward transition.
- **Intermediate Motion (Pre-motion):** Modifying the robot's configuration to prepare for a transition within a specified contact sequence.
- **Transition (Up/Down):** Executing a smooth and safe transition between columns.
- **Intermediate Motion (Post-motion):** Adjusting the robot's configuration to be ready for locomotion again after completing a transition.
- **Locomotion (Go to the safe location):** Navigating the robot to a pre-defined safe location.

These tasks form the basis for a defined state machine, as depicted in Fig. 6. Starting from the initial state, the robot autonomously inspects a single-column tray, capable of repeating the inspection task when goals are added or modified. Following completion of the inspection, the robot positions itself on the manway for upward or downward transitions. Trajectories for the transitions between the column trays are

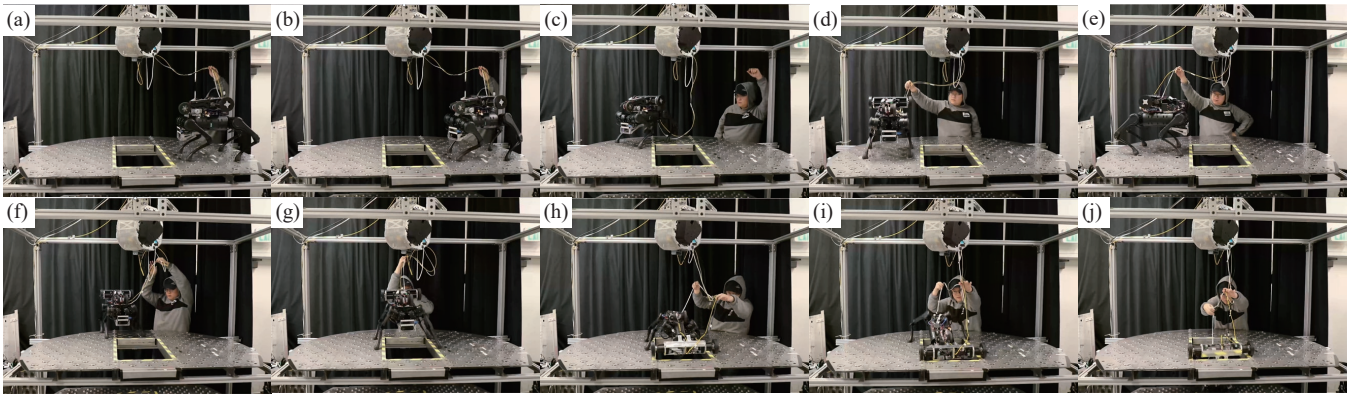


Fig. 7. **Snapshots of Autonomous Inspection:** (a) initial position establishment, (b) manway search, (c) locomotion execution (inspection) towards designated location, (d) inspection task completion, (e) resuming manway search, (f) moving towards waypoint, (g) approach to transition-ready position on the manway, (h) intermediate motion for downward transition preparation, (i) downward transition execution, (j) completion of downward transition.

generated by the optimization approach proposed in [7] with additional CoM constraints for enhancing the balance during the transitions [5]. It is terminated if a stable transition is not achieved during the process. Upon successful transition, the robot readjusts its configuration for subsequent locomotion. The process concludes if the robot reaches the pre-defined safe location or repeats based on pre-defined conditions. This state machine significantly enhances the autonomy of robotic inspection within multi-column trays, minimizing the need for human intervention.

F. Perception in a Loop

Utilizing a Kinect package, we implement a perception stack to discern the manway within the column tray. The primary function of this perception stack is to furnish crucial data regarding the vertices of the manway, denoted as v_1 , v_2 , v_3 , and v_4 . To optimize efficiency and mitigate computation delays in the planning and control loops, we meticulously specify the locations where perceptual data is acquired.

To enhance the precision of our measurements, we conduct 100 data collections and leverage their average. Assuming the relative positions of the manway vertices, $v_1^{\{P\}}$, $v_2^{\{P\}}$, $v_3^{\{P\}}$, and $v_4^{\{P\}}$, are determined with respect to the perception frame, we convert these positions into the global coordinate system:

$$\mathbf{v}_k^{\{G\}} = \mathbf{R}_{\{P|G\}} \mathbf{v}_k^{\{P\}} + \boldsymbol{\varphi}_p^{\{G\}}, \quad \forall k \in \{1, 2, 3, 4\} \quad (12)$$

where $\mathbf{v}_k^{\{G\}}$, $\mathbf{R}_{\{P|G\}}$, and $\boldsymbol{\varphi}_p^{\{G\}}$ represent the position of the k -th vertex in the global coordinate system, the rotation matrix from the global to perception coordinates, and the position of the Kinect package with respect to the global frame, respectively.

Based on the obtained vertices $\mathbf{v}_k^{\{G\}}$, we compute the base's velocity command and determine the landing location of the swing foot, as detailed in Sections III-A and III-B. The center of the polygon formed by these vertices is considered as the center of the column tray. Additionally, we make the assumption that given the radius of the column layer and the dimensions of the manway, it is feasible to ascertain the accuracy of the perceptual information.

IV. EXPERIMENTS

In this section, we show the results of our experiments demonstrating the autonomous inspection capabilities of the Unitree A1 quadruped robot equipped with a roller arm within an industry-grade column tray. The roller arm, boasting 5 DOFs, incorporates 4 actuators and a bidirectional lead screw for arm extension. Experimental setups employed Intel NUC and Nvidia Jetson AGX Xavier devices, connected to the robot via network switches, to execute and demonstrate the experiments.

Concurrently, trajectory computations for transition and intermediate motions were carried out using FROST [39] and MATLAB on a laptop with a 3.4 GHz Intel Core i7 processor. The optimization solver OSQP [40] was employed for solving the QP problem of safety-critical planning and full-body control, executed on the Intel NUC. This integrated approach ensured efficient planning and control mechanisms throughout the inspection process.

The column tray is structured with three layers, each separated by a 22-inch gap, with a tray radius of 35 inches. The manway, a crucial component, features dimensions of 27.5 inches in length and 15 inches in width. Notably, each layer of the tray is distinguished by unique stud patterns.

A. Autonomous Inspection

The overall inspection scenario is depicted in Fig. 7. We describe the detailed inspection process corresponding to the snapshot. Initially, the robot is positioned arbitrarily on the top layer of the column tray for autonomous inspection, facing the manway (Fig. 7(a)). To conduct the search motion, we control the robot's body yaw angle within a range of ± 0.3 radians for a duration of 2 seconds. If perceptual data (vertices of the manway) is not acquired, the robot repeats the search motion until the vertices data is received from the perception stack (Fig. 7(b)). Subsequently, the robot engages in **Locomotion** (inspection) to reach the predetermined position on the opposite side (Fig. 7(c)). Throughout the inspection task, both CBF constraints h_1 and h_2 are considered in our locomotion control alongside the footstep re-planning.

Once the inspection task is completed (Fig. 7(d)), the robot is directed towards the manway and repeats the search

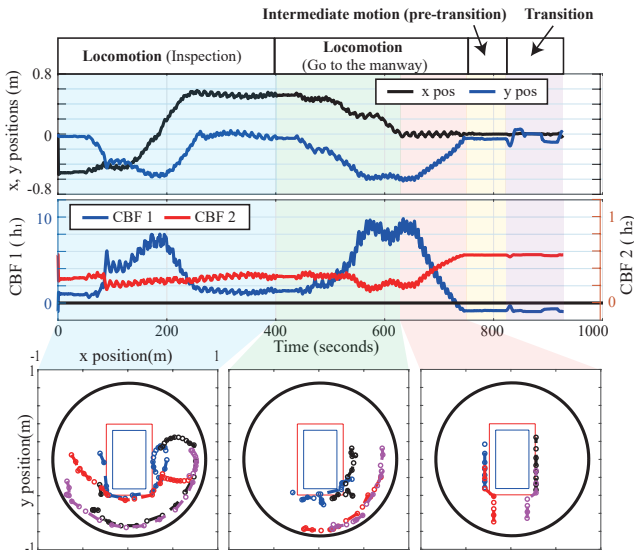


Fig. 8. **Experimental results:** In the upper portion, color-coded zones delineate distinct stages of locomotion: Blue for inspection, green for approaching a waypoint, red for transition-ready location, yellow for pre-motion before downward transition, and purple for the downward transition. In the lower section, circular markers denote the positions of the front-left (blue), front-right (black), back-left (red), and back-right (purple) feet.

motion to acquire perception data for the manway (Fig. 7(e)). The **Locomotion** (Go to the manway) phase comprises two stages: movement towards the waypoint with orientation alignment (Fig. 7(f)) and approach to the transition-ready position on the manway (Fig. 7(g)). While navigating towards the waypoint, the CBF constraints persist to prevent unsafe motions within the manway by ensuring safe foot placement. However, as the robot approaches the transition-ready location, the CBF constraints are lifted, activating only the footstep re-planning component.

Following this, **Intermediate motion** (Pre-motion) is executed to align the joint configuration with the initial value of the transition trajectory (Fig. 7(h)), which is same as the process illustrated in Fig. 4. The robot then moves to the lower column tray through **Transition** (Down), as depicted in Fig. 7(i) and (j). In the event of a successful transition, the robot proceeds to execute **Intermediate motion** (Post-motion), described in Fig. 5. However, if the transition fails, the robot halts, and an additional tether system is employed to pull the robot back to the upper column layer.

B. Discussion

Fig. 8 presents the experimental results of autonomous inspection conducted within the industry-grade column tray. In the upper segment of Fig. 8, the robot’s base position is precisely controlled through full-body control within the confined space without any jerky or unstable motions. In addition, the CBF values for h_1 and h_2 are always greater than 0 within the blue and green zones, ensuring the robot’s safety. Since we remove the safety filter within the red, yellow, and purple zones, negative CBF value (h_1) does not mean any safety violation.

Furthermore, the lower portion of Fig. 8 shows the footstep planning during the inspection. To make footstep re-planning

more conservative, we expand the unsafe region based on the geometry of the manway (blue box) and include a 5 cm buffer margin (red box). All foot placements during autonomous inspection are located within the safe region delineated by the boundary of the column layer (black circle) and the red box, excluding during transition behavior. These results distinctly show that the proposed framework enables safety-critical autonomy of our robotic system within an industry-grade column tray.

However, in the hardware experiments, challenges arose in locomotion control due to the uncertain friction coefficient of the column tray. Initially, we adopted a baseline value of $\mu = 0.7$, which had proven effective for locomotion on flat ground. However, the column tray is much more slippery than the flat ground so we need to tune parameter, resulting in a revised value of $\mu = 0.4$ after iterative hardware experiments. Moreover, the limitation of space rendered the trot gait impractical, leading us to employ the quasi-static gait. Nonetheless, in scenarios where the column tray provides more wider space, employing the trot gait could be feasible instead of the quasi-static gait. To enhance the alignment of transition-ready positions and generate safer motions, it is crucial to achieve more accurate and frequent updates of the perception stack.

V. CONCLUSION

This paper proposes a comprehensive framework for the autonomous inspection of a quadruped robot equipped with a roller arm within an industry-grade column tray featuring multiple layers. Building upon our prior work, we achieve full autonomy in executing inspection missions by employing a state machine to determine a sequence of the robot’s behaviors. Additionally, contact sequences for intermediate motions, connecting locomotion and transition stacks, are derived through the solution of a MIQP problem. The manway and tray dimensions are accurately measured using a perception package, and this perceptual information is leveraged to tailor the safety-critical planning and footstep replanning components for operation in confined spaces.

Looking ahead, our future plans involve integrating generative AI with the proposed safety-critical framework to enhance autonomous inspection capabilities across diverse environments. Rather than manually specifying state machines or behavior trees, our goal is to develop a wrapper that bridges the outputs of generative AI with the framework, adapting to dynamic environments that may involve moving objects or unexpected events.

ACKNOWLEDGEMENT

The authors express their gratitude to collaborators at NASA JPL, particularly for their contributions to the perception component, and the development of the roller arm used throughout the experiments (as documented in [7]).

REFERENCES

- [1] W. Echelmeyer, A. Kirchheim, and E. Wellbrock, “Robotics-logistics: Challenges for automation of logistic processes,” in *IEEE International Conference on Automation and Logistics*, 2008, pp. 2099–2103.

- [2] J. Lee, M. Seo, A. Bylard, R. Sun, and L. Sentis, "Real-time model predictive control for industrial manipulators with singularity-tolerant hierarchical task control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023, pp. 12 282–12 288.
- [3] D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.
- [4] R. S. Lim, H. M. La, Z. Shan, and W. Sheng, "Developing a crack inspection robot for bridge maintenance," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011, pp. 6288–6293.
- [5] J. Lee, J. Kim, W. Ubellacker, T. G. Molnar, and A. D. Ames, "Safety-critical control of quadrupedal robots with rolling arms for autonomous inspection of complex environments," *arXiv preprint arXiv:2312.07778*, 2023.
- [6] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.
- [7] T. G. Molnar, K. Tighe, W. Ubellacker, A. Kalantari, and A. D. Ames, "Mechanical design, planning, and control for legged robots in distillation columns," *Journal of Computational and Nonlinear Dynamics*, vol. 18, no. 6, p. 061001, 2023.
- [8] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 944–951, 2021.
- [9] 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 (ICRA)*. IEEE, 2021, pp. 8352–8358.
- [10] J. Lee, J. Kim, and A. D. Ames, "Hierarchical relaxation of safety-critical controllers: Mitigating contradictory safety conditions with application to quadruped robots," *arXiv preprint arXiv:2305.03929*, 2023.
- [11] 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*, vol. 13. Cambridge, MA, USA, 2017, pp. 1–10.
- [12] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 1041–1053.
- [13] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," *arXiv preprint arXiv:2311.05802*, 2023.
- [14] 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, 2016.
- [15] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.
- [16] J. Kim, J. Lee, and A. D. Ames, "Safety-critical coordination for cooperative legged locomotion via control barrier functions," *arXiv preprint arXiv:2303.13630*, 2023.
- [17] M. Jankovic, M. Santillo, and Y. Wang, "Multiagent systems with cbf-based controllers: Collision avoidance and liveness from instability," *IEEE Transactions on Control Systems Technology*, 2023.
- [18] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, 2012.
- [19] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "Opensoft: a whole-body control library for the compliant humanoid robot coman," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6248–6253.
- [20] D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis, "Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 936–956, 2020.
- [21] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3359–3365.
- [22] S. Zapolsky and E. Drumwright, "Quadratic programming-based inverse dynamics control for legged robots with sticking and slipping frictional contacts," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3266–3271.
- [23] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1085–1090.
- [24] J. Lee, J. Ahn, D. Kim, S. H. Bang, and L. Sentis, "Online gain adaptation of whole-body control for legged robots with unknown disturbances," *Frontiers in Robotics and AI*, vol. 8, p. 788902, 2022.
- [25] J. Lee, H. Dallali, M. Jin, D. Caldwell, and N. Tsagarakis, "Robust and adaptive whole-body controller for humanoids with multiple tasks under uncertain disturbances," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5683–5689.
- [26] F. Farshidian, E. Jelavić, A. W. Winkler, and J. Buchli, "Robust whole-body motion control of legged robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4589–4596.
- [27] P. M. Viceconte, R. Camoriano, G. Romualdi, D. Ferigo, S. Daffarra, S. Traversaro, G. Oriolo, L. Rosasco, and D. Pucci, "Adherent: Learning human-like trajectory generators for whole-body control of humanoid robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2779–2786, 2022.
- [28] D. Ferigo, R. Camoriano, P. M. Viceconte, D. Calandriello, S. Traversaro, L. Rosasco, and D. Pucci, "On the emergence of whole-body strategies from humanoid robot push-recovery learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8561–8568, 2021.
- [29] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6286–6291.
- [30] S. Bertrand, I. Lee, B. Mishra, D. Calvert, J. Pratt, and R. Griffin, "Detecting usable planar regions for legged robot locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4736–4742.
- [31] B. Mishra, D. Calvert, S. Bertrand, S. McCrory, R. Griffin, and H. E. Sevil, "Gpu-accelerated rapid planar region extraction for dynamic behaviors on legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8493–8499.
- [32] I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, and D. G. Caldwell, "Onboard perception-based trotting and crawling with the hydraulic quadruped robot (hyq)," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 6052–6057.
- [33] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [34] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall, "Managing autonomy in robot teams: observations from four experiments," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007, pp. 25–32.
- [35] P. Schermerhorn and M. Scheutz, "Dynamic robot autonomy: Investigating the effects of robot decision-making in a human-robot team task," in *Proceedings of the 2009 international conference on multimodal interfaces*, 2009, pp. 63–70.
- [36] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *The International Journal of Robotics Research*, vol. 17, no. 4, pp. 315–337, 1998.
- [37] M. Iovino, J. Förster, P. Falco, J. J. Chung, R. Siegwart, and C. Smith, "On the programming effort required to generate behavior trees and finite state machines for robotic applications," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5807–5813.
- [38] R. Ghzouli, T. Berger, E. B. Johnsen, A. Wasowski, and S. Dragule, "Behavior trees and state machines in robotics applications," *IEEE Transactions on Software Engineering*, 2023.
- [39] A. Hereid and A. D. Ames, "Frost*: Fast robot optimization and simulation toolkit," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 719–726.
- [40] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: <https://doi.org/10.1007/s12532-020-00179-2>