RL-augmented MPC Framework for Agile and Robust Bipedal Footstep Locomotion Planning and Control

Seung Hyeon Bang*[1], Carlos Arribalzaga Jové*[1,2], and Luis Sentis[1]

***Abstract*— This paper proposes an online bipedal footstep planning strategy that combines model predictive control (MPC) and reinforcement learning (RL) to achieve agile and robust bipedal maneuvers. While MPC-based foot placement controllers have demonstrated their effectiveness in achieving dynamic locomotion, their performance is often limited by the use of simplified models and assumptions. To address this challenge, we develop a novel foot placement controller that leverages a learned policy to bridge the gap between the use of a simplified model and the more complex full-order robot system. Specifically, our approach employs a unique combination of an ALIP-based MPC foot placement controller for sub-optimal footstep planning and the learned policy for refining footstep adjustments, enabling the resulting footstep policy to capture the robot's whole-body dynamics effectively. This integration synergizes the predictive capability of MPC with the flexibility and adaptability of RL. We validate the effectiveness of our framework through a series of experiments using the full-body humanoid robot DRACO 3. The results demonstrate significant improvements in dynamic locomotion performance, including better tracking of a wide range of walking speeds, enabling reliable turning and traversing challenging terrains while preserving the robustness and stability of the walking gaits compared to the baseline ALIP-based MPC approach.**

I. INTRODUCTION

Agile and robust bipedal locomotion is essential for humanoid robots to achieve human-level performance. One of the main challenges in achieving this is designing a footstep policy that enables bipeds to constantly adjust their planned footstep positions to maintain balance as well as to achieve more agile and fast maneuvers, even while traversing adverse environments, such as external disturbances or challenging terrains.

In this paper, we present an RL-augmented MPC framework designed to generate a footstep policy for agile and robust bipedal locomotion. Our framework, as illustrated in Fig. 1, combines model-based optimal control (MBOC) with reinforcement learning (RL) to leverage the strengths of both approaches. Specifically, we adopt a hierarchical control architecture consisting of a high-level (HL) planner, which integrates both MPC and RL policies, and a low-level (LL) tracking controller. The MPC utilizes a simplified model to generate an initial, suboptimal footstep plan. The residual RL policy then refines this plan by leveraging the robot's full-order dynamics model. This approach helps overcome the modelling errors of the simplified model and thus results in an enhanced footstep policy.

* These authors have contributed equally to this work.
[1] The University of Texas at Austin, TX, USA, {bangsh0718, ca36828, lsentis}@utexas.edu
[2] Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

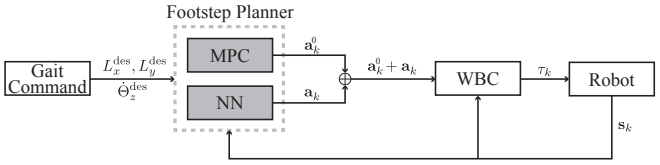

Fig. 1. **Overview of the proposed control framework:** The footstep planner consists of a simplified model-based model predictive controller (MPC) and a neural network (NN). Together, these components generate the footstep policy by integrating the solutions from each module. A whole-body feedback controller (WBC) then tracks the footstep policy.

A. Related Work:

In legged robot locomotion, footstep planning is essential for maintaining stability in the robot's inherently unstable dynamics while navigating through complex environments. The Raibert heuristic [1] has been widely used in legged robot control due to its simplicity and effectiveness.

Alternatively, model-based footstep planning strategies have increasingly been developed. These methods typically use simplified models that approximate the robot's underactuated dynamics to determine foot placements that stabilize the robot's CoM dynamics. [2], [3] utilized the Linear Inverted Pendulum (LIP) model to calculate footstep placement based on the robot's CoM states, aiming to reverse the CoM velocity after a specified duration. On the other hand, the capture point concept [4], [5], based on the divergent component of motion of the LIP model, has been widely used to develop footstep decision-making strategies. It provides a theoretical framework for analyzing a controller's ability to bring robots to a stop within a specified number of steps.

On the other hand, [6] uses the Angular Momentum Linear Inverted Pendulum (ALIP) model to generate a gait for stable walking. This gait is designed by generating the robot's swing foot and vertical CoM trajectories while conserving angular momentum at impact. [7] proposes a one-step ahead gait controller that determines where to step given a desired angular momentum. [8] extended this gait controller with an MPC-based framework [9], [10], [11], considering workspace and friction cone constraints. More recently, more complex dynamic models such as the single-rigid body model [12] and the centroidal dynamics model [13] have been adopted to jointly optimize the states and footsteps. However, the abstractions and assumptions in these simplified models can compromise the robustness of the optimized footsteps or require heuristics and arduous tuning due to the lack of consideration of the full-order model.

In contrast to model-based approaches, model-free RL-based approaches can discover a footstep policy through trial and error, eliminating the need for explicit dynamic models. RL enables the discovery of the policy by maximizing the discounted cumulative reward as an RL agent interacts with the environment. Unlike end-to-end RL approaches [14], [15], [16] which learn locomotion policies by directly mapping sensor data to joint commands, RL-based footstep policies are usually developed within a hierarchical control structure [17], [18], [19] that incorporates a HL and a LL controller. For example, [17] trains a HL controller to generate footsteps using proprioceptive states and terrain information as input. Similarly, [19] proposes an RL method to train a residual footstep policy with an offline gait library. These approaches have enabled the direct exploration of task space actions, yielding more sample-efficient learning. Following this paradigm, [20] formulates observation and action spaces of a Markov Decision Process (MDP) inspired by the ALIP model to train a footstep policy that enables robust walking under adversarial conditions. However, unlike model-based footstep planning, model-free RL approaches do not have access to the underlying robot dynamics and so struggle with sample inefficiency as they must learn to reason about the robot dynamics implicitly. As a result, they require substantial training data since they need to build the policies from scratch.

In recent years, there has been extensive research on integrating MBOC and RL to leverage the benefits from both approaches. [21] proposed an RL framework that incorporated MBOC to generate both reference base and foot motions and employed the motion imitation technique to learn complex locomotion tasks. [22] utilized RL to learn task-optimal parameters of a simplified dynamics model, which was subsequently optimized to generate CoM trajectories and footstep positions using an MPC framework. In [23], an RL footstep policy was trained with guidance from the previously-mentioned Time-to-Velocity Reversal (TVR) footstep planner, based on the LIP model. The training process utilized the TVR planner's solution as suboptimal guidance, enabling the residual RL policy to maximize long-term rewards without requiring excessive training data.

Our approach combines MBOC and RL, similarly to the method introduced in [23]. However, instead of adopting the LIP model, we use an ALIP model-based MPC footstep generator similar to [8]. This approach allows us to obtain more viable footsteps to feed to the RL policy, thanks to the MPC's handling of strict constraints and its use of a prediction horizon. Unlike [23] computing the next desired footstep at the apex of each step, our method enables replanning footsteps multiple times during the foot swing motion. This capability aims to be more versatile for robust locomotion behaviors under external disturbances. Moreover, in contrast to [20], our RL policy leveraging the MBOC process converges faster during training, enabling sample-efficient learning.

B. Contributions:

The main contributions of this paper are the following:

1) We propose the first RL-augmented MPC framework for bipedal footstep generation, designed to significantly enhance the tracking of the robot's walking speed, the robustness to external disturbances, the walking adaptability (transitioning between different velocity commands), and the ability to traverse arbitrary slopes.
2) We designed flexible reward terms for the RL process to effectively learn from the ALIP-MPC process.
3) We demonstrate that our approach achieves more agile, robust, and adaptive locomotion behaviors than using MPC alone by overcoming modelling errors associated with the simplified dynamics used by the MPC footstep planner.

C. Organization:

The remainder of this paper is organized as follows. Section II provides a concise overview of the ALIP-based MPC footstep planner, WBC, and model-free RL. Section III describes the locomotion problem addressed in this study. In Section IV, we describe the proposed RL-augmented MPC approach. Section V validates the effectiveness of the proposed approach through various locomotion scenarios. Finally, Section VI concludes the paper and discusses future works.

II. PRELIMINARIES

In this section, we introduce the ALIP-based MPC footstep planner [8], which we will use to synergize with our RL method. We also present background of the whole-body controller (WBC) and model-free RL approaches that we employ.

A. ALIP-based MPC Footstep Planner

The main objective of the ALIP-based MPC footstep planner is to compute the desired footholds to enable the angular momentum to converge to a desired state at the end of each step. In the following subsection, we briefly describe the main components of the MPC.

1) ALIP Model: Simplified dynamic models offer the advantage of capturing the centroidal states of the robot while reducing the dimensions of the full-order model under several assumptions. Among the models used for footstep design, the ALIP model [6] has increasingly gained attention due to its higher accuracy for state estimations of the model. This advantage is attributed to the use of the robot's angular momentum about the contact point, which is represented as one of its states in the model. Under the model assumptions, with one of them being that the angular momentum about the center of mass ($\boldsymbol{L}^c$) is zero to facilitate linearization [8], the ALIP dynamics model is given by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}_{\text{fp}} \tag{1}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{mz_H} \\ 0 & 0 & -\frac{1}{mz_H} & 0 \\ 0 & -mg & 0 & 0 \\ mg & 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

m denotes the mass of the robot, g is the gravitational constant, and z_H is the constant CoM height. The state and input variables are defined as $\mathbf{x} = [x_c, y_c, L_x, L_y]^\top \in \mathbb{R}^4$ and $\mathbf{u}_{\text{fp}} = [u_x, u_y]^\top \in \mathbb{R}^2$, where x_c, y_c represents the CoM position in the x, y-direction with respect to the stance contact point, L_x, L_y denotes the angular momentum about the x, y-axes of the contact point, and u_x, u_y are the new foot placement positions in the x, y-direction from the current stance contact point.

2) MPC Formulation for Foot Placement: Assuming conservation of angular momentum about the contact point before and after each footstep, the resulting discrete-time dynamics over an N_s footstep horizon, with each step having a fixed time duration of T_s, including the step-to-step and intra-step dynamics, are given as follows:

$$\mathbf{x}_{k+1} = \begin{cases} e^{\mathbf{A}\Delta t}(\mathbf{x}_k + \mathbf{B}\mathbf{u}_{\text{fp},k}), & \text{if } k = iN_{\Delta t}, \\ & 0 \le i \le N_s - 1 \\ e^{\mathbf{A}\Delta t}\mathbf{x}_k, & \text{otherwise} \end{cases} \tag{2}$$

where Δt is the sampling time and $N_{\Delta t} = T_s/\Delta t \in \mathbb{Z}$ is the number of samples for each step.

The MPC process aims to find the desired footholds with the objective of tracking the desired angular momentum $(L_x^{\text{des}}, L_y^{\text{des}})$ during step transitions while satisfying kinematic constraints (e.g., leg length), $\mathcal{X}^{\text{kin}}$, friction cone constraints, $\mathcal{X}^{\text{slip}}$, and foot placement safety constraints, $\mathcal{U}$ [8]. The MPC problem is formulated over N_s footstep as follows:

$$\begin{aligned} \min_{\mathbf{U}_{\text{fp}}} \quad & \|\mathbf{x}_{e,N_{\Delta t}N_s}\|^2_{\mathbf{Q}_f} + \sum_{k=0}^{N_{\Delta t}N_s-1} \|\mathbf{x}_{e,k}\|^2_{\mathbf{Q}_k} \\ \text{subject to} \quad & (2), \\ & \forall \mathbf{x}_k \in \mathcal{X}^{\text{kin}} \cup \mathcal{X}^{\text{slip}} \text{ and } \forall \mathbf{u}_{\text{fp},k} \in \mathcal{U}, \\ & \mathbf{Q}_k = \mathbf{0}, \ \ \forall k \notin \{N_{\Delta t}, \dots, N_{\Delta t}(N_s - 1)\} \\ & \mathbf{x}_0 = e^{\mathbf{A}T_r} R_{\text{MPC}}^\top \mathbf{x}_{\text{cm}} \end{aligned} \tag{3}$$

where $\mathbf{U}_{\text{fp}} = [\mathbf{u}_{\text{fp},0}^\top, \mathbf{u}_{\text{fp},N_{\Delta t}}^\top, \cdots, \mathbf{u}_{\text{fp},N_{\Delta t}(N_s-1)}^\top]^\top$ and $\mathbf{x}_{e,k} := \mathbf{x}_k - \mathbf{x}_k^{\text{des}}$. The desired state at each step transition is constructed via the solution of a periodic orbit, similar to [7], [8], where $L_x^{\text{des}} := L_x^{\text{main}} + L_x^{\text{offset}}$, with L_x^{offset} being an additional lateral angular momentum term. $\mathbf{x}_0$ is the predicted state just before impact during the current step using (1). T_r denotes the remaining duration of the current step, R_{MPC} is the rotation matrix considering the desired yaw orientation of the torso, and $\mathbf{x}_{\text{cm}}$ represents the current state measurement. We also note that the above MPC struggles with continuous turning motions within its N_s steps prediction horizon because the ALIP model does not inherently consider the rotational dynamics [24]. Therefore, we utilize the torso yaw rate command $\dot{\Theta}_z^{\text{des}}$ to compute the desired foot orientation command γ_k^0 as a one-step ahead prediction, similarly to [7]. This MPC formulation is rewritten in the form of a quadratic program (QP). After solving this QP, only the first optimal solution, $\mathbf{u}_{\text{fp},0}$, is utilized.

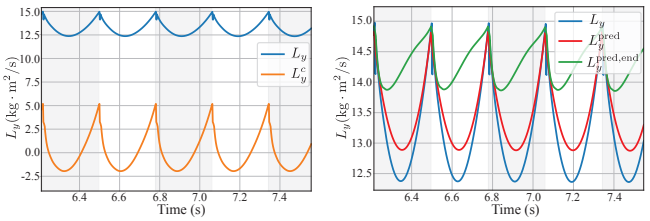

Fig. 2. **ALIP-model limitation:** Comparison of the angular momentum about the robot's contact point, L_y, angular momentum about its CoM (L_y^c) (left), and the predicted evolution L_y^{pred}, using the forward simulation of the ALIP model. The initial state of L_y is taken at the time of the step transition, and the predicted evolution at the end of the step, $L_y^{\text{pred, end}}$, is shown at every time instance during foot swing (right), while the robot walks forward using ALIP-based MPC [8] and WBC [28]. Notice that the blue and red lines are noticeably different and $L_y^{\text{pred, end}}$ fluctuates considerably.

B. Whole-body Control (WBC)

Several approaches [25], [26] exist for computing joint commands to execute high-level objectives (e.g., desired foot positions) in task space. In particular, given multiple task objectives, WBC [3], [27], [28] takes into account the full-order dynamics model of a robot to compute the optimal joint commands that minimize the tracking error of the tasks hierarchically based on state feedback. It also considers several constraints, including contact constraints and actuator limits, and is usually formulated as inverse dynamics with QP to offer real-time computation.

C. Model-free Reinforcement Learning (RL)

The RL problem is described as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, which consists of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a state transition function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a discount factor γ. The goal of RL is to find a policy parameterized by a parameter θ, $\pi_\theta^\star : \mathcal{S} \to \mathcal{A}$, that maximizes the expected discounted reward:

$$\pi_\theta^\star = \underset{\pi_\theta}{\text{argmax}} \ \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) \right], \tag{4}$$

where $s_k \in \mathcal{S}$, $a_k \in \mathcal{A}$, and $\tau \sim \pi_\theta$ is a trajectory drawn from the policy π_θ.

III. PROBLEM STATEMENT

The effectiveness of the ALIP-based MPC process to generate an optimal foot location for agile and robust bipedal locomotion highly depends on the accuracy of the ALIP model relative to the full-order robot model. Although the ALIP model is effective for controlling a variety of bipedal robots [7], the assumptions underlying the model described in Section II-A.1 are invalid for realistic robots, like our DRACO 3 robot, which has a relatively large distal mass on the leg [28], as illustrated in Fig. 2. This model discrepancy results in a significant difference between the actual (L_y simulated with the full-order model) and predicted (L_y^{pred} simulated with the ALIP model) values, causing the predicted value at the end of the step ($L_y^{\text{pred, end}}$) to be inaccurate while the foot is in the swing phase of the walking gait.

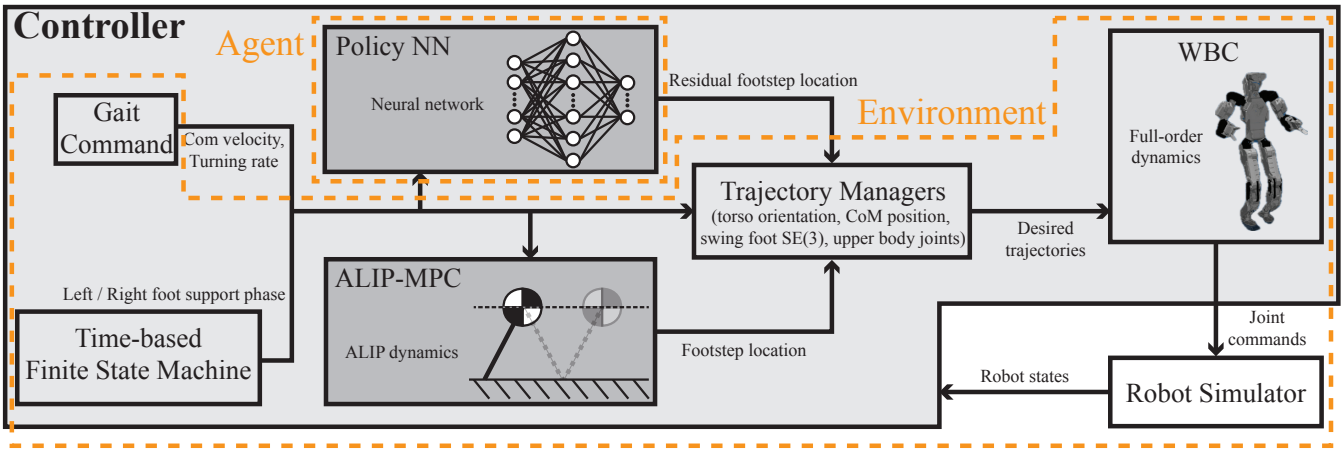

Fig. 3. **Reinforcement learning framework:** The agent learns a policy neural network (NN) in simulation, where the robot is controlled through the modules in the controller. The ALIP-based model predictive controller (MPC) follows a gait command and operates based on a time-based finite state machine (FSM), which manages the footstep timing. The ALIP-MPC process optimizes the desired footstep location, while the trajectory managers generate the swing foot trajectories based on the footstep locations. The trajectory managers also govern the desired task trajectories to harness the full-order model. All desired trajectories are sent to the whole-body controller (WBC) [28] to generate joint commands. In this learning framework, the policy NN generating residual footstep locations is the agent, while everything in the closed-loop system is considered the environment.

Consequently, this renders the ALIP-based MPC footstep planner inadequate for agile and robust locomotion with multiple replanning steps during swing.

Our goal is to circumvent this problem by appropriately designing an RL policy in conjunction with the MPC footstep planner, ensuring that the resulting policy incorporates the residual actions considering the full-order robot model. This approach overcomes the limitations of the ALIP-based MPC caused by modelling simplifications, without the need to solve a computationally challenging nonlinear optimization problem online with more complex dynamic models. Instead, it relies on simulation data collected offline that captures the robot's whole-body dynamics, allowing for efficient online deployment without substantial computation.

IV. APPROACH

This section introduces our proposed approach to design a footstep policy that enables agile and robust bipedal locomotion by combining model-based optimal control and model-free RL methods.

A. System Overview

Our method combines the ALIP-based MPC described in subsection II-A with an RL approach. The RL policy is to impose supplementary actions onto the MPC to amplify the agility and robustness of the locomotion capability. Unlike the end-to-end RL, this synthesis provides sample-efficient and reliable policy learning by utilizing the MPC solution as prior knowledge. Furthermore, this fusion enhances the resulting policy's capability to adapt to challenging scenarios that the MPC process alone cannot address due to modelling simplifications.

The proposed learning framework, illustrated in Fig. 3, devises a residual footstep policy that generates task-space actions based on the robot's current states and gait commands. In each training episode, the gait commands are randomly sampled from a predefined range. Given a swing duration, the MPC process produces a dynamically consistent footstep plan that satisfies the proxy robot's kinematic and friction cone constraints in the ALIP model manifold. Starting from the MPC footstep policy, the RL agent effectively explores the robot's nonlinear whole-body dynamics manifold, which is facilitated by our proposed MDP design. Upon completing the training process, the optimal footstep location, formulated by combining the residual footstep policy with the ALIP-based MPC solution, is utilized for locomotion control.

B. MDP formulation

Our goal is to learn a residual policy π_r to achieve an improved final policy π:

$$\pi(s) = \pi_0(s) + \pi_r(s), \tag{5}$$

where $\pi_0(s)$ is the initial policy generated from the ALIP-based MPC process. Given an MDP $\mathcal{M}$ process with a fixed initial policy $\pi_0(s)$, we define the residual MDP $\mathcal{M}^{(\pi_0)} = (\mathcal{S}, \mathcal{A}, \mathcal{P}^{(\pi_0)}, R^{(\pi_0)}, \gamma)$ similarly to the approach in [29], where:

$$P^{(\pi_0)}(s_k, a_k, s_{k+1}) = P(s_k, \pi_0(s_k) + a_k, s_{k+1}),$$
$$R^{(\pi_0)}(s_k, a_k, s_{k+1}) = R(s_k, \pi_0(s_k) + a_k, s_{k+1}).$$

The residual $\pi_r(s)$ is the policy in the residual MDP, which can then be solved using standard RL algorithms. In the following subsection, we describe the details of the residual MDP.

1) Action Space: Considering the initial footstep policy, $\pi_0(s)$, obtained from the ALIP-based MPC described in subsection II-A where the initial action is defined as $a_k^0 = (\mathbf{u}_{\text{fp},k}^0, \gamma_k^0)$, we design the residual action as follows:

$$a_k = (\mathbf{u}_{\text{fp},k}^{\text{res}}, \gamma_k^{\text{res}}), \tag{6}$$

where $\mathbf{u}_{\text{fp},k}^{\text{res}} \in \mathbb{R}^2$ denotes the position deviation from the footstep position $\mathbf{u}_{\text{fp},k}^0$ in the x and y directions, and $\gamma_k^{\text{res}} \in \mathbb{R}$

represents the angle deviation from the footstep yaw γ_k^0. Note that we only consider a 3-dimensional action space assuming the terrain slope is constant. However, this approach can be extended to a more general action space that includes foot SE(3) representations. This choice of residual action space allows the final policy to learn using the full-order nonlinear dynamics of the bipedal robot, ulinke the use of the initial footstep policy, which relies on the simplified ALIP model.

2) Observation Space: The observation space uses the states given by the ALIP model. We augment it by including additional states related to the robot's task space and a gait-related temporal variable. This state design choice enables the residual policy to better learn from the robot's full-order dynamics. These states are the following:

$$s_k = (\sigma_k, T_{rk}, \boldsymbol{\alpha}_k, \boldsymbol{\psi}_k, \boldsymbol{\beta}_k, a_{k-1}^0 + a_{k-1}), \tag{7}$$

where $\sigma_k \in \{-1, 1\}$ is an indicator of which foot is in stance, $T_{rk} \in \mathbb{R}_+$ is the remaining foot swing time, $\boldsymbol{\alpha}_k = (x_c, y_c, z_c, L_x, L_y, L_z) \in \mathbb{R}^6$ is the state of the ALIP model with addition of the z components, $\boldsymbol{\psi}_k = (\boldsymbol{\phi}_k^{\text{torso}}, \boldsymbol{\phi}_k^{\text{sw}}, \boldsymbol{\omega}_k^{\text{torso}}) \in \mathbb{R}^9$ is the robot's orientation-related states with $\boldsymbol{\phi}_k^{\text{torso}}, \boldsymbol{\phi}_k^{\text{sw}}$ being the orientations of the torso and swing foot in Euler angle respectively, and $\boldsymbol{\omega}_k^{\text{torso}}$ being the angular velocity of the torso. $\boldsymbol{\beta}_k = \left(L_x^{\text{offset}}, L_y^{\text{des}}, \gamma^{\text{des}}\right) \in \mathbb{R}^3$ is the desired commands at the end of the step, $\left(a_{k-1}^0 + a_{k-1}\right) \in \mathbb{R}^3$ is the policy's action taken in the previous step. Note that the linear velocity of the CoM is not included in the observation space as it is implicitly captured in the angular momentum about the contact point in $\boldsymbol{\alpha}_k$.

Including the most recent action from the previous step in the observation space is essential to ensure the smoothness of the swing foot trajectory. During footstep replanning while swinging, significant differences between two consecutive footstep solutions can lead to discontinuous swing foot trajectories, often rendering the robot's locomotion unstable and jerky. Therefore, incorporating the action from the previous step into the observation space allows it to be utilized in the reward function, promoting smoother and more stable locomotion.

3) Rewards: Given the set of early termination states $\mathcal{T}$ in an episode:

$$\mathcal{T} = \left\{ s \in \mathcal{S} \mid (z_c, L_{x,y}) \notin \left[z_c^{\min}, z_c^{\max}\right] \times \left[L_{x,y}^{\min}, L_{x,y}^{\max}\right] \right\}, \tag{8}$$

we design the reward function to ensure velocity tracking of the robot while realizing stable and robust locomotion. Additionally, the reward design enables the RL agent to explore beyond the simplified model (ALIP) state space by formulating the reward associated with the augmented observation and action space discussed in the previous subsection. The reward function is defined as follows:

$$R(s_k, a_k, s_{k+1}) = \begin{cases} 0 & \text{if } s_{k+1} \in \mathcal{T}, \\ r_{\text{sw}} & \text{if } s_{k+1} \notin \mathcal{T} \cap \sigma_{k+1} = \sigma_k, \\ r_{\text{end}} & \text{otherwise} \end{cases} \tag{9}$$

with:

$$\begin{aligned} r_{\text{end}} &= r_a + r_{L_x} + r_{L_y} + r_\gamma + r_{z_H} + r_\phi, \\ r_{\text{sw}} &= \widetilde{r}_{L_x} + \widetilde{r}_{L_y} + r_\pi, \end{aligned}$$

where the details for each term are described as follows:

$$\begin{aligned} r_a &= \text{alive bonus (constant value)} \\ r_{L_x}(s_{k+1}) &= \text{Ker}_{L_x}(e_{L_{x,k+1}}), \\ r_{L_y}(s_{k+1}) &= \text{Ker}_{L_y}(e_{L_{y,k+1}}), \\ r_\gamma(s_k, s_{k+1}) &= \text{Ker}_\gamma(\phi_{\text{yaw},k+1}^{\text{torso}}), \\ r_{z_H}(s_{k+1}) &= -w_{z_H}|r_{k+1,z} - z_H|, \\ r_\phi(s_{k+1}) &= -w_\phi \left\|(\phi_{\text{roll},k+1}^{\text{torso}}, \phi_{\text{pitch},k+1}^{\text{torso}})\right\|_2^2, \\ \widetilde{r}_{L_x}(s_{k+1}) &= \text{Ker}_{\widetilde{L}_x}(e_{\bar{L}_x,k+1}), \\ \widetilde{r}_{L_y}(s_{k+1}) &= \text{Ker}_{\widetilde{L}_y}(e_{L_y,k+1}), \\ r_\pi(s_k, s_{k+1}) &= \text{Ker}_\pi(a_k - a_{k-1}), \end{aligned}$$

with

$$\begin{aligned} e_{L_{\{x,y\},k}} &= \left(L_{\{x,y\},k} - L_{\{x,y\},k}^{\text{des}}\right), \\ e_{\bar{L}_{x,k+1}} &= \max\left(0, \left|L_{x,k+1} - L_x^{\text{offset}}\right| - L_x^{\text{main}}\right), \\ \text{Ker}_v(e) &= \omega_v \exp\left(-\left(e/\sigma_v\right)^2\right). \end{aligned}$$

We have designed the above reward terms in order to enhance the tracking of the dynamics obtained by the ALIP-MPC process shown in (2). Specifically, r_{sw} and r_{end} are utilized during the intra-step and leg switching phases, respectively. The term r_a encourages the robot to take more steps, which is particularly effective at the beginning of the training. The terms r_{L_x}, r_{L_y} and r_γ incentivize the robot to follow the desired gait commands. The terms r_{z_H} and r_ϕ promote tracking the desired CoM height and torso pitch and roll, which in our case are kept constant. Similarly to the terms r_{L_x} and r_{L_y}, $\widetilde{r}_{L_x}$ and $\widetilde{r}_{L_y}$ encourage the tracking of the desired L_x and L_y in the intra-step phase. Finally, the term r_π encourages the final footstep policy to avoid excessive variations between the previous action and the current action.

C. Learning and Policy Network

Because we want to encourage the RL agent to explore the unknown residual dynamics between the full-order and ALIP models, we optimize our policy utilizing Proximal Policy Optimization (PPO) [30]. The residual policy, π_r, is parametrized by the neural network which outputs the 3-dimensional residual footstep location.

During the training phase, in order for the RL agent to encourage exploration, we employ the residual action drawn from the Gaussian distribution:

$$a_k \sim \mathcal{N}(\mu_\theta, \sigma(r)), \tag{10}$$

where μ_θ and $\sigma(r)$ denote the mean and covariance of residual actions. μ_θ is parameterized by θ while $\sigma(r)$ is a scheduled parameter that depends on both initial conditions and the training procedure. Our policy is structured with a multi-layer perception architecture with two fully connected

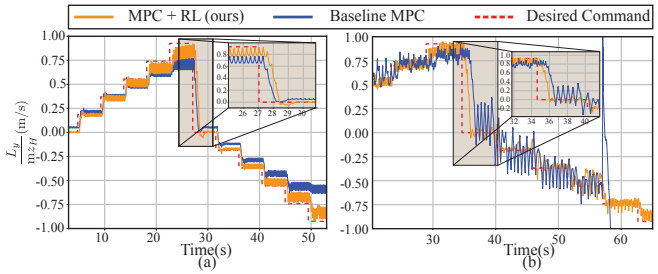

Fig. 4. **Velocity tracking during forward walking:** Comparison of the baseline MPC and our proposed MPC + RL method: (a) Lower frequency policy (b) Higher frequency policy

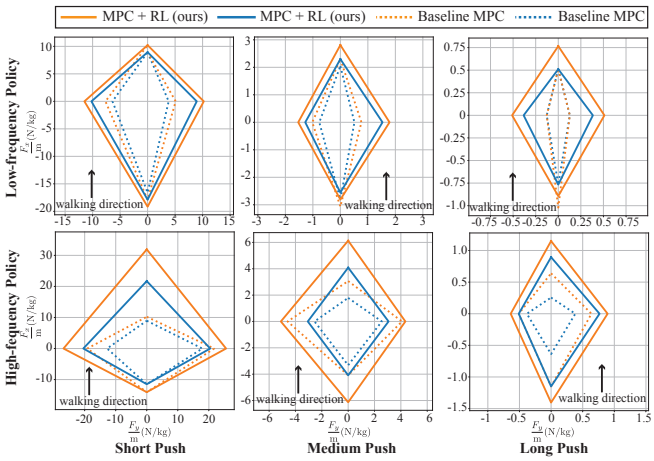

Fig. 5. **Force perturbation tests:** At each evaluation episode, incremental perturbation forces (pushes) were applied until the robot lost balance. The blue curve represents the minimum force at which the robot failed to maintain balance. The orange curve represents the maximum force with which the robot consistently maintained balance when starting the evaluation episode with that force. Unlike the blue curve, independent of previous perturbations. (Top row): Low-frequency policy (Bottom row): High-frequency policy.

hidden layers of 64 units and the $\tanh$ activation function. The output layer is bounded by a scaling factor to constrain the maximum value. To effectively facilitate the RL agent to start exploring around the MPC policy, we initialize the last layer of the neural network to be zero, similar to [29].

V. EXPERIMENTAL RESULTS

To evaluate the efficacy of our RL-augmented MPC footstep planner in achieving agile and robust omnidirectional locomotion as well as in traversing challenging terrains, we conducted several experiments in simulation. The baseline approach against which our proposed method was compared is the state-of-the-art MPC-based footstep planner in [8]. Additionally, to analyze the effects associated with the footstep planning frequency (i.e. the number of times we replan the trajectories during each footstep duration), we formulated a variation of our proposed MDP, that allows for planning footsteps once per foot swing, right at moment of the foot switching. This MDP is formulated by excluding intra-step dynamics-related components (T_{r_k} and $a^0_{k-1} + a_{k-1}$ in (7) and r_{sw} in (9)). For more details on the experiments conducted in this section, please refer to the accompanying video.

A. Experiment Setup

We performed simulations on the 25-DoF Humanoid robot, DRACO 3 [28], using Pybullet [31]. All control modules employed in this paper are written in C++ and integrated with Python using Pybind11 [32], enabling their use during the simulations and RL policy training. Throughout all test cases, we maintained uniformity by adhering to the baseline MPC specifications, including foot swing height, MPC prediction horizon, and MPC weights, and by using WBC [28] with identical task setups and gains.

During training, to ensure that the data is collected at a consistent rate, we maintained a fixed update rate of 114 Hz for the high-frequency MPC and 5 Hz for the low-frequency MPC to generate the desired footholds. Meanwhile, the WBC process calculated joint commands at approximately 600 Hz. The two different RL policies operated at their respective MPC frequencies, labeled as the high-frequency (planning multiple times per step) and low-frequency (planning once per step) policies. These configurations expedited the training process without the need to solve the MPC process at every WBC control loop.

B. Simulation Results

1) Velocity Tracking: This experiment aims to validate that the proposed approach can achieve fast, reliable walking in both sagittal and lateral directions. In the training setup, we randomized the velocity command between $\frac{L_y^{\text{des}}}{mz_H} \in [-0.925, 0.925]$ m/s. We tested our proposed MPC + RL policy for tracking a velocity profile in different walking directions. Fig. 4 shows the velocity tracking performance of our proposed policy against the baseline MPC in the sagittal direction. Unlike the baseline MPC, our policy demonstrated excellent tracking of the gait commands while walking, both in steady-state and through transient commands. We note that training the RL agent for different velocity commands within an episode enabled effective tracking of aggressive changes in the velocity command, as demonstrated in Fig. 4(b). This performance is crucial for enabling the robot to walk at human speeds, rapidly accelerating and decelerating for practical tasks. Note that we did not train for aggressive changes in the policy results shown in Fig. 4(a).

For lateral walking, we used a training setup similar to the one used for sagittal walking, but with L_x^{des} as the target angular momentum component. Our combined MPC+RL policy efficiently enabled the robot to track velocities within the range of $[-0.83, 0.83]$ m/s while avoiding self-collisions. In contrast, the baseline MPC was only able to track lateral velocities within the range of $[-0.6, 0.6]$ m/s, as indicated by the accompanying video.

2) Turning in place: To evaluate the performance of our approach on tasks where the ALIP-based MPC process fails, we consider a continuously turning task. The difficulty of such motion arises from the use of the ALIP model, which

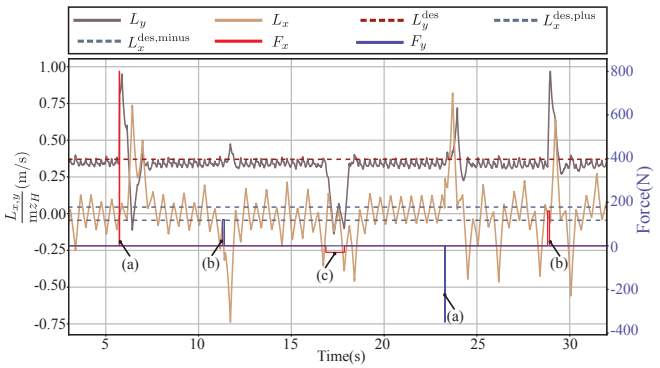

Fig. 6. **Velocity tracking under external disturbance:** The performance of the proposed high-frequency policy under various types of pushes: (a) Short push, (b) Medium push, (c) Long push.

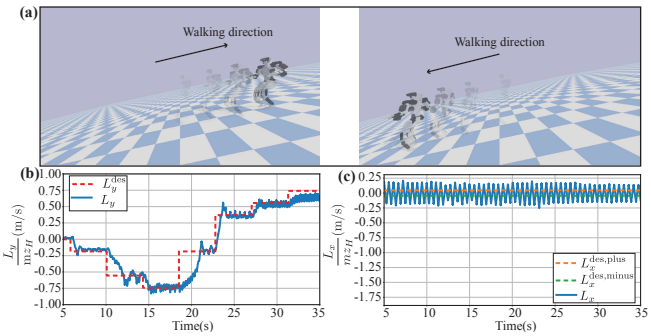

Fig. 7. **Sloppy terrain walking:** The tracking performance of the proposed high-frequency policy on a 11.5-degree slope: (a) Snapshots of walking in the sagittal direction, (b) Velocity tracking performance in the sagittal direction, (c) Velocity tracking performance in the lateral direction.

does not include rotational dynamics. During training, we considered the desired turning rate $\dot{\Theta}_z^{\text{des}} = 1.27$ rad/s corresponding to $\gamma^{\text{des}} = 0.35$ rad/step.

For such scenario, we define the performance metric of the policy by its capability to perform a full 360° turn without losing balance while tracking the desired yaw rate commands. The high-frequency policy enabled the robot to accomplish an average number of 2.7 turning motions per episode with a yaw rate of 1.27 rad/s, which was 350% better than the baseline MPC policy, which consistently failed to achieve one full 360° turn. On the other hand, the low-frequency policy allowed for tracking a yaw rate of up to 1.9 rad/s without falling, while the baseline MPC was only able to perform an average of 7 steps per episode corresponding to 40% completion of a full turn. This improved stability while turning-in-place is achieved thanks to our MPC + RL footstep policy, effectively exploring beyond the base footstep solution space limited by the use of the simplified ALIP model.

3) Robustness against Disturbances: In this experiment, we study the effectiveness of our policies in handling external disturbances. During training, we considered two types of force disturbances (pushes): long push with a duration of 1 s and short push with a duration of 0.0175 s. We selected 35 N for the long push and 350 N for the short push, where the magnitude of the force is close to the robot's weight. We randomly sampled the type of pushes, as well as their direction (either sagittal or lateral) and timing (any time during the foot swing) applied to the robot's torso link. We chose a forward walking gait at 0.37 m/s.

We assessed the robustness of our policy in rejecting external disturbances by applying three types of pushes, including the two types of pushes during training and a medium push with a duration of 0.1 s. Fig. 5 shows the maximum force magnitude (scaled by the robot's mass) for different directions that our policies and the baseline MPC can withstand without losing balance. Our policies outperform the baseline MPC in resisting pushing forces for all directions and all types of pushes while maintaining the forward walking motion. Notably, our higher frequency policy demonstrated superior performance to the lower frequency policy while showcasing remarkable robustness to the extensive range of perturbation not considered during training. This result can be attributed to our novel MDP formulation to recompute new footstep locations during the swing motion. Moreover, the higher frequency policy excelled in tracking velocity effectively under various intensive external pushing forces, as depicted in Fig. 6.

4) Walking in Sloppy Terrain: To assess the applicability of our policy to challenging terrains, we tested it in terrains with slopes up to 11.5 degrees by evaluating the speed tracking performance. In this scenario, we assumed the robot's perception system accurately estimated the slope of the terrain. In contrast to the baseline MPC, our proposed high-frequency policy demonstrated a good speed-tracking performance while walking in a sagittal direction on every slope. Fig. 7 shows a tracking performance of our high-frequency policy on a slope of 11.5 degrees.

5) Sample Efficiency: To validate the sample efficiency of our approach, we compared it with an RL-only approach that learns the task space policy similarly to [20], but with modifications to allow re-planning during the foot swing duration. Our approach demonstrated a faster convergence rate thanks to the initial MPC policy.

VI. CONCLUSION AND FUTURE WORK

This work presents a unique online bipedal footstep planning framework capable of reliably and efficiently adjusting planned footsteps to achieve precise and robust velocity tracking for agile and robust dynamic locomotion. Our approach combines a simplified model-based MPC footstep policy with a model-free RL policy. By leveraging RL's flexibility and adaptability, the resulting policy efficiently handles more complex locomotion tasks that challenge the simplified model-based MPC footstep planning process. We showcase our experimental results during various locomotion tasks, including tracking a wide range of walking speeds while traversing flat and sloppy terrain and exhibiting reliable turning behaviors. This demonstrates a significant enhancement in dynamic locomotion performance compared to the baseline simplified model-based MPC.

While our approach has shown effectiveness in dynamic locomotion, there are exciting areas for future exploration. First, the action space of our policy is limited to a 3-dimensional footstep task space. This action space can be extended to a larger dimensional task space (e.g., by including CoM height and orientation) to further handle modelling errors introduced by the use of simplified models, leading to improved stability and adaptability of locomotion. Second, as an extension of our work, loco-manipulation policies can be similarly constructed by integrating the existing model-based (e.g., centroidal dynamics [33]) controller with an RL policy to enhance the robustness and versatility of the model-based controller.

ACKNOWLEDGMENT

This work was supported by the Office of Naval Research (ONR), Award No. N00014-22-1-2204.

REFERENCES

[1] M. H. Raibert, H. B. Brown, M. Chepponis, E. Hastings, J. Koechling, K. N. Murphy, S. S. Murthy, and A. J. Stentz, "Stable Locomotion," *Order A Journal On The Theory Of Ordered Sets And Its Applications*, no. 4148, 1983.

[2] J. Ahn, D. Kim, S. Bang, N. Paine, and L. Sentis, "Control of a high performance bipedal robot using viscoelastic liquid cooled actuators," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2019-Octob, pp. 146–153, 2019.

[3] 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," *International Journal of Robotics Research*, vol. 39, no. 8, pp. 936–956, 2020.

[4] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 12 2006, pp. 200–207.

[5] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 8 2012.

[6] M. J. Powell and A. D. Ames, "Mechanics-based control of under-actuated 3D robotic walking: Dynamic gait generation under torque constraints," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2016-Novem. IEEE, 10 2016, pp. 555–560.

[7] Y. Gong and J. W. Grizzle, "Zero Dynamics, Pendulum Models, and Angular Momentum in Feedback Control of Bipedal Locomotion," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 144, no. 12, 12 2022.

[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 *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2022, pp. 6724–6731.

[9] J. Lee, S. H. Bang, E. Bakolas, and L. Sentis, "MPC-Based Hierarchical Task Space Control of Underactuated and Constrained Robots for Execution of Multiple Tasks," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2020-Decem, no. Cdc, pp. 5942–5949, 2020.

[10] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[11] S. H. Bang, J. Lee, C. Gonzalez, and L. Sentis, "Variable Inertia Model Predictive Control for Fast Bipedal Maneuvers," 7 2024. [Online]. Available: http://arxiv.org/abs/2407.16811

[12] G. Bledt and S. Kim, "Extracting Legged Locomotion Heuristics with Regularized Predictive Control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 406–412, 2020.

[13] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online Non-linear Centroidal MPC for Humanoid Robot Locomotion with Step Adjustment," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 10 412–10 419.

[14] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. Van De Panne, "Feedback Control for Cassie with Deep Reinforcement Learning," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1241–1246, 2018.

[15] J. Siekmann, S. Valluri, J. Dao, L. Bermillo, H. Duan, A. Fern, and J. Hurst, "Learning Memory-Based Control for Human-Scale Bipedal Locomotion," *Robotics: Science and Systems*, 2020.

[16] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, no. Icra, pp. 2811–2817, 2021.

[17] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning," in *ACM Transactions on Graphics*, vol. 36, no. 4. Association for Computing Machinery, 2017.

[18] K. Green, Y. Godse, J. Dao, R. L. Hatton, A. Fern, and J. Hurst, "Learning Spring Mass Locomotion: Guiding Policies With a Reduced-Order Model," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3926–3932, 4 2021.

[19] H. Duan, J. Dao, K. Green, T. Apgar, A. Fern, and J. Hurst, "Learning Task Space Actions for Bipedal Locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2021-May. IEEE, 5 2021, pp. 1276–1282.

[20] G. A. Castillo, B. Weng, S. Yang, W. Zhang, and A. Hereid, "Template Model Inspired Task Space Learning for Robust Bipedal Locomotion," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2023, pp. 8582–8589.

[21] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "RL + Model-Based Control: Using On-Demand Optimal Control to Learn Versatile Legged Locomotion," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6619–6626, 2023.

[22] Y.-M. Chen, H. Bui, and M. Posa, "Reinforcement Learning for Reduced-order Models of Legged Robots," 10 2023. [Online]. Available: http://arxiv.org/abs/2310.09873

[23] J. Ahn, J. Lee, and L. Sentis, "Data-Efficient and Safe Learning for Humanoid Locomotion Aided by a Dynamic Balancing Model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4376–4383, 2020.

[24] D. E. Orin, A. Goswami, and S. H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 10 2013.

[25] K. Sreenath, H. W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[26] L. Sentis, Jaeheung Park, and O. Khatib, "Compliant Control of Multicontact and Center-of-Mass Behaviors in Humanoid Robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, 6 2010.

[27] 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, 1 2022.

[28] S. H. Bang, C. Gonzalez, J. Ahn, N. Paine, and L. Sentis, "Control and evaluation of a humanoid robot with rolling contact joints on its lower body," *Frontiers in Robotics and AI*, vol. 10, no. October, pp. 1–21, 2023.

[29] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual Policy Learning," 12 2018. [Online]. Available: http://arxiv.org/abs/1812.06298

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[31] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," 2016.

[32] Wenzel Jakob and Jason Rhinelander and Dean Moldovan, "pybind11 — Seamless operability between C++11 and Python," 2017.

[33] M. Murooka, M. Morisawa, and F. Kanehiro, "Centroidal Trajectory Generation and Stabilization Based on Preview Control for Humanoid Multi-Contact Motion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8225–8232, 2022.