Learning Agile Locomotion and Adaptive Behaviors via RL-augmented MPC

Yiyu Chen and Quan Nguyen

***Abstract*— In the context of legged robots, adaptive behavior involves adaptive balancing and adaptive swing foot reflection. While adaptive balancing counteracts perturbations to the robot, adaptive swing foot reflection helps the robot to navigate intricate terrains without foot entrapment. In this paper, we manage to bring both aspects of adaptive behavior to quadruped locomotion by combining RL and MPC while improving the robustness and agility of blind legged locomotion. This integration leverages MPC's strength in predictive capabilities and RL's adeptness in drawing from past experiences. Unlike traditional locomotion controls that separate stance foot control and swing foot trajectory, our innovative approach unifies them, addressing their lack of synchronization. At the heart of our contribution is the synthesis of stance foot control with swing foot reflection, improving agility and robustness in locomotion with adaptive behavior. A hallmark of our approach is robust blind stair climbing through swing foot reflection. Moreover, we intentionally designed the learning module as a general plugin for different robot platforms. We trained the policy and implemented our approach on the Unitree A1 robot, achieving impressive results: a peak turn rate of 8.5 rad/s, a peak running speed of 3 m/s, and steering at a speed of 2.5 m/s. Remarkably, this framework also allows the robot to maintain stable locomotion while bearing an unexpected load of 10 kg, or 83% of its body mass. We further demonstrate the generalizability and robustness of the same policy where it realizes zero-shot transfer to different robot platforms like Go1 and AlienGo robots for load carrying. Code is made available for the use of the research community at `https://github.com/DRCL-USC/RL_augmented_MPC.git`**

I. Introduction

In the quest for the practical deployment of quadruped robots in real-world scenarios, the integration of adaptive behavior into their motion remains a challenge. This adaptive behavior consists of two essential dimensions: 1) real-time adaptation to external perturbations, and 2) self-adjustments such as foot reflection when a robot's foot gets stuck in an obstacle. Current advancements in legged mobility predominantly lean on Model Predictive Control (MPC) and Reinforcement Learning (RL). MPC, which employs real-time optimization over a set horizon to compute the optimal control sequence, often requires substantial computational resources and careful parameter tuning. RL, while notable for exceptional adeptness at navigating uneven terrains and unexpected disturbances, demands extensive offline computation, and careful reward tuning, and often produces policies tailored to specific robots. To combine the benefits of both MPC and RL, we present an innovative approach synthesizing the strengths of model-based control and reinforcement learning. Our central objective is to bolster agility, robustness, and adaptive behavior in blind locomotion through the integration of stance foot control and swing foot reflection using RL.

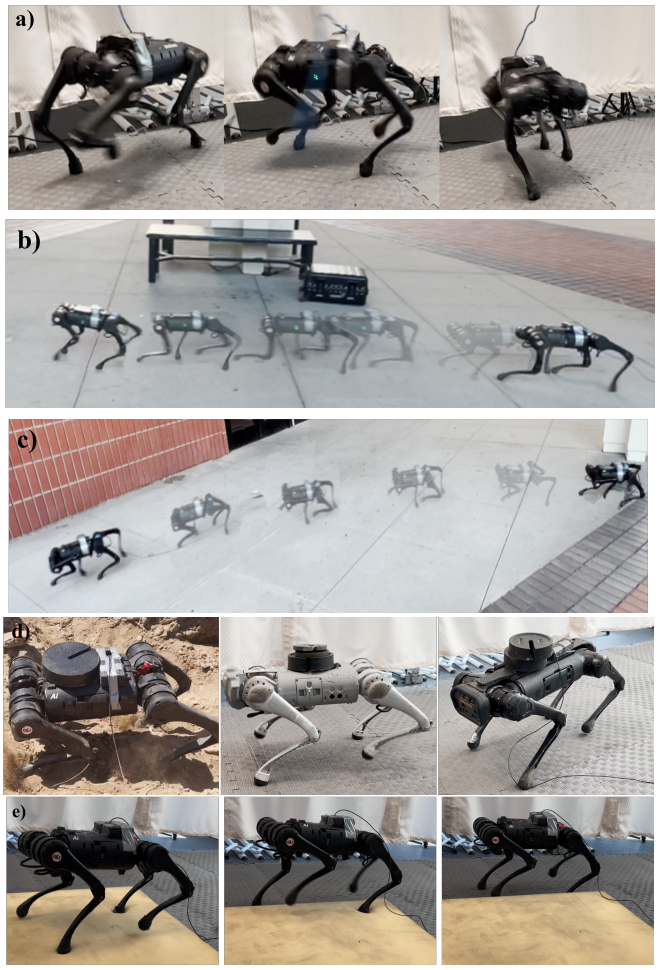

Fig. 1. Experiment result highlights. a) High-speed steering in place; b) High-speed running; c) High-speed running and steering; d) Generalization of the same policy across different robot platforms; e) Transition between soft and hard terrain. Experiment video: `https://www.youtube.com/watch?v=HxSIxTnEw08`

MPC, as validated by multiple studies [1]–[7] has gained traction in the legged robot community for its capability to handle the hybrid nature of quadrupedal locomotion under constraints. Central to the MPC paradigm is its prediction based on simplified dynamics, offering a future state estimation while preserving real-time computational feasibility. Yet, these simplified models inherently come with model uncertainties. For instance, the single rigid dynamics (SRB)

This work is supported in part by National Science Foundation Grant IIS-2133091. The opinions expressed are those of the authors and do not necessarily reflect the opinions of the sponsors.

Y. Chen and Q. Nguyen are with the Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089, email: yiyuc@usc.edu, quann@usc.edu.

model overlooks the resultant dynamics from leg momentum and external disturbance. Further complexities arise when translating these optimized trajectories into joint-level commands. Existing methodologies adopt a hierarchical control structure for this conversion, using techniques like Jacobian mapping [8], control barrier functions[9], and joint-level whole body control[2], [10]. The problem of addressing uncertainties in legged motion has been approached with adaptive control methods [11]–[17], adjusting the control parameters online. In addition, an implicit limitation of the MPC framework is its inherent decoupling of stance foot control from swing foot control due to the intricate modeling challenges of their interplay.

Reinforcement learning offers a tantalizing alternative [18]–[27], implicitly deciphering the dynamics interplay between the stance and swing control for all kinds of locomotion. In this paradigm, agents continually engage with environments, iteratively refining their action strategies based on the reward, resulting in the mastery of complex terrains and adaptive behavior attuned to environmental dynamics. Nevertheless, deploying RL in real-world scenarios raises legitimate concerns about its generalizability and safety. The aforementioned challenges underscore the urgency for a control framework evolution, one that concurrently addresses model uncertainties and optimizes swing foot reflection with regularized motions.

Researchers integrate MPC and reinforcement learning to combine the benefits of RL and model-based control. In [28], the RL framework utilizes model-based optimal control to generate reference motion and then leverages motion imitation technique[29] to learn versatile legged motion. RL [30]–[32] is also utilized to learn parameters or dynamics in the MPC problem. [33] proposes an RL-based control of the accelerations of an SRB model which allows robust sim-to-real transfer. [34] leverages RL to infer the set of unmodeled dynamics for the RMPC framework for adaptive locomotion. Additionally, [35] proposed an online supervised learning technique to derive a residue model to address the model uncertainties for model-based controller. What sets our research apart from these studies is our innovative synthesis of stance foot control and swing foot reflection by leveraging RL, enabling adaptive balancing and adaptive foot reflection within one unified framework.

In this paper, we present a novel RL-augmented MPC framework tailored to enhance blind locomotion for quadruped robots. By leveraging RL, we synthesize stance foot control and swing foot reflection from the convex MPC framework [1], specifically addressing the inherent issues of model uncertainty and the pre-defined swing foot trajectory. By tackling the dual challenges of adapting to model uncertainty and optimizing foot reflection, we successfully demonstrated improved agility, robustness, and adaptive behavior in blind legged locomotion. Notably, our research introduces robot-agnostic action and observation spaces to guarantee the policy's generalizability across various robot platforms. Our proposed framework has the following contributions:

- We introduce a novel RL-augmented MPC framework designed for adaptive blind quadruped locomotion, encompassing high-speed movement, uncertain dynamics adaptation, and reactive obstacle traversal.
- Our contribution uniquely combines stance foot force control with swing foot reflection, addressing model uncertainties and bridging foot swing and force control, overcoming inherent challenges in the nominal MPC framework.
- Our framework provides a robot-agnostic RL module for MPC, realizing zero-shot transfer across various robot platforms, and showcasing state-of-the-art performance on the Unitree A1, Go1, and AlienGo robots.

The paper is organized as follows: Sec. II presents our novel RL augmented MPC framework. Then, experimental validation is presented in Sec. III. Sec. IV provides concluding remarks.

II. Proposed Approach

A. System Overview

Illustrated in Fig. 2 is the overall system architecture, which is built upon [8]. The user provides velocity commands to the robot, while an event-driven finite state machine determines the gait schedule. Within the locomotion control module, the MPC is in charge of stance foot control. In contrast, the swing foot control determines the desired foot positions p_f. Force commands F are converted into joint torques using the Jacobian, and concurrently, the desired foot positions are mapped to corresponding joint angles through inverse kinematics. A Kalman filter facilitates state estimation, delivering proprioception data to both the locomotion control and the adaptive behavior policy.

Central to this system is our innovative adaptive behavior policy. Its primary aim is to impose supplementary actions onto the baseline MPC framework to bring adaptive behavior to the robot while ensuring performance across multiple robot platforms. This policy processes past commands, proprioception, acceleration from MPC force commands and heuristic foot placement. The result is dynamic compensation (explained in Sec. II-B) for the MPC and an offset joint angle Δq for swing foot reflection (explained in Sec. II-C). The adaptive behavior policy (explained in Sec. II-D), which learns to synthesize both the dynamics compensation essential for force control and the reaction for swing foot trajectory, given gait schedule and velocity commands. More than mere compensation, our policy amplifies the agility and robustness of the locomotion. Importantly, it achieves broad generalizability across different robot platforms without resorting to domain randomization.

B. MPC with Dynamics Compensation

To address the challenges of model uncertainties while retaining the generalizability across different robot platforms, we build upon the convex MPC setup in [1]. Model uncertainties inherently sprout from: 1) the model mismatch between the simplified model and hardware when abstracting from full-order dynamics, for instance, leg dynamics, and joint-level torque mapping. and 2) the disturbances applied

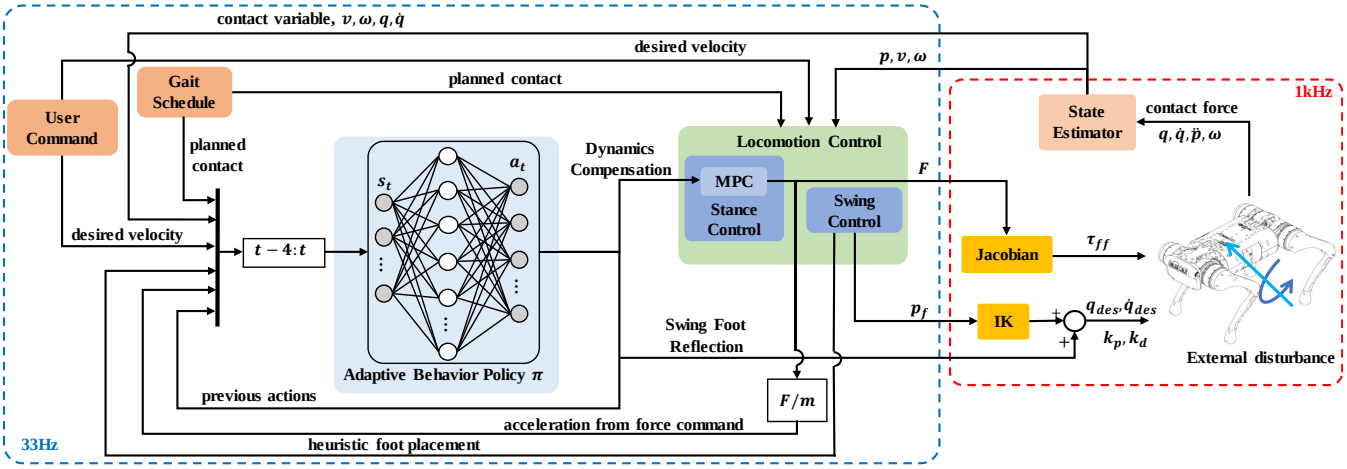

Fig. 2. System architecture of the proposed framework. The high-level module, framed in blue, includes the adaptive behavior policy and locomotion control module, operating at 33Hz. The low-level module, running at 1kHz, includes leg control (using Jacobian and IK), state estimation, and the robot's hardware. The F/m block normalizes the MPC force command into accelerations as a robot-agnostic input to the adaptive behavior policy.

to the robot from unknown disturbances, loads, and terrains. To capture these uncertainties, we introduce time-varying, locally-linear acceleration terms to incorporate into the linearized continuous-time state space equation. In this way, we incorporate a term in the MPC formulation to encompass different types of model uncertainties. The dynamics compensation terms are encapsulated $\Delta\boldsymbol{\alpha}$ and $\Delta\boldsymbol{a}$, which represent angular and linear accelerations respectively in the continuous-time state space equation:

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{\Theta}\\ \boldsymbol{p}\\ \boldsymbol{\omega}\\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{R}_z(\psi) & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{\Theta}\\ \boldsymbol{p}\\ \boldsymbol{\omega}\\ \dot{\boldsymbol{p}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_3 & \dots & \mathbf{0}_3\\ \mathbf{0}_3 & \dots & \mathbf{0}_3\\ \hat{\boldsymbol{I}}^{-1}[\boldsymbol{r}_1]_\times & \dots & \hat{\boldsymbol{I}}^{-1}[\boldsymbol{r}_n]_\times\\ \mathbf{1}_3/m & \dots & \mathbf{1}_3/m\end{bmatrix}\begin{bmatrix}\boldsymbol{F}_0\\ .\\ .\\ \boldsymbol{F}_n\end{bmatrix} + \begin{bmatrix}0_{3\times1}\\ 0_{3\times1}\\ \Delta\boldsymbol{\alpha}\\ \Delta\boldsymbol{a}+\boldsymbol{g}\end{bmatrix} \tag{1}$$

where $\boldsymbol{\Theta}$ represents the robot's orientation as a vector of Euler angels $[\phi, \theta, \psi]^T$, $R(\psi)$ is the rotation matrix corresponding to the yaw angle ψ, $\boldsymbol{p}$ and $\dot{\boldsymbol{p}}$ is the COM position and velocity of the robot, ω is the angular velocity of the robot, r_i is the vector from the robot's COM to foot i, F_i is the ground reaction force for leg i, I is the inertia, m is the mass of the robot, and $\boldsymbol{g}$ is the gravity term.

Equation (1) can be rewritten with an auxiliary state to represent the dynamics into a convenient state-space form:

$$\dot{\boldsymbol{x}}_c(t) = \boldsymbol{A}_c(\psi, \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{a})\boldsymbol{x}_c(t) + \boldsymbol{B}_c(\boldsymbol{r}_{1\dots n}, \psi)\boldsymbol{u}(t) \tag{2}$$

where

$$\boldsymbol{x}_c(t) = \begin{bmatrix}\boldsymbol{\Theta} & \boldsymbol{p} & \boldsymbol{\omega} & \dot{\boldsymbol{p}} & 1\end{bmatrix}^T \in \mathbb{R}^{13}$$

$$\boldsymbol{A}_c(t) = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{R}_z(\psi) & \mathbf{0}_3 & \mathbf{0}_{3\times1}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_{3\times1}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \Delta\boldsymbol{\alpha}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \Delta\boldsymbol{a}+\boldsymbol{g}\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0\end{bmatrix} \in \mathbb{R}^{13\times13} \tag{3}$$

Then, this formulation is discretized and formulated as a QP problem as in [1].

Opting for accelerations over forces and moments offers a broader view of disturbances to scale the dynamics for different robot platforms. This is particularly important considering robots vary in their ability to withstand external forces and moments. Consequently, we choose a metric that stands independent of the unique mass characteristics specific to each robot, recognizing that these attributes play a pivotal role in adaptability. It's noteworthy that even if the inertia of robots is usually minimal, any oversight in compensating moments can critically impair controller efficacy. Angular acceleration $\boldsymbol{\Delta\alpha}$, in this respect, offers a more intuitive and efficient mechanism to modulate the robot's orientation. To delve deeper into this nuance: the process of translating force/moment to acceleration inherently requires knowledge of the robot's mass and inertia. This perspective inherently considers the robot's mass and inertia, allowing our approach to seamlessly generalize across various robotic platforms regardless of mass and inertia differences. Moreover, this design choice also allows a compact formulation of the optimization problem as the size of all the matrices remains the same as in [1], which also facilitates onboard computation.

C. Adaptive Foot Swing Reflection

The swing foot in legged robots involves two main components: 1) foot placement and 2) swing trajectory. The foot placement determines the contact location which is crucial for stance control, while the swing trajectory would help the foot reflection to overcome obstacles. Our goal transcends the conventional scope of adaptive control that solely addresses model uncertainties. Instead, we seek to attain adaptive behavior in both the foot placement and swing trajectory, responding reactively to external disturbances, such as external loads or varying terrain.

In the baseline MPC framework, foot placement follows a predetermined heuristic[2] based on velocity commands, feedback, and stance time:

$$\boldsymbol{p}_{heuristic,i} = \boldsymbol{p}_{hip,i} + \frac{T_{stance}}{2}\boldsymbol{v} + k(\boldsymbol{v} - \boldsymbol{v}_{cmd}) + \frac{1}{2}\sqrt{\frac{z_0}{||g||}}\boldsymbol{v} \times \boldsymbol{\omega}_{cmd} \quad (4)$$

where $\boldsymbol{p}_{hip,i}$ is the hip location in the world frame for leg i, T_{stance} is the scheduled stance phase time, $\boldsymbol{v}$ is the velocity of the robot's COM, $\boldsymbol{v_{cmd}}$ is the velocity command, z_0 is the nominal height of locomotion, $\boldsymbol{\omega}_{cmd}$ is the yaw rate command and in this setup, we used a k of 0.03. We then employ a pre-defined Bezier curve to interpolate the foot swing trajectory, outputting the p_f as the desired foot location to the swing foot.

Our methodology places a heightened emphasis on adaptive swing reflection. Unlike traditional approaches that manually adjust trajectories, our system utilizes an offset joint angle, Δq, to modulate the nominal swing trajectory. This doesn't just modify foot placement; it also dynamically adjusts its swing trajectory over discrete obstacles. By prioritizing adaptive swing control, our system offers a more holistic and responsive solution, synchronously adjusting both the trajectory shape and final foot placement in real time, all under the guidance of one unified adaptive behavior policy. When combined with dynamic compensation, our action space enhances the robustness and agility of legged locomotion.

D. Learning to Synthesize Stance Control and Swing Control for Adaptive Behavior

In this section, we present our novel learning framework that integrates the strengths of the traditional MPC control approach with the dynamic adaptability offered by Reinforcement Learning (RL). While the baseline MPC framework excels in forward-looking predictions, RL is capable of reasoning over past experiences. Our primary goal transcends merely addressing model uncertainties; we strive to synthesize these decoupled control realms (stance foot control and swing foot control) using RL. This method unravels the intricate connections between stance foot and swing foot controls. This synthesis means that when force optimization is constantly evolving, enriched by insights from the swing foot's heuristic and past proprioception data. In parallel, the swing foot's trajectory and placement are fine-tuned based on cues from the force optimizations and past proprioception data. This seamless integration and reciprocal adaptation ensure that the robot exhibits adaptive behavior under different conditions, underscoring the power and efficacy of our proposed approach.

1) Action Space: Considering that the MPC problem intrinsically incorporates the robot's mass properties, we can design robot-agnostic action space to account for model uncertainty while ensuring generalizability. Our learning module computes both dynamics compensation components - namely $[\Delta\boldsymbol{\alpha}, \Delta\boldsymbol{a}]$ - and swing foot reflection in the form of joint angle offset Δq from the nominal swing trajectory. In other words, our adaptive behavior policy seeks to derive sacalable supplementary actions that can be seamlessly layered onto the locomotion controls for different robot platforms. This design ensures improved robustness and agility in locomotion performance and substantially facilitates the generalizability of the framework.

2) Observation Space: To ensure generalizability, the observation space must remain independent of the robot's mass properties, because the MPC problem inherently considers them. At every time t. the policy obtains an observation and performs supplementary actions to the nominal MPC control framework. As presented in Fig. 2, the observation for the policy takes a history window of 5 MPC horizons, capturing parameters including the joint angle and velocities $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, linear and angular velocities of the robot $\boldsymbol{v}_{com}$ and $\boldsymbol{\omega}_{com}$, planned contact boolean of every foot from the given gait schedule $\boldsymbol{s}_\phi$, the actual contact state of each foot from the contact sensor data $\boldsymbol{s}_{actual}$, desired COM state from user input $\boldsymbol{v}_{des}$ and $\boldsymbol{\omega}_{des}$, the heuristic foot placement of every foot $\boldsymbol{p}_{heuristic}$ and the acceleration from force commands of MPC optimization $\boldsymbol{F}/m$. Similar to dynamics compensation, the ground reaction force command is expressed in terms of acceleration. This consideration is pivotal for generalization across different robots, given that robots come with diverse mass properties, prompting the MPC to produce force commands on varying scales. By ensuring the observation space remains agnostic to a robot's unique internal attributes, we substantially facilitate the generalizability of our framework.

3) Training: In this paper, we employ PPO[36] for training and use the Unitree A1 robot in the simulation. The reward function at time t is designed to ensure velocity tracking of the robot while minimizing the energy cost.

$$R(t) = w_1 r_{survival} + w_2 r_{velocity} + w_3 r_{energy} + w_4 r_{height} \quad (5)$$

where

$$\begin{aligned} r_{survival} &= 1 \\ r_{velocity} &= ||\boldsymbol{v}_{des} - \boldsymbol{v}_{COM}|| + ||\boldsymbol{\omega}_{des} - \boldsymbol{\omega}|| \\ r_{energy} &= \sum_{i=1}^{12} ||\tau_i \dot{q}_i|| t \\ r_{height} &= 0.02 - ||z_{COM} - z_{des}|| \end{aligned} \quad (6)$$

and w_i are the corresponding weight factors for each reward term. We use the same reward function by varying weights for the 3 applications we validate our approach. In this work, we approximate the policy using MLPs with hidden layers of [256, 32, 256] neurons with tanh as activation function. The training of the MLP is performed offline with numerical simulation by Pybullet[37].

E. Learning with MPC in Simulation

In the architectural design of our training methodology, the MPC setup holds a central position. The integration of MPC becomes a computation bottleneck, prominently manifested in the form of optimization-induced latency. With the QPOASES[38] solver, the MPC problem's computation time averages $1ms$. In comparison, a simulation step in Pybullet is computed in less than $0.1ms$. The implication here is clear: updating the ground reaction force at each

simulation step would drastically decelerate simulation and consequently slow done training. Therefore, our strategy is to update the MPC problem at a less frequent rate, while the lower-level joint commands receive updates much more frequently. As illustrated in Fig. 2, we update the MPC problem every 30 ms (MPC horizon time) to generate the desired ground reaction force command while the Jacobian maps this force command to joint torques every 1 ms. We also have our policy to be set up with an update rate of 33 Hz. This configuration aligns with our hardware experiments, negating the need to constantly run the MPC solver for updating the ground reaction force at every control step. Thanks to this efficient setup, our approach not only expedites the training but also ensures the sim2real of the framework as it mirrors the exact setup used in our hardware experiments.

III. Experimental Validation

In this section, we detail the experimental validation of our framework, highlighting the enhanced robustness, agility, and adaptive behavior in blind locomotion. Our approach is rigorously tested across various computation platforms and robotic systems, with all policy learning conducted offline using the Unitree A1 robot. For the comparative study, we maintained uniformity by adhering to the baseline MPC specifications, which include gait patterns, foot swing height, MPC weights, and joint PD gains. Throughout the experiments, the robot was maneuvered using a joystick by the author. A seamless transfer of all policies to the actual hardware was made possible thanks to the robustness of the MPC controller and the generalizability of the learning framework.

A. High Speed Locomotion

The primary aim here is to validate our RL-augmented MPC methodology during challenging high-speed maneuvers—both running and turning. Due to their stark dynamical distinctions, these activities provide a rigorous testbed. We use a flying trot gait for these high-speed maneuvers. The learning domain comprises dynamics compensation factors and foot placement offsets, delineated as The angular acceleration within $\pm[2.0, 10.0, 2.0] rad/s^2$, the linear acceleration bounded by $\pm[4.0, 2.0, 3.0] m/s^2$ and the joint angle set to $\pm 0.3 rad$.

1) High Speed Turning: We validated our approach with high-speed turning in place and we are able to achieve state-of-the-art performance in terms of the turning rate. As presented in Fig. 3, we ramp up the yaw rate command to $\pm 7 rad/s$. Impressively, the A1 robot hit a peak turn rate of 8.5 rad/s in the counter-clockwise direction, maintaining an average yaw rate close to $7 rad/s$, while the baseline MPC couldn't survive this command and failed immediately. As demonstrated in the support video, the policy doesn't merely rely on adjusting foot placement to enhance motion. A significant contribution comes from the angular acceleration compensation in the roll direction as shown in Fig. 1. This intelligent adjustment by the policy facilitates smoother

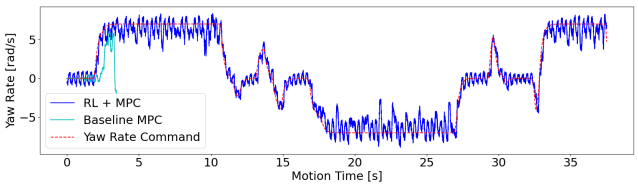

Fig. 3. Yaw rate plot of high speed turning policy from IMU data

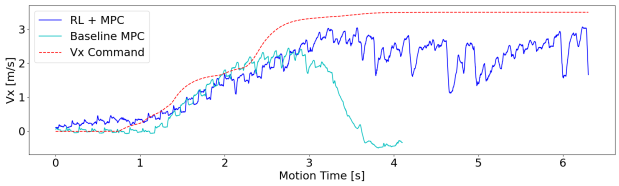

Fig. 4. Linear velocity in the body frame of high-speed running policy

turns, enabling us to register the swiftest turn ever recorded on the Unitree A1 robot.

2) High-Speed Running and Steering: Furthermore, we conducted tests involving high-speed running and steering. In experiments depicted in Fig. 4, when velocity commands were increased to $3.5 m/s$, the baseline MPC failed within a few running steps due to the model uncertainty at high speed. Contrarily, our learned policy enabled the robot to withstand disturbances at high speeds, achieving a peak velocity of around $3 m/s$ based on state estimation data. Further tests involved steering at high speed. We first accelerated the robot to $2.5 m/s$ and then applied a yaw rate command of $0.5 rad/s$. Fig. 5 suggests that our learned policy surpassed the baseline MPC in velocity tracking. Prior to turning, our RL-augmented MPC tracked the intended velocity of $2.5 m/s$, while the baseline lagged, peaking at roughly $2 m/s$. Integrating translation and rotation dynamics, our RL-augmented MPC deftly steered at $2.5 m/s$ as highlighted in Fig. 1. Meanwhile, the baseline MPC decelerates to $1 m/s$ to prevent failure.

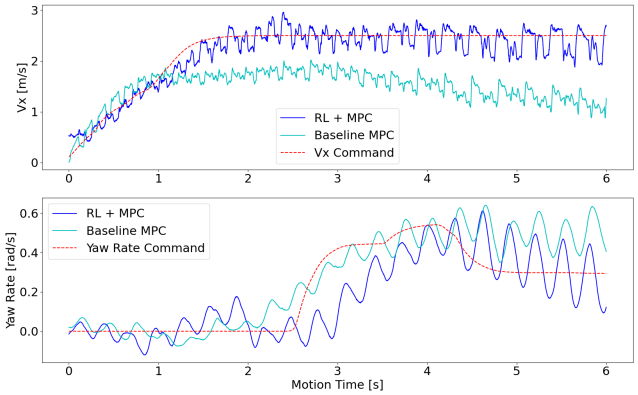

Fig. 5. Linear velocity in body frame and yaw rate of high-speed running and steering policy

B. Walking with Significant Model Uncertainty

In this section, we highlight our framework's effectiveness in managing model uncertainty and enhancing the

TABLE I

LOAD CARRYING CAPACITY OF THE SAME POLICY OVER DIFFERENT ROBOTS WITH TROTTING GAIT

Robot	Maximum Load Capacity
AlienGo	$10kg$
Go1	$7kg$
A1	$10kg$

robustness of locomotion. In the training setup, we randomize the velocity commands for the robot in body frame of $\boldsymbol{v}_x \in [-1, 1]m/s$, $\boldsymbol{v}_y \in [-0.5, 0.5]m/s$ and $\boldsymbol{\omega}_z \in [-2.0, 2.0]rad/s$ given a trotting gait of $0.3s$ gait cycle. We choose trotting as the test gait as it is more difficult than standing or quasi-static walking. Adding to this complexity, random external forces and moments are applied, challenging the robot further. Within this setting, agents learn dynamics compensation and foot placement offsets, with angular acceleration constrained to $\pm[4.0, 10.0, 2.0]rad/s^2$, linear acceleration at $\pm[4.0, 2.0, 8.0]m/s^2$ and the joint angle set to $\pm 0.3rad$.

Notably, even though the policy was primarily trained on flat terrain, it showcased remarkable adaptability on soft, uneven soil terrain, when the A1 robot carries an additional $5kg$ load, as depicted in Fig. 1. Moreover, the generalizability of our approach is evident as the policy learned on the A1 robot realizes zero-shot transfer to different robotic platforms like Go1 and AlienGo while adapting effortlessly to different gait cycle timings (see support video). This generalizability is credited to our scalable action space and observation space, which is independent of the robot's internal properties.

Our framework also excels in compensating for external moments. The exemplary adaptive behavior of our framework is evidenced in Fig. 6, where we introduce loads that impose additional moments on the robot's body. Despite these challenges, our adaptive behavior policy effectively mitigates the uncertainties. This performance can be attributed to the incorporation of angular acceleration as the key dynamics compensation term. Furthermore, our framework's computational efficiency stands out, comfortably running on a Raspberry Pi 4 board on the Go1 robot. The policy inference and MPC optimization respectively clock in at approximately $1ms$ and $3ms$ respectively on the Raspberry Pi 4 board.

C. Blind Walking over Discrete Terrain

We also validated our framework in the realm of adaptive foot swing reflection, specifically on discrete terrain. We trained the policy to traverse over randomly generated discrete terrains ranging from 8 to 12 cm in height, with the robot's foot swing height target fixed at $8cm$. The action space for this scenario are angular acceleration within $\pm[4.0, 10.0, 2.0]rad/s^2$, the linear acceleration at $\pm[2.0, 2.0, 2.0]m/s^2$ and the joint angle set to $\pm 0.3rad$.

In this setup, dynamic compensation handles unexpected ground contact, and adaptive foot behavior prevents foot entrapment. Impressively, this policy seamlessly transitioned to blind stair climbing of a stair height of $13cm$. As presented in Fig. 7, while the standard MPC often led to the robot's foot being trapped, our refined framework adopted adaptive swing trajectories, facilitating immediate foot reaction upon contact with the discrete obstacle(see support video). This emergent behavior is learned by the policy, and the MPC ensures the robot's stability and movement. The seamless integration is a testament to the effectiveness of our RL-augmented MPC approach for adaptive behavior.

(a) Baseline MPC

(b) RL-Augmented MPC

Fig. 6. Comparison between baseline and proposed approach in terms of pitch compensation. The Go1 robot is carrying a load of $5kg$ and the A1 robot is carrying a load of $10kg$. They all run the same adaptive behavior policy trained on the A1 robot.

(a) Baseline MPC (b) RL-Augmented MPC

Fig. 7. Comparison between baseline and proposed approach for blind stair climbing. The stair height is approximately 13cm and the foot swing height is all set to 8cm.

IV. CONCLUSION

In this research, we have presented an integration of Reinforcement Learning (RL) and Model Predictive Control (MPC), improving the agility and robustness of legged locomotion. Through the integration of MPC's forward-looking predictions and RL's ability to reason on past experiences, our framework has unveiled marked improvements in a robot's dexterity to traverse intricate landscapes and handle unexpected perturbations. Notably, the adoption of adaptive swing foot reflection showcases how the blend of these two methodologies can lead to real-world locomotion improvements. The extensive experimental validations presented further underscore the robustness and generalizability of our proposed RL-augmented MPC framework, setting a new benchmark for state-of-the-art robotic control systems. We envision that this synthesized approach will pave the way for future research such as perceptive locomotion, fostering even more resilient and adaptive behavior for legged robots.

REFERENCES

[1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[2] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

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

[4] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, 2023.

[5] W. Chi, X. Jiang, and Y. Zheng, "A linearization of centroidal dynamics for the model-predictive control of quadruped robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4656–4663.

[6] N. Rathod, A. Bratta, M. Focchi, M. Zanon, O. Villarreal, C. Semini, and A. Bemporad, "Model predictive control with environment adaptation for legged locomotion," *IEEE Access*, vol. 9, pp. 145 710–145 727, 2021.

[7] M. Sombolestan and Q. Nguyen, "Hierarchical adaptive control for collaborative manipulation of a rigid object by quadrupedal robots," *arXiv preprint arXiv:2303.06741*, 2023.

[8] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

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

[11] C. Cao and N. Hovakimyan, "L 1 adaptive controller for a class of systems with unknown nonlinearities: Part i," in *2008 American Control Conference*. IEEE, 2008, pp. 4093–4098.

[12] Q. Nguyen and K. Sreenath, "L 1 adaptive control for bipedal robots with control lyapunov function based quadratic programs," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 862–867.

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

[14] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[15] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 565–572, 2021.

[16] M. Sombolestan, Y. Chen, and Q. Nguyen, "Adaptive force-based control for legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7440–7447.

[17] M. Sombolestan and Q. Nguyen, "Adaptive force-based control of dynamic legged locomotion over uneven terrain," *arXiv preprint arXiv:2307.04030*, 2023.

[18] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[19] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *arXiv preprint arXiv:2205.02824*, 2022.

[20] L. Krishna and Q. Nguyen, "Learning multimodal bipedal locomotion and implicit transitions: A versatile policy approach," *arXiv preprint arXiv:2303.05711*, 2023.

[21] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[22] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1479–1486.

[23] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[24] X. Cheng, A. Kumar, and D. Pathak, "Legs as manipulator: Pushing quadrupedal agility beyond locomotion," *arXiv preprint arXiv:2303.11330*, 2023.

[25] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, *et al.*, "Genloco: Generalized locomotion controllers for quadrupedal robots," in *Conference on Robot Learning*. PMLR, 2023, pp. 1893–1903.

[26] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[27] Y. Jin, X. Liu, Y. Shao, H. Wang, and W. Yang, "High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1198–1208, 2022.

[28] 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," *arXiv preprint arXiv:2305.17842*, 2023.

[29] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[30] J. Sacks and B. Boots, "Learning to optimize in model predictive control," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 549–10 556.

[31] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning*. PMLR, 2022, pp. 773–783.

[32] Y. Yang, G. Shi, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots, "Cajun: Continuous adaptive jumping using a learned centroidal controller," *arXiv preprint arXiv:2306.09557*, 2023.

[33] Z. Xie, X. Da, B. Babich, A. Garg, and M. v. de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2022, pp. 523–539.

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

[35] Y. Sun, W. L. Ubellacker, W.-L. Ma, X. Zhang, C. Wang, N. V. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. D. Ames, "Online learning of unknown dynamics for model-based controllers in legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8442–8449, 2021.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[37] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[38] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.