Augmenting Learned Centroidal Controller with Adaptive Force Control

Barath Satheeshkumar
Carnegie Mellon University

Kailash Jagadeesh
Carnegie Mellon University

Tony Tao
Carnegie Mellon University

Nikhil Sobanbabu
Carnegie Mellon University

***Abstract*—We investigate the limitations of the CAJUN framework, a hierarchical learning and control architecture for legged robots that enables continuous jumping with adaptive jump distances. CAJUN employs reinforcement learning (RL) to generate a centroidal policy that determines high-level motion parameters such as gait timing, base velocity, and swing foot position. These parameters are then processed by a low-level controller that computes joint-level motor commands using a Quadratic Program (QP). While CAJUN performs well in nominal simulation settings, we observe significant performance degradation under sim-to-sim transfer with added payloads, indicating a lack of robustness to model uncertainty. To address this, we introduce a modified low-level control strategy that incorporates $\mathcal{L}_1$ adaptive control within the QP formulation. Our approach improves adaptability to dynamic variations and enhances robustness in transfer scenarios, enhancing resilience to model uncertainties during deployment. The codebase for this work can be found in Github:RL-AFQP**

***Index Terms*—Jumping, Legged Locomotion, Adaptive Control**

I. Introduction

Legged robots possess a unique ability to traverse complex and unstructured environments. Unlike wheeled or tracked systems, they can adaptively modulate foot placement and body posture to negotiate steep inclines, irregular surfaces, and constrained spaces [2], [21]. These features make them particularly well-suited for real-world tasks such as disaster response, firefighting, and search-and-rescue missions [18], [19]. However, deploying legged robots in such high-stakes scenarios requires not only agility and versatility, but also robustness to model inaccuracies and unpredictable external disturbances.

Conventional locomotion strategies typically rely on continuous walking gaits, which inherently restrict the robot's reach to within a single body length of its current stance [6]. This limits traversal over large gaps or disconnected footholds, such as stepping stones. Jumping offers a compelling solution, allowing robots to bridge larger distances and expand their operational workspace. However, designing reliable and adaptive controllers for continuous jumping remains a core challenge. While optimization-based methods offer precision, they often require computationally expensive trajectory planning and are limited by simplified online tracking controllers [6], [7]. On the other hand, learning-based approaches provide greater adaptability but are frequently hampered by complex reward shaping and poor transferability from simulation to the real world—particularly for underactuated, high-dynamic tasks like jumping [12].

To overcome these challenges, the CAJun framework [1] introduces a hierarchical control architecture that unites the strengths of optimization and learning. A high-level reinforcement learning (RL) policy predicts centroidal targets—such as desired base velocities, gait parameters, and footstep locations—while a low-level quadratic program (QP) controller tracks these targets at 500 Hz by computing joint torques. This architecture enables continuous, adaptive jumping and has been validated on the Unitree Go1 robot, which can clear 70 cm gaps, outperforming prior methods by over 40% [12].

Although jumping expands the terrain capabilities of legged robots, robustness to uncertainty remains a critical barrier to reliable real-world deployment. Most existing control methods assume access to accurate dynamics models and degrade significantly in the presence of model mismatch or disturbances. The L1 adaptive control framework [15] addresses this limitation by introducing an adaptive nature to address the model uncertainty. Its novelty is the decoupling of robustness and fast adaptation through a low-pass-filtered adaptation law, ensuring guaranteed stability and smooth transient behavior.

The L1 adaptive controller has been validated in both simulation and real-world experiments on the A1 quadruped platform [15]. It has demonstrated impressive robustness to unknown and time-varying payloads - successfully handling loads up to 92% of the robot's weight—where traditional non-adaptive baselines fail [20].

Together, the CAJun and L1 frameworks represent complementary advances in legged locomotion. CAJun improves reachability and agility through hierarchical learning-based jumping, while L1 improves reliability under uncertainty through robust force-based adaptation. This paper benchmarks the two frameworks in simulated and real-world scenarios and proposes hybrid approaches, such as integrating L1 adaptation into learned policies, to advance robust, agile and adaptable locomotion.

II. Related Works

Optimization-Based Control for Jumping. Optimization-based controllers have long been a cornerstone in legged locomotion, enabling behaviors ranging from periodic bounding and pronking to dynamic, high-amplitude leaps [7], [8]. These methods leverage high-frequency control loops to generate robust motions under significant perturbations, but often suffer

from computational bottlenecks that restrict online planning to short horizons and low-displacement jumps [9]–[11]. To address this, several works pre-compute reference trajectories offline via trajectory optimization (TO), extending jump height and distance significantly [6]. However, generalizing beyond these handcrafted trajectories to more adaptive, reactive behaviors remains a challenge. Notably, Park et al. [13] integrated a multi-level planning scheme to perform continuous bounding over hurdles with fixed gaits. In contrast, the CAJun framework introduces a more flexible formulation where the policy simultaneously modulates gait timing, base pose, and swing foot trajectory, allowing it to generalize across terrains and commands.

Hierarchical RL for Legged Locomotion. Recent advancements in legged robot control have combined RL with model-based lower-level controllers in hierarchical frameworks [14]. These systems typically employ a high-level RL policy that outputs gait timings, CoM trajectories, or footstep plans for a lower-level optimizer or controller to track. However, the need to solve optimization problems at every control step often results in prohibitively slow training times. CAJun [1] addresses this bottleneck by introducing a closed-form solution to the foot force optimization problem, significantly reducing per-step computation without sacrificing physical consistency. As a result, CAJun achieves orders-of-magnitude faster training—completing in just 20 minutes on a GPU—while maintaining task versatility and high-performance jumping.

Adaptive Control Adaptive control allows a controller to adjust its parameters online to handle model uncertainties [22], and has been widely used in robotic arms, mobile platforms, and quadrupeds [23]. Classical Model Reference Adaptive Control (MRAC) frameworks were developed for linear systems with parametric uncertainty [24], but they do not offer guarantees on transient response. To overcome this, $\mathcal{L}_1$ adaptive control introduces a low-pass filter in the adaptation loop, enabling separation of adaptation and robustness and offering fast, stable transients [25]. Recent work has combined $\mathcal{L}_1$ control with Bayesian learning to enable safe, online adaptation [26].

In legged robotics, adaptive control has been used for estimating the center of mass [27] and stabilizing locomotion using nonlinear control reference models like Control Lyapunov Functions (CLFs) [28]. These methods were validated on bipedal robots for walking and running, but rely on Hybrid Zero Dynamics [29] with joint-level trajectory tracking. Other works, such as [30], combine CLF-based adaptive constraints with MPC, though this increases computational cost due to added constraints.

III. Methodology

Figure 1 shows the overall control architecture. We begin with some preliminaries on the centroidal model of the quadruped, then present CAJun's Unconstrained QP solver, describe the $\mathcal{L}_1$ augmentation (with premise, continuous-time foundation, and discrete implementation) to the force-based

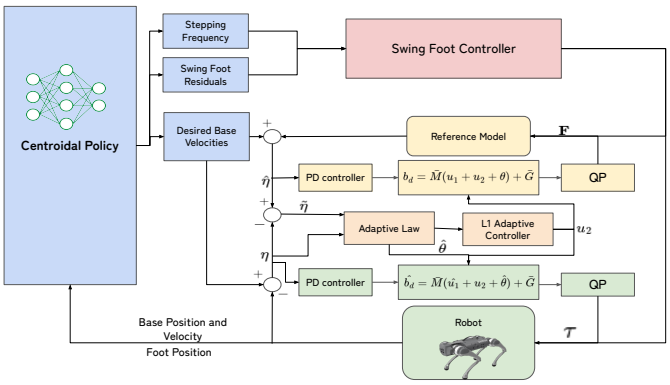


Fig. 1: Overall control architecture: High-Level Centroidal Policy feeding into the low-level QP solver (with $\mathcal{L}_1$ adaptation).

QP formulation, and finally specify some details of the centroidal policy of CAJun

A. Preliminaries: Centroidal dynamics

Let

$$x = (p, \theta) \in \mathbb{R}^6 \quad , \quad \dot{x} = (v, \omega) \in \mathbb{R}^6$$

be the base pose and velocity of the robot. Under rigid-body assumptions, stack the four foot forces into

$$f = \big[f_1; \ldots; f_4\big] \in \mathbb{R}^{12},$$

and write the discrete-time centroidal dynamics:

$$M_c\big(\dot{x}_{k+1} - \dot{x}_k\big) = u_k + d_k, \qquad u_k = \sum_{i=1}^{4} f_{i,k},$$

where

$$M_c = \begin{bmatrix} m^{-1} I_3 & 0 \\ 0 & I^{-1} \end{bmatrix}$$

is the 6×6 inverse inertia matrix (with mass m and inertia tensor I), $d_k \in \mathbb{R}^6$ is an unknown wrench disturbance, and

$$\ddot{x}_{\text{des}} \;=\; \frac{\dot{x}_{k+1} - \dot{x}_k}{\Delta t}$$

is the desired discrete acceleration.

B. CAJun's Fast Unconstrained GRF Solver

During reinforcement-learning in CAJun [1], a closed-form GRF is computed without constraints. Define the block-replicated inverse inertia

$$A = \underbrace{\big[M_c \;\; 0 \;\; 0 \;\; 0\big]}_{6\times 12} \;\Big|\; \underbrace{\big[0 \;\; M_c \;\; 0 \;\; 0\big]}_{6\times 12} \;\Big|\; \underbrace{\big[0 \;\; 0 \;\; M_c \;\; 0\big]}_{6\times 12} \;\Big|\; \underbrace{\big[0 \;\; 0 \;\; 0 \;\; M_c\big]}_{6\times 12}.$$

Given target $\ddot{x}_{\text{des}}$ and gravity wrench $g \in \mathbb{R}^6$, CAJun solves

$$f_b = \arg\min_{f \in \mathbb{R}^{12}} \; \tfrac{1}{2} \left\| Af - (g + \ddot{x}_{\text{des}}) \right\|_U^2 \; + \; \tfrac{1}{2} \left\| f \right\|_V^2, \qquad (1)$$

with diagonal weights $U \in \mathbb{R}^{6\times6}$, $V \in \mathbb{R}^{12\times12}$. Its closed-form is

$$f_b = \left(A^\top U A + V\right)^{-1} A^\top U \left(g + \ddot{x}_{\text{des}}\right).$$

To enforce unilateral contact and friction limits, each foot i is post-processed:

$$f_{i,z} \leftarrow \text{clip}\big(f_{b,i,z},\, f_{\min},\, f_{\max}\big),$$
$$(f_{i,x}, f_{i,y}) \leftarrow \min\Big(1, \frac{\mu\, f_{i,z}}{\|(f_{b,i,x}, f_{b,i,y})\|}\Big)\,(f_{b,i,x}, f_{b,i,y}).$$

C. $\mathcal{L}_1$ Adaptive Control Augmentation

a) Premise for L1 adaptation: Considering a linear system with unknown disturbance:

$$\dot{x} = Ax + Bu + d,$$
$$y = Cx,$$

L1 adaptation proposes the predictor–adaptation laws

$$\dot{\hat{x}} = A\,\hat{x} + B\,u_{\text{ref}} + \hat{d} + L\,(y - \hat{x}),$$
$$\dot{\hat{d}} = -\,\Gamma\,(\hat{x} - y),$$

where L, Γ are observer and adaptation gain matrices. A low-pass filter with cutoff ω_c is applied to $\hat{d}$ to yield $\tilde{d}$ before control.

b) Our discrete centroidal-wrench implementation: We treat the centroidal dynamics

$$M_c\,\ddot{x} = u + d, \qquad u = \sum_i f_i,$$

and discretize via forward-Euler at Δt:

$$\hat{x}_{k+1} = \hat{x}_k + \Big(M_c\,u_k + \hat{d}_k + L\,(y_k - \hat{x}_k)\Big)\,\Delta t,$$
$$\hat{d}_{k+1} = \hat{d}_k - \Gamma\,(\hat{x}_k - y_k)\,\Delta t,$$
$$y_k = \frac{\dot{x}_k - \dot{x}_{k-1}}{\Delta t},$$

with y_k the measured centroidal acceleration. A first-order discrete low-pass filter of cutoff ω_c is then applied:

$$\tilde{d}_k = \alpha\,\hat{d}_k + (1-\alpha)\,\tilde{d}_{k-1}, \quad \alpha = \frac{\omega_c \Delta t}{1 + \omega_c \Delta t}.$$

D. Integration with the Low-Level Force-based QP

We subtract the filtered disturbance $\tilde{d}_k$ from the original target:

$$\ddot{x}_{\text{corr}} = \ddot{x}_{\text{des}} \;-\; \tilde{d}_k,$$

and rerun the QP of the previous subsection with $\ddot{x}_{\text{corr}}$. This dual-solve loop guarantees both accurate tracking of the learned centroidal policy and rapid compensation for unmodeled payloads.

E. Training Centroidal Policy

The centroidal policy is trained using the training pipeline in [1]. The reinforcement learning (RL) problem is represented as a Markov Decision Process (MDP), which includes the state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability $p(s_{t+1} \mid s_t, a_t)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and initial state distribution $p_0(s_0)$. We aim to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward over an episode of length T, which is defined as:

$$J(\pi) = \mathbb{E}_{s_0 \sim p_0(\cdot),\, s_{t+1} \sim p(\cdot|s_t, \pi(s_t))}\left[\sum_{t=0}^{T} r(s_t, a_t)\right].$$

The state space is designed to include the robot's proprioceptive state, as well as related information about the current jump. The proprioceptive information includes the current position and velocity of the robot base, as well as the foot positions in the base frame. The task information includes the current phase of the jump Φ and the location of the target landing position in egocentric frame. The action space includes the desired stepping frequency f, the desired base velocity in sagittal plane v_x, v_z and v_θ, as well as the desired swing foot residuals, which are specified to different modules in the low-level controller. The reward function has 9 terms. At a high level, the reward function ensures that the robot maintains an upright pose, follows the desired contact schedule, and lands close to goal. Table I summarized the rewards and their corresponding weights

TABLE I: Reward Terms and Corresponding Weights

Reward Term	**Weight**
Upright	0.02
Base Height	0.01
Contact Consistency	0.008
Foot Slipping	0.032
Foot Clearance	0.008
Knee Contact	0.064
Stepping Frequency	0.008
Distance to Goal	0.016
Out-of-bound-action	0.01

F. Phase-based Gait Generator

The gait generator determines the desired contact state of each leg (swing or stance) based on a pre-defined contact sequence and the timing information from the centroidal policy. To capture the cyclic nature of locomotion, we adopt a phase-based gait representation, similar to prior works. The gait is modulated by a phase variable ϕ, which increases monotonically from 0 to 2π in each locomotion cycle, and wraps back to 0 to start the next cycle. The propagation of ϕ is controlled by the *stepping frequency* f, which is commanded by the centroidal policy:

$$\phi_{t+1} = \phi_t + 2\pi f \Delta t \tag{2}$$

where Δt is the control timestep. The mapping from ϕ to the desired contact state is pre-defined. We adopt two

types of jumping gaits in this work, namely, *bounding* and *pronging*, where bounding alternates between the front and rear leg contacts, and pronking lands and lifts all legs at the same time. Note that for the experiments we fixed the gait to be pronking gait and tested the performance, our framework allows adaptation to any gait and it can be changed by varying the mapping from ϕ to the contact state.

IV. Evaluation

A. Experiment Setup

The **CAJun** policy, as proposed in the original paper, was trained and evaluated in simulation using *NVIDIA Isaac Gym*. While Isaac Gym offers a high-fidelity environment for legged locomotion, it is still only one approximation of the real world. To investigate how well CAJun generalizes across simulation environments—commonly referred to as the *sim-to-sim transfer gap*—we attempted to deploy the same learned policy in a different simulator: **MuJoCo**. This setting allows us to isolate the impact of model mismatch without the additional complexities of sim-to-real transfer.

To achieve this, we recreated the Go1 robot model in MuJoCo using an XML-based description. This involved carefully converting the URDF and configuration files used in Isaac Gym into MuJoCo's format, preserving joint geometries, motor specifications, control modes, and physical constraints. Special attention was given to torque limits and actuation dynamics to match the conditions under which CAJun was trained.

During this conversion, we identified a number of *inertial and physical parameters* that were crucial to the dynamics observed in the rollout. These included:

- Center of mass (COM) position (x, y, z)
- Principal moments of inertia (I_{xx}, I_{yy}, I_{zz})
- Payload mass capacity and distribution

While the CAJun paper does provide baseline values for these parameters, the physical modeling in MuJoCo is sensitive to even small discrepancies in these values. Based on recommendations from the original paper and standard modeling practices, we defined a feasible range of values for each hyperparameter and conducted a parameter sweep to assess the robustness of the policy under these variations.

What we found was that **CAJun's performance deteriorated rapidly** when the inertial parameters deviated even slightly from their nominal values. With the default parameters from Isaac Gym, the robot frequently failed to complete full rollouts in MuJoCo—often falling mid-jump or becoming unstable before takeoff. This suggests that the policy is tightly coupled to the original simulator's dynamics and does not generalize well to even minor discrepancies.

To study this effect more systematically, we first varied inertial parameters while keeping all other settings fixed. As shown in Figure 2, the distance traveled during the jump was highly sensitive to changes in inertia and COM placement. Next, we introduced payload variation while keeping the default CAJun inertial values. The results, summarized in Figure 2a, were even more concerning—the robot could barely jump with added payload, and many trials failed outright. We also observe substantial variations in the robot's performance when the moment of inertia parameters (I_{xx}, I_{yy}, I_{zz}) are perturbed. Even small deviations from the tuned values lead to noticeable degradation in policy execution, often resulting in incomplete rollouts or reduced travel distance. This trend is evident in subfigures 2c and 2d. These results highlight that the CAJun policy is highly sensitive to inaccuracies in the inertial model and lacks robustness to changes in mass distribution—an essential requirement for real-world deployment.

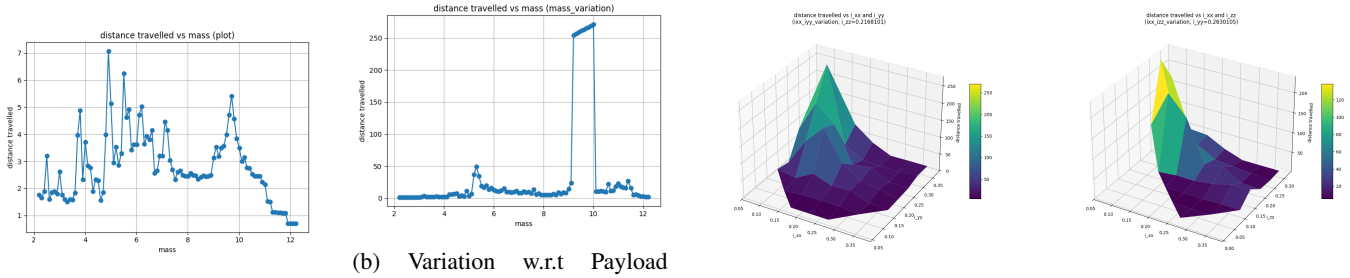
Interestingly, once we manually tuned the inertial parameters—within physically realistic bounds—we observed a significant improvement. The robot could complete more rollouts, and the distance traveled under payload variation increased noticeably. This suggests that with careful tuning, the policy can be coerced into performing better, but it also highlights a major limitation: **CAJun does not adapt to changes in dynamics**. Its performance is highly sensitive to the physical parameters of the simulation, indicating poor sim-to-sim transferability and limited robustness under variable payload conditions.

These observations form the basis for our next experiment, where we incorporate an $\mathcal{L}_1$ adaptive control layer to mitigate these limitations.

B. Robusness to Model Uncertainties

We evaluated the robustness of our framework by varying the base mass of the robot from 5.2kg to 12 Kg(2x the mass) and comparing with the vanilla Cajun implementation.

Figure.3 illustrates the variation in distance traveled as a function of the robot's mass for two control strategies: a standard QP-based controller (**Vanilla QP**) and a controller augmented with an L1 adaptive component (**L1 Adaptive**). The x-axis represents the mass in kilograms, ranging from approximately $5.2\,\mathrm{kg}$ to $12.2\,\mathrm{kg}$, while the y-axis indicates the distance traveled in meters. We observe that the **L1 Adaptive** controller consistently achieves higher travel distances compared to the **Vanilla QP** controller across nearly the entire mass range. Notably, the L1 Adaptive curve demonstrates both greater amplitude and more pronounced peaks, indicating that it is able to maintain and even enhance performance in the presence of varying mass configurations. This robustness is particularly evident in the higher mass regime (above $9\,\mathrm{kg}$), where the Vanilla QP controller performance begins to degrade sharply, falling below $2\,\mathrm{m}$ of travel distance, while the L1 Adaptive controller maintains over $8\,\mathrm{m}$.In the lower mass range (5.5–$7.5\,\mathrm{kg}$), although both controllers exhibit some fluctuation, the L1 Adaptive controller still shows superior consistency and higher average distance. The smoother yet elevated performance curve of the L1 Adaptive controller suggests that it can compensate for changes in dynamics introduced by mass variation, likely due to its real-time model adaptation capabilities. These results validate the efficacy of incorporating L1 adaptive control for mass-invariant performance and underscore the limitations of relying solely on



(a) Variation w.r.t Payload Mass

(b) Variation w.r.t Payload Mass(tuned params)

(c) Variation w.r.t I_{xx}

(d) Variation w.r.t I_{yy}

Fig. 2: Impact of varying payload mass under fine-tuned inertial parameters across different robot model parameters. The distance traveled improves(b) significantly compared to the default CAJun parameters(a), indicating sim-to-sim adaptation gap.

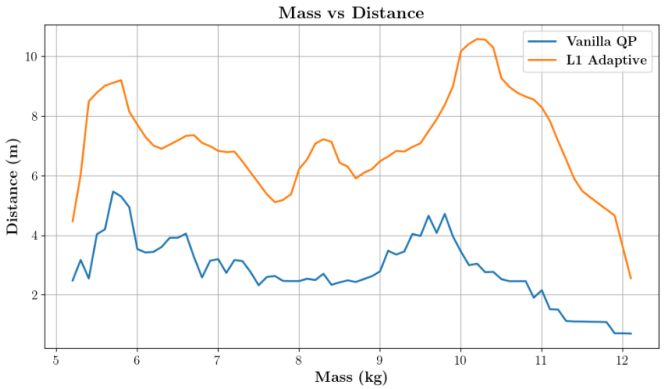


Fig. 3: Comparison of Total Distance Travelled under varying payloads

fixed-model QP controllers in dynamic, uncertain environments.

C. Effect of Low-Pass Filtering on Disturbance Estimates

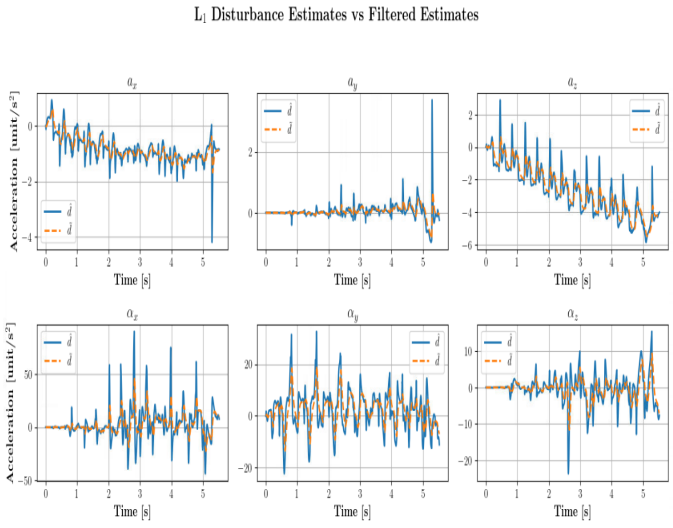


Fig. 4: Low-Pass filtering of Uncertainity estimates

Figure.4 above compares the raw disturbance estimates $\hat{d}$ and their low-pass filtered counterparts $\tilde{d}$ obtained through the L_1 adaptive control architecture. The top row illustrates the linear acceleration components (a_x, a_y, a_z), while the bottom row depicts the angular acceleration components $(\alpha_x, \alpha_y, \alpha_z)$. In both rows, it is evident that the raw estimates $\hat{d}$ contain high-frequency variations, reflecting the rapidly changing dynamics and model discrepancies encountered during motion. However, the filtered estimates $\tilde{d}$, obtained via a low-pass filter intrinsic to the L_1 framework, exhibit significantly smoother profiles across all axes.

This filtering plays a crucial role in enabling stable and high-frequency adaptation. Without such smoothing, directly injecting raw estimates into the control loop could lead to undesirable oscillations or even instability. By isolating the low-frequency components of the disturbances, the L_1 controller ensures that only the structured, slowly varying portions influence the adaptive term—allowing the control loop to run at higher frequencies without sacrificing robustness or stability. The consistency between $\hat{d}$ and $\tilde{d}$, especially in preserving major trends while suppressing noise, validates the efficacy of the L_1 filter. This design choice underpins the core advantage of L_1 adaptive control: fast adaptation with guaranteed robustness margins.

V. Conclusion

This work investigated the robustness limitations of the CAJun framework—a hierarchical reinforcement learning-based controller for continuous quadrupedal jumping—under varying inertial parameters and payload conditions. While CAJun demonstrates impressive capabilities in learning dynamic, high-amplitude jumping behaviors, its reliance on a low-level QP-based controller hinders generalization across sim-to-sim transfer and payload variation scenarios. Through a series of controlled experiments, we showed that even slight mismatches in mass or moments of inertia can significantly degrade the jumping performance, highlighting the sensitivity of the existing policy to model inaccuracies. To address this issue, we proposed augmenting the CAJun low-level controller with an $\mathcal{L}_1$ adaptive control law, designed to compensate for unmodeled dynamics and parametric uncertainty in real time. Our evaluation demonstrates that this modification signif-

icantly improves the stability and performance of the jumping policy across a range of inertial variations, without requiring re-training or additional tuning. The $\mathcal{L}_1$-augmented controller enables more consistent foot placement, higher jump distance, and better recovery across both simulated and transferred domains, underscoring its potential as a plug-and-play module for improving robustness in learned control frameworks. In future work, we aim to extend this approach to full sim-to-real transfer on hardware, and explore integrating adaptive estimation directly into the learning loop for end-to-end robustness.

REFERENCES

[1] 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. [Online]. Available: https://doi.org/10.48550/arXiv.2306.09557

[2] C. Gehring, C. D. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart. Dynamic trotting on slopes for quadrupedal robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5129–5135. IEEE, 2015.

[3] H. Kolvenbach, P. Arm, E. Hampp, A. Dietsche, V. Bickel, B. Sun, C. Meyer, and M. Hutter. Traversing steep and granular martian analog slopes with a dynamic quadrupedal robot. *arXiv preprint arXiv:2106.01974*, 2021.

[4] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[5] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

[6] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, et al. Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 2132–2139. IEEE, 2021.

[7] 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)*, pages 3359–3365. IEEE, 2017.

[8] 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)*, pages 1–9. IEEE, 2018.

[9] 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.

[10] Y. Ding, A. Pandala, and H.-W. Park. Real-time model predictive control for versatile dynamic motions in quadrupedal robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8484–8490. IEEE, 2019.

[11] C. Gehring, S. Coros, M. Hutter, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, et al. Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics & Automation Magazine*, 23(1):34–43, 2016.

[12] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[13] H.-W. Park, P. M. Wensing, and S. Kim. Jumping over obstacles with MIT Cheetah 2. *Robotics and Autonomous Systems*, 136:103703, 2021.

[14] G. Bellegarda and Q. Nguyen. Robust quadruped jumping via deep reinforcement learning. *arXiv preprint arXiv:2011.07089*, 2020.

[15] M. Sombolestan, Y. Chen, and Q. Nguyen, "Adaptive Force-based Control for Legged Robots," in *Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021, pp. 7440–7447, doi: 10.1109/IROS51168.2021.9636393.

[16] A. D. Ames, K. Galloway, J. W. Grizzle, and K. Sreenath. Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics. *IEEE Trans. Automatic Control*, 2014.

[17] 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)*, pages 2245–2252. IEEE, 2018.

[18] C. C. and N. Hovakimyan. Design and analysis of a novel l1 adaptive controller with guaranteed transient performance. *IEEE Transactions on Automatic Control*, 53(2):586–591, March 2008.

[19] C. Cao and N. Hovakimyan. Stability margins of l1 adaptive controller: Part ii. In *Proceedings of American Control Conference*, pages 3931–3936, 2007.

[20] Q. Nguyen and K. Sreenath. L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs. In *2015 American Control Conference (ACC)*, IEEE, 2015, pages 862–867.

[21] H.-W. Park, P. Wensing, and S. Kim. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[22] K. J. Åström, "Adaptive Control," in *Mathematical System Theory: The Influence of R. E. Kalman*, A. C. Antoulas, Ed., Berlin, Heidelberg: Springer, 1991, pp. 437–450.

[23] M. Sombolestan and Q. Nguyen, "Hierarchical Adaptive Loco-manipulation Control for Quadruped Robots," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2023, pp. 12156–12162.

[24] D. Zhang and B. Wei, "A review on model reference adaptive control of robotic manipulators," *Annual Reviews in Control*, vol. 43, pp. 188–198, Jan. 2017.

[25] C. Cao and N. Hovakimyan, "L1 adaptive controller for a class of systems with unknown nonlinearities: Part I," in *Proc. American Control Conf.*, Jun. 2008, pp. 4093–4098.

[26] A. Gahlawat et al., "Contraction L1-Adaptive Control using Gaussian Processes," in *Proc. 3rd Conf. on Learning for Dynamics and Control*, PMLR, vol. 144, Nov. 2021, pp. 1027–1040.

[27] G. Tournois et al., "Online payload identification for quadruped robots," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Dec. 2017, vol. 2017-Septe, pp. 4889–4896.

[28] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots with control Lyapunov function based quadratic programs," in *Proc. American Control Conf.*, Jul. 2015, vol. 2015-July, pp. 862–867.

[29] J. W. Grizzle, C. Chevallereau, and C. L. Shih, "HZD-based control of a five-link underactuated 3D bipedal robot," in *Proc. IEEE Conf. on Decision and Control*, 2008, pp. 5206–5213.

[30] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive CLF-MPC with application to quadrupedal robots," *IEEE Robot. Autom. Lett.*, 2021. doi:10.1109/LRA.2021.3128697.