

Manipulator as a Tail: Promoting Dynamic Stability for Legged Locomotion

Huang Huang¹, Antonio Loquercio¹, Ashish Kumar, Neerja Thakkar¹, Ken Goldberg¹, Jitendra Malik¹

Abstract—For locomotion, is an arm on a legged robot a liability or an asset for locomotion? Biological systems evolved additional limbs beyond legs that facilitates postural control. This work shows how a manipulator can be an asset for legged locomotion at high speeds or under external perturbations, where the arm serves beyond manipulation. Since the system has 15 degrees of freedom (twelve for the legged robot and three for the arm), off-the-shelf reinforcement learning (RL) algorithms struggle to learn effective locomotion policies. Inspired by Bernstein’s neurophysiological theory of animal motor learning, we develop an incremental training procedure that initially freezes some degrees of freedom and gradually releases them, using behaviour cloning (BC) from an early learning procedure to guide optimization in later learning. Simulation experiments show that our policy increases the success rate by up to 61 percentage points over the baselines. Simulation and real robot experiments suggest that our policy learns to use the arm as a “tail” to initiate robot turning at high speeds and to stabilize the quadruped under external perturbations. Quantitatively, in simulation experiments, we cut the failure rate up to 43.6% during high-speed turning and up to 31.8% for quadruped under external forces compared to using a locked arm.

I. INTRODUCTION

Human arms are vital for manipulation, but also improve stability. For instance, humans swing their arms to balance when falling or when being pushed [1]. Similarly, many animals use their tails to improve their stability and agility. A beaver’s wide, flat tail helps it balance its body weight and skillfully maneuver through the water. A cheetah’s tail helps it make sharp turns at astonishing speeds [2].

Prior works [3], [4], [5], [6], [7], [8] have added an arm to legged robots for mobile manipulation and focus on maintaining robustness while the arm performs manipulation tasks. Recent model-based approaches propose whole-body planners that use robotic arms beyond manipulation. By formulating a multi-contact optimal control problem, they show that an arm can help the balance of the robot [4], [7], [9]. However, these approaches are sensitive to the fidelity of the model used for planning [4], require specific self-collision routines [7], and need accurate state estimators (e.g., linear velocity) for trajectory tracking [9]. These requirements constrain these approaches to relatively large robots carrying enough sensors and computing for the required computation.

Inspired by biology, we explore how a low-cost and small robot-manipulator setup can learn to actively use the arm to benefit the stability and agility of locomotion under external perturbations and at high speeds. Using an arm to balance or turn is a complex whole-body control problem requiring synergy between all the robot limbs. It is a highly dynamic

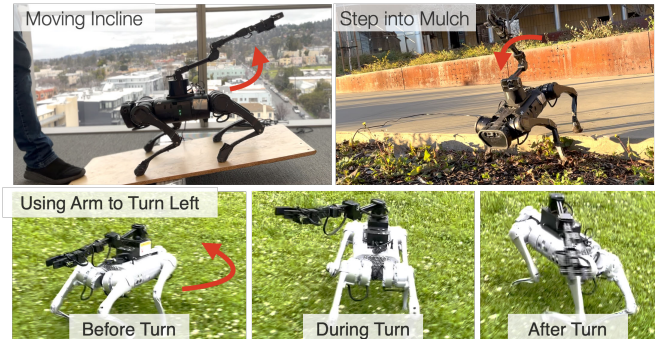


Fig. 1: A learned whole-body control policy increases a legged robot’s stability and agility. We evaluate the dynamic benefits of the arm in two scenarios. **Top:** stabilization under dynamic external forces. The actuated arm responds to an impulse force by swinging quickly away from the impulse direction to generate a balancing impulse force. As shown in section VIII, the arm relies on such dynamic forces instead of the static CoM changes to stabilize. **Bottom:** dynamic agile locomotion, where the arm helps the quadruped turn left at high speeds. Before and during the turn, the arm moves toward the turning direction to increase the centripetal acceleration (toward the turning direction), similar to behaviors observed in animals. After the turn, the arm moves back to the nominal position. Videos and supplementary materials are here: <https://tinyurl.com/2p8edezu>

task: the robot must move its limbs quickly to generate sufficient torques to balance against external perturbations or enough momentum to turn fast.

Learning-based methods have shown success for legged locomotion [10], [11], [12], [13], [14]. We learn a unified sensorimotor policy for both the arm and the quadruped. To simplify the complex optimization problem for a system with many degrees of freedom, we use an incremental stage-wise training procedure that gradually increases the number of controllable degrees of freedom. Specifically, we learn incrementally more complex behavior with RL, using BC from previously learned behaviors as a proxy loss to guide the optimization. Using demonstrations and incremental learning to improve RL-based controllers has been widely explored for locomotion [15], [16], [17], [18], [19]. In this paper, we show the unification of these two concepts is particularly suitable for our problem setting. Empirically, we show up to 30 percentage points higher success rate in walking compared to using demonstrations without incremental learning and up to a 50 percentage points higher success rate than incremental learning without demonstrations (i.e., curriculum learning),

¹ University of California, Berkeley

as shown in Table III.

We deploy the learned policy on a physical robot in the real world and evaluate the emerging locomotion strategy in both simulation and real-world settings for two locomotion tasks: stable locomotion to balance against external perturbations and agile locomotion to turn at high speed. We observe that the actuated arm compensates against pushes and stabilizes under external forces and on a moving incline (Fig. 1). We also find that the learned policy exhibits behaviors that are reminiscent of the motion of geckos [20]: when turning at high speeds (≈ 1.5 m/s), the tail precedes the motion of the body to increase centripetal forces (Fig. 1). In addition, we interpret our results in the light of a first-principle analysis. The analysis shows that while the arm is not useful statically due to its small mass, it clearly benefits the dynamic stability.

II. RELATED WORK

Many biologists and roboticists have studied the utility of animal tails in maneuverability and stability [21]. For instance, cats and squirrels use their tails to facilitate balancing [22], [23]. Lizards and cheetahs use their tails to effectively reorient their bodies at high speeds [20], [2]. These utilities of tails helped the first terrestrial organisms adapt from aquatic environments to various terrestrial ones, by improving their stability and agility on difficult terrains [24].

Tails have also proven beneficial in robotic design. An attached tail can assist a robot in flipping upright and achieving a balanced landing while in free fall [25], [26], [23] or when jumping [27], [28]. Tails can help robots maneuver through turns at higher speeds [2] and navigate difficult terrain that would otherwise trap the robot [29]. The Arque tail helps humans balance by counteracting changes in the wearer’s center of mass and momentum [30]. Inspired by the utility of tails in biology and robotics, we use a manipulator as a tail to enhance locomotion stability and agility. A few other works have also used limbs to counteract disturbances, in similar function to a tail. For instance, [31] proposes a model that plans arm motions in reaction to forces on a humanoid robot, allowing its center of mass to stay within the support polygon. [32] focus on robust locomotion in a quadruped robot with an arm, keeping the base stable as the arm carries out a manipulation task. In [33], a standing quadruped robot with an arm attached is disturbed, and an arm trajectory helps counteract forces. [34] learns a policy to use the arm to prevent the quadruped from falling or help the quadruped to recover from falling. In contrast, our method counteracts forces with dynamic arm movements while the quadruped is moving and uses arm to enhance quadruped agility.

Some prior works focus on mobile manipulation and train the arm separately from the robot base, and combine the two [35], [36], [37]. Other work involves jointly learning locomotion and manipulation on the whole body, reducing engineering and tuning needed to combine the two components and allowing for more complex behaviour to emerge [3], [4]. Our approach similarly jointly learns the behaviour of the robot body and the arm, but carries out this learning in multiple stages to simplify the optimization problem. We

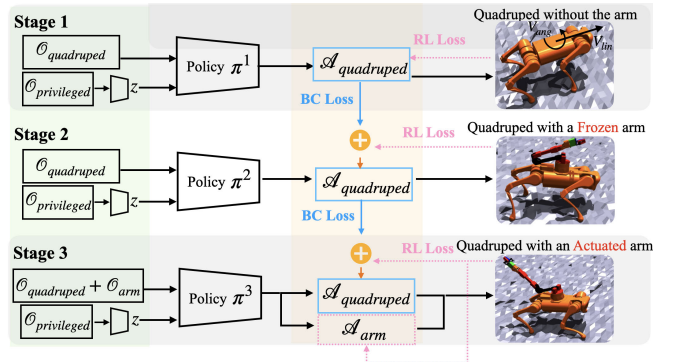


Fig. 2: We learn whole-body control policies for quadruped locomotion with an arm in three stages, each learning a different skill of increasing complexity. Policies at each stage observe $\mathcal{O}_{quadruped}$ or \mathcal{O}_{arm} and an extrinsic vector z with compressed information of privileged observations $\mathcal{O}_{privileged}$, which consists of mass, friction, quadruped base velocity and external perturbations.

show another use of the manipulator beyond manipulation to benefit locomotion, in contrast to prior works [36], [33] that consider the arm as a liability for locomotion.

Using demonstrations to improve the training process of RL algorithms has a long history [38], [39], [15], [16], [17], [18], [40]. These works find that when expert demonstrations are available, training policies with RL is faster and more effective. Additionally, incremental RL learning has proven to be successful [41], [42]. We made the following key changes to achieve the unification of learning from demonstrations, RL and incremental learning to achieve multi-stage incremental learning of whole-body control. We incrementally learn policies trained on simpler skills to provide demonstrations for policies trained on more complex ones. Policies from early stages provide on-policy demonstrations using an annealing weight instead of static external demonstrations for later stage, which is shown to favor training [43], [44], [45].

III. METHOD

We use an iterative optimization algorithm for synthesizing behaviors for a task \mathcal{K} . We assume that the task \mathcal{K} can be decomposed into multiple stages: $\mathcal{K}^1 \dots \mathcal{K}^n$ [46], [47], [48]. The stage $\mathcal{K}^i = (\mathcal{O}^i, \mathcal{R}^i, \mathcal{A}^i)$ has \mathcal{O}^i as the observation space, \mathcal{R}^i as the reward function, and \mathcal{A}^i as the action space. Each stage \mathcal{K}^i includes all the previous stages $\mathcal{K}^{1:i-1}$, along with additional objectives at stage i . These objectives could potentially add a term to reward function for obtaining \mathcal{R}^i , or involve additional sensors or actuators to get \mathcal{O}^i and \mathcal{A}^i respectively. The training scheme works as follows:

Base step (\mathcal{K}^1): We train the policy $\pi^1(o^1)$ to predict a^1 from scratch using RL to optimize the returns defined by the reward function \mathcal{R}^1 .

Inductive step (\mathcal{K}^i): Once we have policy π^{i-1} , we train policy π^i by optimizing:

$$\begin{aligned} \max_{\pi^i} \quad & (J(\pi^i) = \mathbb{E}_{\tau \sim p(\tau|\pi^i)} \left[\sum_{t=0}^{T-1} \gamma^t r_t^i \right]) \\ \text{s.t.} \quad & \mathcal{DW}(\pi^i, \pi^{i-1}) \leq \delta \end{aligned} \quad (1)$$

where \mathcal{DW} is a Wasserstein distance function, τ is the trajectory of the agent when executing policy π^i , and $p(\tau|\pi^i)$ represents the likelihood of the trajectory under π^i . This optimization is similar to the one proposed in [40]. In contrast to [40], we compute the expectation over the visited states τ instead of a fixed dataset \mathcal{D} and do not require any initial reference motions. Using the Kantorovich-Rubinstein theorem [49], we simplify it to a relaxed unconstrained optimization problem (details on website):

$$\mathcal{L}(\pi^i) = J(\pi^i) + \lambda * \mathbb{E}_\tau[\|a_t^i - a_t^{i-1}\|], \quad (2)$$

where $a_t^i = \pi^i(o_t^i)$, $a_t^{i-1} = \pi^{i-1}(o_t^i)$ (computed over the relevant part of o_t^i), and $\|a_t^i - a_t^{i-1}\|$ is a regression loss (computed over compatible part of actions) that encourages π^i to stay close to π^{i-1} . Note that Eq. (2) can be algorithmically optimized by simultaneously minimizing $J(\pi^i)$ using RL and $\mathbb{E}_\tau\|a_t^i - a_t^{i-1}\|$ with BC between π^i and π^{i-1} . We anneal λ to decrease the imitation constraint in the later stages of training and find an overall better solution. Figure 2 shows an overview of our approach.

We learn locomotion policies by dividing the task into a three-stage process (Fig. 2). We first train a walking policy π^1 for locomotion on flat terrain without an arm attached. Then, we train a policy π^2 for locomotion in the presence of task specific conditions, where the arm is attached but locked. Finally, we train a policy π^3 that controls both the legged robot and the arm under task specific conditions. The final policy can utilize the arm to balance under strong external pushes or rough terrain and help the quadruped turn at high speed. For both π^2 and π^3 , we use BC on the policy from the previous iteration as a proxy loss during optimization. The task-specific conditions in stages 2 and 3 are external perturbations for stable locomotion and turning at high speed for agile locomotion.

For stage 1 and 2, we have quadruped observations $\mathcal{O}_{\text{quadruped}}$ and the privileged observation $\mathcal{O}_{\text{privileged}}$. The action space $\mathcal{A}_{\text{quadruped}}$ is target quadruped joint angles. For stage 3, we have the additional arm observation \mathcal{O}_{arm} . The action space is the union of $\mathcal{A}_{\text{quadruped}}$ and \mathcal{A}_{arm} (target arm joint angles). We train our policies in simulation using NVIDIA Isaac Gym [50] and use RMA [10] to transfer the policy trained in simulation to the real world. Details are on website.

IV. EXPERIMENTAL SETUP

In the *stable locomotion task*, we learn a locomotion policy that is stable against external perturbations. During policy training, we sample the external pushing force uniformly from 200 – 300¹ with an offset from the base sampled uniformly from $[-0.12, 0.12]m$ along the x or y axis of the robot quadruped base. The policy is trained to walk with a linear velocity in the range $[0, 0.3]m/s$. We use a Unitree A1 with a WX200 arm. In the *agile locomotion task*, we learn a locomotion policy that can turn fast at high running speed. No external perturbations are applied in this task. The policy is

¹According to the simulator’s documentation, forces are in Newtons. However, we empirically found the unit to be physically unreasonable and possibly correct up to scale.

trained to run with a linear velocity in the range $[0.6, 1.5]m/s$ and an angular yaw velocity in the range $[0.5, 2]m/s$. We use a Unitree Go1 with a K1 arm for this task.

In both tasks, the episode terminates when the legged robot base collides with the ground. Since the last several degrees of freedom of the arm have less effect on changing the inertia and momentum, we only use the first 3 degrees of freedom of the arm. More details are on website.

The following metrics are used to evaluate the results: *Success rate*, the fraction of robots that survived at the end of the episode; *Time to fall* (TTF), the average episode length across all the robot instances divided by the total episode length; *Linear velocity* (V_{lin}) *tracking error*, the L2 distance between the robot base linear velocity and the linear velocity command; and *Angular velocity* (V_{ang}) *tracking error*, the L2 distance between the robot base yaw angular velocity and the yaw angular velocity command.

V. QUANTITATIVE ANALYSIS: BENEFIT OF ARM FOR LOCOMOTION

Stability against perturbation We compare the performance of the robot with an actuated arm, with a locked arm (π_2), and without arm (Table I). The robot with an actuated arm outperforms both baselines, having a survival rate of up to 33 percentage points higher and a linear velocity tracking error up to 88% lower. The success rate and TTF(%) for all policies decrease as the external perturbation magnitude increases. However, the robot with an actuated arm has a relatively lower performance drop while that of a frozen arm has a 45 percentage points performance drop for the survival rate. Not having an arm increases performance compared to keeping the arm locked, but is still outperformed by our approach. This indicates the robot cannot tolerate high perturbations without the arm’s support.

Agile locomotion We compare all methods on the task of high-speed locomotion in simulation. We average the performance over 2000 trials, with a maximum duration of 20s. We sample a yaw angular command of $0.5 rad/s$ for half of the trials and $2 rad/s$ for the rest, and a constant linear velocity command of $1m/s$, $1.25m/s$, and $1.5m/s$. Results are shown in Table II. The policy with an actuated arm has better tracking performance than all the baselines. However, in terms of success rate, the no-arm baseline is slightly better than the actuated arm, with differences increasing as a function of speed. This shows that the arm’s weight and inertia make agile locomotion more difficult. However, actuation provides a significant advantage over a locked arm.

We conduct simulation experiments studying the relation between the arm motion and the quadruped yaw as in Fig. 3a. We study the first arm joint motion as it has the largest effect on inertia. In Fig. 3a, we show the first arm joint state and robot yaw for robot running at $1m/s$ with a changing yaw angular command up to $1rad/s$. From the plot, the arm starts to move right after receiving a non-zero yaw command and the yaw of the quadruped changes after the arm motion. The delay between the quadruped yaw and the arm motion indicates that the quadruped is following the arm motion

Arm	Success % (\uparrow)		TTF % (\uparrow)		V_{lin} Tracking Error (\downarrow)	
	200-300	300-400	200-300	300-400	200-300	300-400
No Arm	95.02 \pm 0.62	68.03 \pm 1.14	0.975 \pm 0.00	0.809 \pm 0.01	0.230 \pm 0.02	0.765 \pm 0.03
Locked	90.61 \pm 0.77	45.83 \pm 1.71	0.950 \pm 0.00	0.627 \pm 0.01	0.067 \pm 0.01	0.224 \pm 0.01
Actuated	96.11\pm0.37	78.20\pm1.77	0.982\pm0.00	0.875\pm0.01	0.027\pm0.00	0.120\pm0.02

TABLE I: Results for locomotion under disturbances. Mean and std are computed over 5 seeds.

V_{lin}	Setting	Success (\uparrow)	V_{ang} TE(\downarrow)	V_{lin} TE (\downarrow)
1	No Arm	90.29\pm0.46	0.33 \pm 0.02	0.014 \pm 0.001
	Locked	82.32 \pm 0.51	0.23 \pm 0.01	0.015 \pm 0.001
	Actuated	89.64 \pm 0.64	0.18\pm0.01	0.010\pm0.001
1.25	No Arm	89.14\pm0.70	0.30 \pm 0.02	0.022 \pm 0.001
	Locked	77.81 \pm 1.01	0.23 \pm 0.01	0.025 \pm 0.001
	Actuated	86.78 \pm 0.77	0.18\pm0.01	0.015\pm0.001
1.5	No Arm	86.48\pm0.85	0.29 \pm 0.02	0.031 \pm 0.001
	Locked	69.77 \pm 0.47	0.23 \pm 0.01	0.037 \pm 0.001
	Actuated	82.96 \pm 0.74	0.18\pm0.01	0.024\pm0.001

TABLE II: Simulation evaluation for high-speed locomotion. Mean and std are computed over 5 seeds. Velocity commands are in m/s , success in percent, and TE denotes tracking error.

and that the arm motion is an active choice of the policy to initiate robot turning, similar to cheetah’s tail behaviour, instead of a passive behaviour caused by the inertia.

We study the relation between the magnitude of the arm motion and the quadruped base motion as in [20]. We record the robot state for robot running at $1.5m/s$ with a changing yaw angular velocity command. We sample 10 angular yaw commands and 3 points from each command to record the first arm joint position and the quadruped yaw position. Details are on website. We plot the quadruped yaw position with respect to the first arm joint position, shown in Fig. 3b. Linear regression indicates a strong positive correlation between the arm motion and the quadruped yaw motion with $R^2 = 0.8877$, consistent with the gecko tail behaviour in [20].

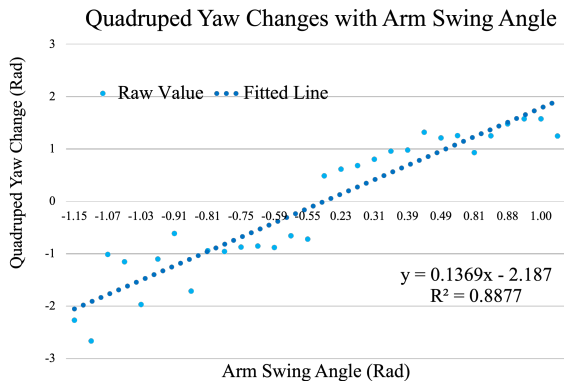
VI. QUANTITATIVE SIMULATION ANALYSIS

We compare our incremental stagewise learning method against five baselines of increasing complexity in the stable locomotion task setting: (i) *Proximal Policy Optimization (PPO)*: a policy trained from scratch on the complete whole-body control task with PPO. (ii) *PPO Curriculum*: a PPO policy trained with the curriculum of a quadruped with a locked arm with no perturbations, to that with perturbations and to that with a moving arm. (iii) *Fine-tuned π^2* : a policy fine-tuned on π^2 for a quadruped with a moving arm. (iv) *Decoupled*: We train the locomotion and arm policies separately. Specifically, we first train a locomotion policy with arm fixed (π^2) and then the arm policy with the latter locomotion policy fixed. At test time, we combine the output of these two policies. (v) *One stage demonstration*: a policy trained with π^1 being a teacher policy on a quadruped with a moving arm directly, without multi-stage learning.

We evaluate with pushing forces in the training distribution (200-300), and with pushing forces beyond what was seen in training (300-400). We average the performance of all methods over 2000 trials, each with a maximum duration of



(a) Simulation results with robot running at $1m/s$. The desired (dotted green) and actual (solid green) first arm joint positions are shown with the quadruped base angular velocity command (blue) and yaw (orange).



(b) Simulation correlation analysis between arm motion and robot yaw. The blue line is a trending line fitted with linear regression on raw values (blue dots).

Fig. 3: Simulation results and analysis of the benefit of arm for agile locomotion.

twenty seconds of simulated time. For 50% of the trials, we sample a standing still command and, for the rest, a forward velocity command of $0.3 m/s$.

Results are shown in Table III. Our method outperforms all the baselines in success rate and TTF(%), achieving up to 16 percentage points higher survival rate than the next best method. PPO trained from scratch has the lowest performance, likely because joint training for the final task is very complex, leading to poor gradient estimates that could lead to a poor local optimum. The PPO performance falls by the largest magnitude compared to any of the other baselines for out of distribution pushing forces. This shows that vanilla PPO has a much poorer generalization. The decoupled baseline performs worse than our method. This is likely

Policy	Success % (\uparrow)		TTF % (\uparrow)		V_{lin} Tracking Error (\downarrow)	
	200-300	300-400	200-300	300-400	200-300	300-400
PPO	64.43	23.10	0.770	0.405	0.028	0.029
PPO cur	76.85	34.40	0.843	0.458	0.032	0.114
Fine-tuned π^2	69.75	37.03	0.807	0.543	0.023	0.066
Decoupled	92.10	68.48	0.930	0.734	0.085	0.222
One stage demonstration	87.93	55.03	0.934	0.692	0.023	0.077
Ours	95.73	84.73	0.975	0.916	0.021	0.081

TABLE III: Simulation evaluation of all methods for stabilizing under external pushes.



Fig. 4: Physical Experiments under different scenarios. Our system, only trained on fractal terrains with external pushes, is successfully deployed on different real-world scenarios. The arm is at different configurations under different scenarios.



Fig. 5: Robot stabilization under external perturbations. The robot with a locked arm fails to maintain balance. For actuated arm, instead of moving the arm to its left for changing the CoM, as shown in section VIII, the arm relies on the dynamic torques to mitigate the impulse perturbation, resulting the arm moving to its right.

because learning a controller for the arm separately from a locomotion controller fails to yield coordinated behavior for better overall performance. This happens despite training the locomotion controller with external pushes, as in all our baselines. The baseline with incremental learning but without using demonstrations from a previous policy (“PPO cur” in Table III) performs better than vanilla PPO but significantly worse than our method. Using demonstrations but without using incremental stage-wise learning (“One stage demonstration” in Table III) shows a significant performance drop for out of distribution pushing forces. This indicates the effectiveness of our method of combining RL, BC and incremental learning. Our method shows a slight performance drop between the two push force ranges, likely because the rotational degree of freedom in the base of the arm allows it to generate a counteracting angular velocity. For linear velocity tracking, our policy performs the best for pushing

forces within distribution. Note that our policy needs to track velocity amidst more extreme perturbations compared to what the baselines experience since it has a higher survival rate.

VII. PHYSICAL EXPERIMENTS

A. Locomotion under Disturbances

We deploy our system, trained only in simulation on fractal terrains with external pushes, in a variety of static and dynamic real-world scenarios, some of which are shown in Fig. 4. We test with perturbations on an incline, mulch, slopes, and down steps, none of which were simulated during training. Our deployment results are shown in the supplementary video.

We evaluate on an incline balanced on a fulcrum positioned in the middle. We put the robot on one end of the incline and rotate the incline by exerting downward pressure on the other end. The arm moves to generate a counter torque on the trunk of the quadruped, ensuring the robot’s pitch stays roughly level during the rotation. However, at equilibrium, the arm does not move back to shift the center of mass towards the incline, since the arm is too light to have a significant impact on the center of mass (see Sec. VIII for details). We vary the incline angle irregularly, causing the arm to respond quickly by changing the direction of rotation in response to the incline rotation direction. We also create an unlevelled ground by keeping the right half of the robot on a pivoting incline, while the other two legs stay on the ground. The robot can handle extreme rotations to the plank despite only having seen fractal variations to flat ground in simulation. The robot can traverse a step, upslope, and downslope, none of which were seen during training. While walking on the mild concrete upslope, the arm maintains a different mean position compared to when walking on flat terrain, and helps it get unstuck while walking up a mild slope.

We additionally conduct quantitative experiments comparing the performance of a quadruped with an actuated and a locked arm. The robot is placed on a moving incline that is pushed down to create an external perturbation. When the external perturbation is high, the quadruped with a locked arm falls on the ground, while the one with a moving arm maintains stable as shown in Fig. 5. The roll changes of the quadruped with the locked arm is twice as high as that of the quadruped with a moving arm (plot on website). This suggests our method of using the arm as a tail can help the quadruped to stabilize under extreme perturbations.

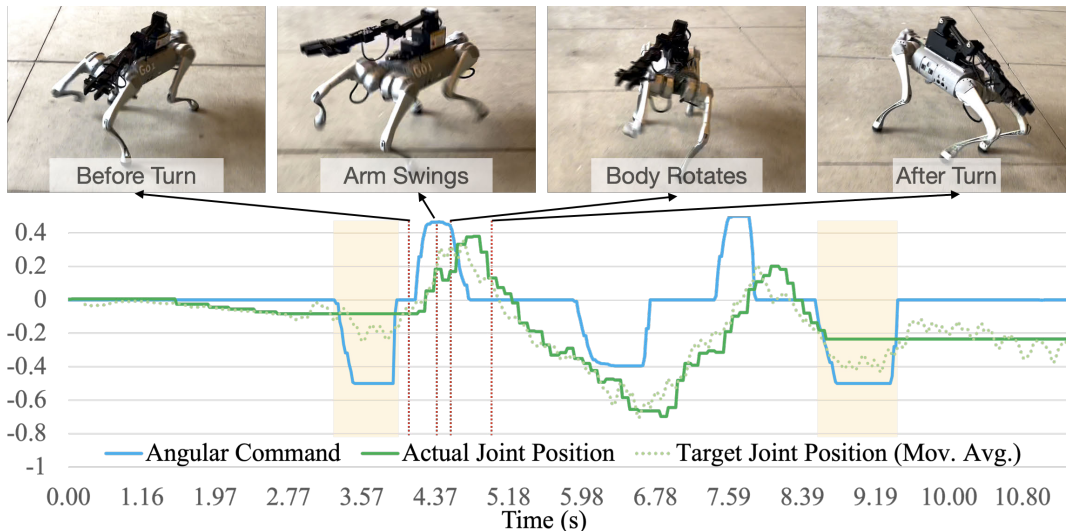


Fig. 6: Physical experiments analysis. The arm anticipates the body during high-speed rotation.

B. Agile Locomotion

We conduct experiments where the robot is running ($1\text{-}2\text{m/s}$) and turning ($0.5\text{-}1\text{rad/s}$) at high speeds on different terrains including grassland (Fig. 1) and cement ground (Fig. 6). We record the the first arm joint state for robot running at 1m/s and turning at different speeds in Fig 6. The video frames on top show the robot state at different time steps. The arm starts to move from right to left when an angular command to the left is received as shown by the first two frames. The quadruped then starts to turn following the arm motion, shown in the third frame. That the arm swings before the quadruped start to turn, indicating the arm motion initiates the quadruped motion by generating momentum, with a delay consistent of what observed in simulation (see Sec. VIII for an analytical analysis of this behavior). However, due to limitations on the arm hardware, the arm can fail to achieve the action given by the policy as indicated by the orange area in the plot, where the desired arm position and the actual arm position has a high discrepancy.

VIII. FIRST PRINCIPLES ANALYSIS OF ARM MOTION

We provide an analytical derivation of the static and dynamic effects of the manipulator motion on the robot body. To make the analysis tractable, we approximate the robot body as a cuboid and the arm as a rod attached to the robot’s center of mass (CoM). First, a static analysis shows that changes in the system’s center of mass $cCoM$ due to the arm rotation θ are minimal. Indeed, under the previous assumption, the latter can be written as:

$$cCoM = \frac{M_a}{M_a + M_b} \frac{l}{2} \cos \theta = 0.022 \cos \theta \leq 0.022m, \quad (3)$$

where M_a, M_b are the arm’s and body’s mass, respectively, and l is the arm length. Second, a dynamic torque analysis shows that the arm’s angular acceleration α_a can significantly affect the body’s angular acceleration α_b . Specifically, it can

be shown that

$$\alpha_b = -\frac{4M_a l^2}{M_b(h^2 + w^2)} \alpha_a = -0.378\alpha_a, \quad (4)$$

where h and w are the torso’s height and width. This acceleration is used by our policy to absorb parts of the torques induced by disturbances or during high-speed locomotion to get faster and sharper turns, resulting in much better angular velocity tracking. We refer to the supplementary material on the project website for a derivation and a similar analysis for changes in the system’s center of pressure.

IX. DISCUSSION AND LIMITATIONS

Attaching an arm to legged robots expands the legged robots capability to mobile manipulation but introduces challenges to robot stability due to increased CoM and additional payload. In this work, we mitigate this challenge by actively controlling the arm for enhanced stability and agility. Specifically, we utilize incremental stage-wise learning to use the arm of a quadruped robot for balancing on uneven terrain, under external pushes and turning at high speeds. Through simulation experiments and real-world deployment, we have shown quantitatively and qualitatively that an arm can be an asset for improving the stability and agility of locomotion. Our results and analytical analysis indicate that an exciting avenue for future research is designing a light “tail” capable of fast accelerations. However, our policy is blind and does not use the environment’s affordances to balance. Using vision and real-world experience to improve could potentially alleviate this problem [11]. The proposed incremental stage-wise learning approach requires manually designing each stage, which could be challenging for more complex systems. Extending the current approach to automatically designed stages could be an interesting future direction.

ACKNOWLEDGEMENTS

This research was performed at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS “People and Robots” (CPAR) Initiative. The authors were supported in part by donations from Toyota Research Institute.

REFERENCES

- [1] L. H. Ting and J. L. McKay, "Neuromechanics of muscle synergies for posture and movement," *Current opinion in neurobiology*, vol. 17, no. 6, pp. 622–628, 2007.
- [2] A. Patel and M. Braae, "Rapid turning at high-speed: Inspirations from the cheetah's tail," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5506–5511.
- [3] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning (CoRL)*, 2022.
- [4] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.
- [5] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, "Alma-articulated locomotion and manipulation for a torque-controllable robot," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 8477–8483.
- [6] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4488–4494.
- [7] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free mpc for whole-body dynamic locomotion and manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4686–4693.
- [8] N. Kumar, T. Silver, W. McClinton, L. Zhao, S. Proulx, T. Lozano-Pérez, L. P. Kaelbling, and J. Barry, "Practice makes perfect: Planning to learn skill parameter policies," 2024.
- [9] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, p. eadg5014, 2023.
- [10] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.
- [11] A. Loquercio, A. Kumar, and J. Malik, "Learning visual locomotion with cross-modal supervision," *arXiv preprint arXiv:2211.03785*, 2022.
- [12] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," 2022. [Online]. Available: <https://openreview.net/pdf?id=Re3NjSwf0WF>
- [13] 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.
- [14] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [15] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [16] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments," *arXiv preprint arXiv:1910.04281*, 2019.
- [17] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [18] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *RSS*, 2020.
- [19] J. Peng and R. J. Williams, "Incremental multi-step q-learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 226–232.
- [20] R. Siddall, V. Ibanez, G. Byrnes, R. J. Full, and A. Jusufi, "Mechanisms for Mid-Air Reorientation Using Tail Rotation in Gliding Geckos," *Integrative and Comparative Biology*, vol. 61, no. 2, pp. 478–490, 06 2021. [Online]. Available: <https://doi.org/10.1093/icb/icab132>
- [21] S. Shield, R. Jericevich, A. Patel, and A. Jusufi, "Tails, Flails, and Sails: How Appendages Improve Terrestrial Maneuverability by Improving Stability," *Integrative and Comparative Biology*, vol. 61, no. 2, pp. 506–520, 05 2021. [Online]. Available: <https://doi.org/10.1093/icb/icab108>
- [22] C. Walker, C. J. Vierck Jr, and L. A. Ritz, "Balance in the cat: role of the tail and effects of sacrocaudal transection," *Behavioural brain research*, vol. 91, no. 1–2, pp. 41–47, 1998.
- [23] T. Fukushima, R. Siddall, F. Schwab, S. L. Toussaint, G. Byrnes, J. A. Nyakatura, and A. Jusufi, "Inertial tail effects during righting of squirrels in unexpected falls: from behavior to robotics," *Integrative and Comparative Biology*, vol. 61, no. 2, pp. 589–602, 2021.
- [24] B. McInroe, H. C. Astley, C. Gong, S. M. Kawano, P. E. Schiebel, J. M. Rieser, H. Choset, R. W. Blob, and D. I. Goldman, "Tail use improves performance on soft substrates in models of early vertebrate land locomotors," *Science*, vol. 353, no. 6295, pp. 154–158, 2016.
- [25] A. M. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. J. Full, and D. E. Koditschek, "Tail assisted dynamic self righting," in *Adaptive Mobile Robotic*. World Scientific, 2012.
- [26] E. Chang-Siu, T. Libby, M. Brown, R. J. Full, and M. Tomizuka, "A nonlinear feedback controller for aerial self-righting by a tailed robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 32–39.
- [27] J. Zhao, T. Zhao, N. Xi, F. J. Cintrón, M. W. Mutka, and L. Xiao, "Controlling aerial maneuvering of a miniature jumping robot using its tail," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3802–3807.
- [28] J. An, T. Chung, C. H. D. Lo, C. Ma, X. Chu, and K. S. Au, "Development of a bipedal hopping robot with morphable inertial tail for agile locomotion," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechanics (BioRob)*. IEEE, 2020, pp. 132–139.
- [29] D. Soto, "Simplifying robotic locomotion by escaping traps via an active tail," Master's thesis, Georgia Institute of Technology, 2022.
- [30] J. Nabeshima, m. y. Saraiji, and K. Minamizawa, "Arque: artificial biomimicry-inspired tail for extending innate body functions," 07 2019, pp. 1–2.
- [31] C. Khazoom and S. Kim, "Humanoid arm motion planning for improved disturbance recovery using model hierarchy predictive control," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6607–6613.
- [32] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar, "Roloma: Robust loco-manipulation for quadruped robots with arms," 2022. [Online]. Available: <https://arxiv.org/abs/2203.01446>
- [33] H. Ferrolho, W. Merkt, V. Ivan, W. Wolfslag, and S. Vijayakumar, "Optimizing dynamic trajectories for robustness to disturbances using polytopic projections," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7477–7484.
- [34] Y. Ma, F. Farshidian, and M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 149–12 155.
- [35] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4488–4494, 2021.
- [36] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022.
- [37] R. C. Quesada and Y. Demiris, "Holo-spok: Affordance-aware augmented reality control of legged manipulators," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 856–862.
- [38] S. Schaal, "Learning from demonstration," *Advances in neural information processing systems*, vol. 9, 1996.
- [39] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, vol. 97, 1997, pp. 12–20.
- [40] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne, "Learning locomotion skills for cassie: Iterative design and sim-to-real," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 317–329. [Online]. Available: <https://proceedings.mlr.press/v100/xie20a.html>
- [41] Z. Wang, H.-X. Li, and C. Chen, "Incremental reinforcement learning in continuous spaces via policy relaxation and importance weighting," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 1870–1883, 2019.
- [42] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online bayesian inference," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 4003–4016, 2021.
- [43] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings*

of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

- [44] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, “Learning high-speed flight in the wild,” *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.
- [45] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, “Deep drone racing: From simulation to reality with domain randomization,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 1–14, 2019.
- [46] S. P. Singh, “Transfer of learning by composing solutions of elemental sequential tasks,” *Machine learning*, vol. 8, pp. 323–339, 1992.
- [47] T. Dean and S.-H. Lin, “Decomposition techniques for planning in stochastic domains,” in *IJCAI*, vol. 2. Citeseer, 1995, p. 3.
- [48] T. G. Dietterich, “Hierarchical reinforcement learning with the maxq value function decomposition,” *Journal of artificial intelligence research*, vol. 13, pp. 227–303, 2000.
- [49] A. L. Gibbs and F. E. Su, “On choosing and bounding probability metrics,” *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.
- [50] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” 2021. [Online]. Available: <https://arxiv.org/abs/2109.11978>