An Adaptable Approach to Learn Realistic Legged Locomotion without Examples

Daniel Ordonez-Apraez[1], Antonio Agudo[1], Francesc Moreno-Noguer[1] and Mario Martin[2,3]

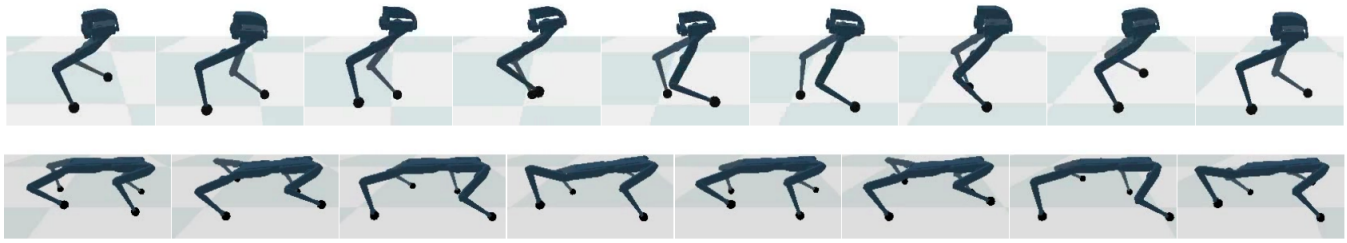

Fig. 1: Control policies for the biped robot Bolt ($1.05[m/s]$) and quadruped robot Solo ($0.96[m/s]$) learned with model-free RL learning guided with the spring-loaded inverted pendulum model. All gaits across training seeds feature periodicity, limbs coordination and natural spring-mass center of mass dynamics.

***Abstract*— Learning controllers that reproduce legged locomotion in nature has been a long-time goal in robotics and computer graphics. While yielding promising results, recent approaches are not yet flexible enough to be applicable to legged systems of different morphologies. This is partly because they often rely on precise motion capture references or elaborate learning environments that ensure the naturality of the emergent locomotion gaits but prevent generalization. This work proposes a generic approach for ensuring realism in locomotion by guiding the learning process with the spring-loaded inverted pendulum model as a reference. Leveraging on the exploration capacities of Reinforcement Learning (RL), we learn a control policy that fills in the information gap between the template model and full-body dynamics required to maintain stable and periodic locomotion. The proposed approach can be applied to robots of different sizes and morphologies and adapted to any RL technique and control architecture. We present experimental results showing that even in a model-free setup and with a simple reactive control architecture, the learned policies can generate realistic and energy-efficient locomotion gaits for a bipedal and a quadrupedal robot. And most importantly, this is achieved without using motion capture, strong constraints in the dynamics or kinematics of the robot, nor prescribing limb coordination. We provide supplemental videos for qualitative analysis of the naturality of the learned gaits[4].**

I. INTRODUCTION

The study and generation of natural legged locomotion is a problem that has been approached both by the robotics and computer animation communities. In robotics, the aim is to enable mobile platforms to locomote in various terrains with the flexibility, energy efficiency, and robustness observed in animals in nature. Alternatively, computer graphics focus on generating animations of legged locomotion that appear natural and are compliant to physical laws and interactions with the virtual environment.

Reinforcement Learning, especially in computer graphics, has shown remarkable progress in generating physics-based controllers for complex animated characters in simulation that showcase highly complex and dynamic motion skills [24, 29, 30]. While demonstrating impressive results, the scalability of these methods to robots of different morphologies (i.e., body size, mass, legs anatomy, and number of legs) is somewhat compromised. Mainly because, to ensure realism, they rely on: (1) Motion capture references [23, 30]. (2) Precise modeling of the kinematic constraints and approximate dynamics of the musculoskeletal system [20, 35]. (3) Learning environments that feature elaborate curriculum learning techniques [35] or meta-learning of multiple motion skills [18].

These approaches aim to aid the learning process by guiding or constraining solution space (i.e., the feasibility region of the optimal policy [24]) to avoid converging to unnatural motions. The necessity to do so comes from the fact that for most legged systems, there exist an extensive set of feasible control policies that maintain locomotion; For instance, a human can move forwards at a speed of $2.0\,[m/s]$ by jogging naturally but also by jumping, limping, using a single leg, and so on. The low variance of gait types in nature across individuals of the same species is a consequence of the tendency of animals to choose the properties of their gaits to maximize stability and robustness while minimizing energy expenditure [1].

For most animals in nature, the selection of the optimal gait that minimizes the Cost of Transport (CoT) (i.e., energy used over distance traveled) is a complex and not well-understood function of the forward velocity, terrain properties, and the dynamics of the animal's musculoskeletal system [1, 26]. For this reason, replicating this optimization

[1] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain [dordonez, aagudo, fmoreno]@iri.upc.edu
[2] Departament de Ciències de la Computació, Universitat Politècnica de Catalunya. [3] Barcelona Supercomputing Center (BSC). mmartin@cs.upc.edu
This work's experiments were run at the Barcelona Supercomputing Center in collaboration with the HPAI group. This work is supported by the Spanish government with the project MoHuCo PID2020-120049RB-I00 and the ERA-Net Chistera project IPALM PCI2019-103386.
[4] Supplemental videos: https://bit.ly/30NtYAu

function in simulation is still an open problem, especially since accurate models of the body dynamics are hard to attain or require considerable design efforts. However, bio-inspired models of legged locomotion are able to approximate valuable properties of the dynamics of locomotion in nature [13, 19], providing substantial queues of the gait properties and gait selection mechanisms in animals.

This work shows that realistic high-speed locomotion can be learned by guiding the control policy with a Spring-Loaded Inverted Pendulum (SLIP) template model. To achieve this, we design an RL environment that employs controlled trajectories of SLIP as reference motions. These are then used to loosely constrain the allowed robot Center of Mass (CoM) trajectories in time, effectively limiting the policy's exploration space to full-body state-action trajectories that feature SLIP CoM dynamics.

This learning environment is designed to mimic several properties of full-body CoM trajectory optimization techniques, as the policy is encouraged to learn full-body control that results in energy-efficient CoM trajectories in the vicinity of the reference motion. The main difference is the use of RL to address the optimal control problem instead of traditional convex trajectory optimization methods [3, 7, 8]. With the use of RL we lose the assurances of convex optimization algorithms but gain flexibility in experimentation by avoiding relying on simplifying assumptions of the gait and body dynamics (e.g., centroidal dynamics, zero CoM angular momentum, knowledge of Centers of Pressure (CoP) during contacts, contact timings). Furthermore, our learning setup is compatible with both model-based and model-free RL and with the use of any feed-forward controller.

Having a generic approach to learning realistic legged locomotion in simulation can be a valuable tool for gait analysis and robot and control design. Some immediate practical applications are the estimation of design requirements (e.g., joint and actuator profiles), identification of passive components, or generation gait information (e.g., centroidal or CoM linear and angular momentum trajectories) to improve the performance of trajectory optimization methods [2]. This work takes a small step towards such a generic system.

II. BACKGROUND

A. Spring Loaded Inverted Pendulum

SLIP is a low-order template model for the dynamics of legged locomotion in the sagittal plane. In its simplest form, the model comprises of a point-mass m representing the body's center of mass and a massless spring-leg that models compliance during contact with the ground (see Fig. 2-left) [12].

Although the model highly simplifies the dynamics of locomotion, it can predict the CoM dynamics' low-frequency components in the sagittal plane (see Fig. 2-right) [12, 13, 19]. The potential of SLIP as a generic model for animal locomotion in nature was proven by Full et al. [11] in a comparative study of the dynamics of high-speed locomotion across different animal species and body sizes. The study revealed that most animals experience similar profiles of

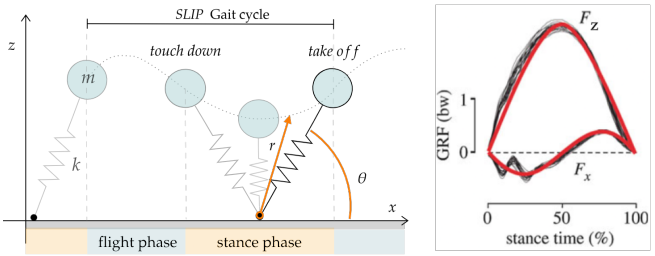

Fig. 2: **SLIP Model parameters and stance cycle. Left:** SLIP gait cycle with flight and stance phases. m and k are the model parameters mass and spring stiffness constant, respectively. **Right:** SLIP predicted ground reaction forces as a function of the stance time is plotted in red. Traces of ground reaction forces observed in human running experiments in black. Right image adapted from [12].

Ground Reaction Forces (GRFs) while running and that these forces could be approximated with SLIP models. Furthermore, there was evidence of patterns in the parametrization of the model for each animal and gait type, allowing to approximate the animal's GRF during locomotion with a SLIP model parameterized with the animal's mass m, hip height r_0, and spring constant k given by:

$$k = \frac{k_{rel}\, m\, g}{r_0} \quad \text{with} \quad k_{rel} = \frac{\frac{F}{mg}}{\frac{\Delta r}{r_0}}, \tag{1}$$

where F is the maximum GRF measured for each animal in the stance phase, Δr indicates an approximated maximum vertical displacement of the CoM during the stance phase, k_{rel} represents a dimensionless spring constant used to compare animals across species, and g is the gravity constant. The parametrization of SLIP models' spring constant is usually guided by the distribution of k_{rel} values observed in nature (see further details in [11]).

Information encoded in SLIP dynamics. The SLIP model dynamics provides insides on the expected time of contact during a stance phase (as a function of the number of legs in contact, the spring constant, and velocity at touch-down), the expected ground reaction forces, and gait period. Additionally, it models the natural fluctuations of the animal's kinetic, potential, and elastic potential (of muscles) energies during locomotion [11, 12].

B. Maximum Entropy Reinforcement Learning

Maximum entropy RL is a framework that exploits the concept of entropy from information theory [4, 17] and introduces it to the classical RL policy objective to address the known challenge of exploration-exploitation [33]. In this framework, agent's policies are modeled by parametric probability distributions, and the agent is tasked with learning to succeed at a given task by maximizing the expected reward while at the same time acting as randomly as possible (i.e., maximize the policy distribution entropy) [17]. The augmented policy objective is therefore defined as:

$$\pi^* = \underset{\pi}{\text{argmax}} \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_{pi}} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\cdot|\mathbf{s}_t) \right], \tag{2}$$

where π represents the RL agent policy, $\mathcal{H}(\cdot|\mathbf{s}_t)$ refers to the entropy of the policy given the present state, α the temperature parameter for entropy maximization, and $r(\mathbf{s}_t, \mathbf{a}_t)$ represents the environment reward as a function of the agent present state $\mathbf{s}_t$ and action $\mathbf{a}_t$.

Maximum entropy RL has been wildly adopted in the robotics community [16, 18, 21, 30] as its reduced sample complexity is promising for real-life applications. In this work, we use Soft Actor-Critic (SAC) [17], an off-policy actor-critic algorithm that has shown remarkable sample complexity, exploration capacities and requires almost no hyper-parameter tuning. This algorithm was used for real-world locomotion learning with success in [16, 22].

C. Space-Time Bounds

Space-Time bounds is a technique introduced in [24] to ensure that a control policy learns dynamic skills while remaining in the close neighborhood of a reference motion. The technique uses an early-termination criteria to stop the RL episodes whenever the agent's state at a given time $\mathbf{s}_t$ violates a fixed or dynamical state constraint (e.g., detection of robot fall or a velocity limit violation). In our work, the reference SLIP motions are in CoM space rather than in joint space or cartesian/system space (see [28] for details).

By constraining the episode state dynamics with a space-time bound $\mathcal{B}$, we effectively ensure that the control policy learned through experimentation generates causal state trajectories that remain within $\mathcal{B}$. This set of possible causal state trajectories allowed by $\mathcal{B}$ is referred to as the feasible region (or solution space) $\mathcal{F}(\mathcal{B})$ of the control policy.

III. RELATED WORK

A. Learning Realistic Legged Locomotion

Previous locomotion learning approaches use various forms of curriculum learning techniques to guide the learning process and reduce exploitation of simulation inaccuracies or unmodeled dynamics by the learned policy. The benchmark tasks on locomotion train an agent to progressively maximize its forward velocity during learning, using the increasing velocity as a mechanism for pruning unstable and undesired gaits [9]. Other approaches rely on modeling more realistically musculoskeletal system dynamics; for instance, in [35], the authors tune the damping and friction coefficients of the DoF of each robot to encourage realism, and aid the training process with a curriculum learning that stabilizes the robot during the early stages of training. Another work [20], approximates realistic state-dependent joint torque limits of humanoid robots to ensure the emergence of realistic motions using both trajectory optimization techniques and model-free RL. Lastly, an alternative technique to encourage realism is to train the robots in parallel on multiple motion tasks in a meta-learning environment, reducing the capacity of the policy to exploit unnatural gaits that are functional for single-tasks environments [18].

			Revolute joints limits		
	r_0[cm]	m[kg]	Pos[rad]	Vel[rad/s]	Torque[N/m]
Bolt	35	1.3	$[-\pi, \pi]$	$[-4\pi, 4\pi]$	[-2.7, 2.7]
Solo	24	2.2	$[-\pi, \pi]$	$[-4\pi, 4\pi]$	[-2.7, 2.7]

TABLE I: Bolt and Solo hip height r_0 and mass m alongside their joints position, velocity and torque limits.

B. Guiding controllers with low-order template models

Using low-order template models to design or guide controllers is not a novel idea. The assumption of centroidal dynamics is a common technique to enable the convex trajectory optimization methods that fuel most state-of-the-art platforms [3, 7]. These controllers are sensitive to initialization [2], require prescribed contact timings, and struggle to perform in unstructured environments [21].

In the context of RL, a bipedal extension of the SLIP model was used to successfully guide the learning process of a walking gait for the Cassie robot [14].

IV. METHODOLOGY

In this work, we evaluate the potential use of the dynamics of the SLIP template model in the problem of learning controllers for legged locomotion that achieve natural and realistic behavior. The motivations to study this problem are the following:

- The SLIP model provides a flexible mechanism for approximating the CoM dynamics of high-speed legged locomotion observed in nature, making it an ideal candidate for generating reference motions that bias the learned control policy to appear natural.
- A well-structured RL environment offers the ideal learning flexibility to learn control policies that fill in the information gap between the template model CoM dynamics and the robot full-body dynamics while making no assumptions about the legged system morphology.
- Realistic and energy-efficient gaits of high-speed locomotion learned in simulation can become valuable assets in studying the role of individual legs and CoM momentum during gaits.

Our experiments focus on learning control policies that maintain a constant forward velocity through a periodic, realistic, and energy-efficient gait. We test our approach on the bipedal robot Bolt (6 DoF) and a quadrupedal robot Solo (8 DoF) [15]. The robots' kinematics can resemble those of an Ostrich and Cat, but in reality, they have dynamics that differ significantly from any animal in nature, as they are abnormally lightweight and possess joints that are greatly under-constrained and overpowered (see table I).

For each robot, we independently learn control policies for moving at constant velocities of medium and high magnitudes (i.e., 3 and 6 times the robot's hip height per second). For Bolt we consider $\dot{x}_{des} = [1.05, 2.10]\,[m/s]$ and for Solo $\dot{x}_{des} = [0.60, 0.96]\,[m/s]$.

A. Enforcing SLIP dynamics

The reference motions for each robot and forward velocities are obtained through the optimal control of a SLIP

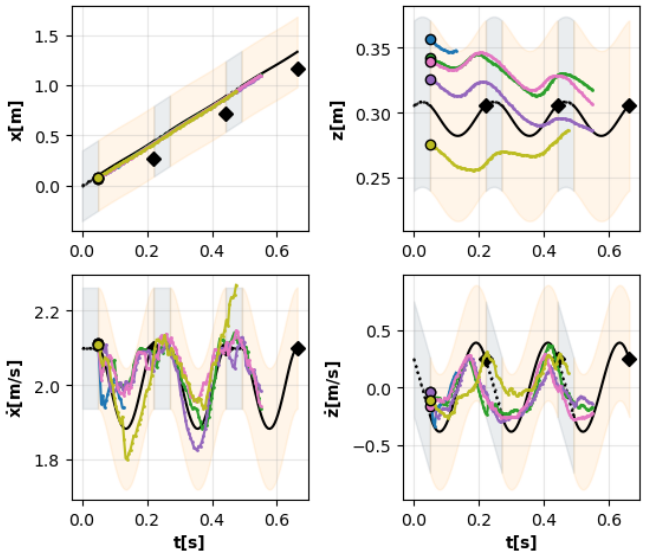

Fig. 3: **Reference SLIP gait CoM trajectory** in the sagittal plane for Bolt maintaining a forward velocity of $2.1[m/s]$ with an Space-Time Bound $\mathcal{B}$ with $\epsilon = 0.75$ (shaded area). Gray and yellow areas represent flight and stance phases of the SLIP gait, respectively. Experimental trajectories in the early stages of training are plotted to illustrate the boundary violation scenarios and episode terminations.

model [5, 27] that is parameterized with the total robot mass (m), the hip height (r_0) [32], and a relative spring constant $k_{rel} = 10.7$ according to [11, 13]. The number of legs expected to contact the ground at the same time during a stance phase is 1 for the bipedal robot Bolt and 2 for the quadrupedal Solo.

Considering that the SLIP model only approximates the low-frequency CoM dynamics during legged locomotion [11, 19] (see Fig. 2 right), during learning, we loosely constrain the dynamics of the robot CoM with the reference motion by terminating the learning episode when the robot violates a space-time bound around the reference SLIP CoM trajectory. This condition is evaluated by considering the bound $\mathcal{B}(t)$ at every time step t in simulation as:

$$\mathcal{B}(t) = \{\mathbf{s}_G(t) \mid \rho > |\mathbf{s}_G(t) - \mathbf{s}_{SLIP}(t)|\}, \quad (3)$$

$$\mathbf{s}_G = [x_G, y_G, z_G, \dot{x}_G, \dot{y}_G, \dot{z}_G], \quad (4)$$

$$\mathbf{s}_{SLIP} = [x_S, y_S, z_S, \dot{x}_S, \dot{y}_S, \dot{z}_S], \quad (5)$$

where $\mathbf{s}_G$ represents the robot's CoM state in the sagittal plane, $\mathbf{s}_{SLIP}$ is the reference SLIP trajectory state, and ρ is a vector that controls the size of the bound (see Fig. 3).

To independently avoid tuning the size of each dimension of the bound for each robot and forward velocity, we parameterize ρ utilizing the robot dimensions and the present gait cycle of the reference trajectory as:

$$\rho = \epsilon \left[\infty, \frac{r_0}{2}, \frac{r_0}{4}, \hat{\dot{x}}_S, \frac{\hat{\dot{z}}_S}{2}, \frac{\hat{\dot{z}}_S}{2}\right], \quad (6)$$

where $\hat{\dot{x}}_S$ and $\hat{\dot{z}}_S$ represent the difference between the maximum and minimum forward and vertical velocities of the SLIP trajectory in the entire present gait cycle, respectively. Lastly, ϵ is a scalar coefficient that defines the size of the space-time bound.

Space/Energy-Time bounds. The bound $\mathcal{B}$ in Eq. (5) and its implications have a clearer intuition in terms of the system energy: By constraining the robot's CoM height and velocity in the sagittal plane, we effectively bound the robot's potential and CoM kinetic energies in time. Therefore, the imposed feasibility region of the control policy $\mathcal{F}(\mathcal{B})$ constrain the possible locomotion gaits to those that behave energetically similarly to the template model, i.e., gaits that have variations in phase of the potential and CoM kinetic energy. If the bound is sufficiently tight, this encourages the robot to use its legs to mimic the energy storage and release mechanisms observed in legged systems in nature during high-speed locomotion [12, 19].

B. Reinforcement Learning Environment

The Markov Decision Process (MDP) used in this work is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, r, \mathcal{T}, \gamma, s_0 \rangle$. $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively. $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}+$ is a positive reward function defined for every state action pair. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ represents the environment dynamics transition function, governed by the physics simulator Bullet [6]. Moreover, γ is the discount factor, and s_0 indicates the initial state distribution.

As in [24], the reward is a positive real function so that the policy is encouraged to remain within $\mathcal{F}(\mathcal{B})$. The reward can be represented by $r = r_S \cdot r_P$, where $r_S \in \{0, 1\}$ is a survival reward taking the value of 1 whenever the robot state is within the bound $\mathcal{B}$ and 0 otherwise, and $r_P \in [0, 1]$ is a multiplicative energy and torque minimization reward, similar to the ones used in [9], defined as:

$$r_P = \left(1 - \frac{\|\tau \dot{q}\|}{\|\tau_{max} \dot{q}_{max}\|}\right) \cdot \left(1 - \frac{\|\tau\|}{\|\tau_{max}\|}\right), \quad (7)$$

where τ is the joint torques and $\dot{\mathbf{q}}$ the joint velocities.

Intuitively, the early episode termination using the bound $\mathcal{B}$ constrains the exploration of the RL agent to remain close to the reference SLIP trajectory in CoM space, i.e., the agent will only accumulate experiences of causal state-action trajectories that maintain the robot's CoM within the $\mathcal{F}(\mathcal{B})$. At the same time, due to the positive reward, as learning progresses, the agent will try to survive by following state-action trajectories of a longer duration, while the energy minimization reward penalizes inefficient trajectories.

This learning environment was designed as a CoM full-body trajectory optimization: The RL exploration (in state-action space) is constrained to the local vicinity of the reference trajectory (in CoM space) and the cost/reward function is composed solely of an energy minimization term.

C. Maximum-Entropy RL and Control architecture

The learning setup of this work is framed as a Maximum Entropy Inverse Reinforcement Learning (IRL) task [25, 37], where the SLIP dynamics are used as the demonstrations and the objective of the optimization is to find the optimal

control policy that maximizes the expected reward and policy distribution entropy (see Eq. (2)). The algorithm used is SAC with automatic temperature adjustment [17].

The policy $\pi_\phi(\mathbf{a}|\mathbf{s})$ and Q functions $Q_\psi(\mathbf{s}_t, \mathbf{a}_t)$ of SAC are parameterized with Fully-Connected Neural Networks ϕ and ψ with two hidden layers of 256 neurons and ReLu activation functions. As in [17], the control policy is modeled as a state conditional $tanh$ transformed normal distribution $\pi_\phi(\mathbf{a}|\mathbf{s}) \sim \tanh\mathcal{N}(\mu_\phi, \Sigma_\phi)$ with mean and diagonal covariance matrix parameterized by ϕ. All experiments are run with $\gamma = 0.995$ and a learning rate of $lr = 3e^{-4}$. Lastly, the control frequency is set to $200Hz$, considering the reactive nature of the policy.

D. State and Action space

We use a reactive policy that directly controls the torques of the robot joints τ, by scaling the action $\mathbf{a} \sim \pi_\phi(\mathbf{a}|\mathbf{s})$ by each DoF torque limits (see table I). The MDP state space $\mathbf{s}$ is defined as:

$$[\mathbf{h}_G, z_G, R, \mathbf{q}, cos(\phi_{SLIP}), sin(\phi_{SLIP}), \dot{x}_{des} - \dot{x}_G, \mathbf{c}], \quad (8)$$

where $\mathbf{h}_G = A_G(\mathbf{q})\dot{\mathbf{q}}$ is the robot's CoM momentum expressed in the robot base coordinates, computed using the Centroidal Momentum Matrix (CMM) $A_G(\mathbf{q})$ [28]. z_G is the robot's base height. R is the 6D continuous representation of the robot's base orientation in world coordinates [36]. As in [21], ϕ_{SLIP} is a gait phase signal increasing linearly from 0 to π during the flight phase and from π to 2π on the stance phase of the SLIP gait cycle. Lastly, $\dot{x}_{des}$ is the desired forward velocity, and $\mathbf{c}$ is a binary vector indicating contact of the robot's feet with the ground as in [35]. Note that the state observation is invariant w.r.t the direction of forward motion.

E. Directional and Sagittal Symmetry

A relevant property of gaits in nature is their symmetry: Animals, including humans, tend to control their body's left and right parts similarly when not injured. In the context of RL, [38] showed that when MDPs have symmetry, the optimal policy, Value, and Q functions are also symmetrical. While in legged locomotion learning [35] reported enhanced performance and realism of gaits when the symmetry was encouraged in the control policy.

In this work, we extended the policy symmetry loss in [35] to function for SAC's policies. The main difference is that [35] uses a Proximal Policy Optimization (PPO) policy which has fixed variance in their action distribution, while in SAC, the variance for the action distribution is also a function of the input state (Σ_ϕ). Therefore our policy symmetry loss is defined as:

$$\begin{aligned}\mathbb{L}_{sym} =& \|\bar{\pi}_\phi(\mathbf{s}) - \Psi_a(\bar{\pi}_\phi(\Psi_s(\mathbf{s})))\| + \\ & \|\hat{\pi}_\phi(\mathbf{s}) - |\Psi_a(\hat{\pi}_\phi(\Psi_s(\mathbf{s})))|\|, \end{aligned} \quad (9)$$

where $\bar{\pi}_\phi = \mathbb{E}[\pi_\phi(\mathbf{a}|\mathbf{s})]$ represents the mean value of the policy distribution, and $\hat{\pi}_\phi = \mathbb{E}[\pi_\phi(\mathbf{a}|\mathbf{s}) - \bar{\pi}_\phi]$ the variance of the action policy distribution. Lastly, Ψ_s and Ψ_a are functions mapping states and actions to their mirrored equivalents.

Furthermore, influenced by [38], we encouraged symmetry in the Critic Q-functions by augmenting each batch sampled from the replay buffer with its symmetric equivalent.

Note that encouraging symmetry in control actions merely requires the control policy to have similar reactions to symmetric states, but not necessarily symmetric gaits trajectories.

V. EXPERIMENTAL RESULTS

We evaluated our approach by learning control policies for both robots with and without enforcing the SLIP dynamics and symmetry. The control policies trained without enforcing the SLIP dynamics use a space-time bound $\mathcal{B}_{const}$ where only the CoM forward velocity and lateral position are bounded. The bound for the forward velocity is centered around the desired constant speed $\dot{x}_{des}$, with $\epsilon = 2.0$ to allow for the expected oscillation around the target velocity proper of animal gaits. Therefore the policy feasibility region $\mathcal{F}(\mathcal{B}_{const})$ contains any locomotion gait that can remain close to the target velocity. In the rest of the document, we refer to the policies trained using the SLIP space-time bound $\mathcal{B}$ as *with-SLIP*, and *without-SLIP* for the ones using $\mathcal{B}_{const}$.

A. Qualitative and quantitative effects of imposing SLIP dynamics

The control policies *without-SLIP* for both robots find stable gait trajectories that maintain forward velocity within the imposed bound $\mathcal{B}_{const}$. However, realistic or natural behavior is not present in any of the training seeds (see comparison video [4]). Due to the large feasibility region, each training seed rapidly converges to unnatural aperiodic gaits that exploit the unmodeled dynamics (e.g., actuator dynamics, joint-space friction, and damping) and the simulation inaccuracies in contact dynamics (specifically the transition from static to dynamic friction regimes). Furthermore, because the elastic properties of muscles are not modeled, the energy minimization reward r_P leads both robots to locomote with short and rapid stepping with nearly fully extended legs (since such a gait minimizes the required energy and torques for compensating gravity).

On the contrary, policies trained *with-SLIP* showcase natural limb coordination and spring-mass dynamics, resulting in realistic and periodic gaits. Because there is no penalization on the torso's linear and angular momentum, the robots learn to locomote with graceful and natural torso motions that sync with leg contacts. Both robots perform irregular footstep sequences when needed to stabilize but tend to converge to near-periodic footstep sequences once they achieve to remain in a limit cycle. In all experiments, Solo converges to use a trot gait, coordinating diagonally opposed legs, while Bolt uses a running gait.

From a quantitative perspective, the control policies without-SLIP showcase similar or better performance in sample complexity, gait robustness, and energetic cost-of transport (see Fig. 4). Therefore, for the RL agent, the unnatural gaits represent suitable and close to optimal control

[4] Videos *with-SLIP* vs. *without-SLIP*: youtu.be/_0Y3sJ9w9F8

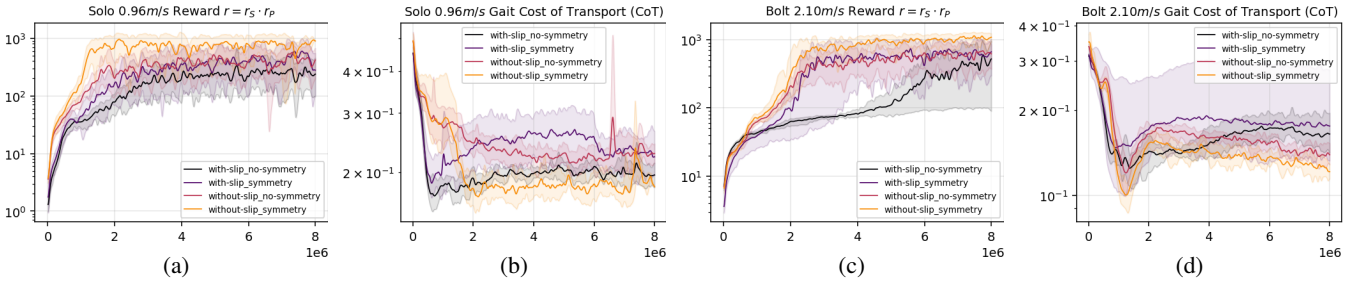

Fig. 4: **Reward curves and gait's Cost of Transport** for high speed locomotion of Solo $0.96[m/s]$ (a and b) and for Bolt $2.1[m/s]$ (c and d). All training curves show the step average, maximum and minimum (shaded region) across 5 training seeds. Exploiting symmetry results in more robust control policies (i.e., higher and faster survival rates) across all training variants. For Bolt exploiting symmetry reduces considerable the sample-complexity. Although quantitatively the variants *with-SLIP* and symmetry appear not to perform better, we highly encourage the reader to compare the gait properties qualitatively in the supplemental video 4.

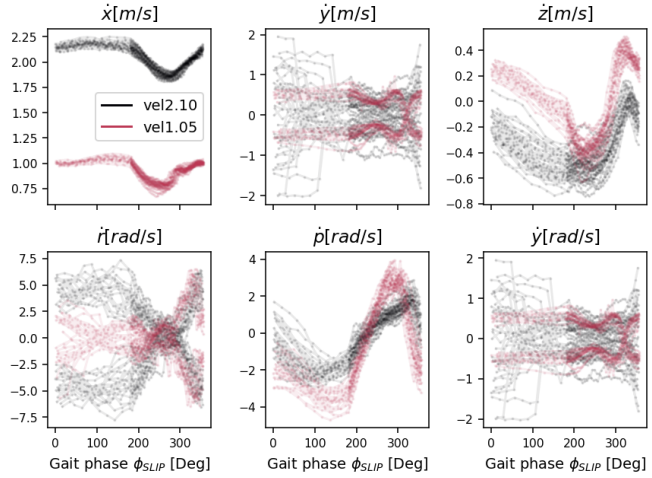

Fig. 5: ***With-SLIP* and symmetry linear and angular gait centroidal velocities** of the torso of Bolt, at medium and high forward velocities. The gaits showcase periodicity and spring-mass dynamics proper of the SLIP model. For the lateral velocity $\dot{y}$, roll $\dot{r}$ and yaw $\dot{y}$ the two trajectory modes correspond to gait cycles with left and right leg contact

policies. This minor difference in quantitative performance highlights the challenges of obtaining realistic behavior with learning techniques without introducing inductive biases in the learning process to encourage realism.

Gait variance across seeds. Perhaps the most relevant impact of enforcing SLIP gait dynamics through space-time bounds is the reduction of the variance of gait properties across training seeds, meaning that final policies for each seed converges to locomotion gaits that feature similar limb coordination and CoM trajectories (see all seed's gaits in supplemental video 4).

Utility of learned gaits. Figure 5 presents the robot base centroidal trajectory during the SLIP gait cycle for the Bolt. This trajectory and those of all training seeds alongside other gait properties can be exploited by centroidal or CoM trajectory optimization methods [8, 28, 31, 34].

B. Impact on symmetry enforcing

The training results in Fig. 4 show a considerable increase in performance by exploiting the symmetry of the robot control and the MDP for both *with-SLIP* and *without-SLIP* variants, as there are signs of improved sample complexity and control robustness (i.e., ability to maintain stable gaits). The impact of symmetry in the realism of the gaits is best observed qualitatively in the comparison video [5].

VI. CONCLUSIONS AND FUTURE WORK

In this work, we show the potential use of the SLIP model as a mechanism for measuring and asserting realism in gaits of high-speed locomotion. Taking a step towards locomotion learning without MoCap references. To achieve this, we designed an RL environment capable of producing realistic full-body periodic locomotion gaits for any robot morphology, using a simple energy minimization reward. We present results with a bipedal and quadrupedal robot in simulation for the model-free control case and a reactive policy acting directly on joint torques.

The advantage of our approach relies on the use of controlled trajectories of the SLIP model as a reference to loosely constrain the policy exploration space. This model can be parameterized to approximate the dynamics of high-speed locomotion of most animals/robots. Enabling applications where no motion capture data is available.

In future work we want to: (1) Enforce symmetry as a hard constraint in the optimization process by using equivariant function approximators for the actor and critic functions [10]. (2) Expand the experiments scope to embrace a larger suite of robot morphologies (and sizes), terrain properties, and energy penalization schemes (based on various models of the cost of transport [1]). (3) Explore the use of extended versions of the SLIP model to address the realism of walking gaits of different robot morphologies. (4) Adapt our approach for Computational Design: By studying the learned robot dynamics at different speeds, we theorize, we can test potential passive components for each DoF to optimize gait energy expenditure, control robustness, and/or the robot's kinematics and dynamics.

[5] Videos *with-symmetry* vs. *without-symmetry*: youtu.be/qN25-XQBqKM

References

[1] R McNeill Alexander. "Models and the scaling of energy costs for locomotion". In: *Journal of Experimental Biology* 208.9 (2005), pp. 1645–1652.

[2] Gerardo Bledt and Sangbae Kim. "Extracting legged locomotion heuristics with regularized predictive control". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 406–412.

[3] Gerardo Bledt et al. "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.

[4] Tom Carter. "An introduction to information theory and entropy". In: *Complex systems summer school, Santa Fe* (2007).

[5] Hua Chen, Patrick M Wensing, and Wei Zhang. "Optimal control of a differentially flat two-dimensional spring-loaded inverted pendulum model". In: *IEEE Robotics and Automation Letters* 5.2 (2019), pp. 307–314.

[6] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. `http://pybullet.org`. 2016–2021.

[7] Jared Di Carlo et al. "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.

[8] Yanran Ding, Abhishek Pandala, and Hae-Won Park. "Real-time model predictive control for versatile dynamic motions in quadrupedal robots". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8484–8490.

[9] Yan Duan et al. "Benchmarking deep reinforcement learning for continuous control". In: *International conference on machine learning*. PMLR. 2016, pp. 1329–1338.

[10] Marc Finzi, Max Welling, and Andrew Gordon Wilson. "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3318–3328.

[11] Robert J Full, Claire T Farley, and Jack M Winters. "Musculoskeletal dynamics in rhythmic systems: a comparative approach to legged locomotion". In: *Biomechanics and neural control of posture and movement*. Springer, 2000, pp. 192–205.

[12] H Geyer et al. "Gait based on the spring-loaded inverted pendulum". In: *Humanoid Robotics: A Reference, A. Goswami and P. Vadakkepat, Eds. Springer Netherlands* (2018), pp. 1–25.

[13] Hartmut Geyer. "Simple models of legged locomotion based on compliant limb behavior — Grundmodelle pedaler Lokomotion basierend auf nachgiebigem Beinverhalten". PhD thesis. 2005.

[14] Kevin Green et al. "Learning Spring Mass Locomotion: Guiding Policies with a Reduced-Order Model". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3926–3932.

[15] F. Grimminger et al. "An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3650–3657.

[16] Tuomas Haarnoja et al. "Learning to walk via deep reinforcement learning". In: *arXiv preprint arXiv:1812.11103* (2018).

[17] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[18] Roland Hafner et al. "Towards general and autonomous learning of core skills: A case study in locomotion". In: *arXiv preprint arXiv:2008.12228* (2020).

[19] Philip Holmes et al. "The dynamics of legged locomotion: Models, analyses, and challenges". In: *SIAM review* 48.2 (2006), pp. 207–304.

[20] Yifeng Jiang et al. "Synthesis of biologically realistic human motion using joint torque actuation". In: *ACM Transactions On Graphics (TOG)* 38.4 (2019), pp. 1–12.

[21] Joonho Lee et al. "Learning quadrupedal locomotion over challenging terrain". In: *Science robotics* 5.47 (2020).

[22] Tianyu Li et al. "Learning generalizable locomotion skills with hierarchical reinforcement learning". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 413–419.

[23] Ying-Sheng Luo et al. "Carl: Controllable agent with reinforcement learning for quadruped locomotion". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 38–1.

[24] Li-Ke Ma et al. "Learning and Exploring Motor Skills with Spacetime Bounds". In: *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, pp. 251–263.

[25] Tien Mai, Kennard Chan, and Patrick Jaillet. "Generalized Maximum Causal Entropy for Inverse Reinforcement Learning". In: *arXiv preprint arXiv:1911.06928* (2019).

[26] AE Minetti and R McN Alexander. "A theory of metabolic costs for bipedal gaits". In: *Journal of Theoretical Biology* 186.4 (1997), pp. 467–476.

[27] Daniel Felipe Ordoñez Apraez. "Learning to run naturally: guiding policies with the Spring-Loaded Inverted Pendulum". MA thesis. Universitat Politècnica de Catalunya, 2021.

[28] David E Orin and Ambarish Goswami. "Centroidal momentum matrix of a humanoid robot: Structure and properties". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2008, pp. 653–659.

[29] Xue Bin Peng et al. “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills”. In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), pp. 1–14.

[30] Xue Bin Peng et al. “Learning agile robotic locomotion skills by imitating animals”. In: *arXiv preprint arXiv:2004.00784* (2020).

[31] Brahayam Ponton et al. “On time optimization of centroidal momentum dynamics”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 5776–5782.

[32] Herman Pontzer. “Effective limb length and the scaling of locomotor cost in terrestrial animals”. In: *Journal of Experimental Biology* 210.10 (2007), pp. 1752–1761.

[33] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[34] Zhaoming Xie et al. “GLiDE: Generalizable Quadrupedal Locomotion in Diverse Environments with a Centroidal Model”. In: *arXiv preprint arXiv:2104.09771* (2021).

[35] Wenhao Yu, Greg Turk, and C Karen Liu. “Learning symmetric and low-energy locomotion”. In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), pp. 1–12.

[36] Yi Zhou et al. “On the continuity of rotation representations in neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5745–5753.

[37] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[38] Martin Zinkevich and Tucker Balch. “Symmetry in Markov decision processes and its implications for single agent and multi agent learning”. In: *In Proceedings of the 18th International Conference on Machine Learning*. Citeseer. 2001.