micromachines

Article

Hybrid Bipedal Locomotion Based on Reinforcement Learning and Heuristics

Zhicheng Wang [1], Wandi Wei [1], Anhuan Xie [2], Yifeng Zhang [3], Jun Wu [1,4] and Qiuguo Zhu [1,4,*]

[1] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China
[2] Zhejiang Lab, Hangzhou 311121, China
[3] Tianjin Jiazi Robot Technology Co., Ltd., Tianjin 300450, China
[4] State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China
* Correspondence: qgzhu@zju.edu.cn

Abstract: Locomotion control has long been vital to legged robots. Agile locomotion can be implemented through either model-based controller or reinforcement learning. It is proven that robust controllers can be obtained through model-based methods and learning-based policies have advantages in generalization. This paper proposed a hybrid framework of locomotion controller that combines deep reinforcement learning and simple heuristic policy and assigns them to different activation phases, which provides guidance for adaptive training without producing conflicts between heuristic knowledge and learned policies. The training in simulation follows a step-by-step stochastic curriculum to guarantee success. Domain randomization during training and assistive extra feedback loops on real robot are also adopted to smooth the transition to the real world. Comparison experiments are carried out on both simulated and real Wukong-IV humanoid robots, and the proposed hybrid approach matches the canonical end-to-end approaches with higher rate of success, faster converging speed, and 60% less tracking error in velocity tracking tasks.

Keywords: legged locomotion; reinforcement learning; humanoid; motion planning

Citation: Wang, Z.; Wei, W.; Xie, A.; Zhang, Y.; Wu, J.; Zhu, Q. Hybrid Bipedal Locomotion Based on Reinforcement Learning and Heuristics. *Micromachines* **2022**, *13*, 1688. https://doi.org/10.3390/mi13101688

Academic Editors: Zhangguo Yu and Marco Ceccarelli

Received: 29 August 2022
Accepted: 3 October 2022
Published: 7 October 2022

1. Introduction

In recent years, various legged robots are developed for traversing through rough and dangerous terrains and working as a replacement for human effort [1]. Specially, dynamic locomotion control, as an essential part of a robust and agile legged robot, has been studied from various aspects.

Model-based methods are first studied and developed [2]. Heuristics methods are widely adopted during early exploration, Raibert proposed hopping controller [2] and extended to bipedal and quadruple robots, and Pratt proposed Virtual Actuator Control [3] and Virtual Model Control (VMC) [4] on bipedal locomotion tasks. These methods adopt reduced modeling and abstract the legged robot into a linear inverted pendulum (LIP) [5] and its derivatives. To further improve the decision, optimization-based methods are introduced to locomotion control [6]. To prevent redundant iterations when searching for the optimal solution, Bledt et al. proposed a regularization-based approach [7]. It uses sub-optimal heuristics to inform the solver through cost function. As a low-cost policy that can approximate optimal solutions with linearization around the working limit cycle, heuristics are still valuable.

Model-based methods have made great advances and are tested to be precise, predictive, and adaptive in various scenes. As they are all based on simplified models that guarantee convergence and simplicity, the accuracy and performance drop when being far away from the working point.

As an emerging and model-free technology, deep reinforcement learning (DRL) has been a popular research field. Previous studies have demonstrated its potential for locomotion tasks. Haarnoja and Tan [8,9] first utilized learning-based gaits on a real robot and

verified the feasibility of the end-to-end route. Subsequently, the work in [10–12] presents learning separate skills such as trotting and fall recovery using a similar framework. As for bipedal locomotion, a harder task, Hurst [13–15] proposed special periodical switching rewards and achieved robust locomotion over planes and stairs. To recombine multiple motor skills into an adaptive multi-modal policy within one end-to-end framework, Yang [16] designed a switching mechanism guided by a phase indicator. Based on that, the multi-expert learning architecture [17] was proposed. It fuses multiple NNs into one according to a gating network's output to produce adaptive behaviors in response to changing situations. Meanwhile, expanding information sources with perceptual sensing such as heightmap scanning [18] and introducing latent encoders are also means to enhance trained policies. ANYmal robot conquered a series of challenging terrains with privilege learning and latent encoder [19,20]. Kumar et al. proposed Rapid Motor Adaptation framework [21] that enables latent space identification and realized quadrupedal locomotion without predefined references or trajectory generators.

On basis of end-to-end frameworks where the action is completely learned, the sum of model-based control signals and the action of learned policies provide external guidance and avoid pointless blind exploration during training. Low-cost model-based controllers such as central pattern generator [19,22,23], model-based gait library [24–26], and heuristic references [27,28] are often adopted in these approaches to achieve agile real-time controlled locomotion. In these frameworks, NN policies learn the residual between optimal decision and reference given by model-based modules. A common issue facing such a summation mechanism is that learned policy might conflict with the other component and cause rigid policy constrained by reference, or the learned policy overwrites the other [22]. Both parts counteract, resulting in a mediocre final performance.

The most similar methodology to ours is linear policy locomotion developed by Krishna [29,30], where the biped's swing leg is controlled by simple NN policy, and the stance leg is controlled by a model-based controller. However, we have an opposite understanding of the traits of walking phases. In previous studies, we observed that the stance leg performs agile movements through acting ground reaction force (GRF) instead of acting specific joint position trajectory. The swing leg works with a relatively low payload and without restriction from the ground, but the foothold decided by it significantly impacts the subsequent body states. Thus, joint-level accuracy and higher stiffness of swinging control are required. Swing control is a rather deterministic tracking task and can be accomplished by heuristic method and similar control policies with fixed-based manipulators [31]. Meanwhile, the stance leg faces more uncertainty during interaction with terrain and is often controlled by torque controllers [25]. Adopting a learning-based method on stance control can improve robustness compared with model-based approaches. It also narrows down the space for exploration and grants higher efficiency in training.

In pursuit of better performance, separated controllers should be activated during the corresponding phase and adapt to the phase's attribute, instead of applying a unified controller to fulfill different demands. Inspired by such philosophy, the paper proposes a novel conflict-free hybrid approach to generate adaptive bipedal locomotion with a higher success rate and is easy to transplant to other biped robots with different joint configurations since the policy produces GRF commands. The proposed method is computationally efficient, and also light-weight since there is no iteration-based numerical solver involved in the framework. The main contributions of this paper are as follows:

1. We proposed an efficient hybrid locomotion policy that divides the task into a stance part using model-free learning-based stance control and a swing part using heuristics swing control. The trained policy can perform controllable and regulated gait.
2. To prevent policy divergence caused by fierce randomization at early stages, we proposed a parallel curriculum learning schema with two-stage progressive domain randomization for challenging locomotion tasks.
3. To further reduce the magnitude of sensor randomization which withdraws training speed, we proposed heuristic-based regularization feedback loops on real-world robot

systems in addition to a two-branch hybrid controller for resisting simulation model mismatch and assisting stabilization.

The organization of this paper is as follows: Section 2 describes the hardware and software platform and notations. Section 3 introduces the overall structure of our method. Section 4 presents simulation and deployment results and comparisons with an end-to-end benchmark. Finally, Section 5 states the discussion and inspiration for future work.

2. Platform and Notations

2.1. Robot Model

The robot platform adopted in our work is Wukong-IV adult-size humanoid. Wukong-IV humanoid is designed and built by our research team. It is 1.4 m tall and weighs 45 kg, actuated by electric motor joints. The robot has 6 degrees of freedom (DoF) on each leg and 4 DoFs on each arm. As a bionic humanoid robot, Wukong-IV's joint configuration and mass distribution bare resemblance to biological humans. A picture of its appearance is shown in Figure 1A, the articulated system model used in simulation training is shown in Figure 1B, and more detailed specifications are listed in Table A1.

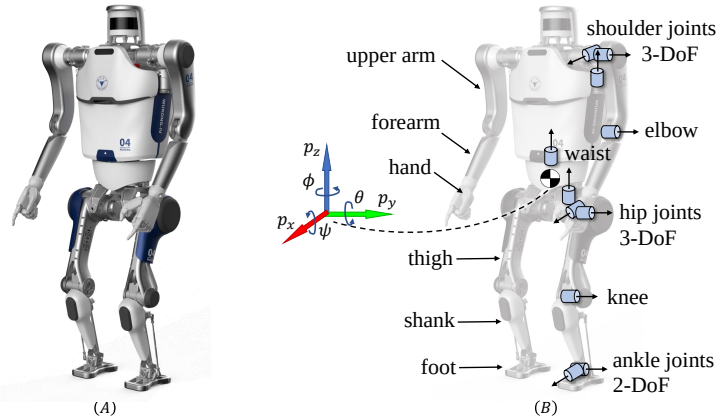

Figure 1. (**A**) Picture and hardware description of Wukong-IV humanoid robot. (**B**) Diagram of joint and link definition for Wukong-IV humanoid robot abstract model.

The simulation environment is built with RaiSim [32]. Policy neural networks and their training algorithm are utilized with PyTorch [33]. All training is performed on a desktop workstation with a central processor of Intel Xeon Gold 6242R and a graphic processor of NVIDIA Geforce RTX 3090.

2.2. Math Notations

In this article, the state of WuKong-IV's floating base is denoted by $q_b = \{p_x, p_y, p_z, \psi, \theta, \phi\}$. The first three elements compose base position, also denoted by p. The last three are roll, pitch, and yaw angle of base. Robot base quaternion is denoted by ξ. Base linear and angular velocity is v and ω. The joint level state is composed of joint angle q and joint angle velocity $\dot{q}$. In stance control, the policy gets input o_t and outputs action a_t. The swing foot's position in the robot frame is denoted by e. Joint torques of legs are denoted by τ_{stance} and τ_{swing}. To represents which foot touches the ground, gait is denoted by c.

For all notations, $(\cdot)_t$ means the quantity in time step t, $(\cdot)^*$ means the expected value of the quantity, and $\delta(\cdot)$ means compensation to be added to the scripted quantity.

3. Materials and Methods

The control frame of hybrid locomotion policy can be described in Figure 2. There are three main branches in the overall structure. The middle branch (blue blocks) shows the learning-based stance control policy (Section 3.1). The network outputs the expected GRF to the stance leg. GRF maps to joint torque via jacobian and then is acted by the robot. The upper branch (orange blocks) shows the heuristic swinging control policy (Section 3.3) which

is a position-based method inspired by capture point method. The next foot placement of swing legs is calculated from a clock signal and robot state, then cosine interpolation generates the air position of swing foot, then inverse kinematics converts the position into joint position, which is tracked by low-level PD. The lower branch (red block) is only used in physical deployment (Section 3.5).

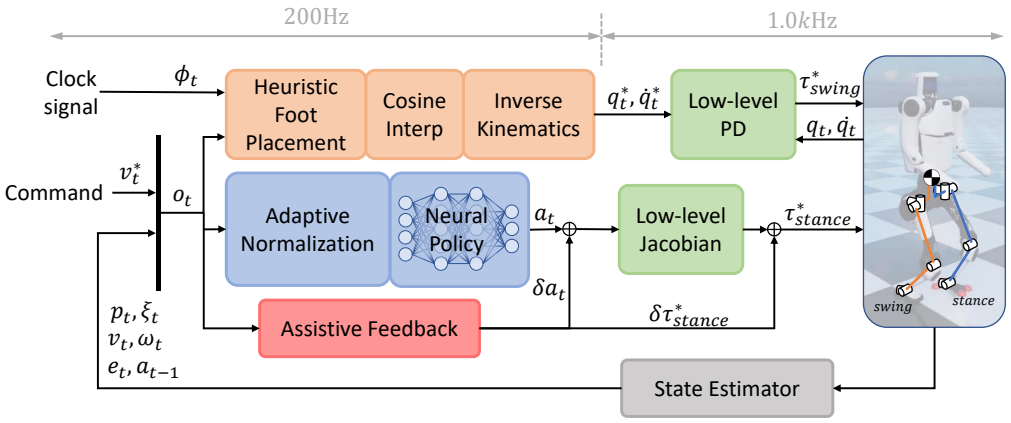

Figure 2. Control framework of hybrid walker. Locomotion task is divided into two different phases and managed by two separate branches. Heuristics swing controller (orange blocks) controls swing leg with a low-cost position-based approach. Learning-based stance controller (blue blocks) adopts NN policy for controlling. Assistive branch (red block) is active only on the real robot for compensating reality gap. Clock signal ϕ is a periodic ramp signal that increases from 0.0 to 1.0 throughout a gait period.

3.1. Learning-Based Stance Control

3.1.1. Observation Space and Action Space

Observation space o_t consists of 33 dimensions of proprioceptive information. The observation vectors are normalized before being given to policy to neutralize the amplitude difference between various observation channels (For normalization parameter update strategy, see Section 3.1.4). Its components are as follows.

$$o_t = \{q_{b,t}, q_b^*, \omega_t, v_t, \omega^*, v^*, e_t, a_{t-1}\} \tag{1}$$

Action space a_t consists of 6-dimension expected GRF which includes reaction force F and torque τ in XYZ directions. F_y component of action is mirrored along zero at the left stance phase so that the policy network only has to explore a simpler region rather than two unconnected regions. Its component is as follows.

$$a_t = \{F_x, F_y, F_z, \tau_x, \tau_y, \tau_z\} \tag{2}$$

3.1.2. Reward Design

Reward function design is also vital for training. Improper reward function would lead to reward hacking, a phenomenon where policy takes unintended behavior to obtain high reward value. Therefore, reward value should be a comprehensive evaluation of task performance and meanwhile dense and smooth enough to learn. For locomotion tasks, the reward function mainly consists of tracking terms that follow a goal and stabilizing terms that prevent falling or collision. In our work, the reward function is composed of three objectives including balancing, command tracking, and gait regulating. The full reward composition is shown below.

$$R_{track} = F_{\alpha_1,\beta_1}(||v_t^* - v_t||) \quad (3)$$
$$R_{pose} = F_{\alpha_2,\beta_2}(1 - \xi_t^{*T} \cdot \xi_t) \quad (4)$$
$$R_{height} = F_{\alpha_3,\beta_3}(|p_{z,t}^* - p_{z,t}|) \quad (5)$$
$$R_{gait} = F_{\alpha_4,\beta_4}(|c_t^* - c_t|) \quad (6)$$
$$R_{slip} = F_{\alpha_5,\beta_5}(||v_{ankle,t}||) \quad (7)$$
$$F_{\alpha,\beta}(x) = \alpha e^{-\beta x^2} \quad (8)$$
$$R_{fall} = \begin{cases} K & \text{for falling} \\ 0 & \text{for other} \end{cases} \quad (9)$$
$$R_{sum} = R_{track} + R_{pose} + R_{height} + R_{gait} + R_{slip} + R_{fall} \quad (10)$$

where $v_{ankle,t}$ means the ankle velocity of the stance leg. R_{pose} and R_{slip} require the robot to maintain balance with firm steps, R_{track} and R_{height} encourage the robot to traverse under certain commands, and R_{gait} regulates the gait frequency to meet the expected one. All vectors above are expressed in the world frame. R_{fall} penalizes major failures. $F_{\alpha,\beta}(x)$ is the Gaussian kernel function, and subscripts α and β are weight coefficients that need to be tuned. Every term except R_{fall} is transformed from negative errors to positive rewards with a Gaussian kernel to prevent proactive termination and set up upper boundaries for each reward term to prevent any term from over-growing and shadowing the others. For specific parameters we used in our experiments, check Table A2.

3.1.3. Policy Representation

In the aforementioned studies [8–12,14,15,19–21,26], Actor-Critic framework [34] is adopted to train the policy for locomotion tasks. Multiple layer perception (MLP) with two hidden layers and tanh as activation function is chosen for policy representation in our study because of its simple structure, which boosts training and transition to non-Python machines. Actor and critic networks share the same structure, configuration, and observation input. The policy is trained with the widely-used Proximal Policy Optimization algorithm (PPO) [35] based on surrogate loss, importance sampling, and AdamW optimizer. No experience replay or any buffer of such kind is used in our framework, and all data collected from simulation would be trained only once before being flushed away. The combination of PPO and MLP has already been proved to be feasible in learning robot locomotion policies by multiple research works [11,12,19,20]. Advanced network structures such as LSTM or Transformer can also fit into the proposed framework as policy representation, but more tuning and training effort will be required. The learning rate and its decay threshold have been optimized to accelerate early convergence and avoid over-fitting. Other hyper-parameters are maintained as default values provided by OpenAI Baselines [36]. The detailed hyper-parameters can be found in Tables 1 and A3.

Table 1. Neural Network Hyper Parameters

Parameters	Value
Network Type	MLP
Latent Layer	2
Latent Node Number	256
Activation	Tanh

3.1.4. Adaptive Normalization

Adaptive normalization is applied before the network (the blue block in Figure 2), where the normalization parameters are updated every epoch according to observation storage as shown in equations below and finally converge to the desired values associated

with the task so that the system designers do not have to re-tune the normalization module when changing task.

$$N_i = N_{i-1} + n_i \tag{11}$$

$$\mu_i = \mu_{i-1} + (\bar{s}_i - \mu_{i-1})\frac{n_i}{N_i} \tag{12}$$

$$\sigma_i^2 = \frac{1}{N_i}[N_{i-1}\sigma_{i-1}^2 + n_i v_i + N_{i-1}\sqrt{\frac{n_i}{N_i}(\mu_i - \mu_{i-1})}] \tag{13}$$

$$s' = \frac{s - \mu_i}{\sigma_i^2 + 10^{-8}}, s \in S_i \tag{14}$$

where s' is normalized observation vector. S_i is the ith batch retrieved from the relevant training episode. n_i is the number of samples in batch S_i. N_i is the total number of samples. $\bar{s}_i$ and v_i are the mean value and sample variance of S_i. μ_i and σ_i^2 are the mean value and variance for normalization after the ith update.

3.2. Curriculum Schema

To improve adaption to various velocity commands and robustness against noisy observations, the policies are trained with a two-stage parallel curriculum as shown in Figure 3.

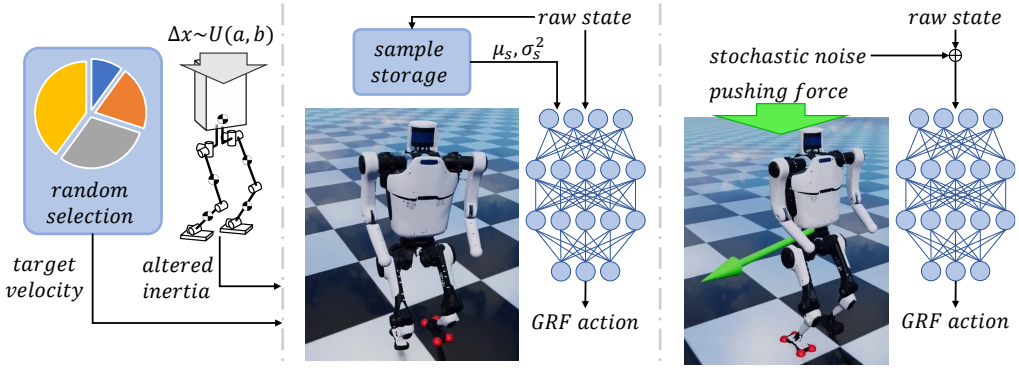

Figure 3. Curriculum schema. Section (**A**) shows parallel randomization curriculum including mass, inertia and task goal randomization. (**B**) shows first curriculum stage with ideal training environment. (**C**) describes second curriculum stage with sensor noise and random force perturbations. Check Section 3.2 for more details.

Before training, parallel environments are built for both accelerating data collection and fostering adaptability. Dynamics randomization is applied to every environment upon setup and remains constant during training. In this way, the policy is exposed to and forced to adapt to a wide range of possible dynamics in training, which reduces the reality gap resulting from dynamics mismatch. Random selection is called in each environment when resetting occurs. It randomly selects a target velocity under world frame from a discrete set. This method is found to be more effective than directly generating continuous random target velocity at every time step. Distribution of random velocity target can be expressed as:

$$P(v^* = i) = \frac{i}{2} + 0.1, i \in \{0.0, 0.2, 0.4, 0.6\} \tag{15}$$

In the first stage, the policy is trained in an ideal environment and adaptively updates normalization parameters μ_i and σ_i^2. The policy network receives raw observation without any external disturbance. Stage one ends when the average overall reward per trajectory reaches a threshold that guarantees velocity tracking without falling.

In the second stage, we add domain randomization to simulate noisy sensors and external force perturbation on the torso. In test runs of some stage one results, instant failure might happen after epoch time ends. Simulation time per epoch per environment is prolonged by half in this stage to prevent this problem. Normalization parameters become fixed in this stage, in case they are violated by randomization and new failure at the beginning of the new stage.

Detailed parameter configurations are listed in Table A4.

3.3. Heuristic Swinging Control

Despite its limitations, heuristics are simple and intuitive enough to be widely used. Inspired by capture point method [37], we adopt a simple linear policy for foot placement planning. The heuristic foot placement policy can be described as follows:

$$e_t = \delta e_t + k_1 \cdot v_t + k_2 \cdot v_t^* \tag{16}$$

where e is step displacement relative to hip expressed in world frame. δe is an offset vector. k_1 and k_2 are both coefficient vectors to be tuned manually.

Foot placement under world frame then has to be converted into joint command. The swing leg is not constrained by contact forces, thus position-based control method can handle it. To avoid collision and singularity, we use trigonometric interpolation in Cartesian space as described below, and transform it into joint space commands via inverse kinematics. We adopted a geometric inverse kinematics computation The planning process is also shown in Figure 4. Joint level position and velocity targets are sent to the low-level controller to be tracked.

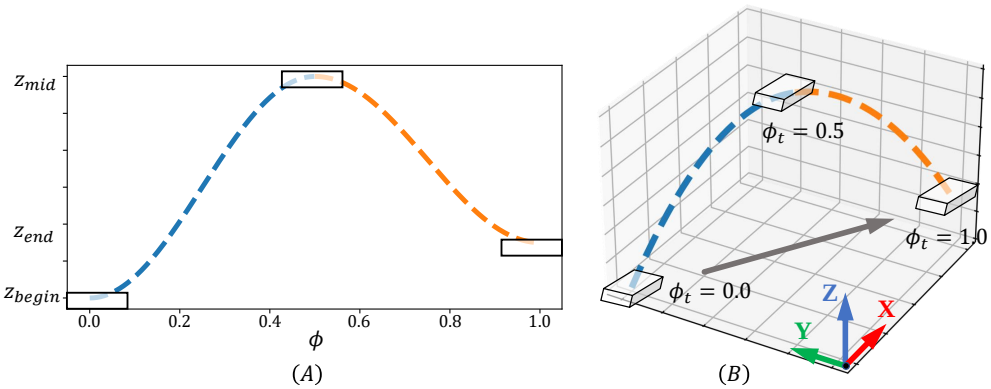

Figure 4. Planned Foot Swinging Trajectory. (**A**) is the plot of phase value and Z axis position of swing foot. White rectangle demonstrates the target position of foot under specific phase value ϕ. (**B**) shows the 3-D curve of the planned spatial trajectory.

$$p_w(\phi_t) = \begin{cases} A_1 \cos 2\pi\phi_t + B_1 & \text{for} \quad 0.0 \leq \phi_t < 0.5 \\ A_2 \cos 2\pi\phi_t + B_2 & \text{for} \quad 0.5 \leq \phi_t < 1.0 \end{cases} \tag{17}$$

$$A_1 = 0.5 \times (z_{begin} - z_{mid}) \tag{18}$$
$$A_2 = 0.5 \times (z_{mid} - z_{end}) \tag{19}$$
$$B_1 = 0.5 \times (z_{begin} + z_{mid}) \tag{20}$$
$$B_2 = 0.5 \times (z_{mid} + z_{end}) \tag{21}$$

where z_{begin}, z_{end}, and z_{mid} refer to the z-axis positions of the swing foot when the swing phase begins, ends, and is in the middle. All positions in the equations are expressed in world frame. ϕ refers to the phase value which increases from 0.0 to 1.0 in a gait period.

Phase switching is also done with heuristic conditions. When both the estimated contact force on the swing foot and the time after the last switch reaches their thresholds, or time reaches a larger threshold, a new switch is triggered.

3.4. Low-Level Controller

Different low-level controllers are placed after mentioned policies to convert their output signals to joint-level torque commands. For the stance leg, when performing stable locomotion, the acceleration of the robot is trivial enough to be ignored, so that the expected GRF can be approximately equal to the opposite of the end-effect force applied by the stance foot. For the swing leg, a simple proportional-derivative controller calculates feedback joint torque commands. The formulas are defined as follows:

$$\tau_{stance} = J_{stance}^{T}(-a_t) \quad (22)$$

$$\tau_{swing} = K_p \cdot (q_t^* - q_t) + K_d \cdot (\dot{q_t}^* - \dot{q_t}) \quad (23)$$

where J_{stance}^{T} is the force jacobian of stance leg, K_p and K_d are coefficient vectors to be tuned.

3.5. Regularized Sim-to-Real Transfer

In addition to randomization mentioned in Section 3.2, dynamics randomization also exists throughout all training stages. Randomized parameters including mass, inertia, and CoM position of links, are generated and fixed when parallel environments are built. See detailed distribution in Table A4.

To maintain the performance of the trained policy, we deploy it with heuristic-based feedback loops to provide regularization on real robots to assist in stabilization around the desired state. Inspired by VMC and Regularized Predictive Control methods, we condition compensation for low-level controllers with floating-base state errors based on heuristic control laws. The feedback formulations are as shown below.

$$\Delta F_z^* = K_z(p_z^* - p_z) \quad (24)$$

$$\Delta \tau_{abduction} = K_y(\theta_t^* - \theta_t) \quad (25)$$

where K_y and K_z are both coefficients. F_z is expected GRF along Z axis, and θ is the roll angle of the robot base.

4. Results

4.1. Simulation Training Results

Figure 5 compares the performance of our proposed method with other learning approaches. Shaded areas refer to the range of a group of curves, and the solid curve refers to the average of the group. All approaches share the same hyper-parameters. As for heuristics and low-level parameters, we roughly tuned them to a near-optimal state that would not cause frequent failure and proactive termination behaviors in the first 50 training epochs. We chose the prior-free end-to-end approach proposed by Hurst [14] as the canonical baseline, and an environment that enables domain randomization from the beginning to demonstrate the practicability of the curriculum algorithm.

The proposed hybrid method earns higher rewards on average than the other approaches in both stages. In the early stage of the proposed training method, the curves have large variance because, in each test run, the curriculum stage switches according to its reward level, and the newly introduced disturbance in the second stage causes more failures and reward drop.

In contrast, using end-to-end baseline approach the policy might occasionally perform as nicely as the converged state of the proposed method, but the overall performance is found to be unstable and bares a lower chance of success in practice.

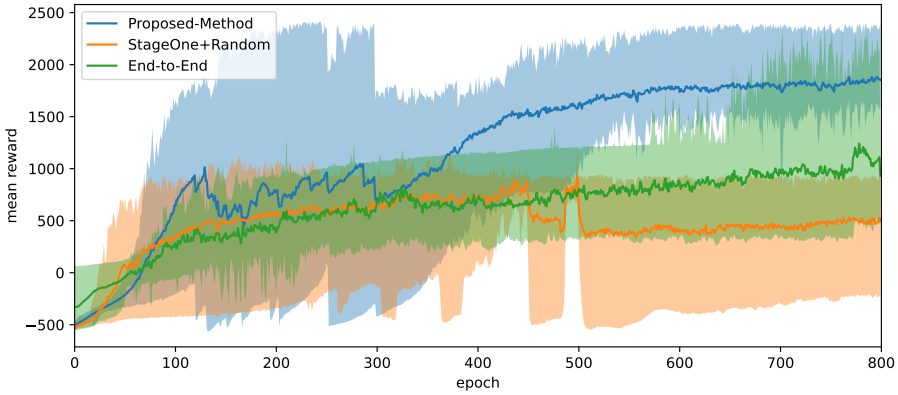

Figure 5. Training reward curves produced by different approaches. Vertical axis represents the average reward gained in one episode among all environments. Blue: proposed approach. Orange: hybrid method without curriculum stage one. Green: end-to-end aproach.

The results of the policy without the curriculum are shown in orange. Corresponding curves reveal relatively weak performance with a high risk of falling. The possible reason is that an over-challenging environment at the beginning leads to frequent failure, then lack of valid data to update policy and normalization parameters, and eventually early convergence before the policy learns to maintain balance.

Figure 6 shows the tracking performances of horizontal velocity. In the 40 s test episode, the velocity reference is composed of a series of step signals. Compared with end-to-end training results, the hybrid policy is proved able to track velocity reference up to 2 km/h with less tracking error. The mean squared error of the hybrid policy and end-to-end baseline policy are 2.232×10^{-3} and 1.282×10^{-2}. We also tested the same trained hybrid policy using robot models with 5 kg heavier torso (purple curve) and 5 kg lighter torso (red curve). Generally, although payload variations cause larger oscillations and slower responding performance, the proposed hybrid policy shows resistance to mass and inertia discrepancy and finishes tracking tasks without falling.

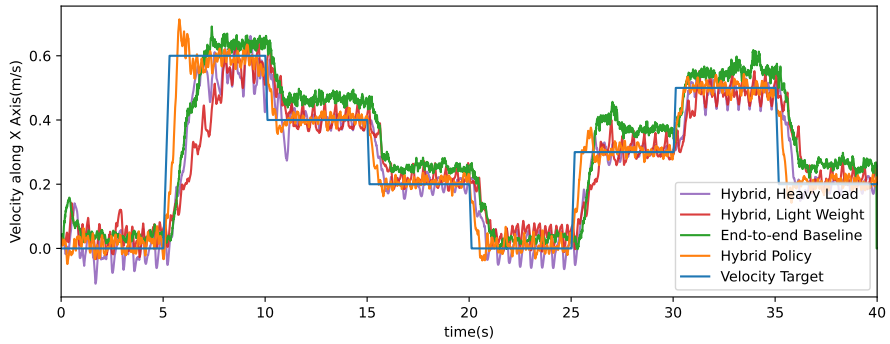

Figure 6. Step respond to step references. Data produced by a simulated robot controlled by different policies to track a predefined reference command signal composed of multiple step signals.

Figure 7 shows the phase plane plot of X velocity tracking error using different policies. All curves contain data of 2 s since velocity references shift. The curves of the hybrid policy (orange and red) display a spiral pattern attracted to a region centering (0, 0). The stability of the proposed hybrid method is displayed. The end-to-end baseline policy converges to an offset region with a relatively random pattern, which suggests larger tracking error and less stability.

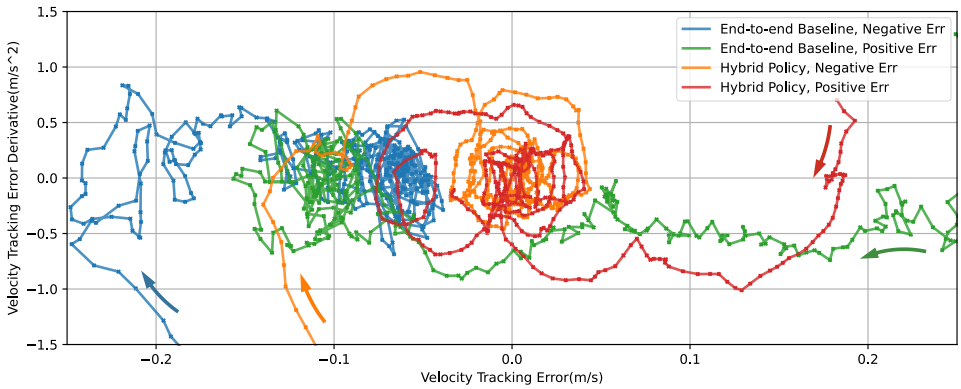

Figure 7. Phase plot of velocity X tracking error. Arrows in different colors indicate the direction of corresponding phase curve. Data obtained from slicing the simulation trajectory in Figure 6.

Figure 8 shows the comparison of contact points and joint-level states. Trained hybrid policy produces a regular gait with a frequency of 2.1 Hz, and steady contact indicates firm contact within the stance phase. Compared with the hybrid policy, the baseline policy's performance is less periodic for its spiky joint angle curves and fitful contacts during the left stance phase.

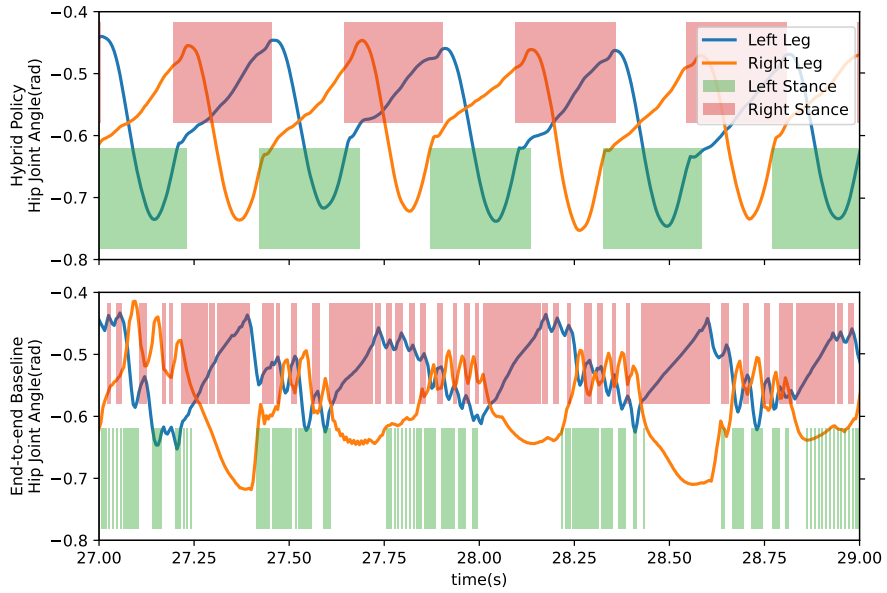

Figure 8. Plot of hip joint position and contact time series while walking. Colored area refers to the foot is having contact with ground at the time. Upper sub-plot's data is generated by proposed hybrid method, and lower sub-plot plots end-to-end baseline data. Both trajectories are recorded from the test episode described in Figure 6.

Figure 9 displays how the robot controlled by hybrid policy regains its balance when being externally pushed. A lateral pushing force of 14 N in world frame is randomly applied on a point on its torso for 0.5 s when the robot is walking 0.6 m/s. After the pose is dragged away from the upright state, two horizontal steps are then taken by the heuristic policy, meantime learned policy maintains the posture of the torso. It is proven that the trained hybrid locomotion controller has enough robustness to endure pushing force and retain balance. In contrast, end-to-end baseline policy cannot endure the same pushing force and falls with brittle actions. In consideration of safety for humans and robots, we did not transfer trained baseline policy onto real robots.

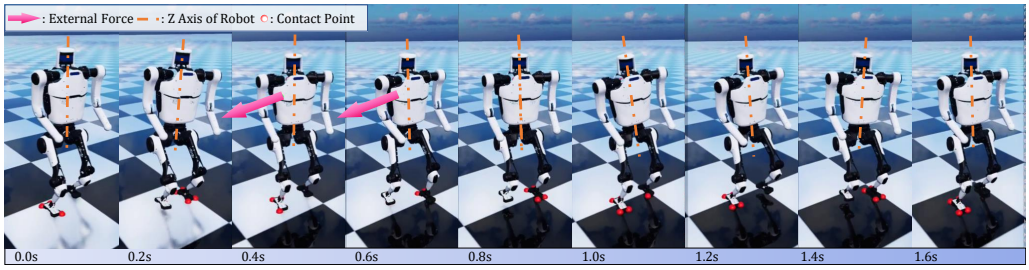

Figure 9. Snapshots of recovering from pushing force. Orange dashed lines refers to the Z axis of robot frame (Corresponding video of the snapshots can be found in Supplementary Materials).

4.2. Sim-to-Real Transfer

Figure 10 shows snapshots of a real Wukong-IV robot controlled with the trained hybrid policy. The robot can maintain balance and step in place. From Figure 11 it is clear that equipped with assistive feedback, the performance of the proposed hybrid policy on a real robot is more stable than ones without extra feedback loops. Unaided policies produce vigorous oscillations and lead to failure soon after.

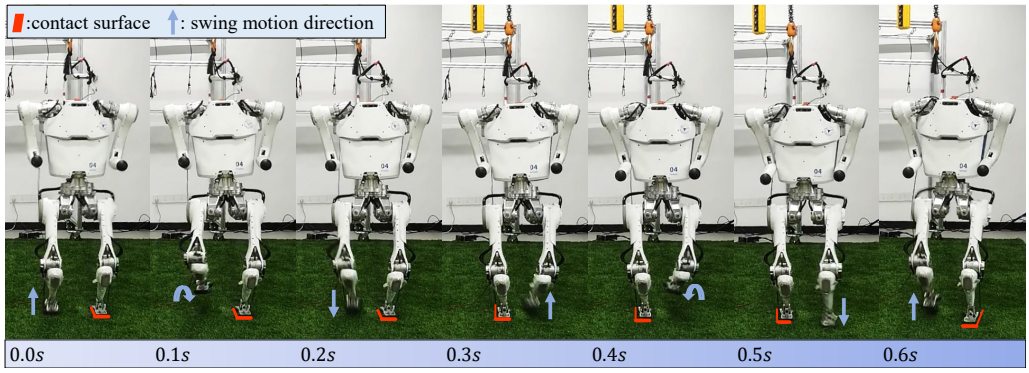

Figure 10. Snapshots of real Wukong-IV robot stepping in place. The trained policy is the same with policy tested in Section 4 (Corresponding video of the snapshots can be found in Supplementary Materials).

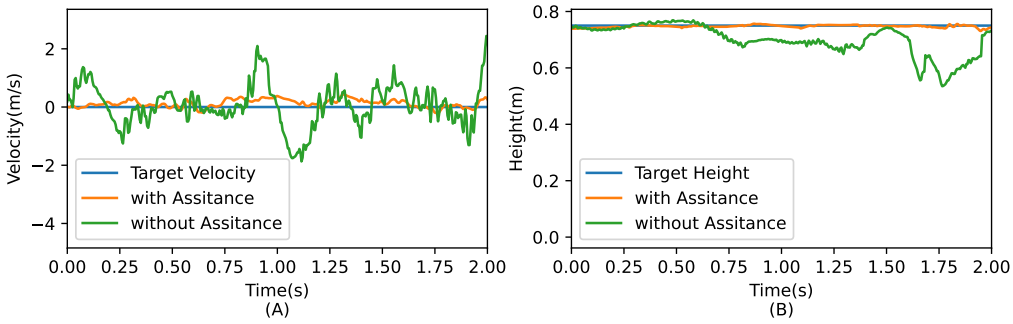

Figure 11. Plots of real robot with or without proposed assistive feedback loop. (**A**) Real robot velocity plot when stepping in place. (**B**) CoM height plot when stepping in place. Orange plots refers to the same experiment as shown in Figure 9. Green plots refers to hybrid policies without assistive feedback.

To prove that assistive feedback is not overwriting the policy, in other experiments, the policy network was disabled and assistive feedback was in charge of actuating the robot. As expected, despite multiple fine-tuning being carried out, the robot still instantly fell and crashed.

5. Discussion

We have presented a hybrid locomotion policy to produce nice results with achieving resilient bipedal walking. Learning and heuristics cooperate within the proposed structure

to finish the locomotion task. In the simulation, the policy is capable of tracking velocity command up to 0.6 m/s. Sim-to-real transfer to Wukong-IV is feasible with assistive feedback loops. The proposed hybrid framework yields higher reward and better performance than similar methods in terms of balance keeping and command tracking accuracy, and thus proved to be a stable and efficient approach to training a locomotion controller with adaption and robustness.

However, there are also limitations in the proposed methodology. Firstly, the performance of policies on a real robot is degraded compared with it in the simulation. The most possible reason is that the simulation and solution results of the software are too ideal, resulting in a large gap with reality. During deployment, policies cannot fully bridge the sim-to-real gap even with assistive feedback loops. In order to improve the performance of real robots, more advanced accurate identification work should be done, and detailed research on more oriented domain randomization combined with the dynamics of legged robots will be demanded in the future. Secondly, the proposed locomotion is still a blind one and just works on flat ground. For a humanoid robot, a highly nonlinear system, relying solely on proprioceptive control, is not conducive for the robot to face complex terrain. Future work is also recommended to adapt to more challenging terrain such as stairs and uneven surfaces with perceptual information.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/mi13101688/s1.

Author Contributions: Conceptualization, Coding RL System and Real Robot Control System, Methodology Development, Early Tuning and Writing Relative Parts are done by Z.W.; Fine Tuning on Both Simulation and Real Robots, Draft Editing are done by W.W.; Funding Acquisition is done by A.X. and Y.Z.; Supervision, Project Administration are done by Q.Z. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (Grant No. 2020YFB1313300), Key Special Project of Science & Technology Assisted Economy 2020 (Grant No. SQ2020YFF0405214), and Key Research Project of Zhejiang Lab (Grant No. 2021NB0AL03).

Data Availability Statement: Not applicable.

Acknowledgments: First, I would like to express my gratitude to Taiwen Yang, Jinpeng Niu and Peng Zhu. They helped throughout software debug and tuning with their precious experience from their locomotion research. Their aid frequently acted as sparkle of inspiration and kept me away from plenty of potential breaches. Second, I wish to give my thanks to Chao Li, Xueyin Zhang, Feng Li, Xiaobo Mo, Kai Xu and Zhiyong Tang from DeepRobotics. As the logistics staff for Wukong-IV, they kept the hardware in running with their effort even after serious damage, and thus created the possibility for experiments on real robots.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Parameter Details

Table A1. Wukong-IV Hardware Specifications.

Parameters	Value
Mass	45 kg
Height	1.4 m
Thigh Length	0.3 m
Shank Length	0.3 m
DoF per Leg	6
DoF per Arm	7
Power Source	Lithium Battery
Joint Actuator	Brushless Motor

Table A2. Reward design and parameters.

Reward Term	Coefficient
R_{track}	$\alpha_1 = 0.3$, $\beta_1 = 40$
R_{pose}	$\alpha_2 = 0.3$, $\beta_2 = 500$
R_{height}	$\alpha_3 = 0.3$, $\beta_3 = 400$
R_{gait}	$\alpha_4 = 0.1$, $\beta_4 = 10$
R_{slip}	$\alpha_5 = 0.3$, $\beta_5 = 100$

For R_{fall}, $K = -600$.

Table A3. Training Hyper Parameters.

Parameter	Value
Parallel Environment	150
Maximum time steps per epoch (Stage One)	2000
Learning Rate	1×10^{-4}
Discount Factor	0.996
Curriculum Reward Threshold	2300

Table A4. Randomization Parameters.

Variable	Distribution
Link Mass	$\Delta m \sim U(0.85m, 1.15m)$
Link CoM Position	$\Delta x \sim U(0.9x, 1.1x)$
Link Inertia	$\Delta I \sim U(0.8I, 1.2I)$
Sensor	$\Delta x \sim U(0.95x, 1.05x)$
External Force along X Axis	$f_x \sim U(-10\mathrm{N}, +10\mathrm{N})$
External Force along Y Axis	$f_y \sim U(-10\mathrm{N}, +10\mathrm{N})$

Random external force is activated for 0.5 s in every 2.5 s.

References

1. Fukuda, T. Cyborg and Bionic Systems: Signposting the Future. *Cyborg Bionic Syst.* **2020**, *2020*, 1310389. [CrossRef]
2. Raibert, M.; Tello, E. Legged robots that balance. *IEEE Expert* **1986**, *1*, 89. [CrossRef]
3. Pratt, J.; Torres, A.; Dilworth, P.; Pratt, G. Virtual actuator control. In Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Osaka, Japan, 8 November 1996; Volume 3, pp. 1219–1226.
4. Pratt, J.; Dilworth, P.; Pratt, G. Virtual model control of a bipedal walking robot. In Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA), Albuquerque, NM, USA, 25 April 1997; Volume 1, pp. 193–198.
5. Gubina, F.; Hemami, H.; McGhee, R.B. On the dynamic stability of biped locomotion. *IEEE Trans. Biomed. Eng.* **1974**, *BME-21*, 102–108. [CrossRef] [PubMed]
6. Henze, B.; Ott, C.; Roa, M.A. Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA, 14–18 September 2014; pp. 3253–3258.
7. Bledt, G. Regularized Predictive Control Framework for Robust Dynamic Legged Locomotion. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2020.
8. Haarnoja, T.; Ha, S.; Zhou, A.; Tan, J.; Tucker, G.; Levine, S. Learning to walk via deep reinforcement learning. *arXiv* **2018**, arXiv:1812.11103.
9. Tan, J.; Zhang, T.; Coumans, E.; Iscen, A.; Bai, Y.; Hafner, D.; Bohez, S.; Vanhoucke, V. Sim-to-real: Learning agile locomotion for quadruped robots. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018; pp. 1–13.
10. Hutter, M.; Gehring, C.; Jud, D.; Lauber, A.; Bellicoso, C.D.; Tsounis, V.; Hwangbo, J.; Bodie, K.; Fankhauser, P.; Bloesch, M.; et al. Anymal-a highly mobile and dynamic quadrupedal robot. In Proceedings of the 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 38–44.
11. Hwangbo, J.; Lee, J.; Dosovitskiy, A.; Bellicoso, D.; Tsounis, V.; Koltun, V.; Hutter, M. Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **2019**, *4*, eaau5872. [CrossRef] [PubMed]
12. Lee, J.; Hwangbo, J.; Hutter, M. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *arXiv* **2019**, arXiv:1901.07517.
13. Siekmann, J.; Valluri, S.; Dao, J.; Bermillo, L.; Duan, H.; Fern, A.; Hurst, J. Learning memory-based control for human-scale bipedal locomotion. In Proceedings of the Robotics: Science and Systems (RSS), Virtual Event, 12–17 July 2020; pp. 1–8.
14. Siekmann, J.; Godse, Y.; Fern, A.; Hurst, J. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 7309–7315.

15. Siekmann, J.; Green, K.; Warila, J.; Fern, A.; Hurst, J. Blind bipedal stair traversal via sim-to-real reinforcement learning. In Proceedings of the Robotics: Science and Systems (RSS), Virtual Event, 12–16 July 2021; pp. 1–9.
16. Yang, C.; Yuan, K.; Heng, S.; Komura, T.; Li, Z. Learning natural locomotion behaviors for humanoid robots using human bias. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2610–2617. [CrossRef]
17. Yang, C.; Yuan, K.; Zhu, Q.; Yu, W.; Li, Z. Multi-expert learning of adaptive legged locomotion. *Sci. Robot.* **2020**, *5*, eabb2174. [CrossRef] [PubMed]
18. Acero, F.; Yuan, K.; Li, Z. Learning perceptual locomotion on uneven terrains using sparse visual Observations. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8611–8618. [CrossRef]
19. Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; Hutter, M. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **2020**, *5*, eabc5986. [CrossRef]
20. Miki, T.; Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; Hutter, M. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **2022**, *7*, eabk2822. [CrossRef]
21. Kumar, A.; Fu, Z.; Pathak, D.; Malik, J. RMA: Rapid motor adaptation for legged robots. In Proceedings of the Robotics: Science and Systems (RSS), Virtual Event, 12–16 July 2021; pp. 1–11.
22. Iscen, A.; Caluwaerts, K.; Tan, J.; Zhang, T.; Coumans, E.; Sindhwani, V.; Vanhoucke, V. Policies modulating trajectory generators. In Proceedings of the 2nd Conference on Robot Learning (PMLR), Zürich, Switzerland, 29–31 October 2018; Volume 87, pp. 916–926.
23. Shao, Y.; Jin, Y.; Liu, X.; He, W.; Wang, H.; Yang, W. Learning free gait transition for quadruped robots Via phase-guided controller. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1230–1237. [CrossRef]
24. Green, K.; Godse, Y.; Dao, J.; Hatton, R.L.; Fern, A.; Hurst, J. Learning spring mass locomotion: Guiding policies with a reduced-order model. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3926–3932. [CrossRef]
25. Xie, Z.; Da, X.; van de Panne, M.; Babich, B.; Garg, A. Dynamics randomization revisited: A case study for quadrupedal locomotion. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 4955–4961.
26. Kumar, A.; Li, Z.; Zeng, J.; Pathak, D.; Sreenath, K.; Malik, J. Adapting rapid motor adaptation for bipedal robots. *arXiv* **2022**, arXiv:2205.15299.
27. Xie, Z.; Berseth, G.; Clary, P.; Hurst, J.; van de Panne, M. Feedback control for cassie with deep reinforcement Learning. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1241–1246.
28. Duan, H.; Dao, J.; Green, K.; Apgar, T.; Fern, A.; Hurst, J. Learning task space actions for bipedal locomotion. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 1276–1282.
29. Krishna, L.; Mishra, U.A.; Castillo, G.A.; Hereid, A.; Kolathaya, S. Learning linear policies for robust bipedal locomotion on terrains with varying slopes. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 5159–5164.
30. Krishna, L.; Castillo, G.A.; Mishra, U.A.; Hereid, A.; Kolathaya, S. Linear policies are sufficient to realize robust bipedal walking on challenging terrains. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2047–2054. [CrossRef]
31. Namiki, A.; Yokosawa, S. Origami Folding by Multifingered Hands with Motion Primitives. *Cyborg and Bionic Systems* **2021**, *2021*, 9851834. [CrossRef]
32. Hwangbo, J.; Lee, J.; Hutter, M. Per-contact iteration method for solving contact dynamics. *IEEE Robot. Autom. Lett.* **2018**, *3*, 895–902. [CrossRef]
33. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 8–14 December 2019; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 1–12.
34. Konda, V.; Tsitsiklis, J. Actor-critic algorithms. In Proceedings of the Advances in Neural Information Processing Systems 12 (NIPS 1999), Denver, CO, USA, 29 November–4 December 1999; Volume 12, pp. 1008–1014.
35. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
36. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
37. Pratt, J.; Carff, J.; Drakunov, S.; Goswami, A. Capture point: A step toward humanoid push recovery. In Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS), Genova, Italy, 4–6 December 2006; pp. 200–207.