# A Hierarchical Reinforcement Learning Approach for Adaptive Quadruped Locomotion of a Rat Robot

Zitao Zhang[1], Yuhong Huang[2], Zijian Zhao[1], Zhenshan Bing[2], Alois Knoll[2] and Kai Huang[1,*]

***Abstract*— Small robots encounter considerable difficulties in learning effective motions on complex terrains owing to their underactuated nature and nonlinear dynamics. In this paper, we present a novel approach for robot motion generation that implements reinforcement learning, based on simplified exploration of the robot's action and time slice conduction. Our approach controls the robot's actions using normalized signals and hierarchical mappings on mathematical space, which facilitates the learning process. We execute action in the timeslice to make efficient interaction with the environment. The effectiveness of our methodology is evaluated across a diverse range of simulated terrain scenarios, supplemented by physics validation. Our results show that our approach performs effective on complex terrains that are designed for small-sized robots.**

## I. INTRODUCTION

The rat robot is a kind of low-cost compact quadrupedal robot designed to closely resemble an actual mouse [1] as Fig. 1 shows, which is promising in bionomics research and disaster response. On the one hand, the drive structure of quadrupedal robots make them more competitive than wheeled robots in rugged terrains and complex environments. On the other hand, small robots [2] exhibit increased flexibility compared with large dog-sized quadrupedal robots. Therefore, they are well-suited for exploring constricted spaces due to the advantage of size, weight and cost.

Reinforcement learning in quadruped locomotion has gained attention as an alternative approach for conventional methods. In principle, conventional systems require manual effort to model both the robotics system and the target task, often leading to laborious tuning. As an alternative, reinforcement learning is a viable approach for generating robot motion for complex terrain through analysis of the status when the robot is walking. However, research of reinforcement learning for small-sized quadrupedal robots is limited. When it comes to rat robots, the limited resource allocation and streamlined structure could hardly support the effectiveness of baseline reinforcement learning methods. Therefore, this paper focuses on the motion generation of the rat robot in a source-limited scenario.

Implementation of reinforcement learning for small-sized quadrupedal robot locomotion is challenging due to varied action scales and continuous control signals. Firstly, various structure designs of robots endow them with complex nonlinear dynamics. Specifically, soft actuated legs of the rat robot, which possess an infinite number of digrees of freedom(DOF) theoretically [3], render kinematic modeling challenging and result in higher control errors. Secondly, a huge high-dimensional action space of multi-joint robots makes rewards sparse because direct control signals to the robot typically fall within a small time scale. For example, it takes dozens of executions of a 60Hz motor signal to allow the robot to make a tiny movement such as lifting its front claw. Excessive frequent instantaneous control would lead to increased environmental noise, which hinders the robot from obtaining effective feedback on its strategies from scene interactions. As a result, *how to simplify exploration of the robot's action is a fundamental problem for reinforcement learning of the rat robot*.

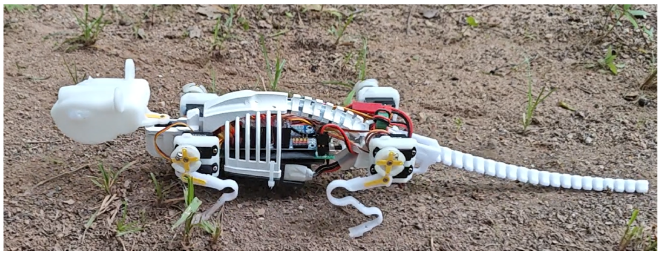


Fig. 1. The rat robot is designed with soft actuated components and supports more flexible robot motions.

To tackle the above challenges, we propose a novel approach for terrain-aware motion generation of the rat robot, based on reinforcement approach. This approach is aimed to simplify exploration of the robot's action and space. The rat robot's motions utilize normalized signals and hierarchical mappings in mathematical space, which significantly improve training speed while maintaining motion flexibility. By partitioning the motion cycle, we apply action signals to different time slices, enabling the robot to interact extensively with the environment at a macroscopic level. We successfully train effective motion policies in four challenging scenarios, and motion patterns are performed on the real rat robot to prove its feasibility. The Main contributions of this work are summarized as follows:

- We propose a novel hierarchical motion generation approach that controls the robot's actions based on normalized signals and hierarchical mappings on mathematical space to simplify the learning process.
- We normalize the execution time of action signals with time slices. Interactions with the environment are promoted to facilitate action learning.

*This work was supported by funding from the Pazhou Lab-oratory Grants with No. PZL2021KF0020. (Corresponding author: Kai Huang. Email: huangk36@mail.sysu.edu.cn)

[1] Authors from the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.

[2] Authors from the Department of Informatics, Technical University of Munich, Munich, German.

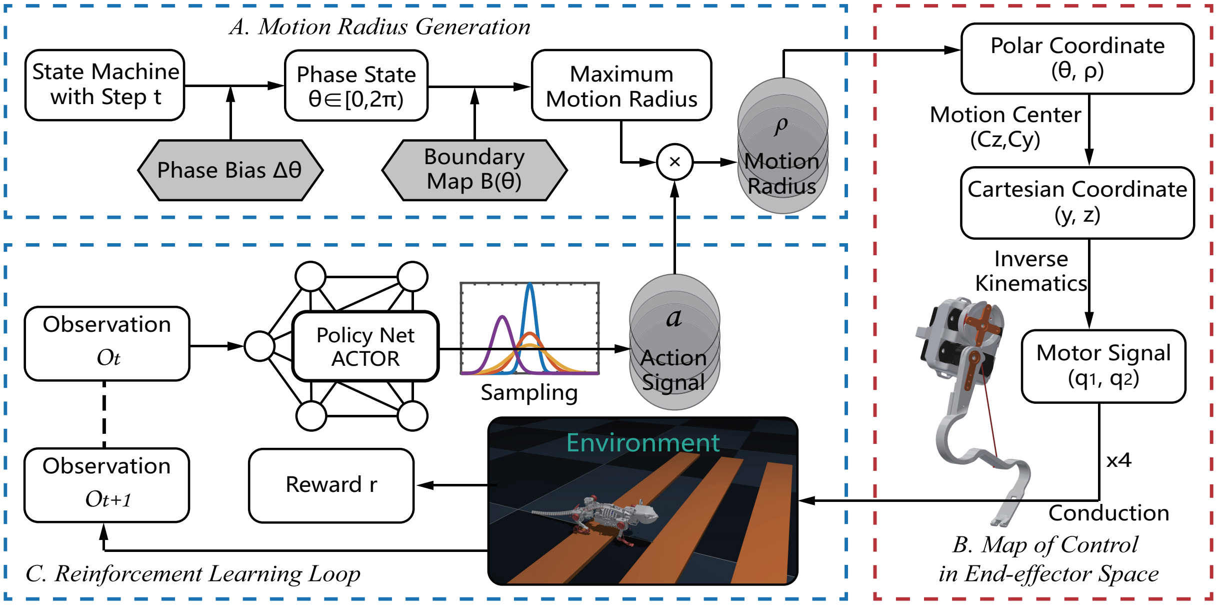


Fig. 2. Architecture for learning adaptive motion of the rat robot on complex terrains.

## II. Related Work

Deep reinforcement learning (DRL) has been extensively utilized for legged locomotion of quadrupedal robots in recent years [4]. DRL methods can be generally divided into model-free ones and model-based ones. The model-free approach [5] employs actions such as torques and joint angles to enable end-to-end control of the robot. However, learning efficient gaits from scratch is challenging in quadrupedal robots due to their complex nonlinear dynamics and sparse rewards. Model-based reinforcement learning methods [6] typically utilize a predetermined model of system dynamics to enhance training. ETG-RL [7] utilizes an evolutionary trajectory generator to optimize the shape of the output trajectory for the given task, thus providing diversified motion priors to guide policy learning. Lee et al. [8] employs neural networks to extract ground information, which is not directly available to the robot's sensors, mined from the robot's proprioceptive information.

Research of reinforcement learning in small-size robots like the rat robot is currently limited. A hierarchical reinforcement learning framework based on SAC is proposed in [9]. The two-level structure reduces the exploration space in learning process, which is worthy of reference for the rat robot. However, the implementation of reinforcement learning on small-sized quadruped robots requires a higher level of perception despite limited sensor information. Full-sized robots like ANYmal [10] utilize machine learning technology to build an altitude map of the surrounding environment by mining data from tactile sensors. However, small-sized robots face additional challenges due to equipment limitations. Maurice et al. [11] implemented linear policies to enable a small middle-sized quadruped robot (with a height of 200mm) to navigate uneven terrain, demonstrating the potential for further processing of IMU data in the locomotion system of small-sized robots.

## III. Architecture Overview

Fig. 2 gives an overview of our proposed approach for the rat robot. Our work mainly involves redesigning decision-making actions for rat robots in order to simplify the action space. In Part A, we build a state machine to implement control in a motion cyclicality. Each leg is given its phase added with different biases. Our boundary map in the end-effector space outputs the Maximum motion radius for each leg at phase $\theta$. The radius of motion is then obtained by multiplying with a normalized action signal. In Part B, We map the control signal from polar to Cartesian coordinate system on the basis of the preset motion center. Then the geometric inverse kinematics designed for the flexible leg maps it into servo control signals for each leg. Part C is a policy-based reinforcement learning control loop. We use a policy network to output action distribution to generate a 4-dimensional action signal at each step.

## IV. Methodology

### A. Motion generation based on a hierarchical mapping

In the end-effector coordinate space, we defined the motion representation in polar coordinates $(\theta, \rho)$. Gait patterns are encoded using the phase $\theta$, and the target landing position of the end-effector can be determined by specifying the radial distance $\rho$. The top-level motion command of the control system is defined as the ratio between the radial distance of the target landing position of the end-effector and the maximum range of motion.

Fig. 3 shows the structure of the rat robot, designed with soft actuated components that differ from dog-size quadrupedal robots. The support spine is constructed from flexible 3D printing materials, providing elastic degrees of freedom during motion. In this work, we do not enable active control of the spine, but focus on the legged locomotion. The bio-inspired legs of our robot are designed to closely mimic

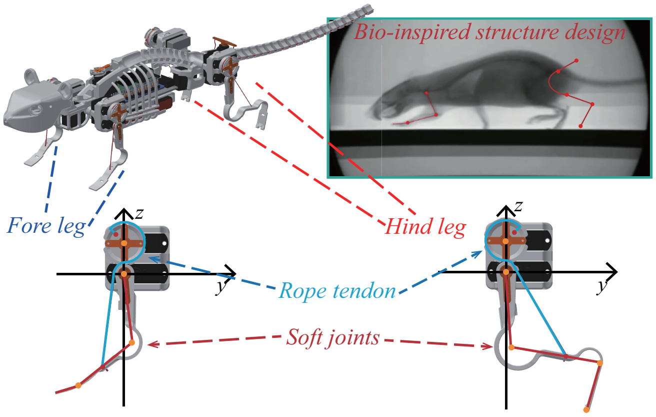


Fig. 3. Structure of the bio-inspired rat robot. The end-effector space is located in the Y-Z plane of its own Cartesian coordinate system.

the body structure and movement pattern of natural mice, while the flexible structures pose challenges to traditional quadrupedal motion control methods, particularly when reinforcement learning is implemented. Each leg of the rat robot is driven by tendon ropes with dual servos on the hip. Each servo has a rotation range of 180 degrees. As a result, the direct control signal in a single time step is denoted as

$$\mathbf{q} \in \mathbb{R}^8_{clip}, \mathbb{R}^8_{clip} := [-\frac{\pi}{2}, \frac{\pi}{2}] \tag{1}$$

which consists of 8 motor signals, each two working together for a leg, and the rat robot gait can be specified by a sequence $\Phi = \{\mathbf{q}\}$.

Most reinforcement learning methods take $\mathbf{q}$ as an action. However, in the context of rat robots, such an action space setup results in discontinuous actions in the end-effector space. This is because numerically similar motor values often correspond to very different locations when mapped to the end-effector space, posing a significant challenge for learning intelligent behaviors. In our work, motion planning is performed in the end-effector space and incorporates gait phase information through coordinate transformation.

As Fig. 4 shows, initially, in the end-effector workspace, we define a maximum reachable motion space $\mathbf{S}_r$ based on the original motion trajectory in our previous work [12], [13], which balanced the step length and foot clearance.

To integrate gait information into the motion signals, the end-effector space is converted from the Cartesian coordinates $(y, z)$ to the polar coordinates $(\rho, \theta)$, with the center of motion $(C_y, C_z)$ as the pole. The maximum range of motion $B(\theta)$ for any phase $\theta$ can be obtained by

$$B(\theta) \leftarrow \partial\mathbf{S}_r. \tag{2}$$

At any phase $\theta$, the location of the end point drop can be determined by specifying the polar radius $\rho$. Now we can define the motion signal of four legs as the ratio between the polar radius of the end point's landing position and the maximum range of motion with format

$$\mathbf{a}_t \in \mathbb{R}^4_{clip}, \mathbb{R}^4_{clip} := [0, 1] \tag{3}$$

where the four components represent four legs for the robot.

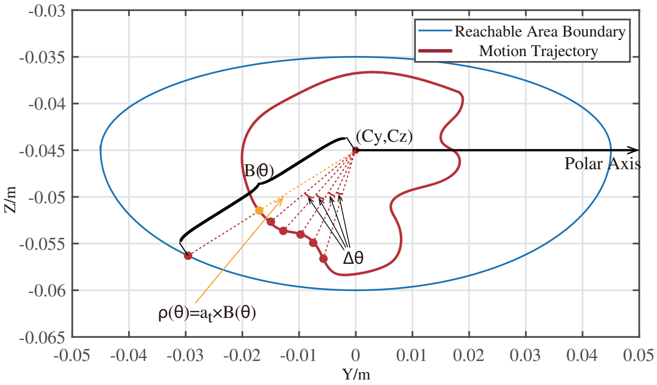


Fig. 4. Motion Signal of Four Legs in Scene Board. $\Delta\theta$ is related to factors including control frequency, response time, and physical constraints.

By adjusting the motion signal $a_t \in [0, 1]$, the corresponding motion position at phase $\theta$ can be controlled by

$$\rho(\theta) = a_t B(\theta), \ (\rho, \theta) \in \mathbf{S}_r \tag{4}$$

Here, phase state $\theta$ is associated with time. In a complete motion cycle $T$, the phase state of leg motion is

$$\theta_i(t) = 2\pi * (t/T + \ \phi_i), t \in [0, T) \tag{5}$$

where $i$ from 1 to 4 represent the left front leg, right front leg, left hind leg, and right hind leg, respectively, and $\phi_i$ is the phase bias of each leg. In the Trotting gait, which is applicable to low and medium speed movements, they are$\phi_1 = 0, \phi_2 = 0.5, \phi_3 = 0, \phi_4 = 0.5$.

$a_t$ can be obtained from the motion generation network:

$$\mathbf{a}_t \leftarrow A(O, \theta_P) \tag{6}$$

where $A(O, \theta_P)$ is the motion generation network and $O$ is the observation of the environment, which can be a high-dimensional vector. By controlling the size of $\mathbf{a}_t$, we can generate different motion trajectories within the reachable region. When $a_t = 1.0, \forall\theta \in [0, 2\pi)$, the motion trajectory degenerates into a pre-defined smooth closed curve.

Back to the Cartesian coordinate system$(y, z)$ in end-effector space, each motion track point can be obtained by coordinate transformation:

$$\begin{aligned} y &= C_y + \rho(\theta)\cos\theta, \\ z &= C_z + \rho(\theta)\sin\theta. \end{aligned} \tag{7}$$

where $C_y$ and $C_z$ are the centor of motion. In equation 7, $(y,\ z)$ represents a point on the trajectory of a single limb.

Subsequently, the coordinates of the end-effector space point are mapped back to the rotation signals of the two motors at the upper and lower hip joints through mathematical inverse kinematic calculations:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = M_{inv} \begin{bmatrix} y \\ z \end{bmatrix}. \tag{8}$$

Here, $q_1$ and $q_2$ represent the rotation signals of the two motors, while $M_{inv}$ is the inverse kinematics matrix obtained through a mathematical bijection. This mathematical model allows for design motions in the end-effector space of the robot quadruped, which has a direct physical meaning for the motion of the robot.

After the aforementioned steps, the motion of the rat robot can be determined by a compact 4D vector. Information of the motion range and gait phase improves the training efficiency, while the action signals defined through hierarchical analytical mapping provide freedom for exploration.

### *B. RL loop with timeslice normalization*

In order to extract effective interaction information for learning between the robot and the environment, we span the action of servo signals to multiple timesteps within a timeslice. During the training, an RL step comprises a segment of end-effector trajectory that spans multiple timesteps. In any arbitrary gait, a complete motion cycle of length $T$ can be divided into multiple trajectory segments, and the number of timesteps in each segment $N_{step}$ is calculated by

$$N_{step} = \left(f T_0 N_{div}\right)^{-1} \tag{9}$$

where $f$ denotes the frequency of periodic motion, $T_0$ stands for the minimum control interval (set to $0.002s$ in our simulation with MuJoCo), and $N_{div}$ refers to the number of period divisions. For instance, with $N_{div} = 8$, a single time slice comprises $1/8$ of the motion trajectory of $T$.

After conduction of a time slice, the observation information state $s_t$ can be extracted from the environment

$$s_t = \left(a_t, i_C, r, \vec{V_{vel}}, \vec{V_{gyro}}\right) \tag{10}$$

where $i_C$ is the index of the phase segment of the time slice in the entire motion cycle $T$, $r$ is the reward value, $\vec{V_{vel}} \in \mathbb{R}^3$ is the filtered velocity, and $\vec{V_{gyro}} \in \mathbb{R}^3$ is the filtered angular velocity. Besides, reward of the interaction within the time slice is obtained as

$$r = K_{vel} \vec{V_{gyro}} \cdot \vec{u_{dir}} \tag{11}$$

where $K_{vel}$is the weighting factor and $\vec{u_{dir}}$ is a unit vector for direction. Our work of reward shaping did not focus on specific body structure or character of environments. Above design of the reward applies for other tasks of quadruped robots as well while remaining room for improvement in specific scenarios.

We parameterize the Control policy as a Gaussian distribution with a diagonal covariance matrix by

$$\pi_\theta\left(a \mid s_t\right) = \mathcal{N}\left(a \mid \mu_\theta\left(s_t\right), \sigma_\theta\right). \tag{12}$$

A deep neural network (DNN) structure is employed to generate the mean $\mu_\theta\left(s_t\right)$ of the distribution, with inputs $s_t$. Additionally, the standard deviation $\sigma_\theta$ is generated by a separate and independent network layer, which facilitates exploration during training. Updating of the policy can be through the common-used policy-based RL algorithms. The complete algorithm is outlined in Algorithm 1.

**Algorithm 1** RL algorithm with the rat robot

```
Require: $S_0$, State with initial environment
    $\beta$, experience replay buffer
    $N$, the number of steps during a time slice
    $\epsilon \sim \mathcal{N}(0,1)$, gaussian noise for exploration
Initiate the policy
while Not Converged do
    Get action signal $a_t \leftarrow \pi_\theta(s_t) + \epsilon$
    Get motion radius $\rho(\theta) = a_t \times B(\theta)$
    Get end-effector space tracking point $(y,z) \leftarrow \rho_t(\theta)$
    Transfer to motor signal $(q_1,q_2) \leftarrow (y,z)$
    for $i = 1$ to $N$ do                ▷ Time slice conduction
        Conduct servo control Signal $(q_1,q_2)$
    end for
    Get observation $s_{t+1}$ and reward $r$
    Append the transition $(s_t, a_t, r, s_{t+1})$ into $\beta$
    if $\beta$ is full then
        Update policy $\pi_\theta$
        Reset $\beta$
    end if
end while
```

## V. EXPERIMENTS

### *A. Experimental setup*

In order to validate the proposed approach, we construct four different terrain scenarios for the robot to train adaptive motions, as Fig. 5 shows. The terrain sizes are designed to simulate real-world environments that are encountered by small robots. After simulation, physics validation of the learning result in the real world is carried out as well.

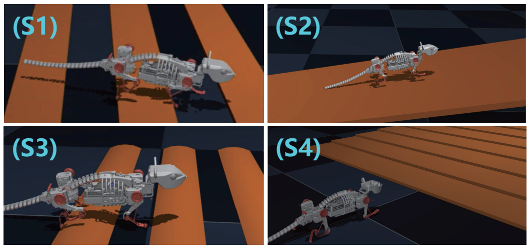
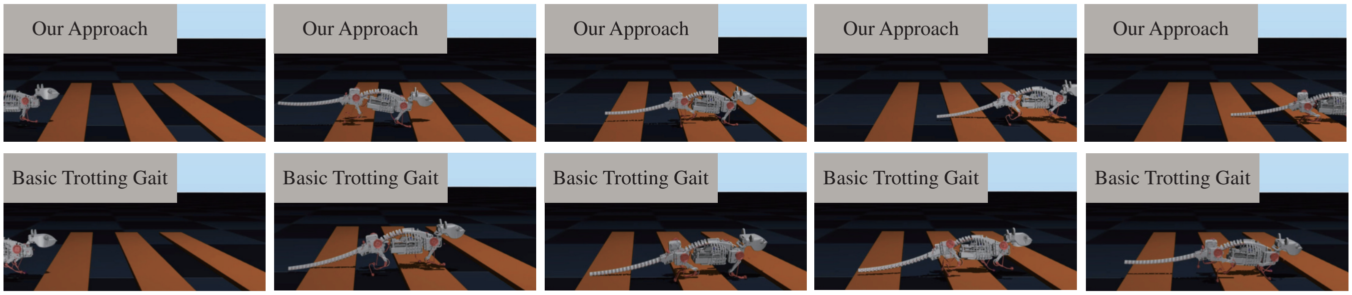


Fig. 5. Four test scenarios for the rat robot. (S1)Planks: Gallop over planks with wide gap(width=10 cm).(S2)Uphill: a 10-degree slope. (S3)Logs: Upon a pile of logs. (S4)Stairs: Micro stairs between two platforms.

The rat robot we use has the size of 40 cm × 25 cm and features 8 DOF for control. Each leg of the robot has a step length of 4 cm and a foot clearance of 1.5 cm. On the basis of the physical platform we have utilized before, we build a digital twin in the MuJoCo platform [14] for simulation. The simulations run on a Ubuntu 18.04 system, with a server equipped with dual Intel Xeon Gold 6234 Processors and eight V100 GPUs. Each training session requires less than 2GB of memory. The base process of reinforcement learning is implemented with the Stable-baselines3 platform [15].

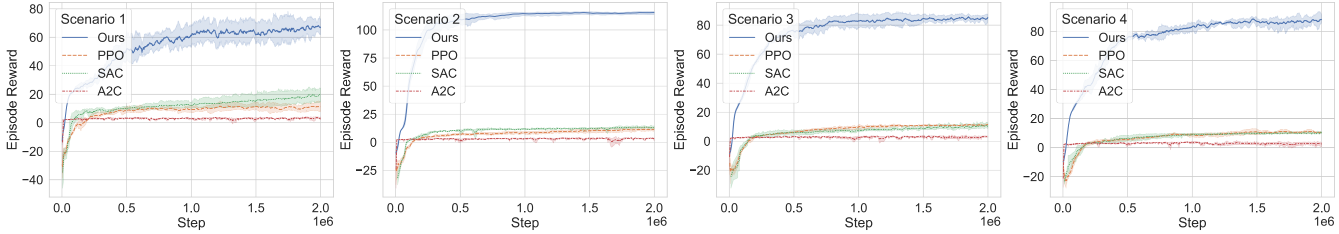
### *B. Motion analysis*

The rat robot successfully completed all four tasks with motion generated by the trained polices. We take Scenario



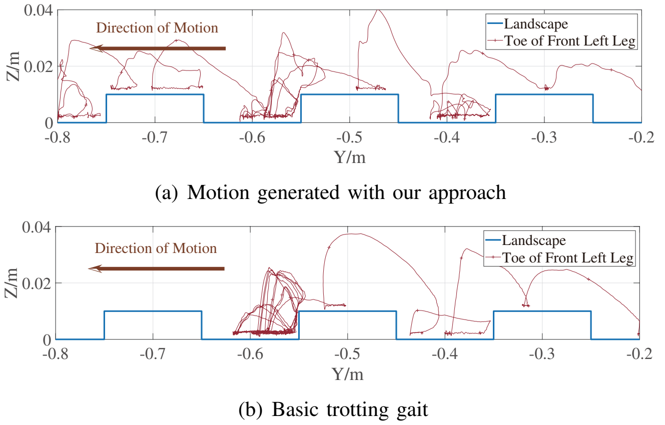
(a)t=0s (b)t=3s (c)t=6s (d)t=9s (e)t=12s

Fig. 6. Montage in ”Planks”. The first line shows the movement of the rat robot controlled with our reinforced gait. The second line shows the movement of the rat robot controlled with a simple trotting gait.



(a) Planks (b) Uphill (c) Logs (d) Stairs

Fig. 7. Training curves on 4 simulation tasks. Each experiment is was run four times. The solid lines represent the average score and the shaded areas represent a standard deviation.



(a) Motion generated with our approach

(b) Basic trotting gait

Fig. 8. End point trajectory of the rat robot’s leg in (S1)Planks.

Planks as an example for analysis and Fig. 6 presents a montage of the rat robot walking forward in it. The first row displays the motion generated by our approach, while the second row shows the basic trotting gait. At the beginning, two methods exhibit similar performance. The fixed trot gait causes the robot to get trapped at the second plank at 6s, failing to overcome the obstacle. That is primarily because the hind legs are unable to surmount the obstacle. In contrast, our approach enables the robot to modify its motion signal and attempt passage. The rat robot touches the middle plank at 3 s and it takes less than 6 s to pass. Experiments in other scenarios perform consistent results.

The end point trajectory of the rat robot is depicted in Fig. 8. The robot successfully overcame the obstacle after adaptive adjustment of our method, but got stuck at the second plank with basic trotting gait. It can be observed that the agent encounters a period of obstruction at the second plank. Following motion adjustments, it generates an adaptive gait thus surmounting the obstacle. In contrast, the robot with a basic trotting gait is unable to overcome the diverse terrain with a fixed motion pattern due to the absence of environmental state feedback. Therefore, there exist overlapping trajectories at the second plank and the robot fails to move on.

### C. Training performance

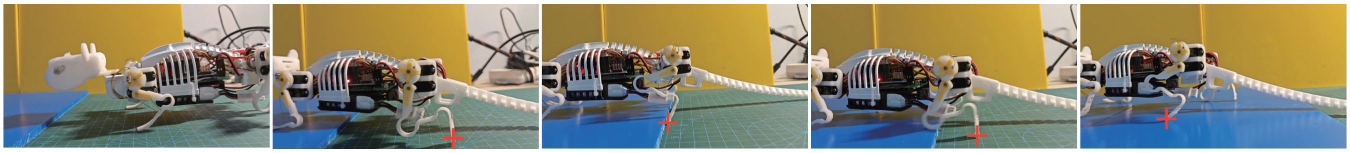
Training curves shown in Fig. 7 illustrate that the episode rewards of our method get converged in all tested scenarios and performance of converged reward is shown in Table I. In order to evaluate the effectiveness of our proposed method, we compare it with three policy-based benchmarks. The proposed method significantly outperforms the benchmark algorithms with higher converged rewards and higher convergence speed. Meanwhile, most benchmark algorithms fail to converge within 2M steps.

TABLE I

PERFORMANCE IN FIVE SCENARIOS.

| Scenario name | Ours | PPO | SAC | A2C |
|---|---|---|---|---|
| Plane | **121.3** | 11.4 | 6.0 | 3.7 |
| (S1)Planks | **66.6** | 11.2 | 19.4 | 3.8 |
| (S2)Uphill | **115.5** | 11.5 | 13.2 | 2.7 |
| (S3)Logs | **85.4** | 11.2 | 10.8 | 2.9 |
| (S4)Stairs | **88.0** | 10.6 | 10.1 | 2.3 |

Note that, aforementioned similar work of quadrupedal locomotion can not be directly compared because of different body construction. For discussion, some methods like the ETG-RL [7] endow robots with remarkable terrain adaptability, while often necessitating timestep-level iterations in the order of tens of millions for convergence in deployment, or extensive pre-training efforts. In contrast, our



(a)t=0s (b)t=5s (c)t=10s (d)t=15s (e)t=20s

Fig. 9. Montage of the robot running in the real world. The obstacle is a 5mm height plank.

approach can effectively obtain gait patterns within hundreds of thousands of step iterations by initiating from scratch.

The experimental results demonstrate that the neural network strategy is capable of learning effective motion through reinforcement learning (RL), facilitated by our model-based motion generation, specifically tailored for small robots. Our training has not yield optimal results due to the absence of complex reward shaping and network design. However, the concept of motion generation is not limited to specific tasks and can be equally applicable to other small robots as well.

### D. Physics validation

Based on the policy learning in simulation as aforementioned, we generate motion signals sequences with the trained policies. In the provided video, we record the performance of the rat robot in uneven terrain and Fig. 9 gives a montage of the robot running at a Plank obstacle. As Table II shows, we test the performance of the robot with the trained motion and the basic trotting gait at different obstacle heights. As height increases, the passage rate of the unadjusted trotting gait decreases dramatically, while that of our trained motion remains high. This is a physical validation of the proposed method which prove that the generated motion is feasible in the real robot. We expect to take further research in conducting online learning where the training loop could also be on the real robot.

TABLE II

PASSAGE RATES AT DIFFERENT OBSTACLE HEIGHTS

| Height(mm) | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Ours trained motion* | 100% | 100% | 70% | 70% | 60% |
| Basic trotting gait* | 100% | 60% | 10% | 0% | 0% |

* Each experiment is repeated 10 times.

## VI. CONCLUSIONS

This paper presents a hierarchical approach for autonomously generating robot motion for complex terrains. Our proposed approach combines the motion generation of our rat robot with model-based methods and a hierarchical mapping to improve learning efficiency. Through timeslice execution of actions in the RL loop, efficient interaction with the environment is guaranteed to facilitate action learning. We conducted a series of case studies, and the results demonstrate the effectiveness and training efficiency of our method. The proposed approach was successful in all challenging scenarios tested, and it can be applied to other robotic applications as well.

## REFERENCES

[1] P. Lucas, S. Oota, J. Conradt, and A. Knoll, "Development of the neurorobotic mouse," in *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE, 2019, pp. 299–304.

[2] S. M. Neuman, B. Plancher, B. P. Duisterhof, S. Krishnan, C. Banbury, M. Mazumder, S. Prakash, J. Jabbour, A. Faust, G. C. de Croon, and V. J. Reddi, "Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 296–299.

[3] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015. [Online]. Available: https://doi.org/10.1038/nature14543

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

[5] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in *2012 American Control Conference (ACC)*, 2012, pp. 2177–2182.

[6] X. Li, W. Shang, and S. Cong, "Model-based reinforcement learning for robot control," in *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2020, pp. 300–305.

[7] H. Shi, B. Zhou, H. Zeng, F. Wang, Y. Dong, J. Li, K. Wang, H. Tian, and M. Q.-H. Meng, "Reinforcement Learning With Evolutionary Trajectory Generator: A General Approach for Quadrupedal Locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3085–3092, Apr. 2022, conference Name: IEEE Robotics and Automation Letters.

[8] 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, Oct. 2020, publisher: American Association for the Advancement of Science.

[9] Y. Wang, W. Jia, and Y. Sun, "A hierarchical reinforcement learning framework based on soft actor-critic for quadruped gait generation," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2022, pp. 1970–1975.

[10] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[11] M. Rahme, I. Abraham, M. L. Elwin, and T. D. Murphey, "Linear policies are sufficient to enable low-cost quadrupedal robots to traverse rough terrain," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8469–8476.

[12] Y. Huang, Z. Bing, F. Walter, A. Rohregger, Z. Zhang, K. Huang, F. O. Morin, and A. Knoll, "Enhanced quadruped locomotion of a rat robot based on the lateral flexion of a soft actuated spine," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2622–2627.

[13] Y. Huang, Z. Bing, Z. Zhang, K. Huang, F. O. Morin, and A. Knoll, "Smooth stride length change of rat robot with a compliant actuated spine based on cpg controller," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[14] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[15] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 1, jan 2021.