

Research Article

Hierarchical reinforcement learning for enhancing stability and adaptability of hexapod robots in complex terrains

Shichang Huang^{a,1}, Zhihan Xiao^{a,1}, Minhua Zheng^{a,b,*}, Wen Shi^c^a School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China^b The Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology, Ministry of Education, Beijing Jiaotong University, Beijing, 100044 China^c Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China

ARTICLE INFO

Article history:

Received 9 December 2024

Revised 2 March 2025

Accepted 3 March 2025

Available online 27 March 2025

Keywords:

Hexapod robot

Central pattern generation

Reinforcement learning

Complex terrains

ABSTRACT

In the field of hexapod robot control, the application of central pattern generators (CPG) and deep reinforcement learning (DRL) is becoming increasingly common. Compared to traditional control methods that rely on dynamic models, both the CPG and the end-to-end DRL approaches significantly simplify the complexity of designing control models. However, relying solely on DRL for control also has its drawbacks, such as slow convergence speed and low exploration efficiency. Moreover, although the CPG can produce rhythmic gaits, its control strategy is relatively singular, limiting the robot's ability to adapt to complex terrains. To overcome these limitations, this study proposes a three-layer DRL control architecture. The high-level reinforcement learning controller is responsible for learning the parameters of the middle-level CPG and the low-level mapping functions, while the middle and low level controllers coordinate the joint movements within and between legs. By integrating the learning capabilities of DRL with the gait generation characteristics of CPG, this method significantly enhances the stability and adaptability of hexapod robots in complex terrains. Experimental results show that, compared to pure DRL approaches, this method significantly improves learning efficiency and control performance, when dealing with complex terrains, it considerably enhances the robot's stability and adaptability compared to pure CPG control.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Legged robots, due to their multiple degrees of freedom and discrete footholds, can flexibly adapt to uneven terrains [1]. Compared to other robots, hexapod robots are widely used in terrain inspection, disaster relief, and natural disaster detection because of their higher stability and flexibility [2,3]. However, the redundant degrees of freedom in hexapod robots impose high demands on control in challenging environments [4]. Enhancing the locomotion capabilities of hexapod robots in complex environments has become a major research focus [5].

Current research methods for legged robots include model-based motion optimization [6,7], biomimetic approaches [8,9], and data-driven methods [7,10]. Model-based motion optimization utilizes precise descriptions of the robot's dynamic model to optimize its motion trajectories and control strategies for specific tasks and environments. By establishing a dynamic model,

the movement of each joint can be accurately calculated, which allows for efficient handling of complex terrains [6]. This method benefits from theoretical interpretability and efficiency but faces challenges due to high model complexity, large computational requirements, and limited adaptability to environmental changes.

In contrast, biomimetic approaches, represented by the central pattern generator (CPG), can produce stable rhythmic movements without external signal feedback. Currently, it is common to establish CPG models using coupled oscillators, with representative models including the Matsuoka oscillator [11] for neural oscillations and the Kuramoto [12] and Hopf oscillators [13] for nonlinear oscillations. Initially applied to the locomotion pattern generation of fish and amphibious robots [14], CPG was later used for legged robot motion control, capable of generating low-dimensional control signals for natural biomimetic gaits [15]. However, CPG can only generate specific movement gaits and is unable to control robot motion in real time based on environmental changes, which limits its further application due to poor adaptability. To compensate for this flaw, researchers have used structural design [16] and sensory feedback [17] to achieve dynamic motion on unstructured terrains. Moreover, a multi-layer CPG control model based on the semi-central CPG model

* Corresponding author at: School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China.

E-mail address: mhzheng@bjtu.edu.cn (M. Zheng).

¹ The two authors contribute equally to this work.

was proposed [18] for rhythmic signal generation, motion pattern determination, and motion trajectory generation to achieve adaptive walking control on sloped terrains. Furthermore, a transitional gait based on CPG bottom-layer feedback was planned using the robot's supporting leg ankle joint angle in relation to body pitch, and a slope gait based on CPG mid-layer feedback was planned according to the relationship between the supporting knee joint angle and the hexapod's pitch angles [19].

In the field of robotic control, deep reinforcement learning (DRL), as a typical data-driven approach, has demonstrated significant adaptability in effectively dealing with various terrains [20, 21] and external disturbances [22]. By applying sensor data from the robot itself to train the DRL model and using the model's output to directly control the joints, effective control has been achieved on both flat and rugged terrains [23,24]. However, learning joint positions directly from external sensor signals faces the challenge of high dimensionality in the state and action spaces, making learning and control difficult. This also requires careful parameter tuning, reward function design, and extensive data collection. In addition, although recent studies have proposed methods to directly learn torque [25] or the desired task space positions [26,27], the problem is that the action signals predicted and output by the model, such as motor torque or joint angles, tend to lack smoothness. To overcome these difficulties, a method (CPG-RL) that uses DRL to optimize CPG parameters has been proposed. This method leverages bio-inspired algorithms to optimize action generation, making the robot's movements smoother and more natural, while improving control accuracy and adaptability in complex environments. Currently, the CPG-RL method has been widely applied in legged robot motion control [28–33]. In the field of bipedal robots, researchers have adjusted the gait generation network to simulate the complex dynamics of human walking, enabling the robot to maintain balance and stability under varying ground conditions [28]. Another study proposed a reinforcement learning method based on CPG, which optimizes control strategies to enable bipedal robots to achieve adaptive gaits on complex terrains, improving both stability and flexibility [29]. In quadruped robots, the CPG-RL method has significantly enhanced the robot's adaptability and mobility on irregular terrains by finely adjusting the timing and force of each foot's contact [30]. For hexapod robots, inverse kinematics is used to convert foot-end positions into desired joint positions, achieving effective motion on soft sand [32] and rough terrain [33]. Additionally, by training a CPG network composed of six Hopf oscillators through DRL, motion control on regular terrains has been realized.

Despite the effectiveness of the CPG-RL method, it still faces challenges such as increased training difficulty due to the complexity of CPG networks and the large volume of sensor data complicating the state and action spaces. Furthermore, some tests have only been conducted on regular terrains, with insufficient training of robots on complex mixed terrains. This paper focuses on enhancing the motion capability of hexapod robots in complex terrains by generating CPG networks and mapping function-related parameters through reinforcement learning. In this paper, CPG-RL refers to a generalized motion control architecture that combines CPG with RL, whereas CPG-RL is our method for controlling hexapod robots. The main contributions of this paper are as follows:

- (1) The “decision-coordination-execution” three-level coupled architecture proposed in this paper enables dynamic parameter adaptation, flexible control strategies, and a significant reduction in network complexity through an efficient hierarchical structure. It addresses the parameter redundancy issue in traditional methods while improving control stability and adaptability.

- (2) This paper introduces a dual-oscillator time-division multiplexing strategy, breaking the traditional fully connected CPG network paradigm for hexapod robots. The mid-level controller adjusts the phase difference θ between the two Hopf oscillators to generate cross-leg coordinated rhythms, reducing the network parameters by 67% while maintaining rhythm stability and avoiding instability from multi-oscillator signal conflicts. Additionally, a parameter-efficient mapping function is proposed for the lower-level controller to reduce the model's training burden.
- (3) The effectiveness of the control method is validated through simulation experiments, demonstrating its superiority in terms of learning performance, stability, and adaptability. Comparison experiments on learning performance and motion control further support this, and physical relocation experiments show excellent motion performance.

2. Method

2.1. Reinforcement learning

In reinforcement learning, at each time step t , the agent selects an action a_t based on the current state s_t , transitions to the next state s_{t+1} through the state transition probability, and receives an immediate reward R_t . From these interactions, the expected return G_t is calculated, as shown in Eq. (1). Ultimately, the agent learns the optimal policy through continuous interaction with the environment to maximize the cumulative future reward. The optimization process in reinforcement learning forms a Markov decision process (MDP). In this study, the motion process of the hexapod robot can be modeled as an MDP, represented as a tuple $\{S, A, T, R, \gamma\}$, where S is the set of all possible states, A is the set of actions, $T(s_{t+1}|s_t, a_t)$ is the state transition probability function, $R(s_t, a_t)$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor used to weigh the importance of future rewards.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

where R_{t+k+1} represents the reward obtained at time step $t + k + 1$. The goal of reinforcement learning is to find a policy $\pi(s)$ for MDP that produces a probability distribution over possible actions, thereby maximizing the expected long-term return. This is achieved by interacting with the environment and learning estimators directly from these interactions.

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim P} G_t \quad (2)$$

where π^* represents the optimal motion policy of the robot in the environment.

2.2. CPG network

The CPG network consists of multiple oscillators arranged in a specific network topology. The selection of oscillators is closely related to the complexity of the CPG network. Compared to neural oscillators, nonlinear oscillators have fewer parameters and simpler dynamic characteristics, which are beneficial for robot control. Additionally, the Hopf oscillator, a type of nonlinear oscillator, can converge from any non-zero state in the state space to a limit cycle, generating stable periodic rhythmic oscillatory signals. Therefore, we choose the Hopf oscillator as the oscillator unit for the CPG. Currently, for hexapod robot motion control, a fully connected network topology consisting of six oscillator units is typically used. In this study, the tripod gait is chosen as the basic gait for hexapod robots, and thus two oscillators are selected to form the CPG network for the hexapod robot. The

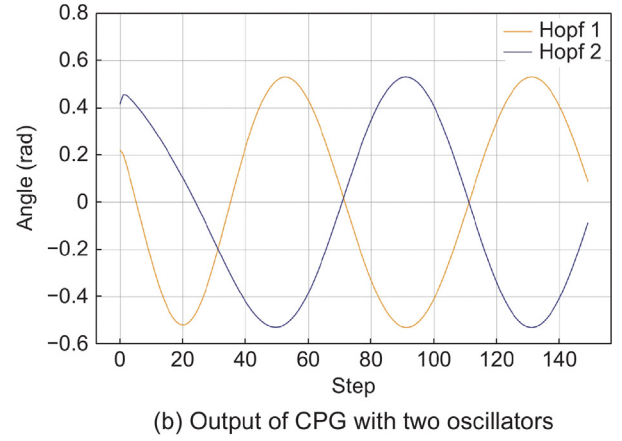
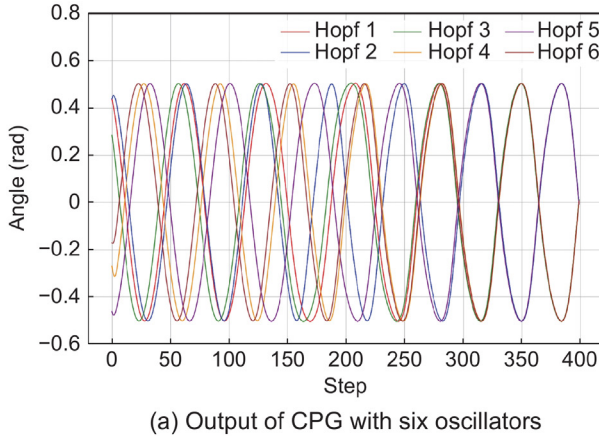


Fig. 1. Output of CPG with different numbers of oscillators.

outputs of CPGs with different numbers of oscillators are in Fig. 1. Fig. 1(a) shows the output of CPG with six oscillators, while Fig. 1(b) presents the output of CPG with two oscillators. From Fig. 1, we can see that the CPG with six oscillators converges at 200 steps, whereas the CPG with two oscillators has already converged at around 70 steps. Moreover, the CPG with two oscillators effectively avoids errors in multiple oscillator signals affecting the robot's movement while significantly reducing the complexity of the CPG network and the difficulty of controlling the hexapod robot. The mathematical model of the CPG network is as follows:

$$\begin{cases} \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ r_i^2 \\ \Delta_{ji} \end{pmatrix} = \begin{pmatrix} \alpha(\mu - r_i^2) & -\omega_i \\ \omega_i & \alpha(\mu - r_i^2) \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \delta \begin{pmatrix} 0 \\ \Delta_{ji} \end{pmatrix} \\ r_i^2 = x_i^2 + y_i^2 \\ \Delta_{ji} = y_j \cos \theta_{ji} - x_j \sin \theta_{ji} \end{cases} \quad (3)$$

where x and y are the output signals of the oscillator. α is the convergence rate parameter of the model, μ is the square of the limit cycle radius, ω is the oscillation frequency, ω_{sw} is the swing phase frequency, ω_{st} is the stance phase frequency, δ is the coupling strength coefficient, and θ_{ji} represents the phase difference between the two oscillators.

2.3. Mapping function

In this study, we define the rising edge of the CPG output signal as the swing phase, and the falling edge as the stance phase. Researchers have observed a distinct pattern of leg movement in hexapod insects during locomotion: during the swing phase, the hip joint rotates forward first, followed by the knee joint rotating to lift the leg and then reversing to return the leg to a balanced position. The ankle joint behaves similarly to the knee joint but in the opposite direction; that is, when the knee joint rotates forward, the ankle joint rotates backward, and vice versa. During the stance phase, the hip joint rotates backward, while the knee and ankle joints remain stationary. Based on these observations, previous researchers [34,35] designed a mapping function that effectively adapts to the motion pattern of insects. However, this function involves many parameters, making it difficult to adjust. Therefore, based on this movement pattern, this study designs a mapping function with relatively fewer parameters. The mathematical model of this function is shown in Eq. (4), and the output of mapping function is illustrated in Fig. 2.

$$\phi(t) = \begin{pmatrix} \phi_1(t) \\ \phi_2(t) \\ \phi_3(t) \end{pmatrix} = \begin{pmatrix} A_1 y(t) \\ \begin{cases} A_2(1 - |y(t)|^2), & \text{if } \dot{y}(t) \geq 0 \\ 0, & \text{if } \dot{y}(t) < 0 \end{cases} \\ -A_3 \phi_2(t) \end{pmatrix} \quad (4)$$

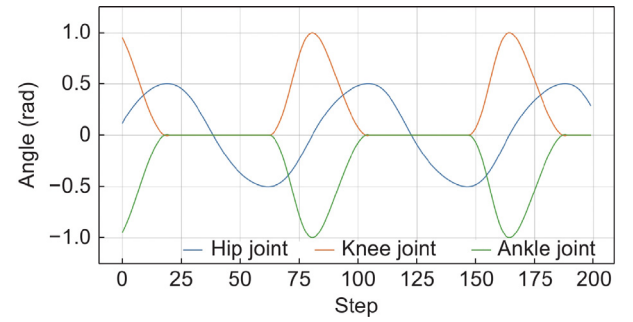


Fig. 2. Output of mapping function.

where $\phi_1(t)$ is the angle of the hip joint. $\phi_2(t)$ is the angle of the knee joint. $\phi_3(t)$ is the angle of the ankle joint. A_1 is the amplitude of the hip joint. A_2 is the amplitude of the knee joint. A_3 is the amplitude of the ankle joint.

2.4. Hierarchical reinforcement learning

The motion control strategy proposed in this paper is based on hierarchical CPG_RL, and the overall framework, as shown in Fig. 3, consists of three main components: the high-level reinforcement learning decision control layer, the mid-level inter-leg coordination layer, and the low-level intra-leg joint coordination layer. At the high-level control, the PPO algorithm [36] is used, which, based on the state space information S_t and environmental reward R_t , outputs an action a_t . These action parameters include ω , which is passed to the mid-level coordination layer, and $A_1[1]$, $A_1[2]$, $A_2[j]$, and $A_3[j]$ (where $j = 1, \dots, 6$, with $A_1[1]$ representing the right leg's hip joint amplitude, $A_1[2]$ representing the left leg's hip joint amplitude, and $A_2[j]$ and $A_3[j]$ representing the knee joint and ankle joint amplitudes for each leg of the hexapod robot, respectively), which are passed to the low-level joint coordination layer. The high-level controller updates the parameters of the mid-level and low-level controllers at each time step based on the current state, including the robot's own state, the parameters of the controllers at all levels, and the reward function. This continuous update improves the control decisions, enabling the high-level controller to make more effective decisions in different environments. The mid-level controller adjusts the step frequency of the legs using the ω parameter passed from the high-level controller, while ensuring the coordination of the legs during movement. The mid-level controller not only sends parameter y to the low-level controller to help it execute specific

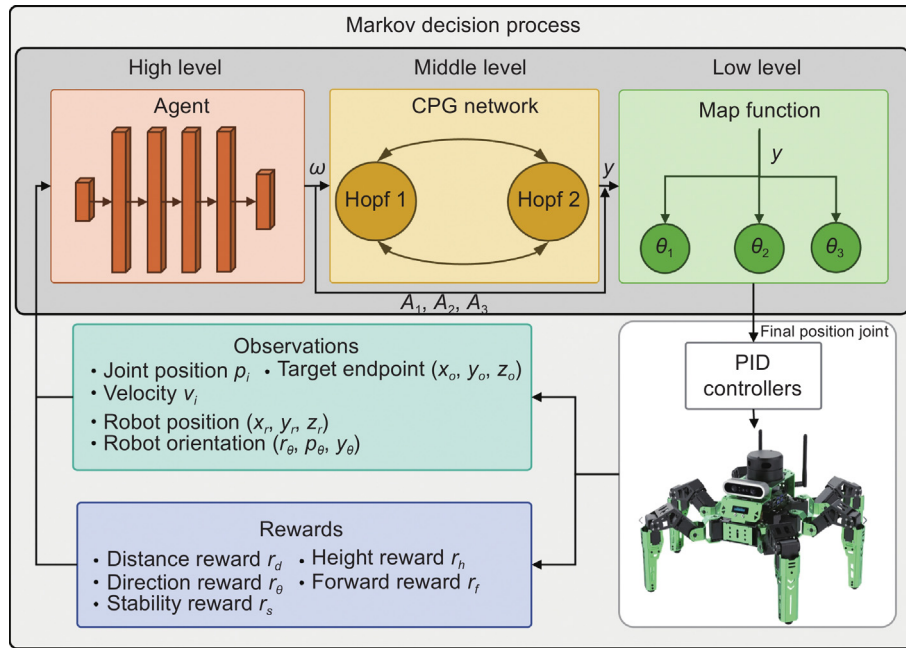


Fig. 3. Framework of hierarchical CPG-RL method. The high-level controller uses the PPO algorithm to learn the parameters of the middle-level controller and the low-level controller according to the observed spatial information and the reward. The middle-level controller generates the inter-leg control signals and passes them to the low-level controller, which generates the joint angles of the hexapod robot through the mapping function.

joint movements accurately, but also acts as a bridge between the high-level and low-level controllers, ensuring smooth information flow and coordination between the layers. The low-level controller relies on the inter-leg coordination signals provided by the mid-level controller and the step length parameters provided by the high-level controller to generate precise joint movements. The performance of the low-level controller directly impacts the robot's motion accuracy and stability, which in turn affects the strategy adjustments made by the high-level controller. The CPG provides a stable gait and adaptive motion generation, while reinforcement learning optimizes the global strategy through intelligent decision-making. The high-level is responsible for decision optimization, the mid-level ensures inter-leg coordination, and the low-level generates specific joint angles. This hierarchical architecture ensures the robot's stability, adaptability, and efficiency across various complex terrains.

2.4.1. State space

All possible state configurations of the robot and environment are considered as the state space for reinforcement learning. Previous research [37,38] often relies on extensive environmental data collected by sensors such as radar and depth cameras, resulting in a high-dimensional state space that makes reinforcement learning difficult to converge. To address this issue, the state space of reinforcement learning in this study only includes the robot's own information and the maximum height of the terrain, effectively reducing the state space to 59 dimensions. These 59 dimensions include: the position s_i and velocity v_i of the hexapod robot's 18 joints, the robot's target positional information (x, y, z) , and the CPG network's parameters $A_1[1]$, $A_1[2]$, $A_2[j]$, and $A_3[j]$, as well as max height h in the robot's environment.

2.4.2. Action space

In this study, the set of all possible actions the hexapod robot can perform in the environment is considered as the action space. The main objective of this study is to enhance the robot's terrain adaptability by training the CPG network's parameters. The action space includes the amplitudes of the hip joint $A_1[1]$,

$A_1[2]$, the knee joint $A_2[i]$, the ankle joint $A_3[i]$ in the lower-level controller, and the frequency ω in the mid-level controller. These parameters, totaling 15 dimensions, allow the hexapod robot to learn and generate gaits that adapt to various terrains by adjusting these values.

2.4.3. Reward function

To enhance the motion stability of the hexapod robot in complex terrains, we design several reward components to ensure that the robot can effectively reach the designated target points and adapt to various environmental conditions. The details of the reward function are shown in Table 1 and the total reward function is as follows.

$$R_t = \omega_h \cdot r_h + \omega_d \cdot r_d + \omega_\theta \cdot r_\theta + r_f + r_s \quad (5)$$

where z_r represents the robot's current height, ω_h is the weight for height reward, and θ denotes the robot's direction angle, the various rewards are calculated to optimize the robot's performance. The vector \mathbf{D} indicates the vector representing the point from the starting point to the target point. The distances d_n and d_t refer to the current and target distances, respectively. Angles θ_y and θ_x measure the robot's tilt, while x_r and x_{pr} represent the robot's current and previous positions. The weights ω_θ , and ω_d correspond to the direction, energy, and distance rewards, respectively. Lastly, r_h , r_θ , r_d , r_f , r_s denote the five different reward functions, each influencing the robot's learning process.

2.4.4. Termination conditions

Termination conditions are used to determine whether the robot has completed the task or met the predefined requirements, ensuring effective operation within reasonable limits. The specific conditions include:

- (1) The task should be immediately terminated when the robot's height is below 0.05 m, or when the roll angle or pitch angle exceeds $\pm \frac{\pi}{3}$, as it indicates that the robot has fallen.

Table 1
Reward components of total reward function and their weights.

Reward type	Reward function	Condition	Weight
Height Reward r_h	$(z_r - 0.11)$	$z_r > 0.12$	ω_h
Direction Reward r_θ	$\frac{(\cos \theta, \sin \theta) \cdot \mathbf{D}}{\ \mathbf{D}\ }$	$\theta > \frac{\pi}{6}$	ω_θ
Distance Reward r_d	$e^{-(d_n - d_t)}$	None	ω_d
Stability Reward r_s	-100	$z_r < 0.05$ or $ \theta_y > \frac{\pi}{6}$ or $ \theta_x > \frac{\pi}{6}$	1
Forward Reward r_f	-0.3	$x_r - x_{pr} < 0.01$	1

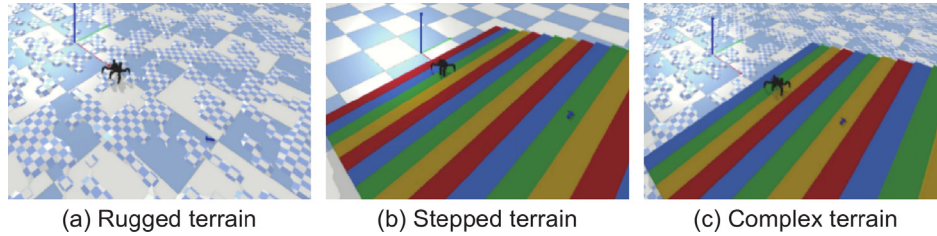


Fig. 4. Three simulation environment terrains in PyBullet. (a) The rugged terrain. (b) The stepped terrain. (c) The complex terrain.

Table 2
Hyperparameters of PPO.

Parameter	Value
Learning rate	1.0×10^{-4}
Batch size	512
Epochs	500 000
Gamma	0.99
Lambda	0.95
Clip parameter	0.2
Neural network architecture	[256, 256, 256, 256]
Activation function	ReLU

- (2) The task should be terminated if the number of steps exceeds 7000, or if the absolute value of the robot's Y-axis position exceeds 1.5 m, to improve efficiency and accuracy and avoid ineffective exploration.
- (3) When the robot is within 0.5 m of the target, it is considered to have reached the goal, and the task is complete.

3. Experiment

To validate the effectiveness of the CPG_RL method in hexapod robot control, this section is organized into the following four parts: experimental setup, comparison of learning outcomes, comparison of motion performance, and Sim2real experiments.

3.1. Experimental setup

In this study, Pybullet [39] is used as the simulation platform to set up the reinforcement learning environment for the hexapod robot. We adapt the OpenAI Gym framework for the reinforcement learning environment and compare the PPO, SAC [40], TD3 [41], and A2C [42] algorithms from the Stable Baselines3 library. To ensure the validity of the comparison experiments, all experiments are conducted with the same hyperparameters, as shown in Table 2. Additionally, three simulation environment terrains in Pybullet showed in Fig. 4 are designed to train the hexapod robot, which includes rugged terrain, stepped terrain, and complex terrain (the combination of rugged and stepped terrain), and their parameters are as follows:

- (1) Rugged terrain consists of randomly generated square protrusions of 4 cm^2 and heights varying between 0 and 6 cm.

- (2) Stepped terrain consists of 15 steps, each step is 6 cm in height, 500 cm in length, and 25 cm in width, with a total height of 90 cm, length of 500 cm and width of 375 cm.
- (3) Complex terrain is the combination of 150 cm width of rugged terrain and 150 cm width stepped terrain.

3.2. Comparison of learning effectiveness

This experiment validates the learning effectiveness of our method through two comparative experiments. In terms of learning efficiency, we evaluate it by comparing the final learning outcomes of different algorithms and their corresponding learning time. These experimental results indicate that the CPG_RL method demonstrates superior learning performance and adaptability on complex terrains.

- (1) **Comparison of training effectiveness:** In this experiment, we compare the training effectiveness of the hexapod robot under the control of hierarchical reinforcement learning (SAC-CPG, A2C-CPG, CPG_RL) across three different environment terrains. The results in Fig. 5(a) show that the CPG_RL method exhibiting the best learning performance, including faster convergence and a higher task completion rate.
- (2) **Comparison of learning outcomes:** This experiment compares the CPG's parameter variations of the hexapod robot under the control of CPG_RL and CPG. The experimental results show that the parameter values of the CPG method remain constant throughout the process, while the CPG_RL method, which integrates reinforcement learning, dynamically adjusts the parameters according to terrain variations, demonstrating greater adaptability.

3.3. Comparison of robot's stability

To evaluate the stability of the CPG_RL method, the hexapod robot is tested on rugged, stepped, and complex terrains, comparing the performance of CPG control and hierarchical reinforcement learning control in reaching the designated target positions. In terms of control performance, the evaluation metrics include changes in the robot's pitch and roll angles during motion, as well as the trajectory curves of the hexapod robot. In addition, we conduct a quantitative analysis of the curves in the figures, measuring stability in terms of the average value, variance, and peak value of the fluctuations.

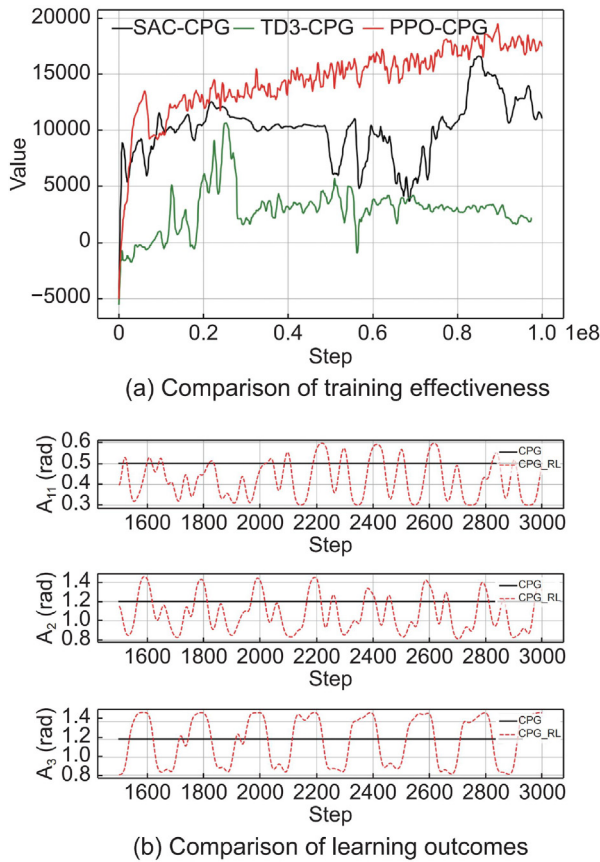


Fig. 5. Comparison of Learning Effectiveness. (a) The leaning reward curve across different algorithms. (b) The curve of parameter variations under the control of CPG and CPG_{RL}.

- (1) **Rugged terrain:** As shown in Fig. 6(a), in terms of pitch angle, the fluctuations of pitch angles under CPG_{RL} and SAC-CPG control are smaller, indicating higher stability, whereas the pitch angle fluctuations under TD3-CPG and CPG control are larger. Furthermore, as shown in Fig. 6(b), the mean value of CPG_{RL} is significantly lower than that of other methods, and the variance of CPG_{RL} and SAC-CPG is noticeably smaller than that of the other two methods. As shown in Fig. 6(c), in terms of roll angle, the angle fluctuations under CPG_{RL} and SAC-CPG control are relatively stable, with no large fluctuations, while larger peaks appear under TD3-CPG and CPG control. Fig. 6(d) further shows that the mean and variance of CPG_{RL} are noticeably smaller than those of other methods. In terms of motion trajectory, as shown in Fig. 6(e), all methods successfully reach the target point on rugged terrain, with Y-axis deviations kept within 0.5 m, ensuring the completion of the specific task. Additionally, as shown in Fig. 6(a) and Fig. 6(c), CPG_{RL} and CPG completed the task with fewer steps, demonstrating faster speed, with CPG_{RL} showing smaller deviation in the Y direction.
- (2) **Stepped terrain:** As shown in Fig. 7(a), the pitch angle fluctuations are similar across all control methods. However, from Fig. 7(b), it is evident that the mean values of SAC-CPG, TD3-CPG, and CPG_{RL} are roughly the same and smaller than that of the CPG method, while the variance of CPG_{RL} is slightly larger than the other three methods. As shown in Fig. 7(c), in terms of roll angle, all control methods exhibit periodic fluctuations, and the amplitudes are similar. Fig. 7(d) shows that the mean values of all

algorithms are quite similar, with the variance of CPG and CPG_{RL} being noticeably smaller than the other methods. In terms of motion trajectory, as shown in Fig. 7(e), only SAC-CPG and CPG_{RL} can reliably complete the task on the stepped terrain, while CPG shows a significant deviation along the Y-axis, and TD3-CPG fails to complete the task. In Fig. 7(a) and Fig. 7(c), CPG_{RL} completes the task in fewer time steps compared to SAC-CPG, and among the two algorithms that stably complete the task, CPG_{RL} does so at the fastest speed. On stepped terrain, due to the faster movement speed of CPG_{RL}, larger angle fluctuations occur, but this does not mean that the robot's performance under CPG_{RL} control is worse.

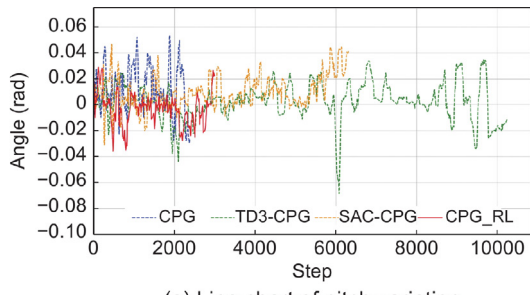
- (3) **Complex terrain:** In terms of pitch angle on complex terrain, as shown in Fig. 8(a), during the rugged terrain phase, all methods show small, irregular pitch angle fluctuations. Upon entering the stepped terrain, the pitch angles under TD3-CPG, SAC-CPG, and CPG_{RL} control exhibit periodic fluctuations, indicating that the robot is climbing the stepped terrain, whereas the pitch angle under CPG control shows large peaks, indicating task failure. Fig. 8(b) further shows that the mean and variance of pitch angles under TD3-CPG, SAC-CPG, and CPG_{RL} control are similar, whereas the mean pitch angle under CPG control is larger, indicating lower stability. As shown in Fig. 8(c), the roll angle follows a similar pattern to the pitch angle, and Fig. 8(d) shows that the mean values under TD3-CPG, SAC-CPG, and CPG_{RL} are almost identical, with CPG control exhibiting a larger mean value and lower stability, while the variances of SAC-CPG, CPG_{RL}, and CPG control are similar, and TD3-CPG exhibits a larger variance. In terms of motion trajectory, as shown in Fig. 8(e), SAC-CPG and CPG_{RL} are able to reach the target point with Y-axis deviation within 0.5 m, while under CPG control, significant deviation occurs when entering the stepped terrain, and TD3-CPG fails to complete the task. Additionally, as shown in Fig. 8(a) and Fig. 8(c), CPG_{RL} completes the task in fewer steps, demonstrating a higher speed.

Based on the above experimental results, the following conclusion can be drawn: the CPG_{RL} control method significantly outperforms other control methods in terms of stability. CPG_{RL} not only exhibits superior performance in minimizing the amplitude and frequency of pitch and roll angle fluctuations, but also effectively reduces deviations along the Y-axis, ensuring a more stable motion trajectory on complex terrains.

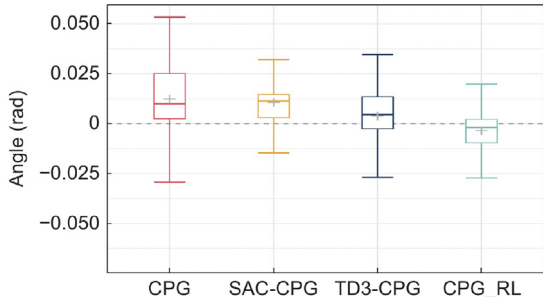
3.4. Comparison of robot's adaptability

To validate the adaptability of the CPG_{RL} control method, we conduct 25 rounds of testing on the hexapod robot across three different terrains using CPG control, CPG_{RL} control, and CPG_{RL} control. The adaptability of the three control methods is evaluated by calculating the success probability of the robot reaching distances of 0.5 m, 1 m, 2 m, and 3 m, and the results are shown in Table 3.

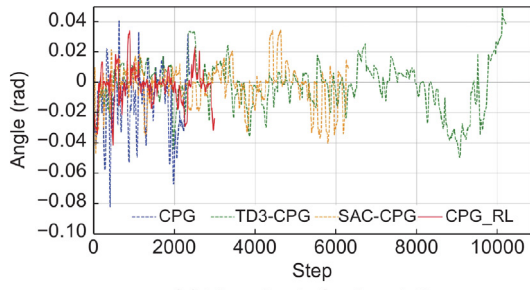
- (1) **Rugged terrain:** The CPG control method achieves a high probability of completing tasks on rugged terrain. The TD3-CPG method performs well between 1 and 2 m but shows a decline in performance at 3 m, although it is still better than the CPG method. The SAC-CPG method nearly reaches a 100% success rate, while the CPG_{RL} method completes all tasks on rugged terrain with a 100% success rate.



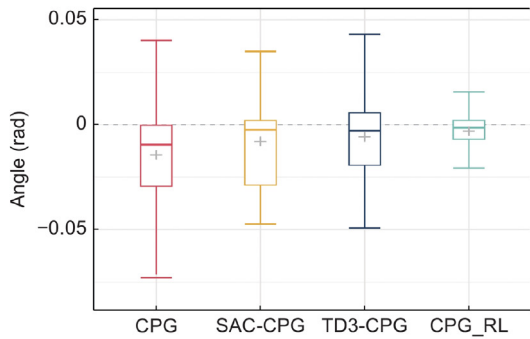
(a) Line chart of pitch variation



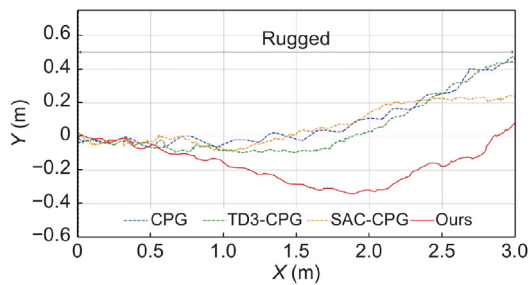
(b) Box plot of pitch



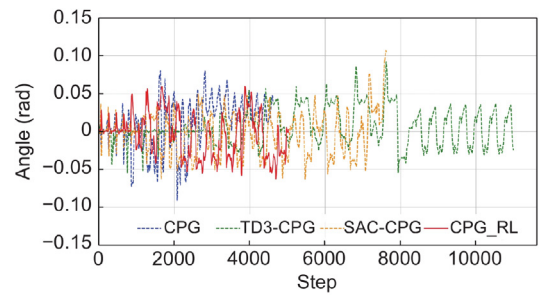
(c) Line chart of roll variation



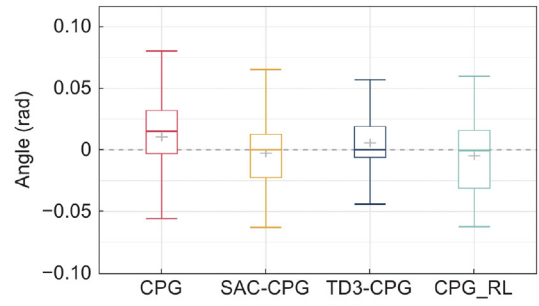
(d) Box plot of roll



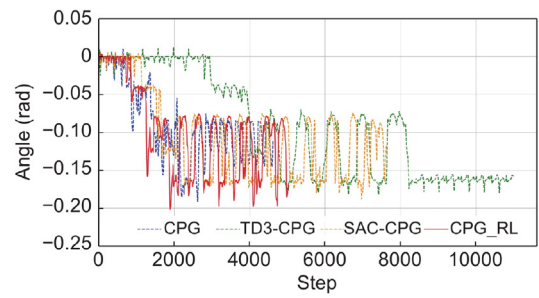
(e) Trajectory on rugged terrain



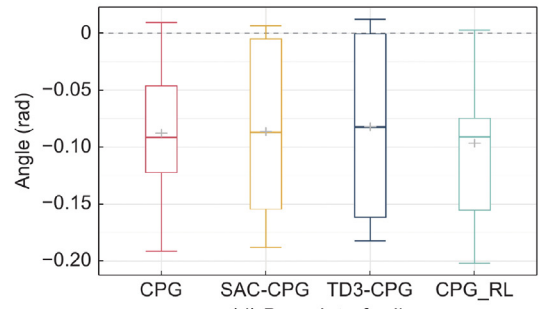
(a) Line chart of pitch variation



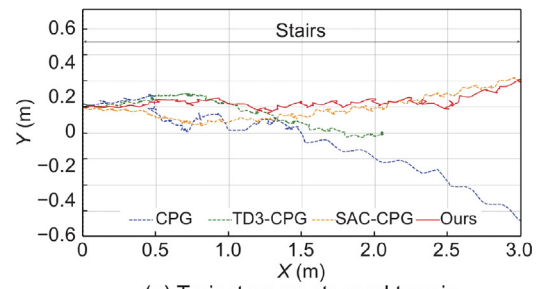
(b) Box plot of pitch



(c) Line chart of roll variation



(d) Box plot of roll



(e) Trajectory on stepped terrain

Fig. 6. Comparison of stability between CPG Method and hierarchical reinforcement learning methods including TD3-CPG, SAC-CPG and ours (CPG_RL) on rugged terrain. (a) Line chart shows pitch angles. (b) Box plot shows pitch angles. (c) Line chart showing roll angles. (d) Box plot shows roll angles. (e) Movement trajectory of the robot.

Fig. 7. Comparison of stability between CPG Method and hierarchical reinforcement learning methods including TD3-CPG, SAC-CPG and ours (CPG_RL) on stepped terrain. (a) Line chart shows pitch angles. (b) Box plot shows pitch angles. (c) Line chart showing roll angles. (d) Box plot shows roll angles. (e) Movement trajectory of the robot.

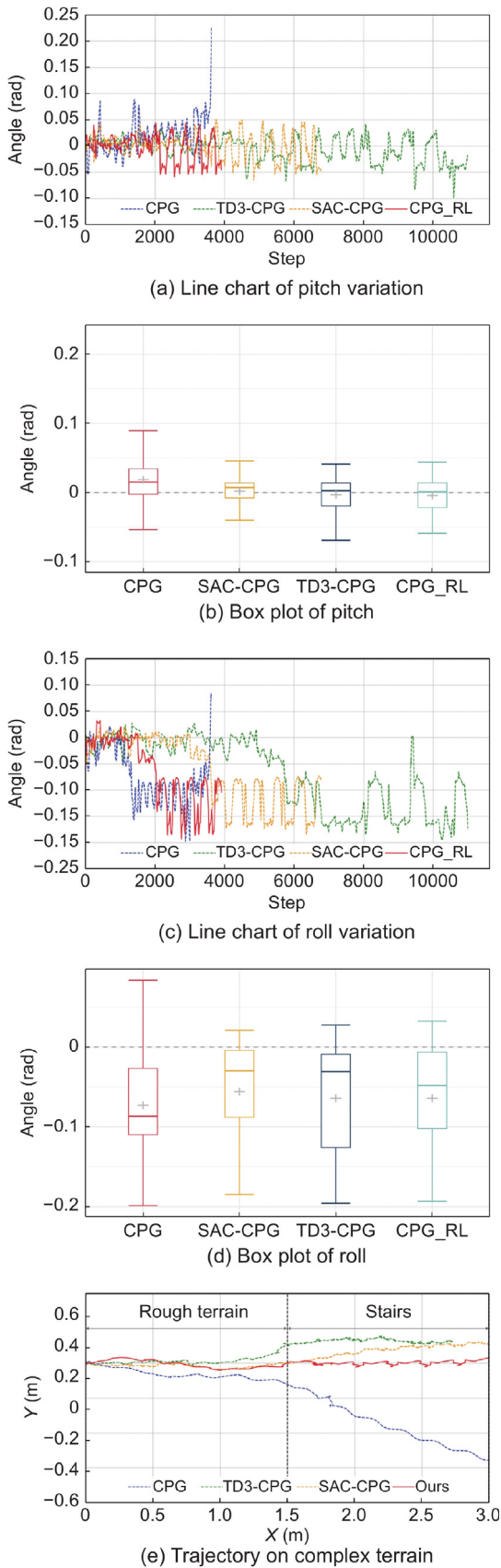


Fig. 8. Comparison of stability between the CPG method and hierarchical reinforcement learning approaches including TD3-CPG, SAC-CPG, and our approach (CPG_RL) on complex terrain. (a) Line chart shows pitch angles. (b) Box plot shows pitch angles. (c) Line chart showing roll angles. (d) Box plot shows roll angles. (e) Movement trajectory of the robot.

Table 3

Success rates of different methods across three terrains at different distances.

Terrain	Method	0.5 m	1 m	2 m	3 m
Rugged	CPG	96%	96%	88%	60%
	TD3-CPG	100%	100%	100%	70%
	SAC-CPG	100%	100%	100%	96%
Stepped	CPG_RL	100%	100%	100%	100%
	CPG	60%	28%	0%	0%
	TD3-CPG	100%	84%	0%	0%
Complex	SAC-CPG	96%	96%	96%	52%
	CPG_RL	100%	100%	100%	100%
	CPG	92%	56%	8%	0%
	TD3-CPG	100%	100%	12%	0%
	SAC-CPG	100%	80%	40%	4%
CPG_RL	100%	100%	100%	100%	

- (2) **Stepped terrain:** The CPG control method completes tasks within 0.5 m on stepped terrain with high probability but has significantly lower success rates beyond 1 m and cannot complete tasks beyond 2 m. The TD3-CPG method performs well within 1 m but also fails to complete tasks beyond 2 m. The SAC-CPG method completes tasks up to 2 m with nearly a 100% success rate, but for tasks over 3 m, the success rate drops to 52%. The CPG_RL control method, however, achieves a 100% success rate for all tasks on stepped terrain.
- (3) **Complex terrain:** The CPG control method has a high success probability for tasks within 0.5 m, but its performance drops significantly beyond 1 m, and it cannot complete tasks beyond 3 m. The TD3-CPG method performs well up to 2 m but struggles beyond that distance. The SAC-CPG method shows good performance beyond 1 m, but its success rate drops to 40% for tasks over 2 m and to only 4% for tasks over 3 m. The CPG_RL method achieves a 100% success rate for all tasks on complex terrain.

Based on these results, the following conclusions can be drawn: The CPG control method, due to its lack of adaptability to complex terrains, exhibits significant deviations in movement direction, making it difficult to complete tasks reliably. The TD3-CPG method performs well on rugged terrain but cannot complete long-distance tasks on stepped terrain due to stalling when crossing steps, indicating it does not learn effective terrain-crossing strategies. The SAC-CPG method performs well on rugged terrain but struggles with long-distance tasks on stepped and complex terrains, with relatively poor performance. The CPG_RL control method demonstrates excellent adaptability and stability, successfully navigating various complex terrains, indicating its superior practicality and reliability in diverse and dynamic environments.

3.5. Sim2real experiments

This study aims to verify the feasibility and effectiveness of the CPG_RL algorithm on a physical hexapod robot, ensuring that the algorithm can operate smoothly in real-world environments while demonstrating superior adaptability and stability. The experiment used a hexapod robot controlled by the ROS operating system. The key parameters of the hexapod robot are as follows: total weight of 2.5 kg, hip length of 50 mm, femur length of 80 mm, and tibia length of 130 mm. The joint angle limits are: hip angle from $-\pi/6$ to $\pi/6$, femur angle from 0 to $\pi/2$, and tibia angle from $-\pi/3$ to 0.

The test terrains shown in Fig. 9 include rugged terrain and stepped terrain. The rugged terrain is 2.5 m long, consisting of obstacles of varying heights with a maximum height difference of 6 cm. This design requires the hexapod robot to traverse various



(a) Movement in rugged terrain



(b) Movement in stepped terrain

Fig. 9. Physical scene demonstration. (a) Robot motion in rugged terrain. (b) Robot motion in stepped terrain.

terrain obstacles, effectively evaluating the algorithm's adaptability to complex terrains. The stepped terrain is 1.5 m long and composed of five wooden boards, each 6 cm in height. This terrain simulates the stepped terrain used in the simulation environment, but with a reduced width to better evaluate the algorithm's performance in real-world conditions. The experimental results indicate that the hexapod robot performed exceptionally well on the stepped terrain, with the CPG_RL control method enabling it to stably traverse each step. Despite the narrow steps increasing the difficulty, the robot successfully completed the task without falling or stalling, demonstrating excellent stability. In the rugged terrain test, the hexapod robot also performed outstandingly, consistently crossing obstacles of varying heights and displaying excellent adaptability and terrain-handling capability. Throughout the process, the robot's movement trajectory remained stable with no significant deviations, fully validating the feasibility and effectiveness of the CPG_RL algorithm in real-world applications.

4. Conclusion

This study proposes a hierarchical control architecture to enhance the adaptability of hexapod robots on complex terrains. The architecture consists of high-level, mid-level and low-level controllers. The hierarchical method's learning efficiency, stability, and adaptability are evaluated through experiments on learning and motion performance validation. The results show that CPG_RL control outperforms traditional CPG control in several aspects, particularly in handling various terrains where it exhibits stronger adaptability and dynamically adjusts its parameters based on terrain changes. Moreover, compared to traditional DRL control, CPG_RL control demonstrates higher learning efficiency and better learning performance. In terms of motion performance, CPG_RL control shows superior stability and adaptability compared to CPG control, successfully completing tasks on a variety of complex terrains. The method is also successfully transferred to real-world experiments, demonstrating excellent motion performance. Future research will focus on integrating

sensors into this method to further improve the adaptability of hexapod robots in complex terrains. In the future, we plan to equip physical robots with visual sensors to enhance their perceptual capabilities, while increasing the terrain complexity to challenge the robots' adaptability and stability in more demanding environments, thus driving further experimental validation and technological optimization.

CRedit authorship contribution statement

Shichang Huang: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Zhihan Xiao:** Writing – review & editing, Writing – original draft, Conceptualization. **Minhua Zheng:** Resources, Funding acquisition. **Wen Shi:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Beijing Natural Science Foundation - Xiaomi Innovation Joint Fund (L243013) and the National Natural Science Foundation of China (62172392).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.birob.2025.100231>.

References

- [1] J. Wu, H. Yang, R. Li, Q. Ruan, S. Yan, Y. an Yao, Design and analysis of a novel octopod platform with a reconfigurable trunk, *Mech. Mach. Theory* 156 (2021).
- [2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots, *Sci. Robot.* 4 (26) (2019) eaau5872.
- [3] P. Čížek, J. Faigl, Self-supervised learning of the biologically-inspired obstacle avoidance of hexapod walking robot, *Bioinspiration & Biomimetics* 14 (4) (2019) 046002, <http://dx.doi.org/10.1088/1748-3190/ab1a9c>.
- [4] B. Xia, K. Che, Z. Tang, J. Wang, M.Q.-H. Meng, Motion planning for hexapod robots in dynamic rough terrain environments, in: 2021 IEEE International Conference on Robotics and Biomimetics, ROBIO, 2021, pp. 1611–1616, <http://dx.doi.org/10.1109/ROBIO54168.2021.9739381>.
- [5] A. Stelzer, H. Hirschmüller, M. Görner, Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain, *Int. J. Robot. Res.* 31 (4) (2012) 381–402.
- [6] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, M. Hutter, Perceptive locomotion through nonlinear model-predictive control, *IEEE Trans. Robot.* 39 (5) (2023).
- [7] A. Torres-Pardo, D. Pinto-Fernández, M. Garabini, F. Angelini, D. Rodriguez-Cianca, S. Massardi, J. Tornero, J.C. Moreno, D. Torricelli, Legged locomotion over irregular terrains: state of the art of human and robot performance, *Bioinspiration Biomimetics* 17 (6) (2022) 061002, <http://dx.doi.org/10.1088/1748-3190/ac92b3>.
- [8] D. Owaki, A. Ishiguro, A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping, *Sci. Rep.* 7 (1) (2017).
- [9] H. Tanaka, O. Matsumoto, T. Kawasetsu, K. Hosoda, Enhancing postural stability in musculoskeletal quadrupedal locomotion through tension feedback for CPG-based controller, *Bioinspiration Biomimetics* (2024) URL <http://iopscience.iop.org/article/10.1088/1748-3190/ad839e>.
- [10] M. Shafiee, G. Bellegarda, A. Ijspeert, Puppeteer and marionette: Learning anticipatory quadrupedal locomotion based on interactions of a central pattern generator and supraspinal drive, in: 2023 IEEE International Conference Robotics Automation, ICRA, 2023.
- [11] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, G. Cheng, Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot, *Adv. Bioinform.* 27 (2) (2008) 213–228.

- [12] H. Kimura, Y. Fukuoka, A.H. Cohen, Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts, *Philos. Trans. Ser. A, Math. Phys. Eng. Sci.* 26 (5) (2007) 475–490.
- [13] L. Righetti, A.J. Ijspeert, Pattern generators with sensory feedback for the control of quadruped locomotion, in: *IEEE International Conference on Robotics and Automation*, 2008.
- [14] A.J. Ijspeert, A. Crespi, D. Ryczko, J.-M. Cabelguen, From swimming to walking with a salamander robot driven by a spinal cord model, *Science* 315 (5817) (2007) 1416–1420.
- [15] Y. Zeng, J. Li, S.X. Yang, E. Ren, A bio-inspired control strategy for locomotion of a quadruped robot, *Appl. Sci.* 8 (1) (2018) <http://dx.doi.org/10.3390/app8010056>, URL <https://www.mdpi.com/2076-3417/8/1/56>.
- [16] D.J. Hyun, S. Seok, J. Lee, S. Kim, High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah, *I. J. Robot. Res.* 33 (11) (2014) 1417–1445.
- [17] L. Righetti, A.J. Ijspeert, Pattern generators with sensory feedback for the control of quadruped locomotion, in: *IEEE International Conference on Robotics and Automation*, 2008.
- [18] C. Liu, L. Xia, C. Zhang, Q. Chen, Multi-layered CPG for adaptive walking of quadruped robots, *J. Bionic Eng.* 15 (2) (2018) 341–355.
- [19] B. Wang, K. Zhang, X. Yang, X. Cui, The gait planning of hexapod robot based on CPG with feedback, *Int. J. Adv. Robot. Syst.* 17 (3) (2020) 172988142093050.
- [20] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain, *Sci. Robot.* 5 (47) (2020).
- [21] K. Li, Y. Xu, J. Wang, M.Q.-H. Meng, SARL: Deep reinforcement learning based human-aware navigation for mobile robot in indoor environments, in: *2019 IEEE International Conference on Robotics and Biomimetics, ROBIO*, 2019, pp. 688–694, <http://dx.doi.org/10.1109/ROBIO49542.2019.8961764>.
- [22] N. Rudin, D. Hoeller, P. Reist, M. Hutter, Learning to walk in minutes using massively parallel deep reinforcement learning, *Comput. Res. Repos.* (2021).
- [23] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, V. Vanhoucke, Sim-to-real: Learning agile locomotion for quadruped robots, *Robot.: Sci. Syst.* XIV (2018) [arXiv:1804.10332](https://arxiv.org/abs/1804.10332).
- [24] A. Kumar, Z. Fu, D. Pathak, J. Malik, RMA: Rapid motor adaptation for legged robots, in: *Robotics: Science and Systems*, 2021.
- [25] S. Chen, B. Zhang, M.W. Mueller, A. Rai, K. Sreenath, Learning torque control for quadrupedal locomotion, in: *2023 IEEE-RAS 22nd International Conference Humanoid Robots, Humanoids*, 2023.
- [26] G. Bellegarda, C. Nguyen, Q. Nguyen, Robust quadruped jumping via deep reinforcement learning, *Robot. Auton. Syst.* (2024) 104799.
- [27] G. Bellegarda, Y. Chen, Z. Liu, Q. Nguyen, Robust high-speed running for quadruped robots via deep reinforcement learning, in: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2022.
- [28] G. Li, A. Ijspeert, M. Hayashibe, AI-CPG: Adaptive imitated central pattern generators for bipedal locomotion learned through reinforced reflex neural networks, *IEEE Robot. Autom. Lett.* PP (99) (2024) 1–8.
- [29] B. Guillaume, S. Milad, I. Auke, Visual CPG-RL: Learning central pattern generators for visually-guided quadruped locomotion, in: *ICRA 2024*, 2024.
- [30] S. Su, Y. Chen, C. Li, K. Ni, J. Zhang, Intelligent control strategy for robotic manta via CPG and deep reinforcement learning, *Drones* 8 (7) (2024).
- [31] S. Huang, M. Zheng, Z. Hu, P.X. Liu, Enhancing hexapod robot mobility on challenging terrains: Optimizing CPG-generated gait with reinforcement learning, *Neurocomputing* 622 (2025) 129328.
- [32] W. Ouyang, H. Chi, J. Pang, W. Liang, Q. Ren, Adaptive locomotion control of a hexapod robot via bio-inspired learning, *Front. Neurobotics* 15 (2021).
- [33] D. Li, W. Wei, Z. Qiu, Combined reinforcement learning and CPG algorithm to generate terrain-adaptive gait of hexapod robots, *Actuators* 12 (4) (2023) 157.
- [34] W. Zhang, Q. Gong, H. Yang, Y. Tang, CPG modulates the omnidirectional motion of a hexapod robot in unstructured terrain, *J. Bionic Eng.* 20 (2) (2023) 558–567, <http://dx.doi.org/10.1007/s42235-022-00290-1>.
- [35] D. Li, W. Wei, Z. Qiu, Combined reinforcement learning and CPG algorithm to generate terrain-adaptive gait of hexapod robots, *Actuators* 12 (4) (2023) <http://dx.doi.org/10.3390/act12040157>.
- [36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, [arXiv:1707.06347](https://arxiv.org/abs/1707.06347). URL <https://arxiv.org/abs/1707.06347>.
- [37] L. Wang, R. Li, Z. Huangfu, Y. Feng, Y. Chen, A soft actor-critic approach for a blind walking hexapod robot with obstacle avoidance, *Actuators* 12 (10) (2023) <http://dx.doi.org/10.3390/act12100393>.
- [38] Z. Zang, M. Kawawa-Beaudan, W. Yu, T. Zhang, A. Zakhor, Perceptive hexapod legged locomotion for climbing joint environments, in: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2023, pp. 2738–2745, <http://dx.doi.org/10.1109/IROS55552.2023.10341957>.
- [39] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, A.P. Schoellig, Learning to fly—a gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control, in: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2021, pp. 7512–7519, <http://dx.doi.org/10.1109/IROS51168.2021.9635857>.
- [40] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018, [arXiv:1801.01290](https://arxiv.org/abs/1801.01290). URL <https://arxiv.org/abs/1801.01290>.
- [41] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, 2018, [arXiv:1802.09477](https://arxiv.org/abs/1802.09477). URL <https://arxiv.org/abs/1802.09477>.
- [42] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, 2016, [arXiv:1602.01783](https://arxiv.org/abs/1602.01783). URL <https://arxiv.org/abs/1602.01783>.