Enhancing Hexapod Robot Mobility on Challenging Terrains: Optimizing CPG-Generated Gait with Reinforcement Learning

Shichang Huang[a], Minhua Zheng[a,b,*], Zhongyu Hu[a], Peter Xiaoping Liu[c]

[a]School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China
[b]School of Mechanical, Electronic and Control Engineering and the Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology, Ministry of Education, Beijing Jiaotong University, Beijing 100044, China
[c]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Abstract

Hexapod robots, with more legs and freedoms, excel in stability and terrain adaptability over wheeled or quadruped robots. They are suited for tasks like geological exploration, outdoor monitoring, and disaster relief. Central Pattern Generators (CPG) enable hexapod robots to walk naturally without external control, but gait adaptability on complex terrains is limited. Deep Reinforcement Learning (DRL), though commonly used in robot motion control, faces more challenges for hexapod robots due to the high-dimensional continuous control problem and their complex dynamics, especially in complex terrains.

This paper presents a novel method that employs DRL to enhance the gait output of CPG, thereby enhancing the terrain adaptability of hexapod robots. Experiments are carried out on three challenging terrains: rough terrains, slopes, and stairs. The results indicate that the presented method provides better learning efficiency and effectiveness than cases when DRL is used alone. It also shows that the presented method allows hexapod robots to traverse complex terrains with enhanced flexibility and efficacy in comparison with cases when only CPG is used.

Keywords: Bioinspired robot learning, Hexapod robot, Reinforcement learning, Central pattern generator

1. Introduction

In recent years, the field of multi-legged robotics has attracted increasing attention from researchers[1][2]. Distinct from traditional wheeled and tracked robots, multi-legged robots demonstrate exceptional adaptability and capability in traversing a variety of challenging terrains [3][4]. Among the diverse legged robots, including biped robots[5], quadruped robots[6] and hexapod robots[7], hexapod robots are particularly notable. They are characterized by multiple redundant degrees of freedom and a rich array of locomotion gaits, which endow them with enhanced fault tolerance and superior stability. These attributes make them highly suitable for critical applications in the fields of disaster relief, material transportation, and military reconnaissance[8][9]. Current research in motion control for hexapod robots predominantly focuses on three areas: biologically-inspired methodologies[10][11], model-based motion control optimization[12][13], and machine learning approaches[14, 15].

Central Pattern Generator (CPG) is a prime example of biomimetic approaches. It is biological neural circuits that can generate stable rhythmic signals without external feedback[16]. Unlike conventional control methods, CPG eliminates the need for precise environmental modeling. Their control systems are adept at producing stable rhythmic signals without reliance on advanced signal inputs or external feedback. CPG is crucial for controlling rhythmic biological motions. These models are capable of autonomously generating rhythmic movements without sensory feedback and are distinguished by their robust adaptability, adjustability, and stability. CPG models are generally classified into two primary categories. The first category encompasses neuron models, such as the Kimura neural model[17] and the Matsuoka neural oscillator[18]. These models are designed to closely mimic biological neural systems, but they often require a lot of control parameters and have a complex structural composition. The second category comprises nonlinear oscillator models like the Hopf oscillator[19] and the Kuramoto phase oscillator[20]. Models in this category are typically more straightforward, featuring more manageable parameters and a simpler overall structure. Initially, CPG was applied to controlling the locomotion of aquatic robots, notably fish [21], and later to amphibious robots[22]. CPG's technology has since been applied to the motion control of legged robots, allowing them to generate natural, biologically inspired gaits derived from low-dimensional control signals[23]. However, the limited adaptability of CPG to varying environmental conditions has restricted their broader implementation in robotic locomotion. To address this limitation, an increasing number of researchers are incorporating structural design modifications[24][25] and integrating sensory feedback mechanisms[26][27] to facilitate dynamic movement across rough terrains. Nonetheless, these adaptations have not

[*] Corresponding author
Email addresses: 22121292@bjtu.edu.cn (Shichang Huang), mhzheng@bjtu.edu.cn (Minhua Zheng), 22121294@bjtu.edu.cn (Zhongyu Hu), xpliu@sce.carleton.ca (Peter Xiaoping Liu)

yet achieved optimal efficacy.

Deep Reinforcement Learning (DRL) stands as a prominent method within the realm of machine learning, showcasing remarkable proficiency in navigating complex terrains and adapting to external disturbances[28][29]. Despite its proven effectiveness as an end-to-end control strategy, DRL faces significant obstacles. It demands direct learning of efficacious policies from environmental and self-states, requiring meticulous parameter tuning, sophisticated reward function design, and extensive data collection. The complexity escalates in high-dimensional action and observation spaces, where the exploration difficulty during training intensifies. This complexity poses substantial challenges for algorithm convergence, especially in hexapod robots endowed with numerous degrees of freedom. To address these challenges, researchers have explored various strategies, such as hierarchical reinforcement learning [30][15], imitation learning[31], and goal-based approaches [32], to enhance learning efficiency and achieve improved outcomes. In addition to this, researchers have made advances in training methodologies. [33] implements a student-teacher architecture to enable precise motion control of robots under extreme conditions. [34] adopts the "Multiplicity of Behavior" (MoB) approach, training a single policy capable of adapting to various tasks and environments, thus facilitating motion control of quadruped robots across diverse unknown terrains. While these methods significantly enhance the adaptability and precision of robots, their training processes are complex, encompassing multiple stages and requiring extended durations. Furthermore, [34] necessitates manual intervention to fine-tune policy parameters for specific real-world applications.

A recent innovation involves the combination DRL and CPG, leading to the development of the CPG-RL approach[35][36][37], which improves the algorithm in reinforcement learning. This combined approach utilizes DRL to optimize CPG parameters, thereby equipping the CPG framework to adapt to more complex environmental conditions. Nevertheless, the terrain adaptability of these approaches largely depends on the inherent generalization capabilities of both DRL and CPG, with most studies being conducted on relatively simple, regular terrains. While CPG demonstrates prowess in simpler terrains, its performance wanes on more challenging terrains. Simultaneously, applying DRL directly to control hexapod robot joint angles introduces significant challenges, involving complex parameter learning and the design regular walking patterns. Considering these challenges, this paper introduces a novel motion control methodology that leverages the strengths of both DRL and CPG, aiming to remedy the limitations inherent in each approach. Unlike optimizing the parameters of the CPG as previous methods do. This strategy allows for more precise control over the robot's movements, enhancing both motion accuracy and efficiency while also improving adaptability to various terrains and environmental conditions. Optimizing joint signals directly also simplifies the learning process, reducing the complexity and time required for training, thereby boosting the overall performance of the robot and ensuring it operates more effectively and stably in complex environments. The incorporation of CPG notably improves the exploration efficiency and training speed of DRL, while DRL enhances the robot's adaptability to diverse terrains. The following outline highlights the primary contributions of this study:

1. A novel hierarchical motion control method is introduced for hexapod robots, which includes a High-Level Terrain Analysis (HLTA) layer and a Low-Level Action Generation (LLAG) layer. The HLTA layer modularizes complex terrains, separating terrain perception from motion control, which effectively minimizes the observational space dimension required for the LLAG layer. This hierarchical approach allows the HLTA layer to precisely guide the LLAG layer, thereby enhancing the accuracy of motion control as well as the efficiency and adaptability of the overall system.
2. The LLAG layer optimizes the rhythmic gaits generated by CPG with DRL. Reward functions are used for different terrains, and curriculum learning is adopted to improve training efficiency. Under identical conditions, this method substantially outperforms DRL alone in terms of learning effectiveness and speed, and outperforms CPG alone in terms of locomotion speed and adaptability to difficult terrains.

Experiments conducted in both simulation and real environments validate the effectiveness of the proposed method. It enables the hexapod robot to traverse on rough terrains with a height difference of 13cm in simulation and 12cm (equal to the height of the center of mass) in real environments, slopes of 18° in simulation and 16° in real environments, as well as stairs of 8cm in both simulation and real environments.

The subsequent sections of this paper are as follow: Section II introduces the robotic platform and provides an overview of the CPG and DRL. Section III explores the integration of DRL with the CPG framework, detailing the design of Markov decision processes and reward functions. Section IV presents experimental results, and section V concludes with a summary of the current work.

2. Preliminaries and problem description

2.1. Hexapod Robot

In this study, we utilize a hexapod robot as depicted in the 'robot' part of Fig. 2 . This robot has six legs: the right front leg (RF), right middle leg (RM), right back leg (RB), left back leg (LB), left middle leg (LM), and left front leg (LF). Each leg of the robot is outfitted with three joints: the hip, knee, and ankle joint, which endow the robot with a total of 18 degrees of freedom, enabling a range of locomotive capabilities.

2.2. CPG Network

Compared to other nonlinear oscillators, the parameters of Hopf oscillator are independent, and it is easy to control output amplitude and frequency, so we employ the Hopf oscillator model in this study. The Hopf oscillator is defined by the following equations:

$$\begin{cases} \dot{x} = \alpha\left(\mu^2 - x^2 - y^2\right)x - \omega y \\ \dot{y} = \alpha\left(\mu^2 - x^2 - y^2\right)y + \omega x \end{cases} \tag{1}$$

where x and y represent the oscillator's output values; α is coefficients dictating the rate of convergence (set to 10 in this study); μ denotes the square of the output signal's amplitude (set to 1 in this study), and ω is the frequency of the oscillator (set to π in this study).

In modeling the gait of a hexapod robot, an enhancement to equation (1) introduces the swing phase frequency ω_{swing} and the stance phase frequency ω_{stance}. These frequencies specifically govern the durations of the swing and stance phases in the robot's gait. Additionally, β is introduced to adjust the value of ω, as shown in (2):

$$\begin{cases} \omega = \frac{\omega_{\text{stance}}}{e^{-by}+1} + \frac{\omega_{\text{swing}}}{e^{by}+1} \\ \omega_{\text{stance}} = \frac{1-\beta}{\beta}\omega_{\text{swing}} \end{cases} \tag{2}$$

where β is the load factor, which is instrumental in determining the proportional relationship between the swing and stance phases (set to 0.5 in this study). ω_{stance} and ω_{swing} represent the stance phase frequency and the swing phase frequency, respectively. The parameter b is utilized to determine the transition ratio between ω_{stance} and ω_{swing} (set to 100 in this study).

Simultaneously, to generate rhythmic gaits, a hierarchical CPG network framework is adopted, as illustrated in the 'Hierarchical structure of CPG' part of Fig. 2. The upper layer employs a circular network topology, with each leg controlled by an individual Hopf oscillator unit. The entire robot uses six oscillator units, forming a coupled network that generates rhythmic gaits, defined by the equation:

$$\begin{cases} \dot{x}_i = \alpha(\mu - x_i^2 - y_i^2)x_i - \omega_i y_i, \\ \dot{y}_i = \alpha(\mu - x_i^2 - y_i^2)y_i + \omega_i x_i + \lambda\Delta_{ji}, \\ \omega_i = \frac{\omega_{\text{stance}}}{e^{-by_i}+1} + \frac{\omega_{\text{swing}}}{e^{by_i}+1}, \\ \omega_{\text{stance}} = \frac{1-\beta}{\beta}\omega_{\text{swing}}, \\ \Delta_{ji} = \sum_{j=1}^{n}(y_j\cos\theta_{ji} - x_j\sin\theta_{ji}) \end{cases} \tag{3}$$

where λ represents the coupling strength coefficient between oscillators (set to 0.1 in this study). The role of the coupling term Δ_{ji} is to synchronize the movement among different joints or parts, ensuring the continuity and coordination of the gait. This is particularly important in simulating natural movements. θ_{ji} indicates the phase difference between oscillators, with the phase differences in this study aligning with the triangular gait pattern.

The lower layer of the CPG network maps CPG output signals to joint angles. the y signal output of the CPG is used as the input of hip joint angle, which is then mapped to the knee and ankle joint angle through the mapping function we design. This mapping ensures that the movements of the joints of the hexapod robot are consistent with those of hexapods, allowing for a coordinated and efficient locomotion of the hexapod robot, the relationship of ϕ_1, ϕ_2 and ϕ_3 is showed in Fig. 1. The mapping function is as follows:

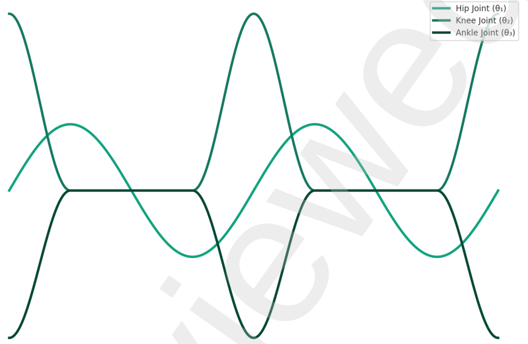


Figure 1: The relationship of ϕ_1, ϕ_2 and ϕ_3 in triangular gait pattern

$$\boldsymbol{\phi}(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \phi_3(t) \end{bmatrix} = \begin{bmatrix} A_1 y(t) \\ \begin{cases} A_2\left(1 - |y(t)|^2\right), & \text{if } \dot{y}(t) \geq 0 \\ 0, & \text{if } \dot{y}(t) < 0 \end{cases} \\ -k_1\phi_2(t) \end{bmatrix} \tag{4}$$

where $\phi_1(t)$, $\phi_2(t)$ and $\phi_3(t)$ are the output angles of the hip, knee joints and ankle joint, respectively, A_1, A_2 and A_3 represent the amplitudes of motion for the hip, knee, and ankle joints, respectively. To ensure each joint's motion falls within an effective yet safe range for locomotion, they are set as follows: $A_1 = 0.45$ for the hip joint, $A_2 = 1.2$ for the knee joint, and $A_3 = 1$ for the ankle joint. The coefficient $k_1 = A_3 \,/\, A_2$ represents the ratio of the amplitudes between the ankle and knee joints, dictating the relationship between the movements of these two joints. All CPG's parameters are obtained through trials to achieve the most suitable results for our robot.

2.3. Reinforcement learning

In exploring robotic locomotion, this study models the learning behavior of a hexapod robot through a Markov Decision Process (MDP). This model is represented as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma\}$. At a discrete time instance t, the robot gleans an observation $s_t \in \mathcal{S}$, executes an action $a_t \in \mathcal{A}$, and subsequently receives a reward $r_t = \mathcal{R}(s_t, a_t, s_{t+1}) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. This action induces a transition to the subsequent state s_{t+1}, governed by the transition probability function $\mathcal{T}(s_{t+1}|s_t, a_t)$. The expected cumulative return from these interactions is calculated as $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$, with $\gamma \in (0, 1)$ representing the discount factor. The primary goal in DRL, within this framework, is to devise a policy that optimizes this cumulative reward.

Here, the agent's actions are determined by the policy network $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$, where θ denotes the neural network weights. The objective is to fine-tune these policy parameters to maximize the accumulated reward over a defined time period T, formalized by the objective function:

$$\theta' = \arg\max_{\theta} \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right], \quad (5)$$

where θ' represents the optimal policy parameters. This approach can address the learning challenges faced by the hexapod robot in complex environments.

3. Methodology

This section presents our methodology of implementing motion control of the hexapod robot. The overall framework, as illustrated in Fig. 2, consists of two parts: the HLTA and the LLAG. In the HLTA, the hexapod robot captures images of the current terrain through its camera and uses a classifier based on LeNet to determine the terrain type such as rough terrains, slopes or stairs. Based on the terrain type detected, the corresponding LLAG strategy is adopted. The LLAG includes the DRL optimization module and the CPG gait generation module. The CPG controller generates rhythmic gaits while DRL produces actions to optimize these gaits based on the robot's current states. The combined action from the two modules is then deployed and executed on the robot.

3.1. Terrain Classifier

Training robots in simulation environments with DRL to navigate complex terrains is a challenging and time-consuming task under limited equipment conditions. To address this challenge, we adopt a modular approach, segmenting complex terrains into three primary categories: rough terrains, slopes, and stairs. The terrain classifier employs a LeNet network, incorporating a self-attention network to improve terrain recognition accuracy in images, the structure of network is show in the 'Terrain Classifier' part of Fig. 2. The training dataset, comprising 600 images (200 for each terrain type), is collected from both real and simulated environments at a 4:1 ratio. 80% of the data is used for training, the other 20% is used for test. After training, the terrain classifier is able to recognize the terrains with 97.5% accuracy. Through the guidance of the terrain classifier, hexapod robot adopts corresponding movement strategies.

3.2. Action Space

As the CPG generates periodic rhythmic signals, the robot might struggle to respond accurately in complex terrains. To enhance adaptability in such terrains, we optimize the CPG-generated gait using DRL. The action generated by DRL is thus utilized to adjust and optimize joint angles generated by the CPG, enabling the robot to more effectively adapt to various terrains. The expression is as follows:

$$a_t = a_c + \epsilon a_r \quad (6)$$

where a_t represents the final joint position transmitted to the robot. a_c is the joint position information outputted by the CPG, and a_r is the joint position outputted by the DRL. Additionally, ϵ is a modulation factor that balances the contribution of the CPG output and DRL output in the final joint position.

To accurately determine the optimal value of ϵ for each type of terrain, this study implement a detailed parameter search method. We set each terrain type to its highest level of complexity and conducted 100 trials for each, with each trial consisting of 100,000 steps to ensure a comprehensive exploration of the parameter space. During each set of trials, the modulation factor ϵ is incrementally increased within the range[0,1], which increases by 0.1 every 10 experiments, until reaching 1. The optimal value of ϵ for each terrain is selected based on the average of the rewards obtained over these 10 trials. Experimental results indicate significant variations in the optimal values of ϵ for different types of terrain: 0.6 for rough terrains, 0.3 for stairs, and 0.4 for slopes. These differences reflect the impact of terrain complexity on control strategies: rough terrain, due to its irregularity, relies more on DRL to adjust the gait of CPG; by contrast, slopes and stairs, being more regular, depend more on the angular outputs from the CPG. These terrain-specific ϵ values have been integrated into our system, allowing the robot to dynamically adjust its behavioral strategy based on the specific characteristics of the terrain, thereby significantly enhancing the flexibility and robustness of our adaptive locomotion strategy.

3.3. Observation Space

High-dimensional continuous state is one of the major practical issues in DRL[38]. To address this issue, our study adopts a method of situating the robot in specific environments, such as rough terrains, slopes, and stairs, for policy training. In these distinct environments, the robot's movements rely solely on its intrinsic information, independent of external sensors like vision or radar. This approach significantly reduces the dimensionality of the observation spaces required in DRL process. The observation spaces considered in this study includes only the robot's joint position p_i, velocity v_i, and torque M_i (for $i = 1, ..., 18$), as well as its world coordinates (x_r, y_r, z_r), orientation (r_r, p_r, y_r), and target endpoint (x_o, y_o). Position information of the physical robot is obtained through the IMU. Furthermore, to enhance the robustness, noise ξ is introduced into the observation spaces. This simplification strategy greatly improves training efficiency and effectively reduces reliance on high-dimensional observation spaces, allowing reinforcement learning to more effectively adapt to various challenging terrain environments.

3.4. Reward Function

Different reward functions are utilized for various terrains to adapt to the distinct characteristics and challenges of each terrain type. By customizing the reward functions for the unique conditions and requirements of each terrain, we are able to let the algorithm's optimization more precisely, enhancing the robot's performance across various terrains. The selection of these reward functions considered not only the physical properties of the terrain but also the complexity of tasks and diversity of objectives, ensuring the efficient and stable operation of the

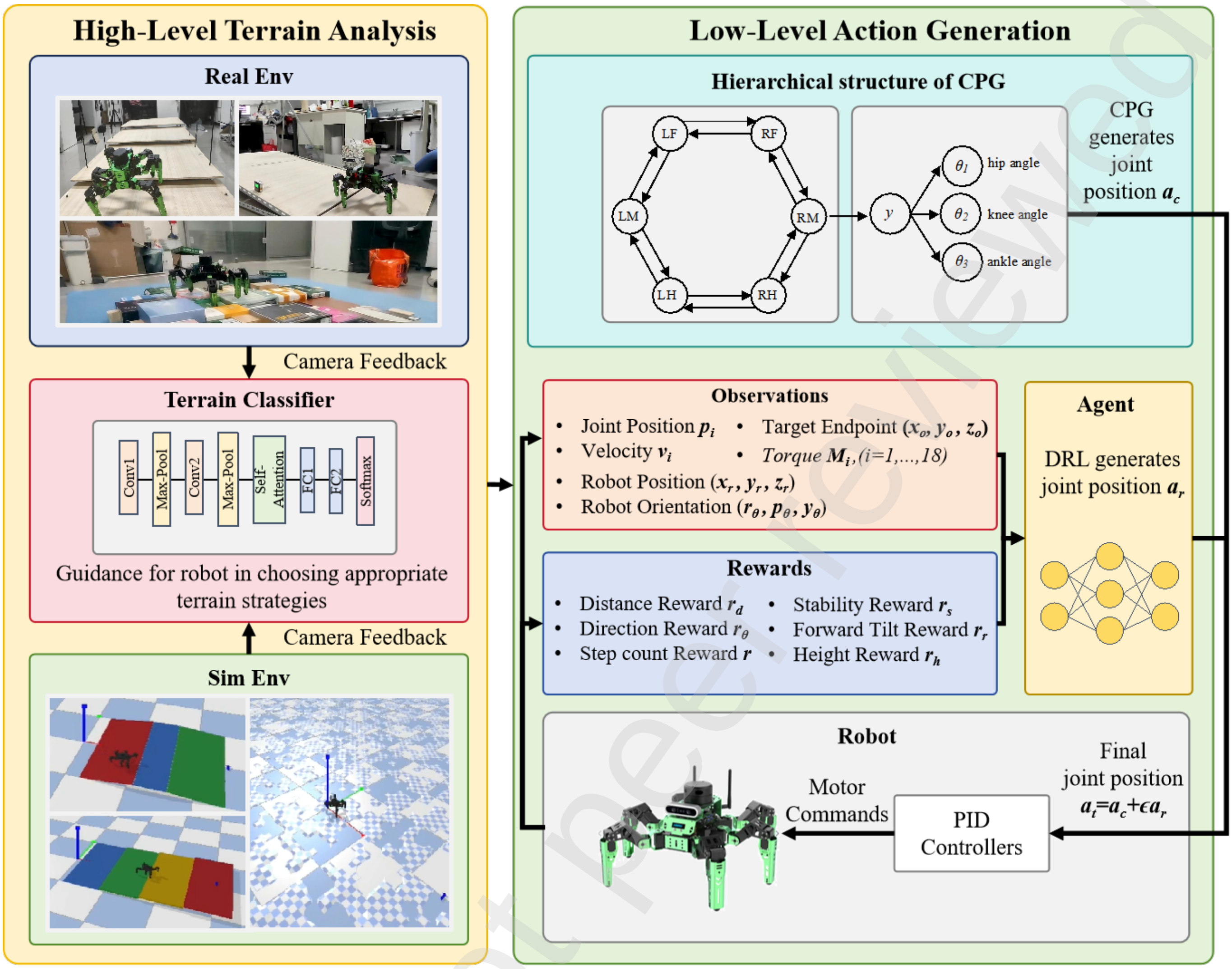

High-Level Terrain Analysis
Real Env
Camera Feedback
Terrain Classifier
Conv1
Max-Pool
Conv2
Max-Pool
Self-Attention
FC1
FC2
Softmax
Guidance for robot in choosing appropriate terrain strategies
Camera Feedback
Sim Env
Low-Level Action Generation
Hierarchical structure of CPG
LF
RF
LM
RM
LH
RH
y
θ_1 hip angle
θ_2 knee angle
θ_3 ankle angle
CPG generates joint position a_c
Observations
Joint Position p_i
Velocity v_i
Robot Position (x_r, y_r, z_r)
Robot Orientation $(r_\theta, p_\theta, y_\theta)$
Target Endpoint (x_o, y_o, z_o)
Torque $M_i, (i=1,...,18)$
Agent
DRL generates joint position a_r
Rewards
Distance Reward r_d
Direction Reward r_θ
Step count Reward r
Stability Reward r_s
Forward Tilt Reward r_r
Height Reward r_h
Robot
Motor Commands
PID Controllers
Final joint position $a_t = a_c + \epsilon a_r$

Figure 2: Framework of the hexapod robot's terrain adaptation strategy. The left part shows the HLTA layer, where the robot captures images of current terrain and identifies the terrain type using a LeNet-based classifier. The right part shows the LLAG layer, which consists of two modules. The upper is a hierarchical CPG module for generating rhythmic gaits a_c, and the lower is the DRL module that generates actions a_r to optimize the rhythmic gaits based on current states. a_t is the optimized actions published on the robot.

intelligent system in diverse environments. Concrete reward functions are shown in Table I and total reward function is as follows:

$$R_{\rm rt} = w_1 \cdot r_d + w_2 \cdot r_\theta + w_3 \cdot r_s + w_4 \cdot r_r + w_5 \cdot r_h + w_6 \cdot r \quad (7)$$

where r_d is the distance reward, reflecting the proximity of the robot to the target point, with positive rewards when the distance d is less than 1 m. r_θ is the directional reward, related to the alignment of the robot with the target direction, and it becomes positive when the angular difference θ is less than $\frac{\pi}{6}$. r_s is the stability reward, which becomes negative when the robot falls. r_r is the forward tilt reward, encouraging the robot to maintain a proper forward posture during climbing, The reward is related to the alignment of the tilt angle ψ with the slope angle ψ_s, κ is set to 5°, when the alignment exceeds κ, the reward becomes negative. r_h is the height reward, encouraging the robot to reach a certain height when climbing stairs, with rewards given when the robot's height z exceeds 12 cm. r is the step count reward, which becomes negative when the exploration steps exceed 4000.The weights of the rewards are denoted by w_1, w_2, w_3, w_4, w_5 and w_6. Otherwise, if the reward term meets the above reward conditions, rewards or penalties are applied; if not met, it is zero.

Beside, we employe curriculum learning in this paper, and a difficulty factor is integrated into the reward function. the equation is as follows:

$$R'_{\rm rt} = (1 + \delta) \cdot R_{\rm rt} \quad (8)$$

where $R'_{\rm rt}$ is the total reward in different difficult terrain. δ is defined as a difficulty factor that increases as the complexity of the terrain increases. Specifically, in rough terrains, complexity is reflected by the height difference of the terrain, and δ increases by 0.1 for each 0.01m increase in height difference. In slopes, complexity is reflected by the angle of the slope, and δ increases

Table 1
Reward Functions for Three Terrains

Reward Types	Reward Functions	Conditions	Weights		
			Rough terrain	Slopes	Stairs
r_d	$\frac{1}{d} - 1$	None	3	2	1
r_θ	$1 - \frac{6\theta}{\pi}$	$\theta \leq \frac{\pi}{6}$	1	0.2	0.2
r_s	-100	Robot falls	0.1	0.1	0.2
r_r	$1 - \frac{\lvert\psi - \psi_s\rvert}{\kappa}$	None	0	0.01	0
r_h	$z - 12$	$z > 12cm$	0	0	0.01
r	-10	$n > 4000$	1	1	1

by 0.1 for each 3° increase in the angle of slope. In stairs, complexity is measured by the stairs' height, and δ increases by 0.1 for each $0.005m$ increase in the height of stairs.

4. experiments and results

In this section, we present the training details and validate the experimental results of the proposed terrain adaptation method. The aspect to be validated is the ability of the hexapod robot to achieve self-adaptation. Therefore, we validate four aspects including validation of locomotion effects, validation of learning outcomes, comparison of motion effects and experiments with real robots.

4.1. Training Details

We first validated our approach using a hexapod robot in the PyBullet simulation environment [39] before implementing it on a real robot. 60 parallel training environment was used in this study for simulations since PyBullet enables parallel training of simulations. The algorithm used to train the robots is Proximal Policy Optimization (PPO)[40], the relevant hyperparameters of which are shown in Table II, the robot we used is the Jethexa robot whose body parameters are shown in Table III.

Table 2
PPO Hyperparameters

Hyperparameters	Value
Learning Rate	1×10^{-4}
Batch Size	256
Number of Epochs	4×10^{4}
Gamma	0.99
GAE Lambda	0.95
Clip Range	0.2
Network Architecture	[256, 256, 256, 256]
Activation Function	ReLU

Besides, we adopt the curriculum learning approach in three different modular terrains to assist training and improve efficiency. In the curriculum learning structure, we created different environments and varying difficulty levels within the same environment. Training terrains are as shown in 'Simulation Environment' part of Fig. 2. The types of terrains and difficulty levels are as follows:

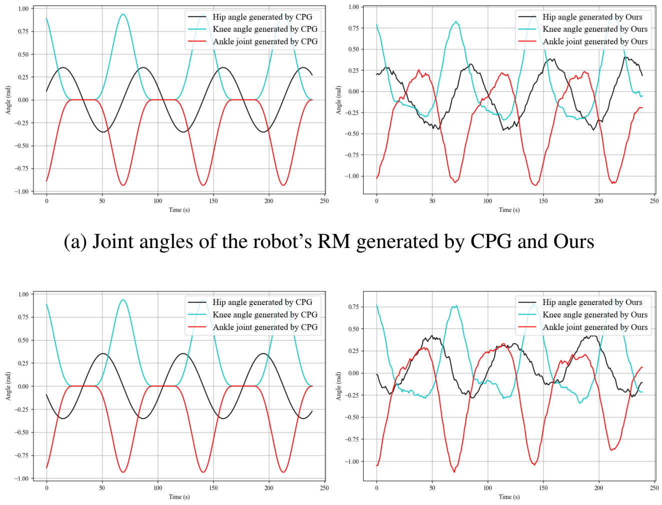
Table 3
Body Parameters of Jethexa

Parameter	Value
Coxa Length	5cm
Femur Length	8cm
Tibia Length	13cm
Body Length	39.7cm
Body Width	42.6cm
Height of Center of Mass	12cm
Mass	2.5kg

- **Rough terrain**: Rough terrains begins with a randomly generated rough terrain featuring a $0.05m$ initial height difference and the target is to move forward $2.5m$. For every 20 successful arrivals at the target, the difficulty level of the terrain will be increased, raising the height difference by $0.01m$, up to $0.13m$ maximum.
- **Slopes**: Slopes include an ascent, flat area, and descent, starting with a 5° slope and the target is to move forward $2.5m$. For every 20 successful arrivals at the target, the difficulty level of the terrain will be increased, raising the slope by 3°, up to 18° maximum.
- **Stairs**: Stairs include four steps, each initially $0.05m$ high and $0.4m$ wide and the target is to move forward $2.5m$. For every 20 successful arrivals at the target, the difficulty level of the terrain will be increased, raising step's height by $0.005m$, up to $0.08m$ maximum.

4.2. Validation of Motion Effects

In this experiment, we show the joint angles of the robot's RM and LM when the robots are let move using CPG and our method in rough terrain (height difference of $0.06m$) in Fig. 3. It can be seen that the CPG outputs periodic joint signals. DRL modifications enhance these signals, allowing the CPG signals, while maintaining a certain periodicity, to be either enhanced or attenuated, which make the robot adapt to rough terrains. This demonstrates the hexapod robot's improved adaptability through the combined use of CPG and DRL.



(a) Joint angles of the robot's RM generated by CPG and Ours

(b) Joint angles of the robot's LM generated by CPG and Ours

Figure 3: Optimization process of the joint angles. Fig.(a) displays the optimization process of the joint angles for the RM of the robot, while Fig.(b) shows the same for the LM. The left section illustrates the joint angles generated by the CPG, and the right section shows the final joint angles optimized by DRL that are processed with noise reduction

4.3. Validation of Learning Outcomes

To evaluate the impact of integrating CPG with the DRL framework, we compare the reward curves of pure DRL (TD3, SAC, PPO), hierarchical reinforcement learning (HRL), and methods of combining reinforcement learning with other reinforcement learning methods(TD3-CPG, SAC-CPG) and our method(PPO-CPG) on three different terrains, as shown in Fig. 4. All methods employed the same hyperparameters and reward functions, and as for HRL, DRL is utilized to learn the parameters of CPG, such as $\omega_{\text{swing}}\in[pi/2, \pi]$, $A_1\in$[-1, 1], $A_2\in$[-0.5, 1], $A_3\in$[-1, 0.5]. The results indicate that our methods significantly outperform both pure DRL and HRL across all tested terrains, and PPO-CPG is best. Specifically, HRL showed intermediate performance, while pure DRL performed the worst. This is primarily because our approach directly optimizes the joint angles generated by the CPG, rather than optimizing the CPG parameters and then applying the output of CPG to the robot as HRL does. Our direct optimization strategy proves to be more efficient. Furthermore, pure DRL failed to effectively train the robot's locomotion gaits.

4.4. Comparison of Motion Effects

4.4.1. Verification of Motion Performance

In this experiment, we conduct 20 experiments each using the CPG, HRL and our method(PPO-CPG) to evaluate the stability of the robot's control over different terrains. We use three control methods to control the hexapod robot to move forward 2.5m across three terrains. Simultaneously, we evaluate the motion performance of the robot using its pitch angles, roll angles, and Y-Direction deviation. Here, we set the height difference of the rough terrain to 5cm, the inclination of the slope to 8°, and the step height of the stair to 4cm in order to get a clear results of the motion stability, and they are shown in Fig. 5.

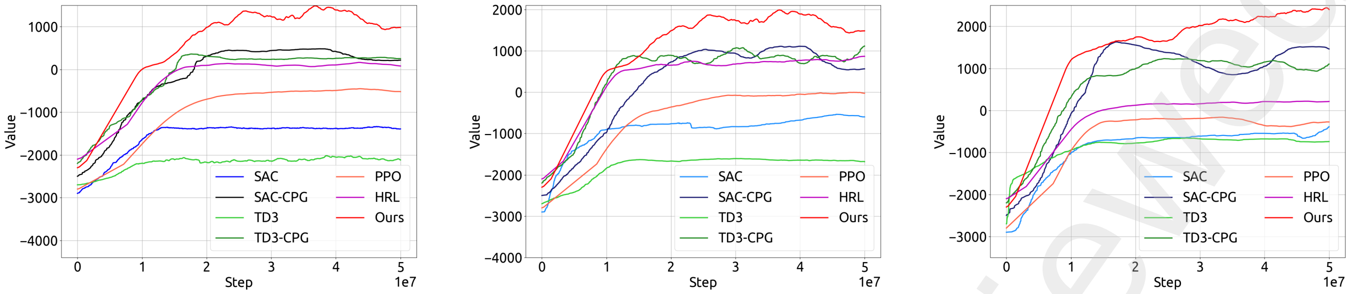
- **Rough Terrain:** According to Fig. 5, all methods can maintain stable roll and pitch angles in the rough terrains, proving that the robots' motion stability is consistent in rough terrain with height difference of 5cm. As for Y-Direction, the CPG method exhibits a greater deviation in the Y direction. Additionally, our method allows the robot to advance a distance of 2.5m in a much shorter period of time, thus demonstrating an improvement in robot motion velocity.
- **Slopes:** According to Fig. 6, all methods maintain a stable Y-Direction deviation and roll angle in slopes. However, our method produces a significantly larger roll angles which is intended to foster a forward leaning posture for better slope climbing. Besides, our method allows the robot to advance 2.5m faster, demonstrating faster velocity, which mirrors the performance in rough terrain.
- **Stairs:** According to Fig. 7, all methods achieve stable roll angles, pitch angles, and Y-Direction deviation. According to Fig. 5(c) , both methods achieve stable roll angle, pitch angle, and Y-Direction deviation. However, our method outperforms the CPG and HRL in terms of speed, again demonstrating that the robot moves faster.

Overall, Our method demonstrates significant advantages in stability across challenging terrains, as evidenced by the comparative results on rough terrain, slopes, and stairs. These enhancements in performance are attributed to the integration of DRL with CPG, where DRL optimizes the rhythmic gait patterns generated by CPG for complex environments. This synergy allows the robot to adjust its movements more precisely to the specifics of each terrain, enhancing both speed and stability.

4.4.2. Observing Motion Adaptability

To evaluate the robot's terrain adaptation capabilities, we compare the CPG with our method in terrains of increasing complexity, conducting 20 experiments for each terrain type of increasing complexity. In this experiment, we similarly have the robot advance 2.5m with a Y-Direction deviation limit of 0.8m. Simultaneously, the success rate of each experiment is recorded, where we consider advancing 2.5m as a success.

The results in Table IV show that our method has significant advantages in all terrain types. In rough terrains, the success rate of CPG starts to decrease when the height difference is 0.07m and drops to zero at 0.11m and the success rate of HRL starts to decrease when the height difference is 0.09m and drops to zero at 0.12m. Our method still has a success rate of 100% at a height difference of 0.11m m and can traverse terrains with a maximum height difference of 0.13m . In slopes, the success rate of the CPG begins to decrease at 10° and drops to zero at 13° and the success rate of the HRL begins to decrease at 11° and drops to zero at 14°. Our method can efficiently climb slopes up to 15° , with a maximum capacity of 18°. For stairs, the success rate of CPG begins to decrease at a step height of



(a) Rough terrains (b) Slopes (c) Stairs

Figure 4: Comparison of training reward curves between DRL Method and Ours Across Different Terrains. Fig. (a) shows the reward comparison curve on rough terrain; Fig. (b) shows the curve on slopes; Fig. (c) shows the curve on stairs.

Table 4
Comparison of Success Rates of CPG and Our Method in Three Terrains with Various Difficulty Levels

	Level(cm)	3	4	5	6	7	8	9	10	11	12
Rough Terrains	CPG	100%	100%	100%	100%	100%	75%	40%	10%	0%	0%
	HRL	100%	100%	100%	100%	100%	100%	100%	85%	60%	20%
	Ours	100%	100%	100%	100%	100%	100%	100%	100%	100%	70%
Slopes	Level(°)	6	7	8	9	10	11	12	13	14	15
	CPG	100%	100%	100%	100%	100%	80%	45%	0%	0%	0%
	HRL	100%	100%	100%	100%	100%	100%	60%	35%	0%	0%
	Ours	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Stairs	Level(cm)	2	3	4	5	5.5	6	6.5	7	7.5	8
	CPG	100%	100%	100%	100%	100%	35%	0%	0%	0%	0%
	HRL	100%	100%	100%	100%	100%	65%	25%	0%	0%	0%
	Ours	100%	100%	100%	100%	100%	100%	100%	100%	80%	65%

0.055*m* and drops to zero at 0.06*m* and the success rate of HRL begins to decrease at a step height of 0.055*m* and drops to zero at 0.065*m*. Our method has a 100% success rate until a height of 0.075*m* and is able to traverse steps as high as 0.08*m*. The success rate of CPG is also higher for stairs. Thus, the experimental results show that our method has a great advantage over CPG alone in terms of terrain adaptability.

Besides, rough terrains feature elements such as stairs and slopes, characterized by differences in height and angle of slopes. Therefore, we conduct adaptability tests using a rough terrain model on stairs and slopes, employing the same methods used in other terrain adaptability tests, and results are in V. This model is used to control robots moving over stairs and slopes while recording the success rate. Results show that this model exhibits good adaptability on slopes; However, performance drops sharply when the angle reaches 14°, and declines to zero at 15°. It also performs well on stair terrains, but begins to decrease in performance when the stair height exceeds 0.06m, dropping to zero at 0.07m. In conclusion, the rough terrain model performs well when the angle of slopes is less than 14° and the stair height is below 0.06m. Training specifically on slopes and rough terrains aims to enable robots to effectively complete tasks in more challenging terrains.

Table 5
Success Rates of Our Method in Slpoes and Stairs with Rough Terrain's model

Level(°)	10	11	12	13	14	15
Slopes	100%	100%	100%	100%	40%	0%
Level(cm)	5.5	6	6.5	7	7.5	8
Stairs	100%	70%	30%	0%	0%	0%

4.5. *Experiments with Real Robot*

In the real experiment, the trained model in the simulation environment is ported to the real robot to verify its performance on the real terrain. For this purpose, rough terrain, slopes and stairs are constructed as shown in the Fig. 6 and full video of the experiment is here [1]. Throughout the sim2real process we employ a master-slave architecture based on the ROS operating system. The slave machine handles the robot's sensor data and generates corresponding actions, while the master machine executes these actions and relays the sensor data back to the slave. This architecture not only optimizes distributed computing but also reduces the computational load on cost-effective robots, thereby enhancing the overall system flexibility.

[1] `https://www.bilibili.com/video/BV1dZ421j7PG/?spm_id_from=333.999.0.0`

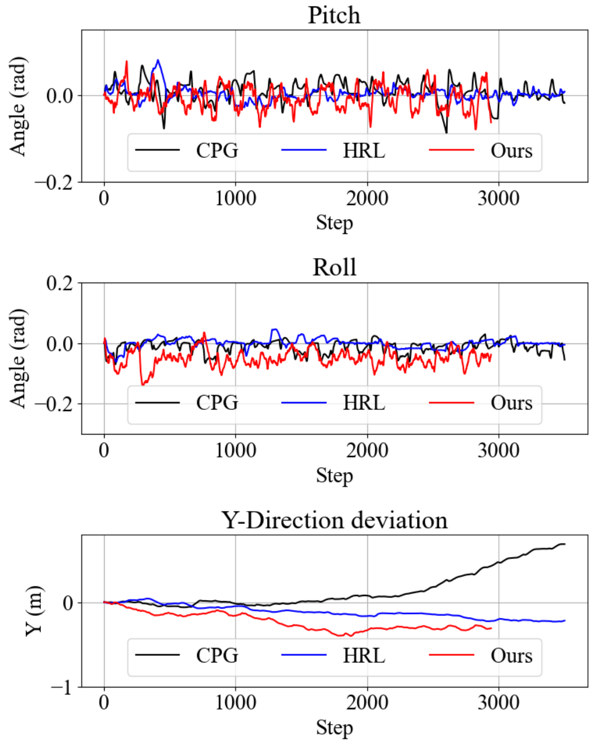


Figure 5: Robot stability over time across rough terrain: Roll, Pitch, and Y-Direction Deviation.

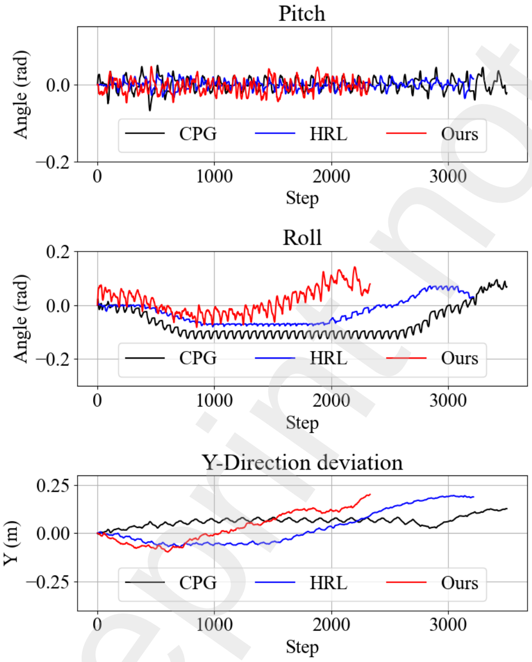


Figure 6: Robot stability over time across slopes: Roll, Pitch, and Y-Direction Deviation.

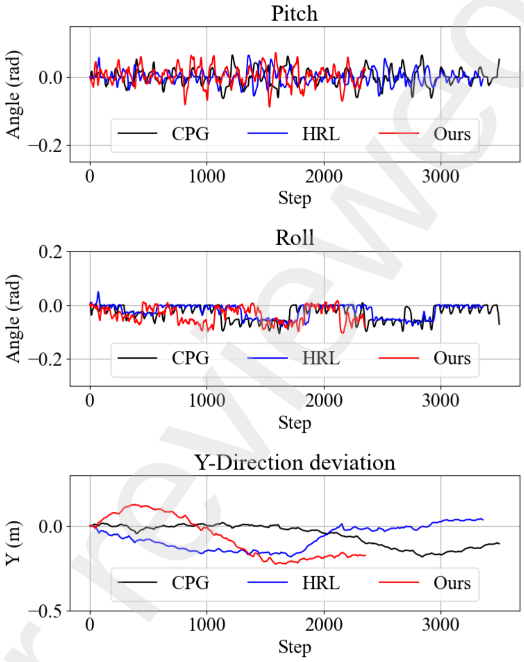
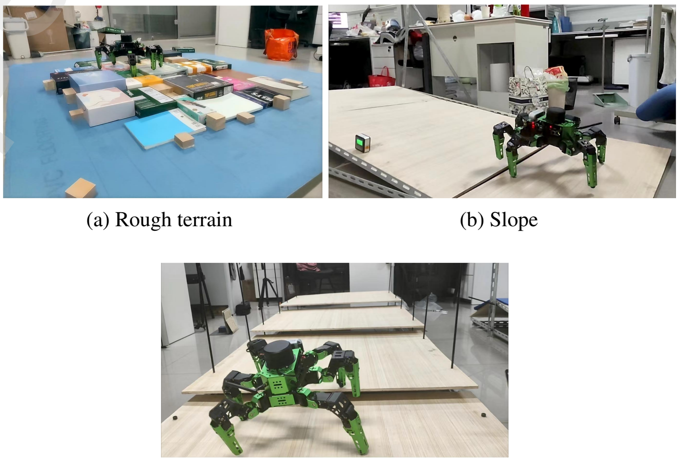


Figure 7: Robot stability over time across stairs: Roll, Pitch, and Y-Direction Deviation.



(a) Rough terrain

(b) Slope

(c) Stair

Figure 8: Locomotion of hexapod robots on three terrains. Fig. (a) shows the locomotion on rough terrain; Fig. (b) shows locomotion on slopes; Fig. (c) shows locomotion on stairs.

- **Rough Terrain:** Illustrated in Fig. 8(a), a $1.5m \times 1.5m$ area is populated with objects of varying heights to simulate rough terrain, with heights ranging from $0m$ to $0.12m$, an average height variation of about $0.06m$. The robot is tested from eight different directional starting positions, conducting two trials from each, for a total of 16 experiments. It successfully navigated the terrain in all trials.

- **Slopes:** Shown in Fig. 8(b), the experimental setup included a $2m$ long slope with an adjustable incline. The robot's performance on slope is tested 20 times, with results presented in Table VI.

- **Stairs:** As seen in Fig. 8(c), a 3.2m long stairs with four adjustable-height steps is used. The robot is tested on various step heights 20 times, with results presented in Table VI.

Table 6
Success Rates of Our Method on Real Robot in Slopes and Stairs with Various Difficulty Levels

Level(°)	8	10	13	14	15	16
Slopes	100%	100%	100%	100%	100%	80%
Level(cm)	4	5	6	7	7.5	8
Stairs	100%	100%	100%	100%	65%	50%

The experimental outcomes reveal that the robot's adaptability on rough terrains in real-world settings aligns closely with its simulated performance. In slopes, the robot manage to climb inclines of up to 18° in simulations, but in reality, it could handle slopes up to 16°, indicating a slight reduction in capability outside the simulated environment. For stairs, the robot ascended steps up to 0.08m high, mirroring its simulated capabilities, though with a reduced success rate in actual scenarios. Overall, the robot demonstrated its ability to perform tasks in real-world environments comparable to those in simulations, albeit with marginally decreased effectiveness.

5. Conclusion

This paper proposes a hierarchical structure approach to enable the hexapod robot to traverse complex terrains such as rough terrains, slopes, and stairs. The HLTA layer adopts a LeNet network and the self-attention mechanism to achieve terrain classification. The LLAG layer combines DRL with CPG, optimizes the rhythmic gaits generated by CPG through DRL to improve the adaptability to complex terrains, and introduces curriculum learning to improve training efficiency. Experimental results show that compared with DRL alone, the proposed method achieves higher learning efficiency, and can complete tasks that cannot be accomplished by DRL alone. Compared with CPG alone, the proposed method enables faster locomotion speed, better movement stability, and stronger adaptability to more difficult complex terrains. When applied to real robots, this method also achieves good motion control effects. Although our approach demonstrates good adaptability on complex terrain, the CPG parameters optimized by DRL are fixed, which limits the exploration capability in unpredictable dynamic environments. In the future, we plan to apply real-time sensor data such as cameras and radar to dynamically learn CPG parameters to improve the robot's perception and adaptation under unpredictable conditions.

References

[1] Y. Gao, B. Su, L. Jiang, F. Gao, Multi-legged robots: progress and challenges, National Science Review 10 (2023) nwac214.

[2] M. Naya-Varela, A. Faina, R. Duro, Engineering morphological development in a robotic bipedal walking problem: An empirical study, Neurocomputing 527 (2023) 83–99. URL: https://www.sciencedirect.com/science/article/pii/S0925231223000115. doi:https://doi.org/10.1016/j.neucom.2023.01.003.

[3] L. Yang, Y. Yin, F. Gao, Z. Wang, L. Wang, Q. Sun, H. Gao, Design and control of a novel six-legged skating robot with skateboards, IEEE/ASME Transactions on Mechatronics (2023).

[4] J. Li, C. Yang, L. Ding, B. You, W. Li, X. Zhang, H. Gao, Trilateral shared control of a dual-user haptic tele-training system for a hexapod robot with adaptive authority adjustment, IEEE Transactions on Automation Science and Engineering (2024) 1–16. doi:10.1109/TASE.2024.3410522.

[5] K. Lobos-Tsunekawa, F. Leiva, J. Ruiz-del Solar, Visual navigation for biped humanoid robots using deep reinforcement learning, IEEE Robotics and Automation Letters 3 (2018) 3247–3254.

[6] J. Sheng, Y. Chen, X. Fang, W. Zhang, R. Song, Y. Zheng, Y. Li, Bio-inspired rhythmic locomotion for quadruped robots, IEEE Robotics and Automation Letters 7 (2022) 6782–6789.

[7] F. Ma, W. Yan, L. Chen, R. Cui, Cpg-based motion planning of hybrid underwater hexapod robot for wall climbing and transition, IEEE Robotics and Automation Letters 7 (2022) 12299–12306.

[8] Z. Gao, Q. Shi, T. Fukuda, C. Li, Q. Huang, An overview of biomimetic robots with animal behaviors, Neurocomputing 332 (2019) 339–350.

[9] I. Kostavelis, A. Gasteratos, Robots in crisis management: A survey, in: Information Systems for Crisis Response and Management in Mediterranean Countries: 4th International Conference, ISCRAM-med 2017, Xanthi, Greece, October 18-20, 2017, Proceedings 4, Springer, 2017, pp. 43–56.

[10] P. Lopez-Osorio, A. Patiño-Saucedo, J. P. Dominguez-Morales, H. Rostro-Gonzalez, F. Perez-Peña, Neuromorphic adaptive spiking cpg towards bio-inspired locomotion, Neurocomputing 502 (2022) 57–70. URL: https://www.sciencedirect.com/science/article/pii/S0925231222008165. doi:https://doi.org/10.1016/j.neucom.2022.06.085.

[11] C. Bal, Neural coupled central pattern generator based smooth gait transition of a biomimetic hexapod robot, Neurocomputing 420 (2021) 210–226. URL: https://www.sciencedirect.com/science/article/pii/S0925231220313187. doi:https://doi.org/10.1016/j.neucom.2020.07.114.

[12] M. R. C. Qazani, H. Asadi, S. Khoo, S. Nahavandi, A linear time-varying model predictive control-based motion cueing algorithm for hexapod simulation-based motion platform, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (2019) 6096–6110.

[13] M. Bjelonic, N. Kottege, T. Homberger, P. Borges, P. Beckerle, M. Chli, Weaver: Hexapod robot for autonomous navigation on unstructured terrain, Journal of Field Robotics 35 (2018) 1063–1079.

[14] Z. Jin, A. Liu, W.-A. Zhang, L. Yu, C.-Y. Su, A learning based hierarchical control framework for human–robot collaboration, IEEE Transactions on Automation Science and Engineering 20 (2023) 506–517. doi:10.1109/TASE.2022.3161993.

[15] X. Wang, H. Fu, G. Deng, C. Liu, K. Tang, C. Chen, Hierarchical free gait motion planning for hexapod robots using deep reinforcement learning, IEEE Transactions on Industrial Informatics (2023).

[16] A. J. Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, Neural networks 21 (2008) 642–653.

[17] H. Kimura, Y. Fukuoka, A. H. Cohen, Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts, The International Journal of Robotics Research 26 (2007) 475–490.

[18] 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, The International Journal of Robotics Research 27 (2008) 213–228.

[19] L. Righetti, A. J. Ijspeert, Pattern generators with sensory feedback for the control of quadruped locomotion, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 819–824.

[20] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, R. Spigler, The kuramoto model: A simple paradigm for synchronization phenomena, Reviews of modern physics 77 (2005) 137.

[21] B. Cafer, O. K. Gonca, K. Deniz, A. Z. Hakan, A. Mustafa, Cpg-based autonomous swimming control for multi-tasks of a biomimetic robotic fish [j], Ocean Eng. 189 (2019) 106334.

[22] T. Matsuo, T. Yokoyama, D. Ueno, K. Ishii, Biomimetic motion control system based on a cpg for an amphibious multi-link mobile robot, Journal of Bionic Engineering 5 (2008) 91–97.

[23] Y. Zeng, J. Li, S. X. Yang, E. Ren, A bio-inspired control strategy for locomotion of a quadruped robot, Applied Sciences 8 (2018). URL: `https://www.mdpi.com/2076-3417/8/1/56`. doi:`10.3390/app8010056`.

[24] Q. Zhou, J. Xu, H. Fang, A cpg-based versatile control framework for metameric earthworm-like robotic locomotion, Advanced Science (2023) 2206336.

[25] M. Thor, P. Manoonpong, A fast online frequency adaptation mechanism for cpg-based robot motion control, IEEE Robotics and Automation Letters 4 (2019) 3324–3331. doi:`10.1109/LRA.2019.2926660`.

[26] H. Yu, H. Gao, Z. Deng, Enhancing adaptability with local reactive behaviors for hexapod walking robot via sensory feedback integrated central pattern generator, Robotics and Autonomous Systems 124 (2020) 103401.

[27] A. Borgmann, S. L. Hooper, A. Büschges, Sensory feedback induced by front-leg stepping entrains the activity of central pattern generators in caudal segments of the stick insect walking system, Journal of Neuroscience 29 (2009) 2972–2983.

[28] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Processing Magazine 34 (2017) 26–38.

[29] 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. doi:`10.1109/ROBIO49542.2019.8961764`.

[30] T. Zhang, S. Guo, T. Tan, X. Hu, F. Chen, Adjacency constraint for efficient hierarchical reinforcement learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2022) 4152–4166.

[31] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, S. Levine, Learning agile robotic locomotion skills by imitating animals, arXiv preprint arXiv:2004.00784 (2020).

[32] G. Xiang, J. Su, Task-oriented deep reinforcement learning for robotic skill acquisition and control, IEEE transactions on cybernetics 51 (2019) 1056–1069.

[33] C. Xuxin, S. Kexin, A. Ananye, P. Deepak, Extreme parkour with legged robots, ICRA 2024 abs/2309.14341 (2024).

[34] G. B. Margolis, P. Agrawal, Walk these ways: Tuning robot control for generalization with multiplicity of behavior, in: Conference on Robot Learning, PMLR, 2023, pp. 22–31.

[35] G. Bellegarda, A. Ijspeert, Cpg-rl: Learning central pattern generators for quadruped locomotion, IEEE Robotics and Automation Letters 7 (2022) 12547–12554.

[36] G. Li, A. Ijspeert, M. Hayashibe, Ai-cpg: Adaptive imitated central pattern generators for bipedal locomotion learned through reinforced reflex neural networks, IEEE Robotics and Automation Letters 9 (2024).

[37] B. Guillaume, S. Milad, I. Auke, Visual cpg-rl: Learning central pattern generators for visually-guided quadruped locomotion, ICRA 2024 (2024).

[38] A. K. Shakya, G. Pillai, S. Chakrabarty, Reinforcement learning algorithms: A brief survey, Expert Systems with Applications 231 (2023) 120495. URL: `https://www.sciencedirect.com/science/article/pii/S0957417423009971`. doi:`https://doi.org/10.1016/j.eswa.2023.120495`.

[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), IEEE, 2021, pp. 7512–7519.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).