frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 26 January 2021
doi: 10.3389/fnbot.2021.627157

Adaptive Locomotion Control of a Hexapod Robot via Bio-Inspired Learning

Wenjuan Ouyang[1], *Haozhen Chi*[1], *Jiangnan Pang*[1], *Wenyu Liang*[2] *and Qinyuan Ren*[1]*

[1] College of Control Science and Engineering, Zhejiang University, Hangzhou, China, [2] Department of Electrical and Computing Engineering, National University of Singapore, Singapore, Singapore

In this paper, an adaptive locomotion control approach for a hexapod robot is proposed. Inspired from biological neuro control systems, a 3D two-layer artificial center pattern generator (CPG) network is adopted to generate the locomotion of the robot. The first layer of the CPG is responsible for generating several basic locomotion patterns and the functional configuration of this layer is determined through kinematics analysis. The second layer of the CPG controls the limb behavior of the robot to adapt to environment change in a specific locomotion pattern. To enable the adaptability of the limb behavior controller, a reinforcement learning (RL)-based approach is employed to tune the CPG parameters. Owing to symmetrical structure of the robot, only two parameters need to be learned iteratively. Thus, the proposed approach can be used in practice. Finally, both simulations and experiments are conducted to verify the effectiveness of the proposed control approach.

Keywords: hexapod robot, two-layer CPG, reinforcement learning, adaptive control, bio-inspired

OPEN ACCESS

Edited by:
Zhan Li,
University of Electronic Science and Technology of China, China

Reviewed by:
Ning Tan,
Sun Yat-Sen University, China
Mingchuan Zhou,
Technical University of Munich, Germany

***Correspondence:**
Qinyuan Ren
latepat@gmail.com

Received: *08 November 2020*
Accepted: *04 January 2020*
Published: *26 January 2021*

Citation:
Ouyang W, Chi H, Pang J, Liang W and Ren Q (2021) Adaptive Locomotion Control of a Hexapod Robot via Bio-Inspired Learning. Front. Neurorobot. 15:627157. doi: 10.3389/fnbot.2021.627157

1. INTRODUCTION

In the past decades, a big step has been taken toward the study of legged robots, such as the study of biped robot (Kim et al., 2020), quadruped robot (Hyun et al., 2014), hexapod robot (Yu et al., 2016), octopod robot (Grzelczyk et al., 2018), and etc. Most of these legged robots have exhibited astonish maneuverabilities in a typically structured environment. Among these legged robots, the hexapod robots have been increasingly attracting attention from scientists and a lot of hexapod robotic prototypes have been successfully developed (Stelzer et al., 2012; Li et al., 2019; Sartoretti et al., 2019; Lele et al., 2020; Zhao and Revzen, 2020). Even though these hexapod robots in shape look very much like the arthropod that the scientists are animating, such as ants or spiders, the robots developed hitherto are still pretty away from real arthropods. One of the main challenges lies in the difficulty of controlling the multi-legs of the robots with coordination to a complex dynamic environment.

To control the locomotion of hexapod robots, from a perspective of cybernetics, two methods are generally adopted, namely kinematics-based and bio-inspired. The former models the locomotion patterns via kinematics analysis. As pointed from the study of Ramdya et al. (2017), three basic locomotion patterns of Drosophila melanogaster have been extracted through biological study, namely tripod locomotion, quadruped locomotion, and five legs support locomotion. Based on the analysis of these three basic locomotion patterns, a foot-force distribution model is established for a hexapod robot walking on an unstructured terrain (Zhang et al., 2014). The study of Zarrouk and Fearing (2015) investigates the kinematics of a hexapod robot using only one

actuator and explores the turning issue of the robot. In the work of Sun et al. (2018), the inverse kinematics of an 18-degree-of-freedom (DoF) hexapod robot is calculated to control the dynamicly alternating tripod locomotion of the robot. Since it is hard to accurately model the kinematics of all the six-leg crawling modes, most obtained locomotion patterns from kinematics analysis are rough and trail-and-error strategy is usually necessary for tuning the rough patterns applied on the robots. The study from Delcomyn (1980) indicates that center pattern generators (CPGs), which are mainly located in the central nervous system of vertebrates or in relevant ganglia of invertebrates, are primarily responsible for generating coordinated, rhythmic locomotion patterns of animals in real time, such as crawling, flying, swimming, and running. Inspired by the characteristics of the stability and self-adaption of biological CPGs, artificial CPGs have been extensively studied, namely the bio-inspired approach, for locomotion generation of hexapod robots. The notable examples include the studies in Chung (2015), Zhong et al. (2018), Yu et al. (2020), and Bal (2021). Through these previous studies, it can be found out that the bio-inspired method can greatly simplify the locomotion control problem underlying coordination of multiple legs.

Although the bio-inspired method has been widely and fruitfully applied in locomotion control of many biomimetic robots, it still remains a challenge for modulating the CPG parameters to generate adaptive locomotion for hexapod robots. The CPG parameters in many studies are determined by experiences and some researchers adopt data-driven optimization methods, such as particle swarm optimization (PSO) method and reinforcement learning (RL), to tune the parameters. In the work of Juang et al. (2011), a symbiotic species-based PSO algorithm is proposed to automate the parameter design for evolving dynamic locomotion of a hexapod robot, but reducing the computing complexity of the PSO algorithm is still under research. In addition, the study of Kecskés et al. (2013) points out that PSO method easily suffers from the partial optimism and causes the loss of accuracy in a coordinate system. In locomotion control, there has been recent success in using RL to learn to walk for hexapod robots. In the work of Barfoot (2006), a cooperative Q-learning RL approach is utilized to experimentally learn locomotion for a 6-DoF hexapod robot, but this RL approach may be unable to deal with the hexapod robots that have higher DoF. The researchers in Sartoretti et al. (2019) employ A3C RL algorithm to learn hexapodal locomotion stochastically. Nevertheless, the performance of the learned controller proposed in the study is dependent on a large number of iterations. For the different terrains, the locomotion of a hexapod robot is controlled through training several artificial neural networks via RL method separately (Azayev and Zimmerman, 2020), but the training scenario is limited to the expert policies and thus the adaptivity of the controller may be inflexible for a dynamic environment.

In this paper, a bio-inspired learning approach is proposed for locomotion control of a hexapod robot with environment change. The proposed bio-inspired learning approach can be characterized by the structure of the learning mechanism. Biologists have proved the motor patterns of animals are controlled by neuro systems hierarchically (Fortuna et al., 2004) and functional configuration of CPGs can be regulated according to sensory feedback to produce different motor outputs (Hooper, 2000). Therefore, inspired from biological control systems, a two-layer CPG motion control system is firstly proposed in this paper to generate locomotion for the robot and then the parameters of the CPG tuning issue is explored to enhance the adaptability of the robot. In the proposed bioinspired control system, the outputs of the first layer of the CPG are

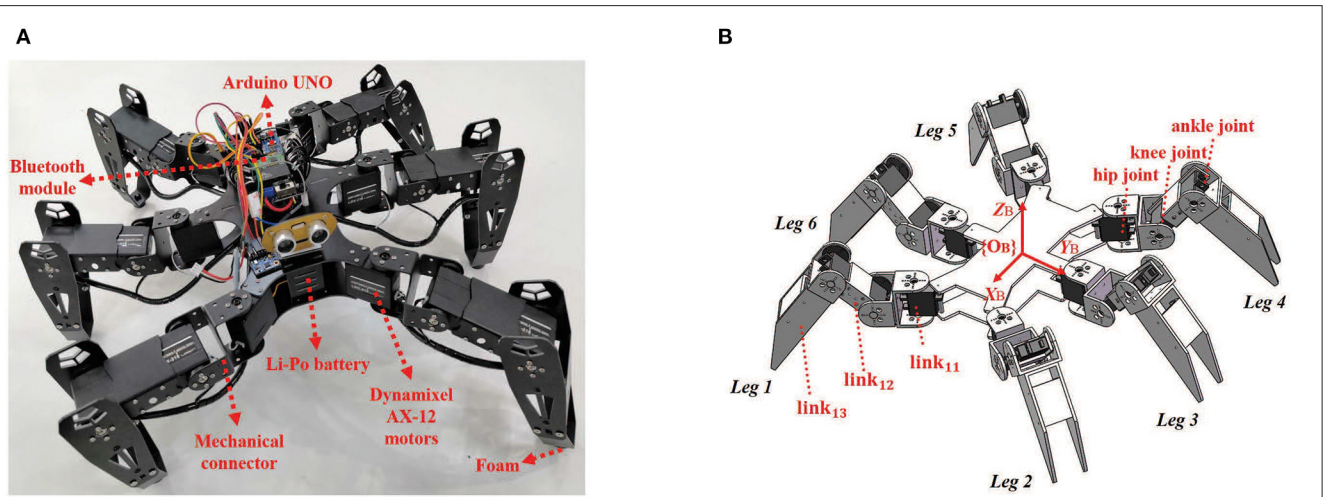

FIGURE 1 | (A) The hexapod robot prototype. **(B)** The mechanical schematic of the hexapod robot. $\{O_B\}$ is the body frame whose origin is fixed on the body's mass center. The robot possesses a rectangular body trunk and six mechanical legs numbered from leg 1 to leg 6. There are three joints and three links in each leg. The joints are named as hip joint, knee joint and ankle joint from the direction of body to foot tip. The legs can be labeled as $link_{i1}, link_{i2}, link_{i3}$ with the leg number $i = 1, 2, \cdots, 6$. For example, the $link_{11}$ (between hip joint and knee joint), $link_{12}$ (between knee joint and ankle joint) and $link_{13}$ (between ankle joint and foot tip) are the three links of leg 1.

TABLE 1 | Technical specifications of the prototype.

Parameter		Prototype	
		Value	Unit
Number of servo motor		18	\
Power supply		7.4	DC(V)
Total weight		1.995	kg
Body dimension	Length	24	cm
	Width	18.5	cm
	Height	4.5	cm
Limb $link_{i1}$	Weight	18.6	g
	Length	4.5	cm
Limb $link_{i2}$	Weight	128	g
	Length	7.5	cm
Limb $link_{i3}$	Weight	56.3	g
	Length	13.5	cm

Where $i = 1, 2, \cdots, 6$ is the number of six legs.

responsible for generating the basic locomotion patterns, such as tripod locomotion, quadruped locomotion, and five legs support locomotion. The second layer of the CPG acting as a Behavior Layer controls the limb motion of the hexapod robot. In order to adapt to environment change, through sensory feedback, basic locomotion patterns can be switched accordingly, and the limb behavior of the robot is regulated via a RL-based learning approach. Compared to the pure data-driven locomotion control approach, only few of the CPG parameters involved with the limb behavior control need to be learned iteratively. Hence, the proposed locomotion control approach can be adopted to the robot practically.

The rest of this paper is organized as follows. Section 2 introduces the model of the hexapod robot. Section 3 details the two-layer CPG controller and explores its dynamics with numerical studies. Following that, the RL-based learning approach for refining the CPG parameters is presented in section 4. In section 5, both simulations and experiments are conducted to verify the proposed locomotion control approach. Finally, the conclusions and future work are given.

2. MODELING OF A HEXAPOD ROBOT

2.1. The Prototype of the Hexapod Robot

The prototype of the hexapod robot is investigated in this paper shown in **Figure 1A**, and the specifications are given in **Table 1**.

Figure 1B illustrates the mechanical schematic of the hexapod robot, which consists of 18 servo motors, a microprocessor, a Bluetooth communication module, a set of mechanical connectors and several other peripherals. Three motors (Dynamixel AX-12) equipped in a leg are concatenated together to act as three joints. A microprocessor (Arduino UNO) is used for processing sensor data, transferring diagnostic information via the Bluetooth module, making decisions and controlling servo motors. Besides that, an external camera (Logitech C930) is employed to track the position of the robot as feedback signals.

2.2. Modeling

To establish the kinematic/dynamic model of the hexapod robot, the joint coordinates of each leg i are defined as depicted in **Figure 2**.

The kinematic model is represented by Denavit-Hartenberg (DH) parameters for resolving inverse kinematic of the leg. According to these fixed frames, the transformation parameters and DH parameters are demonstrated in **Tables 2**, **3**, respectively.

The relative translation and rotation between the $(j-1)th$ and the jth joint coordinates are computed by the transformation matrix (1):

$$ {}^{i}\mathbf{T}_{j}^{j-1} = \begin{bmatrix} \cos\theta_{ij} & -\cos\alpha_{ij}\sin\theta_{ij} & \sin\alpha_{ij}\sin\theta_{ij} & a_{ij}\cos\theta_{ij} \\ \sin\theta_{ij} & \cos\alpha_{ij}\cos\theta_{ij} & -\sin\alpha_{ij}\cos\theta_{ij} & a_{ij}\sin\theta_{ij} \\ 0 & \sin\alpha_{ij} & \cos\alpha_{ij} & d_{ij} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1} $$

where especially, the transition matrix between the body coordinate $\{O_B\}$ and the hip joint coordinate $\{O_{i0}\}$ is represented by (2):

$$ {}^{i}\mathbf{T}_{0}^{B} = \begin{bmatrix} \cos\varphi_i & -\sin\varphi_i & 0 & d_{xi} \\ \sin\varphi_i & \cos\varphi_i & 0 & d_{yi} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2} $$

Consequently, the foot tip coordinate $\{O_{i3}\}$ can be transformed into the body coordinate $\{O_B\}$ by multiplying the previous matrixs sequentially shown in (3):

$$ \begin{aligned} {}^{i}\mathbf{T}_{3}^{B} &= {}^{i}\mathbf{T}_{0}^{B} \cdot {}^{i}\mathbf{T}_{1}^{0} \cdot {}^{i}\mathbf{T}_{2}^{1} \cdot {}^{i}\mathbf{T}_{3}^{2} \\ &= \begin{bmatrix} \cos(\varphi_i+\theta_{i1})\cos(\theta_{i2}+\theta_{i3}) & -\cos(\varphi_i+\theta_{i1})\sin(\theta_{i2}+\theta_{i3}) & \sin(\varphi_i+\theta_{i1}) & d_{xi}+\cos(\varphi_i+\theta_{i1})(l_{i1}+l_{i2}\cos(\theta_{i2})+l_{i3}\cos(\theta_{i2}+\theta_{i3})) \\ \sin(\varphi_i+\theta_{i1})\cos(\theta_{i2}+\theta_{i3}) & -\sin(\varphi_i+\theta_{i1})\sin(\theta_{i2}+\theta_{i3}) & -\cos(\varphi_i+\theta_{i1}) & d_{yi}+\sin(\varphi_i+\theta_{i1})(l_{i1}+l_{i2}\cos(\theta_{i2})+l_{i3}\cos(\theta_{i2}+\theta_{i3})) \\ \sin(\theta_{i2}+\theta_{i3}) & \cos(\theta_{i2}+\theta_{i3}) & 0 & l_{i2}\sin(\theta_{i2})+l_{i3}\sin(\theta_{i2}+\theta_{i3}) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{3} $$

Thus, the position of the foot tip with respect to the body coordinate $\{O_B\}$ can be derived as given below:

$$ \begin{aligned} &\begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \end{bmatrix} \\ &= \begin{bmatrix} d_{xi}+\cos(\varphi_i+\theta_{i1})(l_{i1}+l_{i2}\cos(\theta_{i2})+l_{i3}\cos(\theta_{i2}+\theta_{i3})) \\ d_{yi}+\sin(\varphi_i+\theta_{i1})(l_{i1}+l_{i2}\cos(\theta_{i2})+l_{i3}\cos(\theta_{i2}+\theta_{i3})) \\ l_{i2}\sin(\theta_{i2})+l_{i3}\sin(\theta_{i2}+\theta_{i3}) \end{bmatrix}, \end{aligned} \tag{4} $$

where $[p_{xi}\ p_{yi}\ p_{zi}]^T$ is the position coordinate of the ith foot hip and θ_{ij} is the joint angle.

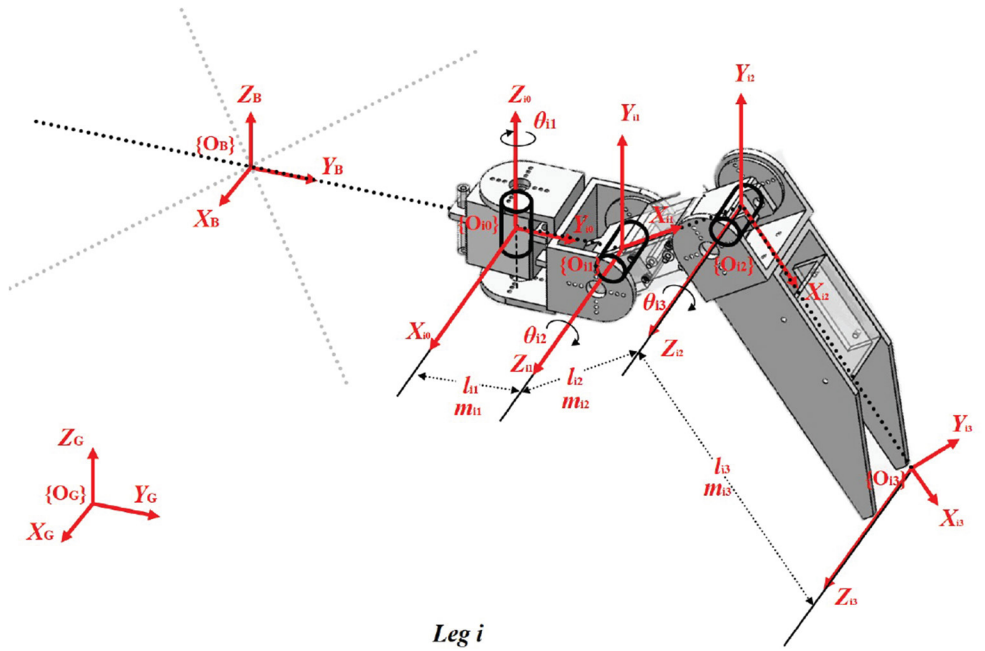

FIGURE 2 | Coordinates at different joints of each leg *i*. $\{O_G\}$ denotes the global coordinate, and $\{O_{ik}\}$ $(k = 0, 1, 2, 3)$ represents the floating frame whose origin fixed on the joints or the foot tip. l_{i1}, l_{i2}, l_{i3} and m_{i1}, m_{i2}, m_{i3} are the length and mass of the leg links, respectively. And $\theta_{i1}, \theta_{i2}, \theta_{i3}$ are the rotational joint angles around Z_{i0}, Z_{i1}, Z_{i2} axis of the leg.

TABLE 2 | Transformation parameters from the $\{O_{i0}\}$ to the $\{O_B\}$.

Leg *i*	**1**	**2**	**3**	**4**	**5**	**6**
$d_{xi}(mm)$	33.5	67	33.5	−33.5	−67	−33.5
$d_{yi}(mm)$	58	0	−58	−58	0	58
$\varphi_i(^\circ)$	−60	0	60	120	180	−120

The hip joint position is defined as (d_{xi}, d_{yi}) and φ_i denotes the direction angle in the body frame $\{O_B\}$.

TABLE 3 | Denavit-Hartenberg parameters.

Joint *j*	α_{ij}	a_{ij}	d_{ij}	θ_{ij}
1	$\pi/2$	l_{i1}	0	θ_{i1}
2	0	l_{i2}	0	θ_{i2}
3	0	l_{i3}	0	θ_{i3}

Where j =1,2,3 is the joint number from hip joint to knee joint of each leg. And α_{ij} is the link twist indicating the angle from $Z_{i(k-1)}$ to Z_{ik} around X_{ik}, a_{ij} is the link length representing the distance from the intersection of $Z_{i(k-1)}$ and X_{ik} to the origin of X_{ik}, d_{ij} is the joint distance meaning the distance from the intersection of $Z_{i(k-1)}$ and X_{ik} to the origin of $Z_{i(k-1)}$, θ_{ij} is the joint angle showing the angle from $X_{i(k-1)}$ to X_{ik} around $Z_{i(k-1)}$.

The leg of the hexapod robot is a complex joint-link system connecting the body trunk with the ground. Hence, closed kinematics chains can be found in the robot system. Since forces and moments propagate via the kinematics chains among different legs (Roy and Pratihar, 2013), the kinematics and dynamics are coupled. The dynamic model of such a coupled hexapod robot with 18 actuators is derived via Lagrangian-Euler method as follows:

$$\boldsymbol{\tau}_i = \mathbf{M}_i(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}_i + \mathbf{H}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}_i + \mathbf{G}_i(\boldsymbol{\theta}) - \bar{\mathbf{J}}_i^T \mathbf{F}_i \quad , \tag{5}$$

where $\boldsymbol{\tau}_i = [\tau_{i1}\ \tau_{i2}\ \tau_{i3}]^T \in \mathbb{R}^3$ is the joint torque vector of the *ith* leg consisting of hip joint torque τ_{i1}, knee joint torque τ_{i2} and ankle joint torque τ_{i3}. $\boldsymbol{\theta}_i = [\theta_{i1}\ \theta_{i2}\ \theta_{i3}]^T \in \mathbb{R}^3, \dot{\boldsymbol{\theta}}_i = [\dot{\theta}_{i1}\ \dot{\theta}_{i2}\ \dot{\theta}_{i3}]^T \in \mathbb{R}^3, \ddot{\boldsymbol{\theta}}_i = [\ddot{\theta}_{i1}\ \ddot{\theta}_{i2}\ \ddot{\theta}_{i3}]^T \in \mathbb{R}^3$ are joint angle, joint angle acceleration, and joint angle jerk vector of the *ith* leg, respectively. $\mathbf{M}_i(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ is a inertia matrix of the *ith* leg. $\mathbf{H}_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{3\times3}$ is Coriolis forces matrix of the *ith* leg. $\mathbf{G}_i(\boldsymbol{\theta}) \in \mathbb{R}^3$ is a link gravitational forces vector of the *ith* leg. $\mathbf{F}_i = [f_{ix}\ f_{iy}\ f_{iz}]^T \in \mathbb{R}^3$ represents ground reaction forces of the *ith* support foot tip with the coordinate $\{O_{i3}\}$. $\bar{\mathbf{J}}_i \in \mathbb{R}^{3\times3}$ is the Jacobian matrix of the *ith* leg, computed by (6). Moreover, the position and velocity of the hexapod robot in this work are transformed to the global coordinate $\{O_G\}$.

$$\bar{\mathbf{J}}_i = \begin{bmatrix} -(l_{i1} + l_{i2}\cos(\theta_{i2}) + l_{i3}\cos(\theta_{i2}+\theta_{i3}))\sin(\theta_{i1}) & -(l_{i2}\sin(\theta_{i2}) + l_{i3}\sin(\theta_{i2}+\theta_{i3}))\cos(\theta_{i1}) & -l_{i3}\sin(\theta_{i2}+\theta_{i3})\cos(\theta_{i1}) \\ (l_{i1} + l_{i2}\cos(\theta_{i2}) + l_{i3}\cos(\theta_{i2}+\theta_{i3}))\cos(\theta_{i1}) & -(l_{i2}\sin(\theta_{i2}) + l_{i3}\sin(\theta_{i2}+\theta_{i3}))\sin(\theta_{i1}) & -l_{i3}\sin(\theta_{i2}+\theta_{i3})\sin(\theta_{i1}) \\ 0 & l_{i2}\cos(\theta_{i2}) + l_{i3}\cos(\theta_{i2}+\theta_{i3}) & l_{i3}\cos(\theta_{i2}+\theta_{i3}) \end{bmatrix}. \tag{6}$$

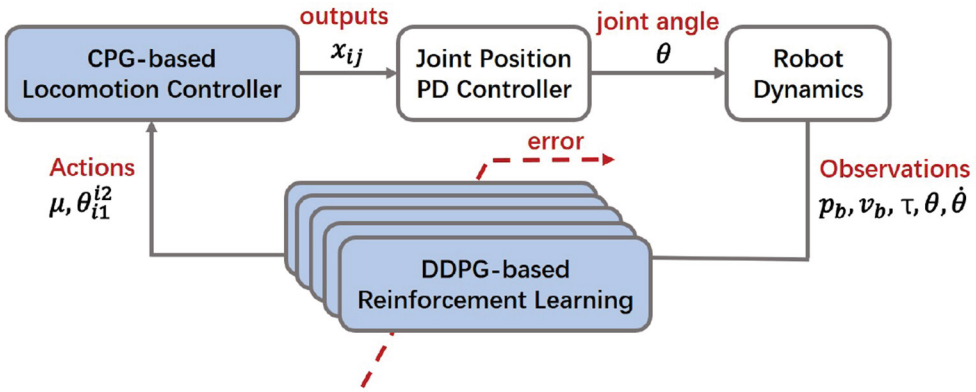

FIGURE 3 | Diagram of the proposed bio-inspired control scheme. The proposed control scheme has a cascaded structure with a feedback loop. It consists of three parts: (1) A dynamic model (with a embedded PD controller) that computes torque commands to handle robot dynamics subject to mechanical constraints. The dynamics parameters $p_b = [p_x, p_y, p_z]^T$, $v_b = [v_x, v_y, v_z]^T$ are the robot body position and velocity vector,respectively; τ, θ, $\dot{\theta}$ indicate the joint torque, angle and angle velocity, respectively. (2) A two-layer CPG locomotion controller that outputs coordinated signals to generate the basic locomotion. The CPG parameters μ and θ_{i1}^{i2} are the inputs representing the amplitude and the phase difference between the hip joint *i*1 and the knee joint *i*2 of the leg i, respectively; x_{ij} is the output signal. (3) A DDPG-based RL motion controller that trains the optimal locomotion via the cost function.

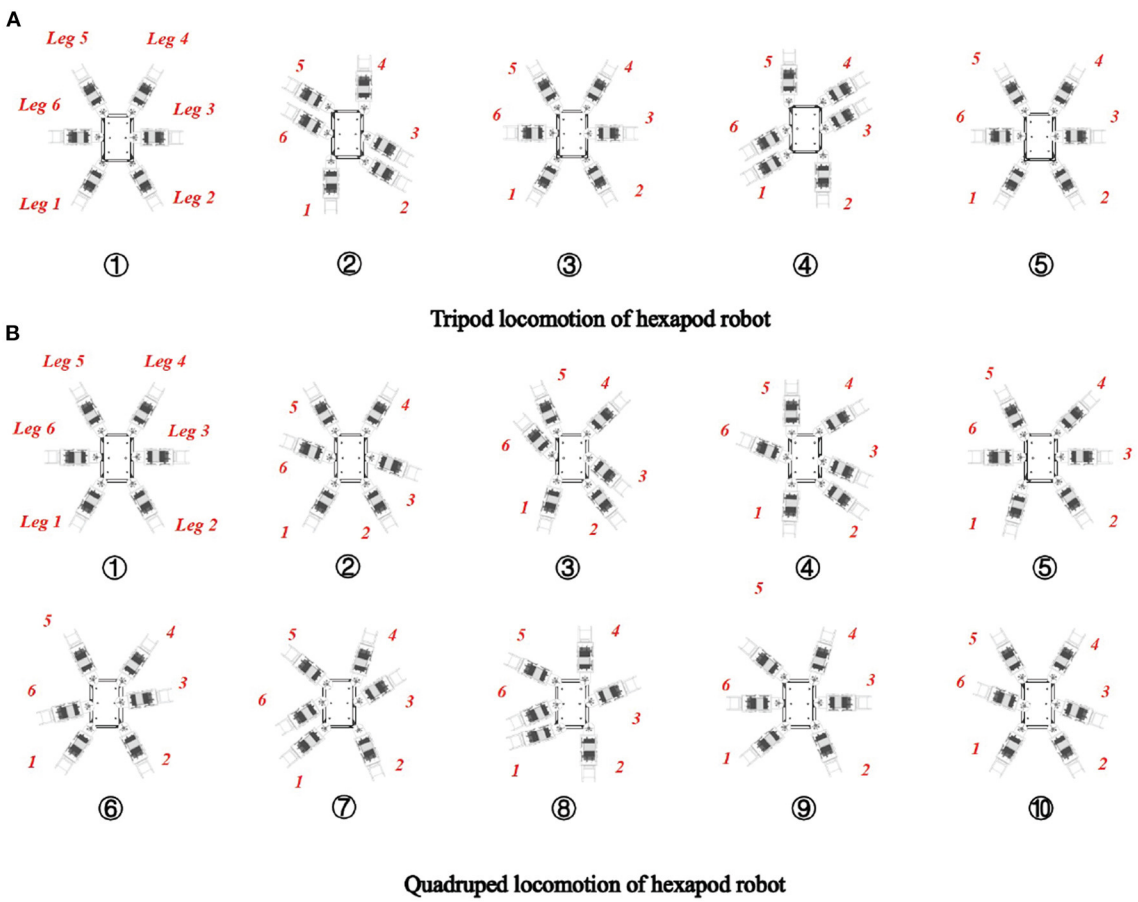

FIGURE 4 | Locomotion patterns via the body layer. During each cycle, six legs in the tripod locomotion are separated into two sets of {*leg* 1, *leg* 3, *leg* 5} and {*leg* 2, *leg* 4, *leg* 6} moving alternately, while in the quadruped locomotion are separated into four sets of {*leg* 3}, {*leg* 1, *leg* 4}, {*leg* 6}, and {*leg* 2, *leg* 5} to move successively.

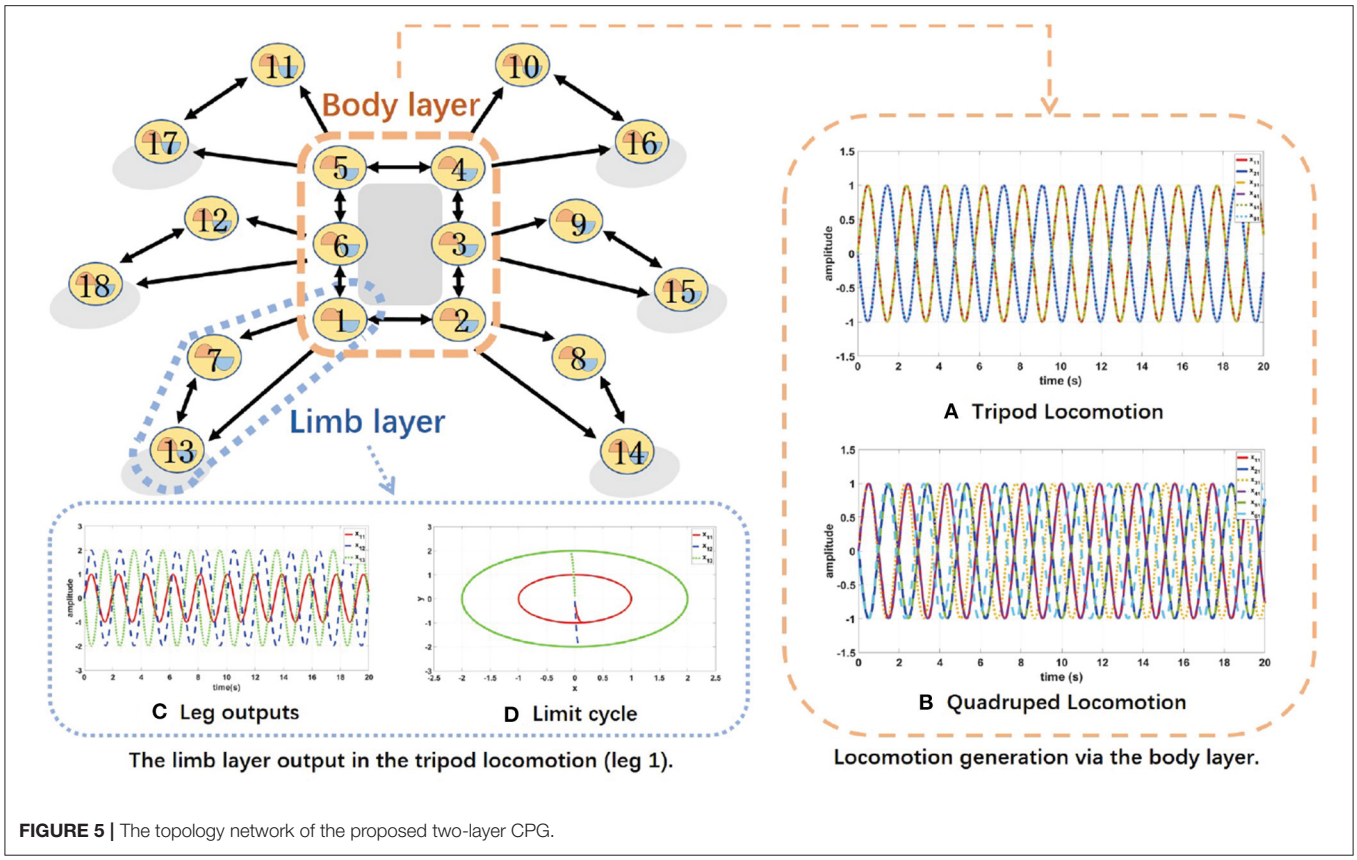

FIGURE 5 | The topology network of the proposed two-layer CPG.

3. LOCOMOTION CONTROLLER VIA CPG

Based on the analysis of the aforementioned mathematical model, the whole control scheme is proposed as shown in **Figure 3**. Inspired by biological arthropods, a hexapod robot is supposed to exhibit various locomotion in different terrains, such as tripod locomotion, quadruped locomotion, and five legs support locomotion (Zhong et al., 2018). Among these locomotion patterns, the tripod locomotion can achieve the fastest movement, while the quadruped and five legs support locomotion are more flexible. In this work, the locomotion patterns can be judged by velocity criterion according to the change of terrains.

In nature, CPGs are mainly used for generating coordinated and rhythmic movements for the locomotion of animals. Based on the similarity between biological legged animals and hexapod robots as well as the attractive capability of the CPG-based model on coupling the dynamics of robots, artificial CPG-based locomotion controllers are widely adopted to generate the locomotion behaviors of the biological counterparts. The basic locomotion patterns of the hexapod robot and the phase relations of the locomotion patterns are illustrated in **Figures 4A,B**.

3.1. Two-Layer CPGs Model

Due to complicated couplings and high degrees of freedom on the hexapod robot, the proposed CPG-based locomotion control is decomposed into two layers: (1) The body layer consists of six hip oscillators with bidirectional couplings. (2) The limb layer includes three oscillators in association with the hip joint, the knee joint and the ankle joint in every leg, where the knee joint oscillator and ankle joint oscillator are interconnected with bidirectional coupling, but the oscillator pair is unidirectionally controlled by the corresponding hip oscillator in the body layer.

Therefore, the body layer acting as a Conscious Layer shown in **Figures 5A,B** provides knowledge to determine the locomotion mode of the hexapod robot, while the limb layer acting as a Behavior Layer shown in **Figures 5C,D** has a major impact on final motion states and performance.

Considering the stable limit cycle and the explicit interpretable parameters, Hopf oscillator is a suitable element to construct CPGs for robotic locomotion (Seo et al., 2010). Hence, in this work, our CPG model can be described as a set of coupled Hopf oscillators and each Hopf oscillator is formulated by (7):

$$\begin{cases} \dot{x} = \alpha(\mu^2 - x^2 - y^2)x - \omega y \\ \dot{y} = \beta(\mu^2 - x^2 - y^2)y - \omega x \end{cases}, \tag{7}$$

where x and y are two state variables, ω is the frequency, α and β are the positive constants which determine the convergence rate of the limit cycle. In this paper, x is defined as the output signal of the oscillator.

Since the hexapod robot in this work has six legs and each leg has 3 DoF, a network consisted of 18 Hopf oscillators is proposed. According to the proposed CPG model shown in **Figure 5**,

to achieve desired motion of the hexapod robot, multiple oscillators are needed to be coupled together to guarantee robotic system synchronization and coordination. Motivated by the work presented by Campos et al. (2010), the proposed CPG model connected by the diffusive coupling is described by:

$$\begin{bmatrix} \dot{x}_{ij} \\ \dot{y}_{ij} \end{bmatrix} = \begin{bmatrix} \alpha(\mu^2 - x_{ij}^2 - y_{ij}^2) & -\omega_{ij} \\ \omega_{ij} & \beta(\mu^2 - x_{ij}^2 - y_{ij}^2) y_{ij} \end{bmatrix} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + k \cdot \sum_{mn \neq ij} \overline{R}(\theta_{mn}^{ij}) \begin{bmatrix} 0 \\ \frac{x_{mn}+y_{mn}}{\sqrt{x_{mn}^2+y_{mn}^2}} \end{bmatrix}, \quad (8)$$

where x_{ij}, y_{ij} with $i = 1, 2, \cdots, 6$ and $j = 1, 2, 3$ denote two state variables. The constant coupling strength $k = 0.1$ and the convergence coefficients $\alpha = \beta = 100$ are set for all oscillators, which are determined through a trial-and-error simulation on the stability of limit cycle in this work. The oscillator frequencies are unified as $\omega_{ij} = \omega$ for simplifying the high-level optimization. Besides, θ_{mn}^{ij} with $m = 1, 2, \cdots, 6$ and $n = 1, 2, 3$ represents the phase difference between the joint ij and the joint mn, then an associated 2D rotation matrix $\overline{R}(\theta_{mn}^{ij})$ is defined as:

$$\overline{R}(\theta_{mn}^{ij}) = \begin{bmatrix} \cos(\theta_{mn}^{ij}) & -\sin(\theta_{mn}^{ij}) \\ \sin(\theta_{mn}^{ij}) & \cos(\theta_{mn}^{ij}) \end{bmatrix}. \quad (9)$$

Compared with (7), the coupling relations among different Hopf oscillators are embedded into the artificial CPG model. This proposed 3D two-layer CPG model not only can regulate the basic locomotion patterns of the hexapod robot, but also fine-tune the motion performance for adapting to environment change. More information about the superiority of the 3D topology are demonstrated in our previous work (Niu et al., 2014). Through this CPG-based locomotion controller, the coordination can be adjusted with fewer parameters, which effectively reduce the control dimension and complexity of the system.

3.2. Simulation of Locomotion Generation

To verify the performance of the proposed CPG-based locomotion controller, several simulations are conducted. In the first layer of the network, the phase differences of the body layer among different hip joints are set as shown in **Table 4** to generate the tripod locomotion or quadruped locomotion. The six body oscillator parameters in the tripod and quadruped locomotion are set as *amplitude* = 1 and *frequency* = 3.14.

As can be seen from **Figures 5A,B**, the outputs of the body layer network are stable and periodic, while the phase differences between the neighboring oscillators maintain strictly 180 *deg* for tripod locomotion and 90 *deg* for quadruped locomotion, respectively.

Take the tripod locomotion patterns in leg 1 as an example, the limb layer network firstly receives the corresponding hip joint signal from the body layer. Secondly, the limb network outputs two signals to control the knee joint and the ankle joint interacting with environment. The phase difference between the knee joint and the ankle joint is limited to 180 *deg* in each leg with *amplitude* = 2 and *frequency* = 3.14.

As shown in **Figure 5C**, the phase difference between the knee joint and the ankle joint is locked in the limb layer. Moreover, **Figure 5D** presents the stable limit cycle of the coupled Hopf oscillators, which alleviates the influence of disturbances and ensures the smooth tuning of the robot locomotion. These simulation results show that the proposed CPG-based locomotion controller carry the potential of excellent controllability and robustness in unknown and unstructured terrains via online adjustment.

It should be noted that several parameters play important roles in the two-layer CPG controller, namely, amplitudes, frequencies, and phase differences. The CPG allows direct modulation of these parameters to enhance locomotion adaptability of the hexapod robot, but the manual tuning process still remain a challenge. Motivated by the movement of the six-legged arthropods modulated further via higher controller from brain-stem level (Yu et al., 2020), a RL-based controller is proposed to optimize the specialized locomotion patterns automatically in the next section.

4. LOCOMOTION OPTIMIZATION VIA REINFORCEMENT LEARNING

4.1. Problem Statement

Locomotion of a hexapod robot can be considered as a Markov Decision Process (MDP), which is described as an agent interacting with the environment in discrete time steps. At each time step t, the state of the agent and the environment can be jointly described by a state vector $s \in S$, where S is the state space. The agent takes an action $a_t \in A$, after which the environment advances a single time step and reaches a new state s_{t+1} through an environment state-transition distribution $\mathscr{P}: S \times A \times S \rightarrow [0, 1]$. Each state-action transition process is evaluated by a scalar reward function $\mathscr{R}: S \times A \rightarrow \mathbb{R}$. At the beginning of each episode, the agent finds itself in a random initial state $s_0 \sim \rho(s)$. Thus, the MDP is defined as a tuple $(S, A, \mathscr{R}, \mathscr{P}, \rho)$ (Tan et al., 2018).

In the MDP, the agent selects the action a under the state s through a stationary policy $\pi : S \rightarrow P(A)$, which is defined as a function mapping states into probability distributions over actions. The set of all stationary policies is denoted as Π. Give a performance measurement function as:

$$J(\pi) = \underset{\varsigma \sim \pi}{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})], \quad (10)$$

where $\gamma \in [0, 1)$ is the discount factor and ς denotes a trajectory under the policy π. The objective of the RL is to select a optimal policy π^* that maximizes $J(\pi)$, i.e.,

$$\pi^* = \underset{\pi \in \Pi}{argmax}\ J(\pi). \quad (11)$$

However, lack of complete freedoms when training the hexapod robot could suffer from some failed actions, such as collisions, falls, and inaccessible locomotion for a real robot. To tackle these issues, the actions of the hexapod robot should be constrained by several conditions such as acceleration, velocity, and torque constraints, which ensures the robot safe exploration.

TABLE 4 | Phase differences in corresponding locomotion.

Phase differences	Locomotion patterns			
	Tripod *(deg)*		Quadruped *(deg)*	
θ_{11}	0	$\theta^{ij}_{mn} = 360 \cdot \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \end{bmatrix}$	0	$\theta^{ij}_{mn} = 360 \cdot \begin{bmatrix} 0 & \frac{2}{4} & \frac{1}{4} & 0 & \frac{2}{4} & \frac{3}{4} \\ -\frac{2}{4} & 0 & -\frac{1}{4} & -\frac{2}{4} & 0 & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} & 0 & -\frac{1}{4} & \frac{1}{4} & \frac{2}{4} \\ 0 & \frac{2}{4} & \frac{1}{4} & 0 & \frac{2}{4} & \frac{3}{4} \\ -\frac{2}{4} & 0 & -\frac{1}{4} & -\frac{2}{4} & 0 & \frac{1}{4} \\ -\frac{3}{4} & -\frac{1}{4} & -\frac{2}{4} & -\frac{3}{4} & -\frac{1}{4} & 0 \end{bmatrix}$
θ_{21}	180		180	
θ_{31}	0		90	
θ_{41}	180		0	
θ_{51}	0		180	
θ_{61}	180		270	

Similar to the MDP, a constrained Markov Decision Process (CMDP) is defined as a tuple $(S, A, \mathscr{R}, C, \mathscr{P}, d, \rho)$. The difference between the MDP and the CMDP is that the policies are trained under additional cost constrains C in the latter. Each cost function $C_l : S \times A \times S \rightarrow \mathbb{R}$ maps transition tuples into costs with the limit c_l. Thus, the discounted cost of policy π with cost function C_l (Achiam et al., 2017) is represented by:

$$J_{C_l}(\pi) = \underset{\varsigma \sim \pi}{E}[\sum_{t=0}^{\infty} \gamma^t C_l(s_t, a_t, s_{t+1})]. \tag{12}$$

where l is the number of the constraints.

The set of feasible stationary policies in a CMDP is

$$\Pi_C = \{\pi \in \Pi : \forall l,\ J_{C_l}(\pi) <= c_l\}, \tag{13}$$

and the reinforcement learning problem in a CMDP is formulated as:

$$\pi^* = \underset{\pi \in \Pi_C}{argmax}\ J(\pi). \tag{14}$$

4.2. Deep Deterministic Policy Gradient Algorithm

Hexapod robots are multiple-input-multiple-output (MIMO) systems, so generally both the state space and the action space are high-dimensional and continuous. While many of stochastic policy gradient-based RL methods require massive and time-consuming search in such a vast space, deterministic policy greatly improve learning rates without sampling in the action space. Deep deterministic policy gradient (DDPG) (Lillicrap et al., 2016) as a model-free, off-policy RL algorithm, which could deal with unprocessed, high-dimensional sensory inputs and learn policies in a high-dimensional continuous action space via deep function approximators, has been widely accepted for robot control. Adaptive locomotion control of a hexapod robot is a challenging task due to the high-dimensional observations and continuous actions. In this work, the DDPG-based reinforcement learning optimization approach is proposed and illustrated in **Figure 6**.

Significantly, the DDPG combines an actor-critic method with deep neural networks (DNNs), and it shows stable performance in tough physical control problems including complex multi-joint movements and unstable contact dynamics. Besides, compared with the on-policy and stochastic counterparts such as proximal policy optimization (PPO), the off-policy and deterministic feature of DDPG ensures a more sample-efficient learning owing to the ability of generating a deterministic action.

The proposed DDPG algorithm is applied on learning the adaptive control policy π for the hexapod robot. The control policy π is assumed to be parameterized by θ^{π}. Specifically, the RL problem of learning the optimal control policy is converted into learning the optimal value θ^{π}. Considering that Policy Gradient is utilized for continuous action space, DDPG algorithm actually combines Policy Gradient with an actor-critic network. The parameter vector θ^{π} is updated in the direction that maximizes the performance measurement function $J(\pi)$. The direction, defined as the action gradient, is the gradient of $J(\pi)$ with respect to θ^{π} which can be calculated as follows:

$$\nabla_{\theta^{\pi}} J(\pi) = E[\nabla_a Q(s, a) \nabla_{\theta^{\pi}} \pi(s)], \tag{15}$$

where the action gradient of the performance measurement function $J(\pi)$ depends on the action-value function $Q(s, a)$, which is unknown and need to be estimated. To achieve the estimation, a critic network Q parameterized by θ^{Q} is used to approximate the action-value function and an actor network based on the current state offers control policy π that outputs the deterministic action in continuous space. In DDPG, the actor network and critic network are approximated by DNNs which can be learned via policy gradient method and error back propagation, respectively.

The use of neural networks for learning action-value function and control policy tends to be unstable. Thus, DDPG employs two important ideas to solve this problem.

REMARK 1. *A copy of the critic network and actor network: a target critic network and a target actor network are constructed and parameterized by vector $\theta^{Q'}$ and $\theta^{\pi'}$, respectively. These two target networks are adopted to calculate the target values, and the parameters $Q'$and π' in the two target networks slowly track the parameters Q and π in the original critic and actor network as follows:*

$$\theta^{Q'} = \kappa\theta^{Q} + (1 - \kappa)\theta^{Q'}, \tag{16}$$

$$\theta^{\pi'} = \kappa\theta^{\pi} + (1 - \kappa)\theta^{\pi'}, \tag{17}$$

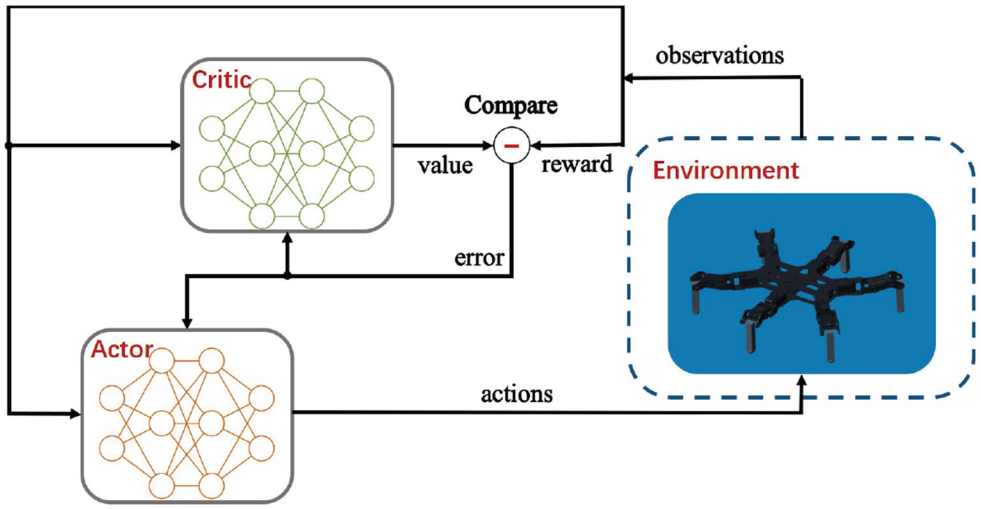

FIGURE 6 | The DDPG-based reinforcement learning structure.

where κ *is positive and* $\kappa << 1$. *The updating mechanism is called soft update, which avoids non-stationary target values and enhances the stability.*

REMARK 2. *Another challenge using neural networks for RL is the assumption that the samples are independently and identically distributed. Obviously, when the samples are generated from sequential exploration in an environment for the robot locomotion, this assumption is violated. To solve this, the replay buffer is used in DDPG. The replay buffer is a finite-size cache filled with transition samples. At each time step, both the actor network and the critic network are updated by sampling a mini batch uniformly from the buffer. Since DDPG is an off-policy learning algorithm, the replay buffer can be large where the algorithm benefits from a set of uncorrelated transitions. At each time step, the critic network* θ^Q *is updated by minimizing the loss:*

$$L = \frac{1}{H}\sum_{h=1}(Y_h - Q(s_h, a_h|\theta^Q))^2, \tag{18}$$

where

$$Y_h = r_h + \gamma Q'(s_{h+1}, \pi'(s_{h+1}|\theta^{\pi'})|\theta^{Q'}), \tag{19}$$

and h is the time step. H is the size of the mini batch sample.

The actor network θ^π *is updated using the sampled policy gradient:*

$$\nabla_{\theta^\pi} J = \frac{1}{H}\sum_{h=1} \nabla_a Q(s, a|\theta^Q)|_{s=s_h, a=\pi(s_h)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s_h}, \tag{20}$$

4.3. Observation Space

The hexapod robot interacts with the environment through observations and actions. In order to apply DDPG on a practical system, the observation space is required to match the real robot and provides enough information for the agent to learn the task. In this work, a MDP observation vector o_t at time t is defined as:

$$o_t =< p_b, v_b, \mathscr{O}, \theta, \dot{\theta}, \tau, A >, \tag{21}$$

where $\mathscr{O}$ is the body orientation. A is the policy output including the amplitude and phase difference in the limb layer of the CPG.

In addition, the observation space consists of only part of the states, so the MDP can not be fully described. For example, the hexapod robot can not identify terrain types without any use of exteroceptive sensors. Hence, the process is referred as a Partially Observable Markov Decision Process (POMDP). Since in our work, the hexapod robot interacts with the environment through a continuous trajectory rather than a discrete action, we find that our observation space is sufficient enough for learning the desirable tasks.

4.4. Action Space

The control policy outputs the coupling parameters of the limb layer of the CPG which determine the intra-limb coordination as well as the adaptation to different terrain types. The action space is defined as follows:

$$a_t =< \mu, \theta_{i1}^{i2} >, \tag{22}$$

The action vector is transmitted as the input of the CPG network which generates the joint positions for each joint actuators.

The two-layer CPG network is chosen as the locomotion generator instead of learning joint position commands directly like most of the other studies (Hwangbo et al., 2019; Tsounis et al., 2020). There are three reasons for this: (1) the CPG network constrains the basic locomotion of the robot, which reduces the search space and accelerates the learning; (2) compared to 18 joint position or joint torque commands, learning symmetric CPG coupling parameters lowers the dimension of the action space; (3) the CPG network outputs smooth joint position commands, which are easier to be realized in the real robot.

TABLE 5 | Reward terms.

Term	Expression
Forward velocity	v_x
Energy consumption	$\|\tau \cdot \dot{\theta}\| + \tau^2$

4.5. Network Architecture

In DDPG, the critic network and actor network are parameterized as deep neural networks. **Figure 7** provides a graphical depiction of the NN model. The model is similar to the network architecture implemented in Fujimoto et al. (2018) and is proved to perform well. The critic network is composed of five hidden layers including three fully-connected (FC) layers and two ReLU layers. The actor network consists of six hidden layers including three fully-connected (FC) layers, two ReLU layers and a Tanh layer. The output modulates the proposed two-layer CPG parameters.

4.6. Reward Function

In this work, the environmental adaptability of the hexapod robot is measured by two criteria: one is the heading velocity of the robot and the other is the energy consumption of the robot. In general, precise reward function in robotics is one of the main challenges in solving the RL problem. Due to the constraints in RL, the reward function is simplified to encourage faster heading velocity and penalize higher energy consumption.

Table 5 shows the detailed reward terms. The velocity reward term motivates the robot to move forward as fast as possible, and it is tuned so that the robot receives a reward for a positive velocity up to a certain point. The penalty term is used to optimize the energy consumption of the robot. Hence, the reward term and the penalty term is intergraded into the reward function r_t as follows:

$$r_t = K_v \cdot v_x - K_e \cdot (|\tau \cdot \dot{\theta}| + \tau^2), \tag{23}$$

where K_v and K_e are the positive weights.

4.7. Guided Constrained Costs

In this work, two types of constrains are introduced into the proposed RL method (Gangapurwala et al., 2020). The first is the performance constraint, which restricts the hexapod robot into the region with potential good performance. The second is the safety constraint to avoid the robot exploring the region where damages may occur.

(1) Performance Constraint Costs: These costs are directly added to the reward function, as shown in **Table 6**. The constraint costs are guided by the kinematic model of the hexapod robot and help to improve the locomotion performance. For example, the Joint Speed term and Torque term are the limits of the actuator performance of the robot. In our control scheme, each supporting leg of the hexapod robot moves symmetrically, so the Orientation term and Height term are given to limit the robot from swinging too much.

TABLE 6 | Performance constraint costs.

Term	Expression
Joint speed	$\|max(\|\dot{\theta}\| - \dot{\theta}^{max}, 0)\|^2$
Torque	$\|max(\|\tau\| - \tau^{max}, 0)\|^2$
Orientation	$\|\mathcal{O}\|^2$
Height	$\|z_c - z_{c0}\|^2$

$\dot{\theta}^{max}$ and τ^{max} refer to the maximum outputs of the robot actuator. z_c and z_{c0} are the current and original body height of the hexapod robot. Orientation and Height limit the swing range of robot body.

TABLE 7 | Safety constraint costs.

Term	Expression
Joint speed	$\text{bool}(\dot{\theta} > \dot{\theta}^{max})$
Torque	$\text{bool}(\tau > \tau^{max})$
Fall	$\text{bool}(z_c < 0)$
Roll	$\text{bool}(\mathcal{O} > \mathcal{O}^{limit})$

bool(·) is a general Boolean judgement function.

(2) Safety Constraint Costs: For the cases when control policy outputs actions that cause the robot to land on unstable and unrecoverable states and damage the robot, the safety constraints are introduced in **Table 7**. The Fall term and Roll term are given to judge whether the robot falls or rolls over. If the control policy outputs the commands that robot can not carry on (see Joint Speed and Torque) or the robot falls and rolls over (see Fall and Roll), the training episode is terminated directly. The training steps explored in this episode are abandoned and a negative terminal reward is added to the last training step in the reformatted episode samples. This control policy avoids inefficient explorations of some constrained regions because the training episode is terminated if any safety constraint costs is true.

5. SIMULATIONS AND EXPERIMENTS

The proposed bio-inspired learning scheme is used to shape the hexapod robot locomotion. We evaluate the feasibility and performance of the motion policy via four different terrains in both simulations and experiments.

5.1. Simulations

The aim of these simulations is to guarantee the convergence of RL algorithm and obtain the theoretical maximum velocity of the hexapod robot in forward motion under different terrains.

The hexapod robot is modeled corresponding to the dimensions and mass of the actual hexapod robot prototype where the contact friction dynamics are determined by material surfaces. There are five main parameters for simulations: (1) learning rates for actor-critic network are set as 0.005 and 0.001, respectively; (2) the maximum number of episodes and steps in an episode are set as 1,400 and 200, respectively; (3) the sampling time is similar to the CPG cycle time which is 1 s; (4) the

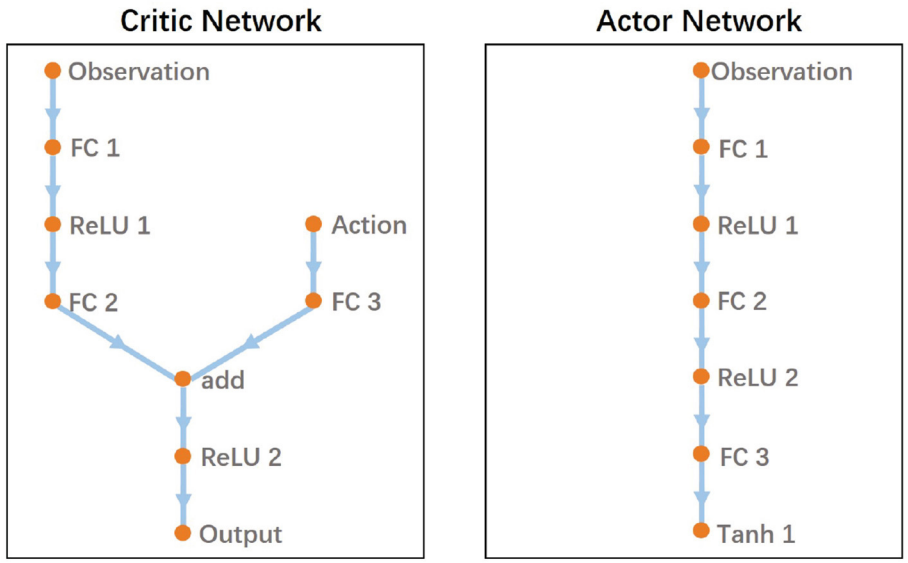

FIGURE 7 | The network architecture of the proposed DDPG-based RL.

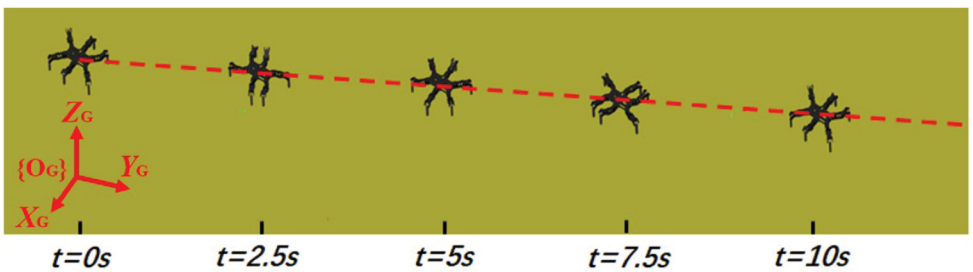

FIGURE 8 | The RL training simulation in an episode.

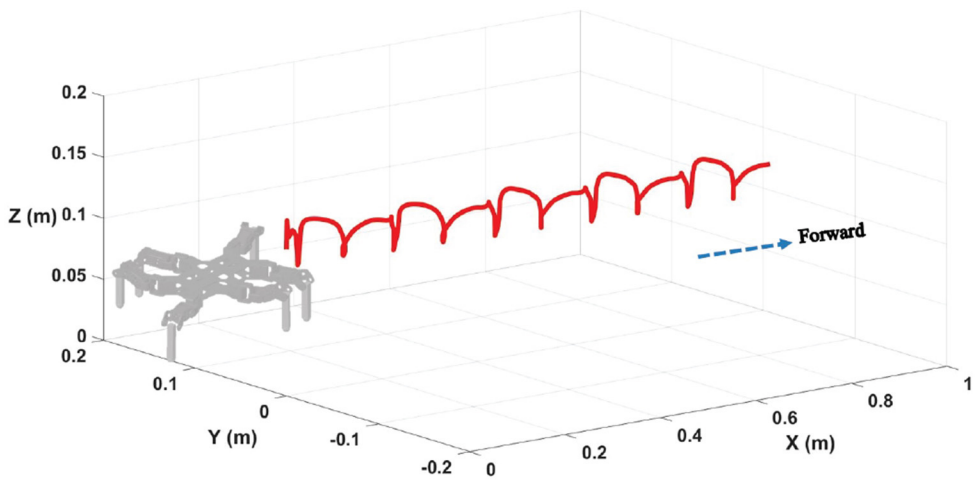

FIGURE 9 | The motion trajectory of the hexapod robot.

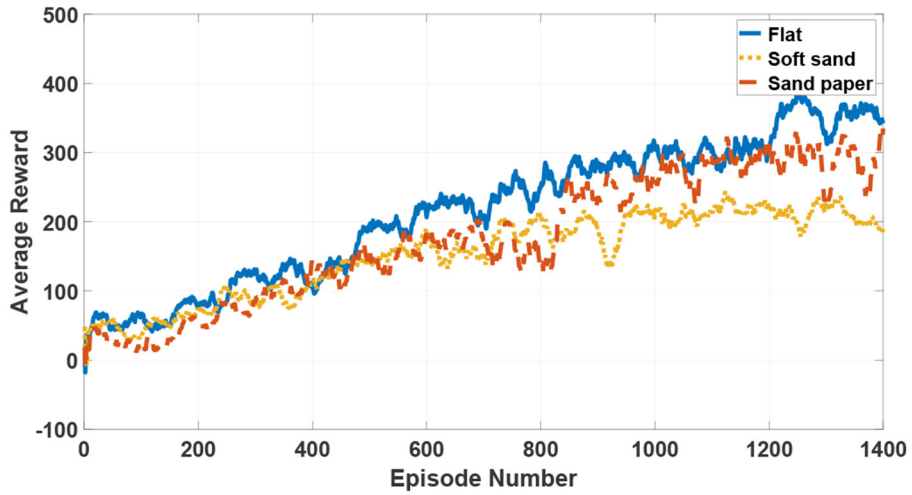

FIGURE 10 | The average reward of reinforcement learning in the tripod locomotion.

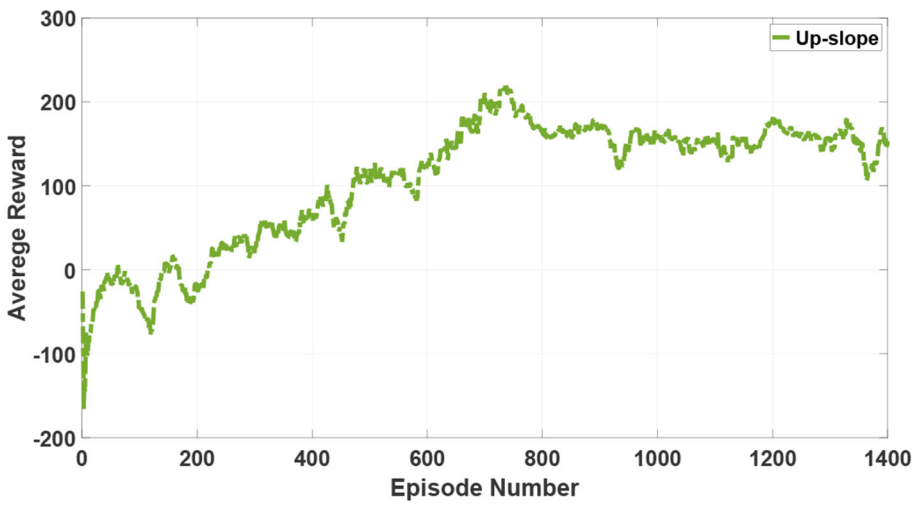

FIGURE 11 | The average reward of reinforcement learning in the quadruped locomotion.

frequency of the proposed two-layer CPG network is fixed on 0.5. The contact friction coefficients are modified according to different ground materials. The training process of an episode is randomly chosen in **Figure 8** and the motion trajectory is displayed in **Figure 9**.

From the two figures, it is noted that the hexapod robot walks well in the tripod locomotion without obvious offset in the direction of Y and Z axis. For a forward motion, we would like to emphasize that the sideslip in the vertical direction will cause an extremely uncertain deviation for the whole movement direction and posture. Therefore, the nearly tiny offset in Y axis illustrates the effectiveness of the proposed motion control scheme. As can be seen on Z axis, the slight fluctuation with the body height also reflects the control stability of the robot under the benefit of physical constraints.

In the training process, the observations are acquired from the hexapod dynamic model and the actions directly modulate the amplitude μ and phase difference θ_{mn}^{ij} in the limb layer of the CPG network. The reward function is given in the aforementioned section. At the end of 1,400 episodes, the average reward converges to a stable value in three terrain types as shown in **Figure 10**. The average training time in these terrains is approximately 6 h.

During learning processes, zero initial values drive the hexapod robot to swing around the origin and the value of reward function is equal to zero. After postures adapting and actions updating, the robot locomotion continuously becomes smoother. As the motion stability performance is improved, the reward value increases over time. Finally, the accumulative data samples help the robot reach the best motion state under the

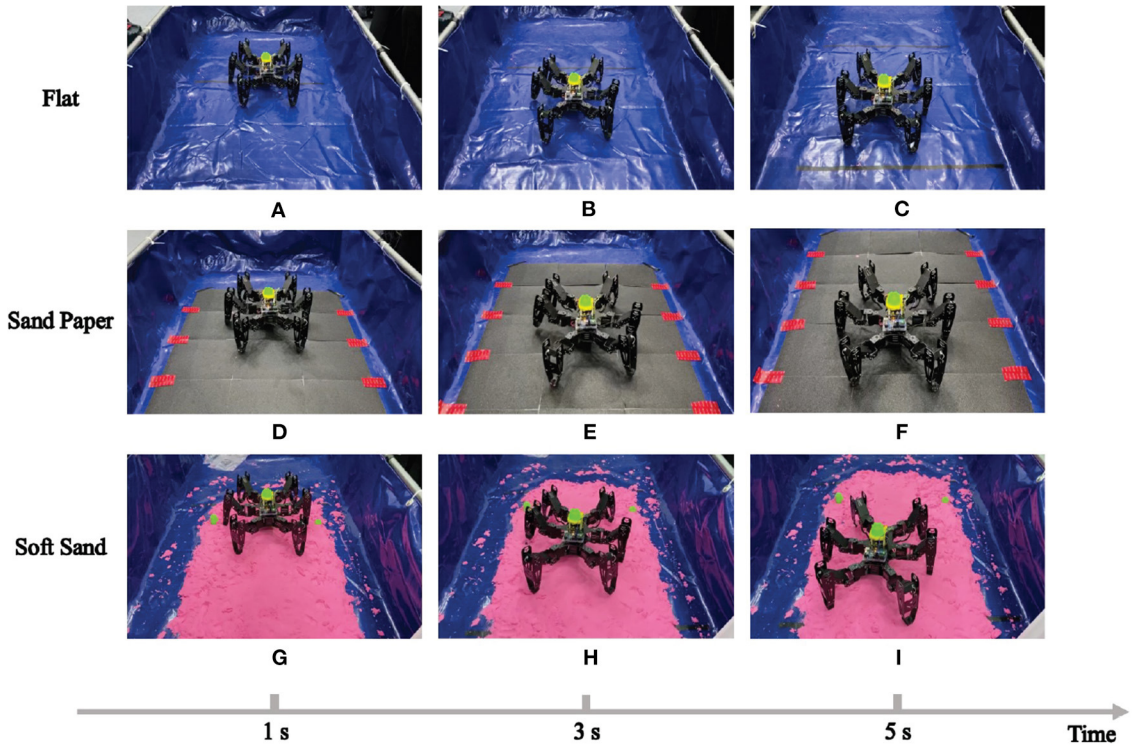

FIGURE 12 | The RL training experiment in different terrains.

specific locomotion mode and the reward function also converges to the biggest value. Compared with the movements in a sand paper and a soft sand, the steady velocity in a flat environment is a bit faster, but the convergence rate is conversely slower. As can be seen from the whole learning episodes, there are no obvious differences of the learning trend among three terrains. Besides, although the learning processes may suffer from several asymmetric and non-natural looking, even defective locomotion, the hexapod robot will finally converge to a stable and optimal locomotion under the limitation of several presetting constraints in the proposed DDPG-based learning approach.

As mentioned in section 3, tripod locomotion can be the fastest but inflexible. Therefore, when encountering complex and harsh terrains such as a slope or stones, the robot will switch to flexible locomotion modes such as quadruped locomotion or five legs support locomotion. In order to exhibit the locomotion flexibility derived from the proposed 3D two-layer CPG controller, an up-slope (10 deg) terrain is simulated and the quadruped locomotion is generated for repeating the aforementioned training process. The result of the average reward is represented in **Figure 11**.

Although the hexapod robot also accomplishes a fast convergence in an up-slope after 1,400 episodes, the average reward in such a tougher terrain is obviously lower than the stable value in a flat. Moreover, based on excellent adjustment characteristics of the proposed 3D two-layer CPG controller, the hexapod robot is endowed with the capability of locomotion transitions for adapting to complicated and unstructured terrains.

5.2. Experiments

Similarly, four experiments on different terrain surfaces, namely, flat, sand paper, soft sand, and up-slope, are conducted to validate the adaptivity and robustness of the proposed bio-inspired learning approach in practical scenarios. The training time is set as 5 s in each episode and other parameters set in these experiments are the same as simulation parameters. In addition, the simulation results can reduce the experimental training time through offering the hexapod robot an initial policy that performs the best in the simulations.

Firstly, environmental adaptability under individual locomotion mode has been tested. Here, the most common locomotion, tripod locomotion, is adopted as the training mode in three different terrain surfaces. The neural network in RL-based algorithm evolves three corresponding policies to make the robot perform well on the specific surfaces. The experiment scenes are shown in **Figures 12A–I**, where the same robot crawls on different contact surfaces.

During the repetitive episodes in the three mentioned terrains, the actual velocity of the hexapod robot with regard to the CPG tuning parameters, namely, the amplitude μ and phase difference θ_{mn}^{ij} in the limb layer are recorded. All these raw data are fitted and the learning results are shown in **Figures 14A–C**.

As can be seen from these figures, the results in different terrains have similar characteristic (the convex surface), but there are obvious differences in specific nodes. For example, when the amplitude is 1 with phase difference is 0.1, the velocity of the robot crawling on the flat will drop significantly from its maximum speed, but it decreased slowly on the soft sand.

Next, the quadruped locomotion in an up-slope (10 deg) is trained to evaluate the adaptability of different locomotion mode generated by the 3D two-layer CPG network (especially the body layer). The experimental platform and the learning result are illustrated in **Figures 13A–C**, **14D**, respectively. As observed in this experiment, the quadruped locomotion performs well and stably in up-slope environment showing a different trend compared with tripod locomotion, which explains the impact of basic locomotion patterns to the robot behavior.

Finally, the maximum velocities in the four terrains are calculated as listed in **Table 8**. It is noticed from **Table 8** that the hexapod robot runs fastest on the flat under tripod locomotion while it runs slowest under quadruped locomotion on the up-slope. Compared with the body length (BL) of the hexapod prototype (24 cm), the maximum velocity is 1/3 ∼ 1/2 BL from 7.35 to 13.10 (cm/s) in the preset terrains. As can be seen, since the surface friction coefficients of the chosen Sand Paper and Soft Sand terrain belong to the category of sand which may be similar to some extent, the difference of maximum velocity is not obvious.

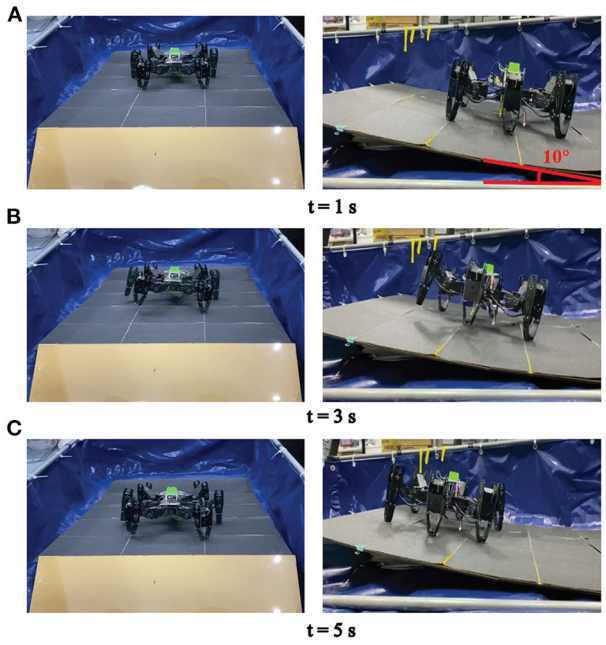

FIGURE 13 | The RL training experiment in an up-slope.

TABLE 8 | The maximum velocity in different terrains.

Terrain types	Velocity(cm/s)	
	Simulations	Experiments
Flat	22.27	13.10
Sand paper	19.87	11.86
Soft sand	15.40	11.72
Up-slope	13.05	7.35

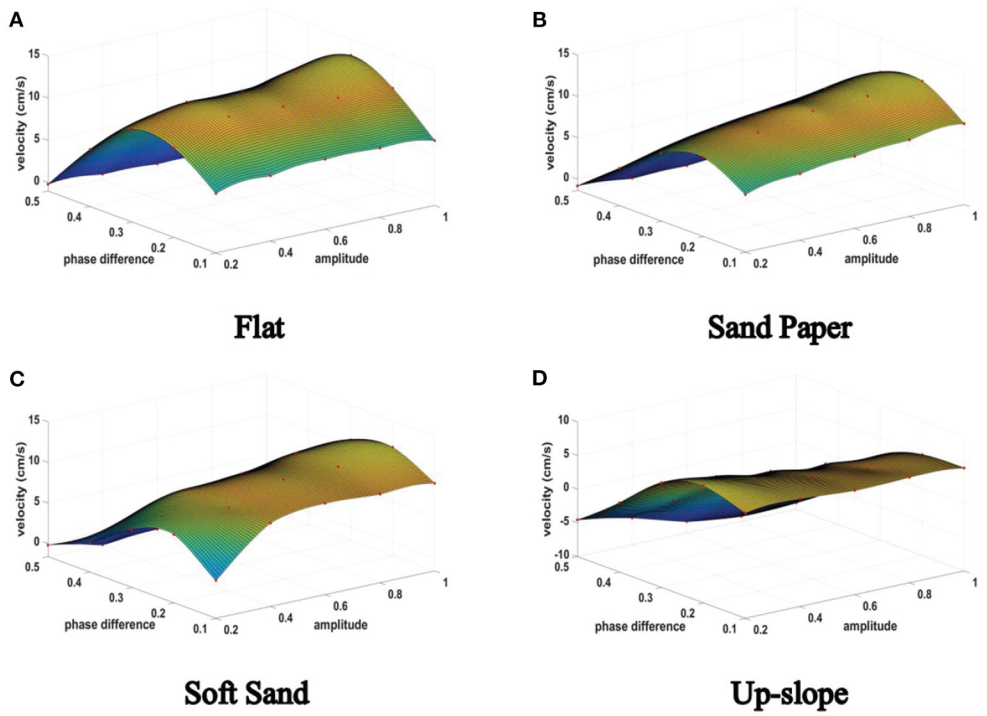

FIGURE 14 | The hexapod robot velocity in different terrains.

It should be also emphasized that, suffering from inaccuracies in modeling process as well as environment construction uncertainties, several deviations between simulation model and actual experiment exist inevitably. For instance, the parametric variables in simulation are fixed, while all the variables are inherently floating in the experiments. Nevertheless, the simulation results still have a certain association with the experimental results, which effectively offer prior information at the beginning of experiment settings and greatly accelerate the convergence rate in the actual system.

In summary, it can be concluded from the experimental results that the proposed 3D two-layer CPG network and the DDPG-based RL algorithm can provided the hexapod robot with excellent maneuverability and environmental adaptability performance while the stability and robustness of the overall control scheme can be also achieved.

6. CONCLUSION

This paper aims to investigate an adaptive locomotion control approach for a hexapod robot. Inspired by biological neuron control systems, the proposed locomotion controller is composed of a set of coupled oscillators, namely an artificial CPG network. The novelty of the CPG controller lies in its 3D two-layer. The first layer of the CPG is able to control the basic locomotion patterns according to the environment information, while a RL-based learning algorithm is adopted for fine-tuning the second layer of the CPG to regulate the behavior of robot limbs. Several numerical studies and experiments have been conducted to demonstrate the valid and effectiveness of the proposed locomotion controller. The navigation of the robot in a complex and dynamic environment will be explored in the next research phase.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

All authors contributed to the theory and implementation of the study. WO designed the whole locomotion control scheme, proposed the two-layer CPG, and wrote the first draft of the manuscript. HC modeled the hexapod robot and carried on the experiments. JP offered the simulation of the reinforcement learning part. WL corrected the paper formation and text required for the journal. QR determines the final Abstract, Introduction, and Conclusion. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by the National Natural Science Foundation of China (No. 61773271) and Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT20066).

REFERENCES

Achiam, J., Held, D., Tamar, A., and Abbeel, P. (2017). "Constrained policy optimization," in *2017 34th International Conference on Machine Learning (ICML)* (Ningbo), 30–47.

Azayev, T., and Zimmerman, K. (2020). Blind hexapod locomotion in complex terrain with gait adaptation using deep reinforcement learning and classification. *J. Intell. Robot. Syst.* 99, 659–671. doi: 10.1007/s10846-020-01162-8

Bal, C. (2021). Neural coupled central pattern generator based smooth gait transition of a biomimetic hexapod robot. *Neurocomputing* 420, 210–226. doi: 10.1016/j.neucom.2020.07.114

Barfoot, T. D., (2006). Experiments in learning distributed control for a hexapod robot. *Robot. Auton. Syst.* 54, 864–872. doi: 10.1016/j.robot.2006.04.009

Campos, R., Matos, V., and Santos, C. (2010). "Hexapod locomotion: a nonlinear dynamical systems approach," in *2010 36th Annual Conference on IEEE Industrial Electronics Society (IECON)* (Glendale, CA), 1546–1551. doi: 10.1109/IECON.2010.5675454

Chung, H.-Y., Hou, C. C., and Hsu, S. Y. (2015). Hexapod moving in complex terrains via a new adaptive cpg gait design. *Indus. Robot* 42, 129–141. doi: 10.1108/IR-10-2014-0403

Delcomyn, F. (1980). Neural basis of rhythmic behavior in animals. *Science* 210, 492–498. doi: 10.1126/science.7423199

Fortuna, L., Frasca, M., and Arena, P. (2004). *Bio-Inspired Emergent Control of Locomotion Systems*. Singapore: World Scientific. doi: 10.1142/5586

Fujimoto, S., Van Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *35th International Conference on Machine Learning* (Stockholm), 2587–2601.

Gangapurwala, S., Mitchell, A., and Hacoutis, I. (2020). Guided constrained policy optimization for dynamic quadrupedal robot locomotion. *IEEE Robot. Autom. Lett.* 5, 3642–3649. doi: 10.1109/LRA.2020.2979656

Grzelczyk, D., Szymanowska, O., and Awrejcewicz, J. (2018). Kinematic and dynamic simulation of an octopod robot controlled by different central pattern generators. *Proc. Instit. Mech. Eng. Part I J. Syst. Control Eng.* 233, 400–417. doi: 10.1177/0959651818800187

Hooper, S. L. (2000). Central pattern generators. *Curr. Biol.* 10, 176–177. doi: 10.1016/S0960-9822(00)00367-5

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., et al. (2019). Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* 4:eaau5872. doi: 10.1126/scirobotics.aau5872

Hyun, D. J., Seok, S. O., Sang, O., Lee, J., and Kim, S. (2014). High speed trot-running: implementation of a hierarchical controller using proprioceptive impedance control on the MIT cheetah. *Int. J. Robot. Res.* 33, 1417–1445. doi: 10.1177/0278364914532150

Juang, C.-F., Chang, Y. C., and Hsiao, C. M. (2011). Evolving gaits of a hexapod robot by recurrent neural networks with symbiotic species-based particle swarm optimization. *IEEE Trans. Indus. Electron.* 58, 3110–3119. doi: 10.1109/TIE.2010.2072892

Kecskés, I., Székics, L., Fodor, J. C., and Odry, P. (2013). "PSO and GA optimization methods comparison on simulation model of a real hexapod robot," in *2013 IEEE 9th International Conference on Computational Cybernetics, Proceedings (ICCC)* (Tihany), 125–130. doi: 10.1109/ICCCyb.2013.6617574

Kim, D., Jorgensen, S., Lee, J., Ahn, J., Luo, L., and Sentis, L. (2020). Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control. *Int. J. Robot. Res.* 39, 936–956. doi: 10.1177/0278364920918014

Lele, A. S., Fang, Y., Ting, J., and Raychowdhury, A. (2020). "Learning to walk: spike based reinforcement learning for hexapod robot central pattern generation," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Genova), 208–212. doi: 10.1109/AICAS48895.2020.9073987

Li, T., Lambert N., Calandra, R., Meier, F., and Rai, A. (2019). Learning generalizable locomotion skills with hierarchical reinforcement learning. *arXiv:1909.12324v1*. doi: 10.1109/ICRA40945.2020.9196642

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2016). "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations* (San Juan).

Niu, X., Xu, X., Ren, Q., and Wang, Q. (2014). Locomotion learning for an anguilliform robotic fish using central pattern generator approach. *IEEE Trans. Indus. Electron*. 61, 4780–4787. doi: 10.1109/TIE.2013.2288193

Ramdya, P., Thandiackal, R., Cherney, R., Asselborn, T., Benton, R., Ijspeert, A. J., et al. (2017). Climbing favours the tripod gait over alternative faster insect gaits. *Nat. Commun*. 8:14494. doi: 10.1038/ncomms14494

Roy, S. S., and Pratihar, D. K. (2013). Kinematics, dynamics and power consumption analyses for turning motion of a six-legged robot. *J. Intell. Robot. Syst*. 74, 663–688. doi: 10.1007/s10846-013-9850-6

Sartoretti, G., Paivine, W., Shi, Y., Wu, Y., and Choset, H. (2019). Distributed learning of decentralized control policies for articulated mobile robots. *IEEE Trans. Robot*. 35, 1109–1122. doi: 10.1109/TRO.2019.2922493

Seo, K., Chung, S. J., and Slotine, J. J. E. (2010). CPG-based control of a turtle-like underwater vehicle. *Auton. Robots* 28, 247–269. doi: 10.1007/s10514-009-9169-0

Stelzer, A., Hirschmüller, H., and Görner, M. (2012). Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *Int. J. Robot. Res*. 31, 381–402. doi: 10.1177/0278364911435161

Sun, Q., Gao, F., and Chen, X. (2018). Towards dynamic alternating tripod trotting of a pony-sized hexapod robot for disaster rescuing based on multi-modal impedance control. *Robotica* 36, 1048-1076. doi: 10.1017/S026357471800022X

Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., et al. (2018). Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv [Preprint] arXiv:1804.10332*.

Tsounis, V., Alge, M., and Lee, J. (2020). Deepgait: planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robot. Autom. Lett*. 5, 3699–3706. doi: 10.1109/LRA.2020.2979660

Yu, H., Gao, H., and Deng, Z. (2020). Enhancing adaptability with local reactive behaviors for hexapod walking robot via sensory feedback integrated central pattern generator. *Robot. Auton. Syst*. 124:103401. doi: 10.1016/j.robot.2019.103401

Yu, H., Gao, H., Ding, L., Li, M., Deng, Z., Liu, G., et al. (2016). Gait generation with smooth transition using cpg-based locomotion control for hexapod walking robot. *IEEE Trans. Indus. Electron*. 63, 5488–5500. doi: 10.1109/TIE.2016.2569489

Zarrouk, D., and Fearing, R. S. (2015). Controlled in-plane locomotion of a hexapod using a single actuator. *IEEE Trans. Robot*. 31, 157–167. doi: 10.1109/TRO.2014.2382981

Zhang, H., Liu, Y., Zhao, J., Chen, J., and Yan, J. (2014). Development of a bionic hexapod robot for walking on unstructured terrain. *J. Bion. Eng*. 11, 176–187. doi: 10.1016/S1672-6529(14)60041-X

Zhao, D., and Revzen, S. (2020). Multi-legged steering and slipping with low dof hexapod robots. *Bioinspir. Biomimet*. 15:045001. doi: 10.1088/1748-3190/ab84c0

Zhong, G., Chen, L., Jiao, Z., and Li, J. (2018). Locomotion control and gait planning of a novel hexapod robot using biomimetic neurons. *IEEE Trans. Control Syst. Technol*. 26, 624–636. doi: 10.1109/TCST.2017.2692727

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.