Neuro-Inspired Motion Control of a Soft Myriapod Robot

Qinyuan Ren, *Senior Member IEEE*, Wenxin Zhu, *Member IEEE*, Jiawei Cao, *Member IEEE*
Wenyu Liang, *Member IEEE*

***Abstract*—Centipedes exploit soft structures to move efficiently in complex environments. These abilities have promoted scientists to develop centipede-like robots with soft robot technologies. In this paper, a centipede-like robot constructed by multiple body segments connected in series is designed. Each body segment features a pair of antagonistic artificial muscles and four electro-adhesive feet, which plays the role of the basic motion unit of the centipede-like robot and can be regarded as a soft myriapod robot. To accomplish the desired movement tasks for the soft myriapod robot, this paper proposes a neuro-inspired hierarchical motion control scheme according to the functions of the centipede brain. In the proposed control scheme, a mushroom-body-inspired controller with an error-based learning mechanism is adopted for the soft actuator motion control, and a central-complex-inspired deep reinforcement learning algorithm is designed for the robot motion selection. At last, the effectiveness of the proposed motion control scheme is verified via both numerical studies and experiments.**

***Index Terms*—Soft robots, motion control, neuro-inspired, bioinspiration, deep reinforcement learning.**

I. Introduction

BIOLOGY offers a wealth of examples of small creatures with extraordinary locomotion capabilities, the study of which serves as a guide for designing bio-inspired myriapod robots. These myriapod robots offer an ideal alternative to sending humans into hazardous environments with potential applications, such as search-and-rescue, bio-monitoring, environment exploration, etc. Many of the myriapod robots take inspiration from cockroaches and notable examples include Rhex [1], RoACH [2] and DASH [3]. Another category of myriapod robots takes a centipede-like morphology, characterized by a long deformable body and multiple pairs of legs [4]–[7]. These centipede-inspired robots possess the advantages of motion adaptation, allowing smooth transitions from horizontal to vertical surfaces, as well as robust locomotion even when some legs are disabled.

Most of the current centipede-like robots feature rotational joints between several rigid-body segments, actively controlled by motors [4], [5]. As a result, these robots always face the problem of losing contact between the legs and the ground due to their rigid-body structures. To enhance body flexibility, Koh *et al.* designed a centipede robot that adopted a spring connecting with the neighbouring body segments in [6], which has been demonstrated in locomotion over various obstacles. Compared to the one-spring design, Hoffman and Wood used two springs to simulate the lateral flexor muscles that run along the body of an actual centipede to produce enhancing undulation locomotion in [7]. This spring-linkage mechanism endows the centipede robot with intrinsic compliance. Furthermore, to achieve intrinsic compliance similar to the biological muscles of centipedes, incorporating artificial muscles into a robotic design is a feasible approach.

Artificial muscles can be regarded as a class of soft actuators that mimic the properties of biological muscles to generate large and compliant deformations. Various soft actuators have been developed in the past decades, such as piezoelectric actuators, shape memory alloy (SMA) actuators, electroactive polymer (EAP) actuators, etc. Among these actuators, a type of EAP actuator, known as dielectric elastomer actuators (DEAs), stands out due to the characteristics of fast response and high power density. These characteristics contribute to the great potential of DEAs as the ideal actuators for various mobile soft robots, such as crawling robots [8], swimming robots [9], flying robots [10], etc. Although DEAs enable soft robots to achieve versatile movements, the development of such robots hitherto is still far from practical applications, and achieving accurate motion control remains a significant challenge. One of the difficulties in motion control of soft robots lies in obtaining precise kinematic/dynamic models of DEAs. These DEAs exhibit strong nonlinear characteristics, including time-variance, hysteresis, and viscoelasticity. To address this challenge, many efforts have been conducted for the control of DEAs, such as employing feedforward control to compensate for nonlinearities and vibration [11], and adopting feedback control for enhancing robustness and accuracy [12]. Another difficulty in motion control of soft robots involves the selection of motion behaviours during interactions with the environment. In [13], a simulation tool was adopted to describe the behaviour of several soft robots and therefore the motion controllers are successfully developed. Nevertheless, a general controller for soft robots that can select optimal gaits in real-time is still under exploration.

In this paper, the main objective is to design and develop a neuro-inspired motion control scheme for a DEA-driven soft myriapod robot. Here, a centipede-like robot constructed by multiple body segments connected in series is designed. Each body segment features a pair of antagonistic DEAs and four

This work was supported by the National Natural Science Foundation, China (No.62173300) and in part by A*STAR C211518005. (Corresponding author: *Wenyu Liang*).

Q. Ren and W. Zhu are with the College of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang, China.

J. Cao is with the Temasek Laboratories, National University of Singapore, Singapore 117411.

W. Liang is with the Institute for Infocomm Research, A*STAR, Singapore 138632.

electroadhesive feet. Moreover, one segment can be regarded as a soft myriapod robot which plays the role of the basic motion unit of the centipede-like robot. With the coordination of body deformation and foot adhesion, this soft myriapod robot can achieve basic straight and turning locomotion. Following that, a hierarchical neuro-inspired motion control scheme is proposed, which mimics the functions of the centipede brain. Previous studies on the motion function of the centipede brain focus on two important neuropils, namely, the mushroom body (MB) and the central complex (CX) [14]. In the motor process of a centipede, the MB is involved in learning, particularly of precisely timed motor movements [15], which can be considered as the adaptive sensory filter [16]. In addition, the CX plays an important role in the selection of motor actions [17] and the modulation of reward behaviour. It has been proven in the robotic experiments [18] that the models of the centipede brain are critical to motor-skill learning, and the neuro-inspired models have shown the adaptive motion capabilities over traditional control strategies [19], [20]. However, these insect-inspired neural models applied to robotics have been rarely studied.

To our best knowledge, this paper is the first study on controlling a soft mobile robot based on an insect brain-inspired model. The main contributions of this paper include: i) A neuro-inspired actuator controller is proposed to compensate for the nonlinearities of the DEAs in the actuation control loop. The controller consists of a model inversion-based linear controller and an adaptive MB-inspired forward model. ii) A neuro-inspired motion controller is developed to select the optimal locomotion for the soft robot movement tasks. Inspired by the reward-modulation mechanism of the CX model for motor-skill learning, a reinforcement learning-based (RL-based) method is proposed to autonomously prioritize control actions for the soft robot in dynamic environments. Without the accurate model of the soft myriapod robot, the desired movement tasks are implemented successfully in real-world scenarios. iii) Practical validation of the proposed neuro-inspired motion control scheme has been conducted on the soft myriapod robot.

The remaining of this paper is organized in the following. The development of the soft myriapod robot is presented and analyzed in Section II. Next, the proposed motion control scheme for the soft robot is detailed in Section III. Then, the numerical study and the experiments are conducted and their results are discussed in Section IV and Section V, respectively. Lastly, the conclusions are given in Section VI.

II. Robot Development

In this section, the robot prototype is introduced first and followed by the analysis of the robot locomotion.

A. *Robot Prototype*

Figure 1 illustrates the proposed centipede-inspired soft robot, which is constructed by multiple DEA-actuated body segments. In nature, the body segments are essential deformable components of biological centipedes to exhibit versatile motions. Hence, the objective of this paper is the design and control of the body segment of the centipede-inspired soft robot which can be regarded as a soft myriapod robot as well.

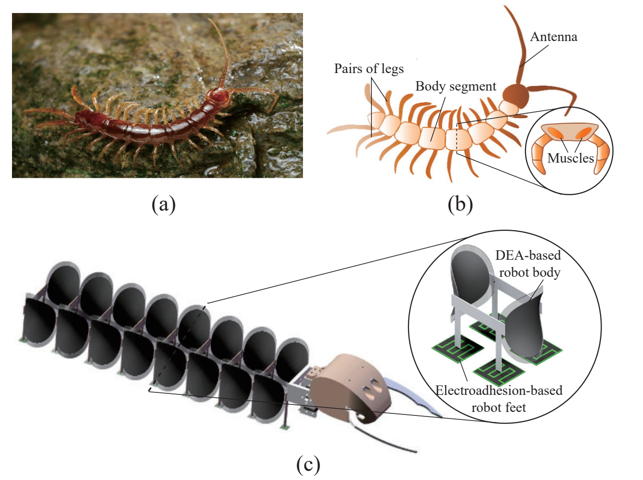


Fig. 1. Morphology and design of the soft robot: (a) biological centipede: the structure of a centipede with one pair of legs attached to each body segment; (b) morphological model: a pair of dorsal antagonistic muscles in the cross section; (c) prototype: a centipede-inspired modular soft robot constructed by multiple DEA-actuated soft robot units.

The soft myriapod robot is mainly comprised of one robot body and four robot feet. For the robot body, a pair of antagonistic DEAs is constructed in the parallel setup. Moreover, the endpoints of both DEAs are connected to two 3D-printed frames via four compliant joints. All the joints are designed to passively restore the angles between the DEAs and the frames to be orthogonal. For the robot feet, four electroadhesion actuators are designed to grasp the surface firmly when they are powered. Every two electroadhesion actuators are connected with one DEA, and they are located at the two end points of each DEA. Remarkably, the purposes of adding these robot feet in the soft robot segment are to provide the support of the DEAs and create the necessary friction for the robot motion.

1) DEA-Based Robot Body: Dielectric elastomers (DEs) are referred to as artificial muscles due to relatively large electrically induced deformations with fast response and high energy density [21]. DEs are generally thin films of elastomeric materials sandwiched between compliant electrodes. When applying a voltage to the electrodes, a compressive stress named Maxwell stress is induced by the resulting electric field. The incompressibility of DEs results in area expansion thickness contraction on the DEs.

The simple mechanism allows DEs to be tailored into soft actuators with different configurations. One way to fabricate DEAs is by attaching compliant elastic frames to pre-stretched DE membranes. The design and the general fabrication process of the DEAs used in this soft robot can be found in [22]. Two annular frames made of polyethylene terephthalate (PET) are attached to two membranes made of VHB 4910 films. The two frames are with the same shape but different thicknesses, where the inner diameters are 75 mm, the outer diameters are 90 mm, and the thicknesses are 0.4 mm and 0.8 mm, respectively. When a voltage is applied, the strain energy of the membrane increases, which results in a decrease of the

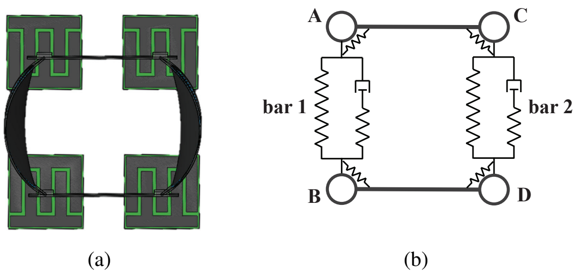


Fig. 2. Soft robot model: (a) robot top view; (b) pseudo-rigid-body model.

PET frames' elastic energy, and the actuator thus expands.

2) Electroadhesion-Based Robot Feet: Among the adhesion techniques, electroadhesion is more promising than other techniques such as suction cup, magnetic adhesion, mechanical grasping (e.g., spines, claws) and elastomeric adhesion (e.g., gecko skin). The reason lies in the following advantages of electroadhesion: suitable for various surfaces of different conditions, electronically controllable operation, low power consumption, lightweight, and no noise [23]. Electroadhesion works by applying a voltage to an array of conductive electrodes embedded in a dielectric.

For the robot feet, flexible material and a comb electrode pattern are used in the design and fabrication to ensure strong electroadhesion performance on different surfaces. The robot feet are made up of a multi-layer structure with paper, graphite electrode and Ecoflex™ rubber. The multi-layer structure is designed for stable electrical adhesion. When connected to a voltage, two electrodes accumulate a positive and a negative charge. The induced charge on the floor is not neutralized with the induced charge on the electrode due to the presence of the dielectric layer of the paper and the captured air gap between the papers. These opposite charges attract each other, resulting in an electrostatic force between the actuator and the floor surface. Thus, the adhesion force is the electrostatic force between the opposite charges, which is generated once the electrodes are powered. With the proper design, the generated adhesion force can be strong enough to ensure the robot foot be attached to the surface firmly without any slipping, which also makes the robot capable of climbing slopes and even vertical walls.

B. Locomotion Generation

To explore the locomotion of the designed soft robot, the pseudo-rigid-body model is used as illustrated in Fig. 2. The designed soft robot can be represented as a four-bar linkage, where the compliant mechanisms can be represented as the appropriately located pin joints and the torsional springs. The torsional springs represent the compliant joints with resistance to motion, which will help the four-bar linkage restore to the original rectangular shape if the soft robot is powered off. Significantly, this soft robot actually can be further simplified to a simple four-bar linkage with mechanical constraints, where two adjustable-length bars, namely bar 1 and bar 2, represent the DEAs. The simplified model will be used to illustrate the following soft robot locomotion analysis.

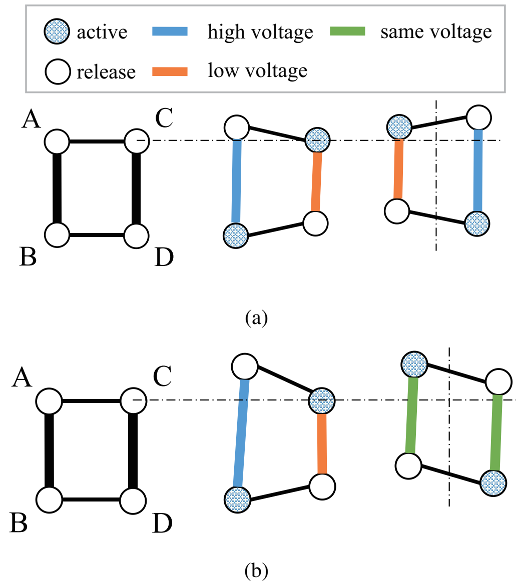


Fig. 3. Linear walking locomotion: (a) straight move; (b) slight torsion. (colour images are available online)

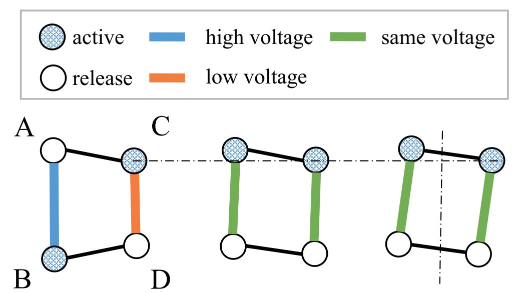
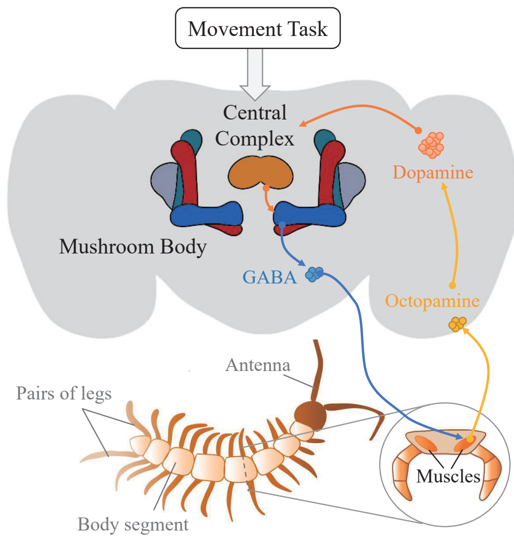


Fig. 4. Turning locomotion. (colour images are available online)

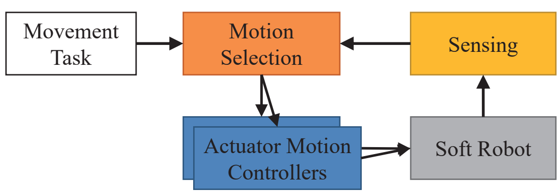
Inspired by the locomotion of centipedes, two fundamental modes of locomotion are demonstrated as followings: linear walking and turning locomotion. Each locomotion is first divided into several stages, and then the body configuration and feet anchoring strategy are designed for each stage.

1) Linear Walking Locomotion: In the linear walking gait, the robot feet are paired diagonally. As can be seen from Fig.3, the linear walking locomotion involves two steps: in step 1, one diagonal pair of feet (feet pair A-D) is moved simultaneously while the other pair (feet pair B-C) is kept still by activating the corresponding electroadhesion actuators; in step 2, the feet pair B-C is moved while the feet pair A-D is remained not moving. Cycling the actuation sequence, the soft robot walks continuously. Considering the structure of the designed soft robot, the phase shift of the two DEAs is 180°. Furthermore, the soft robot will achieve an approximately straight move while the step sizes of both DEAs are the same, as depicted in Fig. 3(a). On the other hand, the soft robot will have a slight torsion while the step sizes of both DEAs are different, as depicted in Fig. 3(b).

2) Turning Locomotion: In the turning gait, the robot feet are paired according to the connecting rod. As illustrated in Fig. 4, the effective turning locomotion involves two steps: In step 1, the pair of feet located at the front bar (feet pair A-C) is free to move while the pair (feet pair B-D) is kept still. In



(a)



(b)

Fig. 5. A block scheme of motion control: (a) biological scheme in motor control; (b) overview of the control scheme for the soft robot

step 2, the pair of feet located at the front bar (feet pair A-C) is kept still while the pair (feet pair B-D) is free to move. Remarkably, the soft robot can achieve different turning angles through different elongations/contractions of the two DEAs in step 2.

III. Motion Control

Figure 5(a) illustrates the motor information flow within an arthropod brain. First, the movement task is sent to the central complex, which is modulated by dopamine (influenced by sensory information) to select motor commands. Then, this motor command is sent to the mushroom body for precise motor control of the arthropod. Inspired by the biological motor control system, a bioinspired motion control scheme is proposed in Fig. 5(b), the main purpose of which is to supervise the designed soft robot to reach the target position with accurate movements (steering or forward). According to the study on the biological system, mimicking the reward-based learning mechanism of the insect central complex, the motion selection module is designed to manage and select the optimal locomotion according to the robot movement task and the proprioception (i.e., the sense of body position). At the low level, based on the error-based learning mechanism of the insect mushroom body, several actuator motion controllers are utilized to control precise actions and movements of DEAs and robot feet.

In a nutshell, the desired moving target location and the measured proprioception information are processed as *state*. Following that, the motion selection module receives such state information and computes the corresponding *action*, i.e., the optimal motion commands. Lastly, the actuator motion control module sets this action as the desired action and coordinates the DEAs and robot foot actuators to achieve the desired action.

A. Actuator Motion Control

The function of the actuator motion control module has two parts: i) motion control of two DEAs, and ii) actuation control of four robot feet, where an on/off controller is designed to manage when and which robot feet are needed to be actuated. Significantly, due to the strong nonlinearities and viscoelasticity of soft actuators, the main challenge of the actuator motion control module lies in the control of the DEAs, thus becoming the focus of this subsection. Since a reliable soft actuator model is essential for control design and experimental analysis, the soft actuator modelling and the controller design for the DEAs, are detailed as follows, respectively.

1) Actuator Modelling: The distance from one endpoint of a single DEA to the other endpoint is defined as the length of the DEA, namely L, and then, a DEA model that represents the relationship between the length change of the DEA and the applied force can be built. According to [22], the DEA model can be the combination between the model of the PET frames and the model of the DE membranes. Thus, the DEA model can be described by

$$m\ddot{z}(t) = -kz(t) - \sum_{i=1}^{4} k_i z_i(t) - \delta(z(t),\dot{z}(t)) + G(z(t))u(t),$$
$$\dot{z}(t) = \frac{k_i}{c_i} z_i(t) + \dot{z}_i(t),\ \ y(t) = z(t), \tag{1}$$

where m is the actuator's inertia, k, k_i, and c_i are the model parameters, z_i is the length change of the i-th spring, z is the length change of the actuator, y is the system output, $\delta(z,\dot{z})$ is the nonlinear term of the system, and $G(z(t))u(t)$ is the actuation force applied to the DEA, $G(z(t)) = az(t) + b$ is a function related to the length change of the actuator with two coefficients a and b, $u = \mathcal{V}^2$ is the control input and $\mathcal{V}$ is the voltage input to the DEA.

Then the state-space form of the model can be equivalently obtained, which is

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + Bu + D\delta(\mathbf{x}) \\ y &= C\mathbf{x} \end{aligned} \tag{2}$$

with

$$A = \begin{bmatrix} -k_1/c_1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -k_2/c_2 & 0 & 0 & 0 & 1 \\ 0 & 0 & -k_3/c_3 & 0 & 0 & 1 \\ 0 & 0 & 0 & -k_4/c_4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -k_1/m & -k_2/m & -k_3/m & -k_4/m & -k/m & 0 \end{bmatrix},$$

$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \mathcal{G}(\mathbf{x}) \end{bmatrix}^T$, $D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}^T$, and $C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$,
where $\mathbf{x} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z & \dot{z} \end{bmatrix}$ is the system state.

2) Motion Controller Design: On the basis of the system model, an inversion-based nonlinear feedforward controller can be designed in (3) so that the system can follow the desired reference.

$$u_{ff}(t) = \mathcal{P}^{-1}(y_d(t)), \tag{3}$$

where y_d is the desired reference, u_{ff} is the feedforward controller output, $\mathcal{P}$ represents the forward model of the system (i.e., $y(t) = \mathcal{P}(u(t))$) and $\mathcal{P}^{-1}$ represents the system's inverse model.

Significantly, due to the system nonlinearity, it is difficult to obtain an accurate system model and design the nonlinear feedforward controller as described in (3). Therefore, the actuator motion controller is designed to approximate such a nonlinear function $P(\cdot)^{-1}$.

Figure 6 illustrates the proposed motion control scheme for a single DEA. This control scheme mainly consists of two parts: an inversion-based linear feedforward controller and a mushroom-body-inspired forward model. For the forward model, an adaptive learning filter inspired by the mushroom body of the insects is designed to compensate for the nonlinearities of the DEAs. Meanwhile, the adaptive filter acted in parallel with an approximate plant inverse model controller to ensure stability in learning and control.

Significantly, considering that a high-level overall feedback loop is available above the actuator motion control module, only a feedforward control mechanism is used in the actuator motion controller design to achieve a fast response without considering the feedback loop.

Inversion-Based Linear Feedforward Controller

Referring to (1) and (2), the DEA can be approximately described by the following high-order transfer function when the complex geometry is ignored. In the previous work shown in [24], the PET frames can be modelled as a high-order system, converting from the state-space model shown in (2).

$$G_n(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{i=0}^{4} a_i s^i}{\sum_{j=0}^{6} b_j s^j}, \tag{4}$$

where $G_n(s)$ is the approximate linear model of the DEA, $U(s)$ is the input to the DEA, $Y(s)$ is the output length change, a_i and b_j are the coefficients of the linear model.

Based on the linear model shown in (4), the following feedforward controller is designed,

$$\hat{u}_{ff}(t) = q_{ff}(t) g_n^{-1}(t) x_d(t), \tag{5}$$

where q_{ff} is a low-pass filter to make the feedforward controller be strictly proper, $g_n^{-1} = \mathcal{L}^{-1}(G_n^{-1})$ is the approximate inverse model which is the inverse Laplace transform of G_n^{-1},

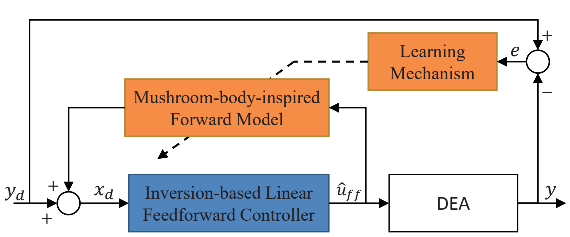


Fig. 6. Motion controller of the single DEA

$\hat{u}_{ff}$ is the linear feedforward controller output, and x_d is the input to the controller.

Furthermore, the designed feedforward controller can be described in the following form,

$$\hat{u}_{ff} = \mathcal{B}(x_d). \tag{6}$$

Mushroom-Body-Inspired Forward Model

The basic function of the mushroom body is to process several modalities of sensory information and regulate locomotion. Figure 7(a) illustrates a simplified schematic of the key cellular components and information flow in the process of sensory inputs to the mushroom body. In the insect brain, the sensory information is delivered to the mushroom body by projection neurons (PNs) from the antennal lobe (AL). A small number of PNs connect with the Kenyon cells (KCs) whose parallel axonal fibres form the MB lobes (including α/β lobe and α'/β' lobe), where KCs terminate onto the dendrites of mushroom body output neurons (MBONs). In the mushroom body, instructive/modulatory signals are provided by the dopaminergic neurons (DANs), and dopamine is thought to act locally to modify KC-MBON synapses.

The mushroom-body-inspired adaptive filter control scheme is shown in Fig. 7(b), which consists of three layers. These three layers are the input layer **U**, the association cell layer **K**, and the output layer **M**. Therefore, the controller can be represented by a pair of mappings mathematically, which are

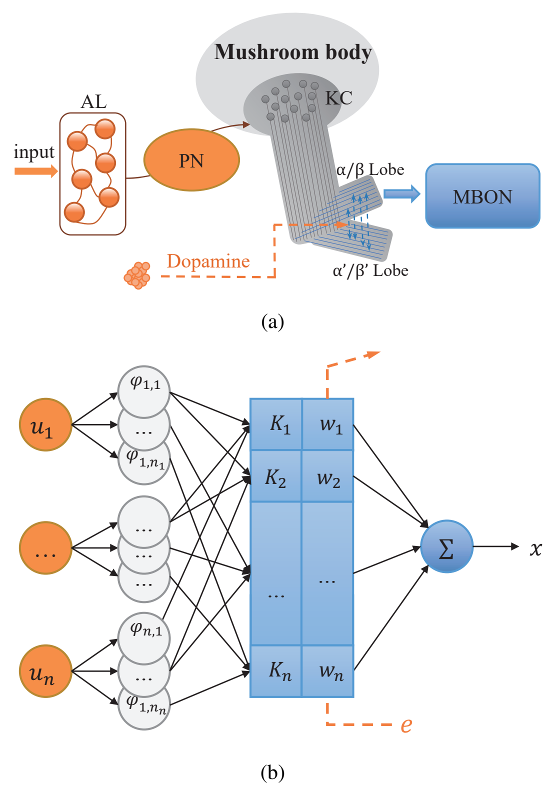


Fig. 7. The mushroom-body-inspired adaptive filter control scheme: (a) a simplified schematic of the key cellular components and information flow in the process of sensory inputs to the mushroom body; (b) the basic structure of the mushroom-body-inspired adaptive filter controller.

$m_I(\mathbf{U} \Rightarrow \mathbf{K})$ and $m_O(\mathbf{K} \Rightarrow \mathbf{M})$:

$$\begin{tikzcd} \mathbf{U} \arrow[r, Rightarrow, "\text{PN}"] & \mathbf{K} \arrow[r, Rightarrow, "\alpha/\beta Lobe"] & \mathbf{M} \\ & & e \arrow[u, "\text{DAN}"'] \end{tikzcd} \quad , \tag{7}$$

where $\mathbf{U}$, $\mathbf{K}$ and $\mathbf{M}$ are the vectors defined as

$$\begin{aligned} \mathbf{U} &:= \{\text{input vector}\}, \\ \mathbf{K} &:= \{\text{association cell (Kenyon cell, KC) vectors}\}, \\ \mathbf{M} &:= \{\text{output (Mushroom body output neuron, MBON) vector}\}, \end{aligned} \tag{8}$$

and e is the instructive signal (representing the DAN).

For the input layer, the input vector $\mathbf{U} = \{u_1, u_2, \cdots, u_n\}$ are set related to the linear feedforward controller output in this paper.

For the association cell layer, the Gaussian function given in (9) is employed as the basis function to locate the regions that are needed to be activated corresponding to the current input vectors,

$$\varphi_{p,q_p}(u_p) = \exp\left(-\frac{(u_p - \mu_{p,q_p})^2}{\sigma^2_{p,q_p}}\right), \quad \text{for } p = 1,2,...,n \text{ and } q_p = 1,2,...,n_p, \tag{9}$$

where $\varphi_{p,p_p}(u_p) \in [0,1]$ are the distance of the p-th input u_p to the q_p-th region with center location at μ_{p,q_p} (i.e., mean) and the width of σ_{p,q_p} (i.e., variance). Remarkably, if $\varphi_{p,q_p}(u_p)$ is closer to 1, u_p is nearer to μ_{p,q_p} and the corresponding cell will be more strongly activated.

Let $\phi_p(u_p) = \begin{bmatrix}\varphi_{p,1} & \varphi_{p,2} & \dots & \varphi_{p,n_p}\end{bmatrix}^T$, then the high-dimensional basis function (two inputs for illustration in the following equations) is represented by the below vector form:

$$\phi(\mathbf{u}) = \phi_1 \otimes \phi_2 = \begin{bmatrix}K_1 & K_2 & \dots & K_N\end{bmatrix}^T, \tag{10}$$

where $N = n_1 \cdot n_2$.

According to (10), a map that contains N regions has been constructed. The location related to the desired reference described by $\mathbf{U}$ in the map can be determined via (10). Furthermore, an unique weight w_l for $l = 1,2,...,N$ in the weight memory $\mathbf{w}$ connects to each region.

For the output layer, the neural network output is expressed in the following general form:

$$\mathcal{M}(\hat{u}_{ff}) = \sum_{l=1}^{N} w_l K_l = \mathbf{w}^T \phi. \tag{11}$$

Lastly, to minimize the mean square error (MSE) between the desired value and the actual value, the gradient descent algorithm is applied to the following loss function (12) for updating the weights in the neural network.

$$J(\mathbf{w}) = \frac{1}{2}(y_d - y)^2 = \frac{1}{2}e^2. \tag{12}$$

In the following, the update rule is derived using the stochastic gradient descent algorithm.

$$\mathbf{w}_{v+1} = \mathbf{w}_v - \lambda \nabla J(\mathbf{w}), \tag{13}$$

where $\lambda > 0$ is the learning rate, $v \geq 0$ and $(v+1)$ denote the v-th and the $(v+1)$-th iterations, respectively. Here, the gradient of $J(\mathbf{w})$ is given by

$$\nabla J(\mathbf{w}) = \begin{bmatrix}\frac{\partial J(\mathbf{w})}{\partial w_1} & \frac{\partial J(\mathbf{w})}{\partial w_2} & \cdots & \frac{\partial J(\mathbf{w})}{\partial w_N}\end{bmatrix}. \tag{14}$$

From Fig. 6 and (6), we can have

$$\hat{u}_{ff} = \mathcal{B}(y_d + \mathcal{M}(\hat{u}_{ff})). \tag{15}$$

Since the inverse of $\mathcal{B}(\cdot)$ is existed, then

$$y_d = \mathcal{B}^{-1}(\hat{u}_{ff}) - \mathcal{M}(\hat{u}_{ff}). \tag{16}$$

Assume that there exist optimal weights $\mathbf{w}^*$ to construct an optimal neural network $\mathcal{C}^*$ such that the actual system output y matches the desired reference y_d, i.e., $y = y_d$, then we have

$$y = \mathcal{P}(\hat{u}_{ff}) = \mathcal{B}^{-1}(\hat{u}_{ff}) - \mathcal{C}^*(\hat{u}_{ff}). \tag{17}$$

From (16) and (17), the error e can be represented by

$$e = y_d - y = \mathcal{C}^*(\hat{u}_{ff}) - \mathcal{C}(\hat{u}_{ff}) - = (\mathbf{w}^* - \mathbf{w})^T \phi. \tag{18}$$

Thus, we have

$$\frac{\partial e}{\partial w_l} \approx -K_l, \text{ and } \frac{\partial J(\mathbf{w})}{\partial w_l} = \frac{\partial J(\mathbf{w})}{\partial e}\frac{\partial e}{\partial w_l} \approx -eK_l. \tag{19}$$

Hence, the approximate gradient descent learning law is given by

$$\mathbf{w}_{v+1} = \mathbf{w}_v + \lambda e \phi. \tag{20}$$

Remarkably, the nonlinear feedforward controller (3) can be learned and approximated by applying mushroom-body-inspired controller, i.e.,

$$\mathcal{P}^{-1}(y_d(t)) \approx \mathcal{B}(y_d(t) + \mathbf{w}^T \phi). \tag{21}$$

B. Reinforcement Learning-Based Robot Motion Selection

As shown in Fig. 8, the robot target reaching task can be divided into two phases: the steering phase and the forward-moving phase. In each phase, two types of locomotion presented in Section II are selected properly with the motion selection module to achieve the movement task. The steering phase is the first phase, where the walking locomotion and turning locomotion are arranged properly so that the orientation of the soft robot can approach the desired orientation that is parallel to the line from its centre to the target location. Then, the soft robot can approach the target location by walking locomotion in the forward-moving phase.

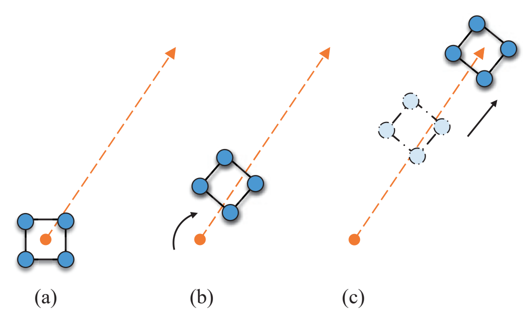


Fig. 8. Motion selection: (a) start location; (b) steering phase; (c) forward-moving phase (approaching to the target location)

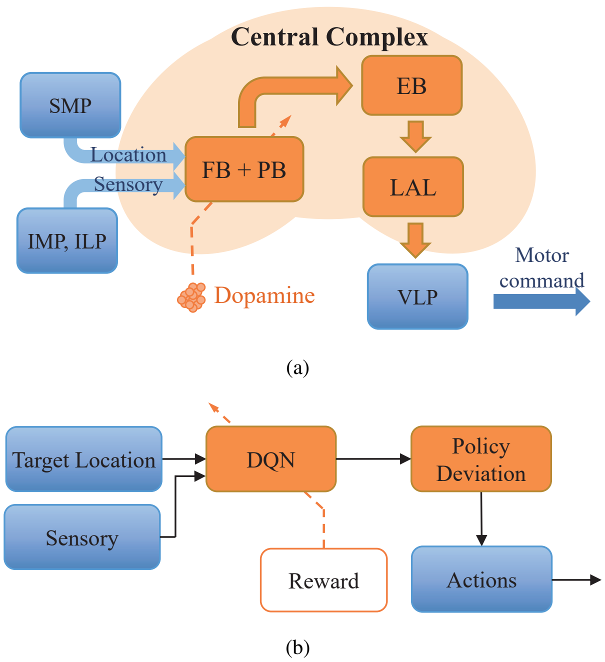


Fig. 9. The DQN-based motion selection controller: (a) a simplified schematic of the key cellular components and information flow in the process of motion selection of the arthropod central complex; (b) the structure of the DQN-based motion selection controller.

For the target-reaching task of the designed soft robot, the main challenge lies in the locomotion arrangement in the steering phase, which requires to design of the motion selection module. As pointed out from [25], the selection and maintenance of adaptive behavioural actions have been attributed to the arthropod CX, which consists of three interconnected midline centres: the protocerebral bridge (PB), fan-shaped body (FB), and ellipsoid body (EB) leading to the paired lateral accessory lobes (LAL). Figure 9(a) illustrates the key cellular components and information flow in the arthropod CX. The inputs of the CX derive from the location information from associative superior medial protocerebrum (SMP) and the sensory information from intermediate and inferior lateral protocerebra (IMP, ILP). These inputs are then delivered to the fan-shaped body and protocerebral bridge (FB+PB), which are involved in the modulation of reward behaviour with dopamine. The behaviour selection is executed by the EB and through LAL sent to the ventrolateral protocerebra (VLP), which plays the role of the motor centre. Inspired by the arthropod CX, an RL method has been proposed for the autonomous skill learning of the soft robot, which has been proven efficient for robotic systems [26]. Figure 9(b) illustrates the structure of the Deep Q-Network-based (DQN-based) motion selection controller, and the design details are as follows.

1) Reinforcement Learning Design: The objective of the RL is to find an optimal policy π^* that helps the soft robot select and take an action a_ι corresponding to different states. Then, the robot transits from the current state s_ι to another state $s_{\iota+1}$ and receives a reward $r_{\iota+1}(s_\iota, a_\iota)$. Therefore, the policy that maximizes the expectation of the cumulative discounted reward R_ι defines the optimal policy π^*, which can be represented by

$$R_\iota = \sum_{\tau=0}^{T} \gamma^t r_{\tau+\iota+1}(s_\iota, a_\iota), \tag{22}$$

where γ is the discount factor. Also, the optimal policy can be expressed as

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}[R_\iota|\pi], \tag{23}$$

where π denotes the policy.

In RL, a function can be used to measure the quality $Q^\pi(s_\iota, a_\iota)$ of a specific state-action pair $(s_\iota,\, a_\iota)$ under a given policy π, which is

$$Q^\pi(s_\iota, a_\iota) = \mathbb{E}[R_\iota|s_\iota, a_\iota, \pi]. \tag{24}$$

From (24), the optimal policy is now the policy that maximizes the values of (24) over all possible state-action pairs (s, a). Hence, an optimal function denoted by Q^* is defined, which is the maximum expected cumulative reward achievable from a given state-action pair (s_ι, a_ι):

$$Q^*(s_\iota, a_\iota) = \max_{\pi} Q^\pi(s_\iota, a_\iota). \tag{25}$$

Significantly, $Q^\pi(s_\iota, a_\iota)$ satisfies the Bellman optimality equation and thus (25) can be rewritten as

$$Q^*(s_\iota, a_\iota) = \mathbb{E}[r_{\iota+1}(s_\iota, a_\iota) + \gamma \max_{a_{\iota+1}} Q^*(s_{\iota+1}, a')|s_\iota, a_\iota], \tag{26}$$

where a' is the possible action taken at state $s_{\iota+1}$.

2) Deep Reinforcement Learning Design: To achieve the optimal function (25), an online learning neural network can be used to approximate the function (24) iteratively via a learning mechanism.

Define an approximation to (24) as $Q(s_\iota, a_\iota; \theta)$ with a set of parameters θ, then the objective of the neural network is to approximate the optimal function (25), i.e.,

$$Q(s_\iota, a_\iota; \theta) \approx Q^*(s_\iota, a_\iota). \tag{27}$$

Here,

$$Q(s_\iota, a_\iota; \theta) = \bar{\mathbf{w}}^T \bar{\phi}, \tag{28}$$

where $\bar{\mathbf{w}}$ is the Q-network weights, $\bar{\phi}$ is the Q-network basic functions.

Furthermore, such an objective can be converted to the optimization problem of minimizing the below loss function.

$$L_n(\theta_n) = \frac{1}{2}\mathbb{E}[(\upsilon_n - Q(s_\iota, a_\iota; \theta_n))^2] \tag{29}$$

with

$$\upsilon_n = r_{\iota+1} + \gamma \max_{a'} Q(s_{\iota+1}, a'; \theta_{n-1}). \tag{30}$$

where n is the number of iterations.

Significantly, because the successive samples are correlated, combining the neural network into the simple Q-learning can result in unstable or divergence. To solve this problem, a simple framework is employed referring to the methods introduced in [27]. In this framework, υ_n is modified and used in the following,

$$\nabla_{\theta_n} L_n(\theta_n) = -(\upsilon_n - Q(s_\iota, a_\iota; \theta_n))\nabla_{\theta_n} Q(s_\iota, a_\iota; \theta_n). \tag{31}$$

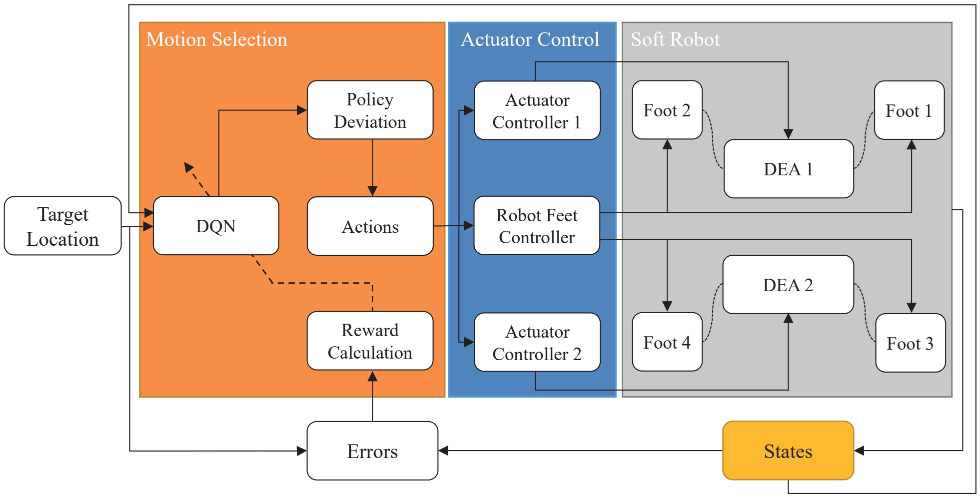


Fig. 10. Detailed overall motion control scheme

Then, the gradient $\nabla_{\theta_n} Q(s_\iota, a_\iota; \theta_n)$ at each weight $\bar{w}_l$ for $l = 1, 2, ..., N$ is given by

$$\nabla_{\theta_n} Q(s_\iota, a_\iota; \theta_n) = \left[\frac{\partial Q(s_\iota, a_\iota; \theta_n)}{\partial \bar{w}_1} \;\; \frac{\partial Q(s_\iota, a_\iota; \theta_n)}{\partial \bar{w}_2} \;\; \cdots \;\; \frac{\partial Q(s_\iota, a_\iota; \theta_n)}{\partial \bar{w}_N} \right]^T . \tag{32}$$

Hence, the update rule is given by

$$\begin{aligned} \bar{\mathbf{w}}_{n+1} &= \bar{\mathbf{w}}_n + \lambda \left(\upsilon_n - Q(s_\iota, a_\iota; \theta_n) \right) \bar{\phi}, \\ \upsilon_n &= r_{\iota+1} + \gamma \max_{a'} Q(s_{\iota+1}, a'; \bar{\theta}_n), \end{aligned} \tag{33}$$

where $\bar{\theta}_n$ is the target parameter used in the computation of the Q-function at the n-th iteration. Such a target parameter is only updated periodically every D step with θ_n. The reinforcement learning law implemented by updating the weight $\bar{\mathbf{w}}$ according to the previous value in the last iteration.

For the designed soft robot, an action set A ($a_\iota \in A$) is designed elements which include 1 type of turning locomotion, and 21 types of walking locomotion with various step size settings on DEAs (i.e., the same step size and different step sizes). Furthermore, the reward r is defined by the following error function.

$$r = \begin{cases} r_{max}, & \text{if } ||\beta_L|| \le \delta_L \text{ and } ||\beta_{AC}|| \le \delta_{AC} \\ -p_1 ||\beta_L||^2 - p_2 ||\beta_{AC}||^2, & \text{otherwise} \end{cases} \tag{34}$$

where r_{max} is a positive constant, β_L is the angle between the desired orientation and the axis of the left DEA ($\overline{AB}$), β_{AC} is the angle between the direction perpendicular to the desired orientation and the front bar ($\overline{AC}$), δ_L and δ_{AC} are two costive constants that represent the steering requirements, and p_1, p_2 are the weightings, It worths noting that the goal is to minimize β_L and β_{AC} as small as possible. Moreover, a designed maximum reward will be earned when both β_L and β_{AC} are extremely small that are within the acceptable ranges.

Additionally, the ε-greedy exploration is employed in the RL algorithm to balance exploitation and exploration.

C. Overall Motion Selection and Control Framework

Finally, by integrating both the robot motion selection and the actuator motion controllers, the overall motion control scheme is detailed in Fig. 10.

IV. Numerical Study

To verify the effectiveness of the proposed motion control scheme, the numerical study is carried out by simulations with model simplification in this section. The simulations are run in MATLAB and Simulink with a sampling time of 1 ms.

A. Actuator Motion Control

To implement the motion controller and test the motion controller in simulation, the model parameters shown in (4) are identified based on the collected data on one DEA via several experiments. In the experiments, one end of the DEA is fixed while the displacement of the other end is measured by a laser sensor so that the length change can be measured and recorded. Then, a square wave with a frequency of 0.05 Hz and an amplitude of 2 to 4 kV is applied to the DEA for actuation.

With the help of the MATLAB System Identification Toolbox, the model parameters are estimated via the input and output data. The identified transfer function is given by (35) which is also validated by two sine wave input signals with the frequency of 0.05 and 0.1 Hz, respectively, and the same amplitude of 2 to 4 kV.

$$G_n(s) = \frac{1.14s^4 + 97.12s^3 + 13.82s^2 + 16.03s + 0.78}{s^6 + 11.97s^5 + 186.20s^4 + 252.40s^3 + 58.49s^2 + 36.74s + 1.26} \tag{35}$$

The fit level of the model means the goodness of fit or coefficient of determination, which represents the error norm between test and reference data sets. The identified linear model fits the actual output over 80%. It implies that the identified linear model can mostly represent the DEA. However, there exist differences between the model and the DEA. These

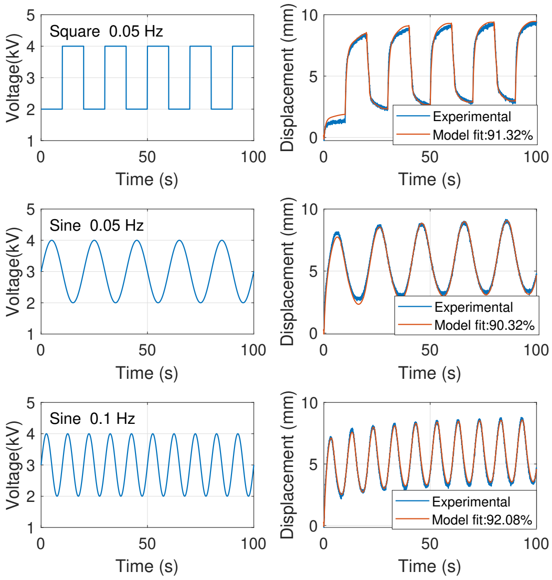


Fig. 11. System identification and model validation of DEA (colour images are available online).

remaining differences are mainly due to the nonlinearity in the DEA. Significantly, a long-short-term memory (LSTM) neural network is built to approximate these remaining differences (i.e., nonlinearity) and combined with the linear model as a whole model in the simulation. As shown in Fig. 11, this combined model can match and represent the actual system in a higher fit level (over 90%).

The simulation results with the desired sine trajectories of 0.1 Hz and 0.2Hz are illustrated in Fig. 12 and Fig. 13, respectively. As can be seen, iterations are needed for the DEA to learn the control signal. However, in just a few iterations, the error converges to be less than 0.5 mm and the learning speed of the proposed controller is fast and acceptable. Furthermore, it can be observed from the figures that the proposed control scheme can track the desired sine wave trajectory better than the linear feedforward controller.

B. Motion Selection

In this subsection, the proposed RL-based motion selection module is trained. To train the Q-network, a set of desired orientations $B_d = \{-30°, -20°, -10°, 10°, 20°, 30°\}$ is chosen in the training. In each episode during the training, the desired orientation is randomly chosen from the set B_d. By applying the proposed RL method in the system model, the training results are shown in Fig. 14.

As can be seen from the results, both the average reward and the norm of network weight vector $||\mathbf{w}||$ can converge after certain episodes. Significantly, it can be found that the average reward starts from a negative value and then is increased gradually under the learning algorithm. Therefore, it can be concluded that the RL is effective in motion selection, which is able to help the robot to generate a good locomotion sequence among different locomotion to achieve the desired steering angle.

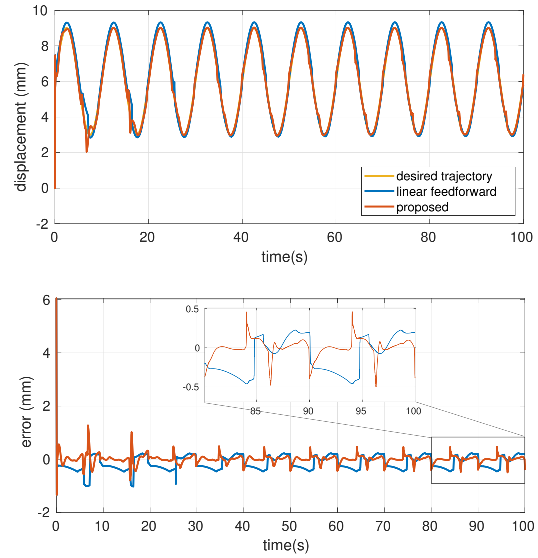


Fig. 12. Simulation results on sine wave trajectory (0.1 Hz) tracking (colour images are available online).

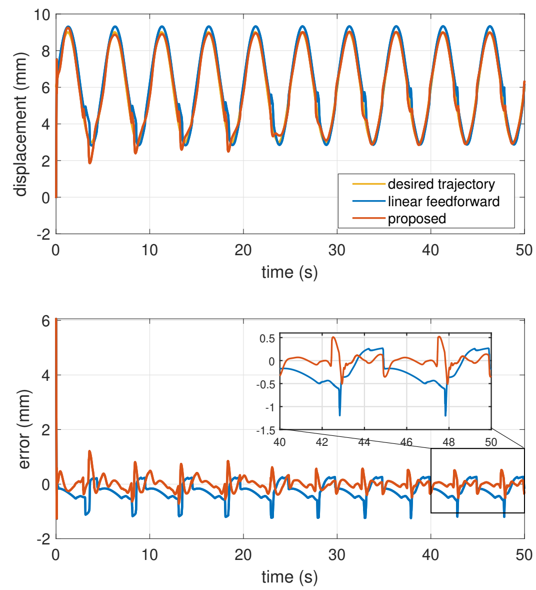


Fig. 13. Simulation results on sine wave trajectory (0.2 Hz) tracking (colour images are available online).

C. Movement Task Test

Furthermore, in this simulation, a target location is set and the two-phase motion strategy mentioned in Section III-B is implemented. In the steering phase, the trained network is applied and the robot action is selected according to the criteria shown as follows:

$$a_t = \operatorname*{argmax}_a Q(s_t, a; \theta). \tag{36}$$

After the robot orientation meets the steering requirement, the robot will be switched into the forward moving phase automatically, where the straight moving trot gait is implemented.

Figure 15 shows the simulation result of the movement task test by using the proposed motion selection module. As can be seen, the robot is able to achieve the desired steering angle and then reach the target location successfully. This result verifies that the proposed scheme is effective in managing and regulating the soft robot to conduct the desired planar movement accurately.

In addition, the proposed motion selection method is compared with a method named "naive motion selection method" which combines one walking locomotion with one turning locomotion as a body-turn motion primitive in the steering phase. In this selection method, the selection of walking locomotion follows the idea of feedback control where the robot selects the control strategy corresponding to the steering angle closest to the angle between the body direction and the target direction. As illustrated in Fig. 15, the comparison results show that the proposed motion selection method is faster than the naive motion selection method in the steering phase (i.e., the proposed method requires fewer steps in the steering phase).

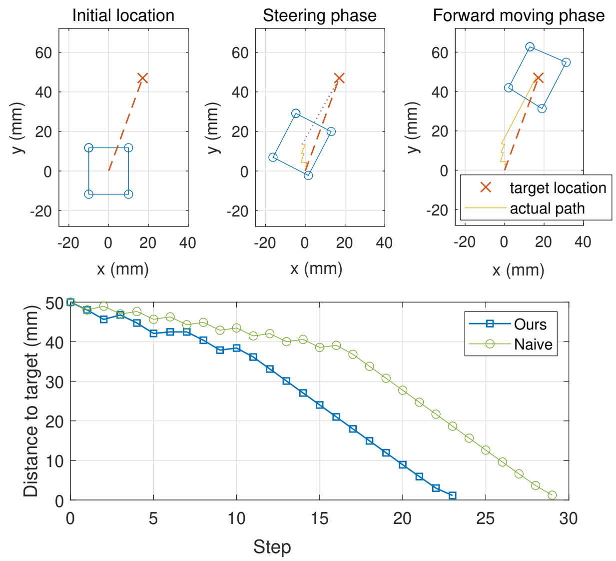


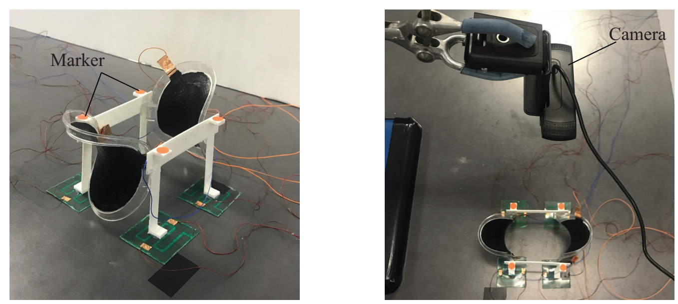
Fig. 15. Simulation result of movement task (top: the process of target-reaching for the soft robot using the proposed method, where the motion strategy is divided into the steering phase and the forward-moving phase; bottom: comparison results in the steering phase).

V. Experiments and Results

To validate the effectiveness of the proposed motion control scheme in the practical robotic system, several experiments have been conducted with the soft myriapod robot.

A. Experimental System Setup

Figure 16 illustrates the configuration of the experiment system. In the experiments, each DEA is driven by an adjustable high-voltage amplifier while each foot is powered by a DC-DC converter, and the soft robot motion is controlled by a dSPACE control board (DS-1104) embedded in a computer. Furthermore, four orange markers are attached to the four connection points between the robot bodies and feet while a high-definition camera (Logitech C922 Pro) is placed on top of the robot to build a vision system that can capture the feedback of the robot motion in real-time. The camera is connected to another computer where the vision algorithm and the motion selection module are implemented via MATLAB. Finally, the motion commands determined by the motion selection module will be sent to the dSPACE control board (as the desired motion references to the DEAs). For the motion selection module, the network parameters obtained in the simulation are used in this practical system.



(a) (b)

Fig. 16. Soft robot and experimental system setup: (a) soft robot; (b) experimental system setup

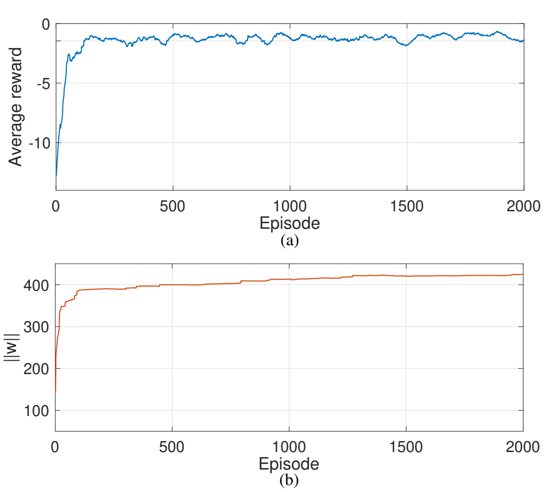


Fig. 14. Learning network training results: (a) average reward (processed with moving average filter); (b) norm of the network weight vector.

B. Experimental Results

Two experiments are conducted and the results are discussed in the following.

1) Steering: In this experiment, the movement task of the soft robot is to steer itself to the desired orientation, i.e., the DEAs need to be parallel to the desired orientation. The frequency of each step is set at 0.2 Hz. Figure 17 shows the experimental results while the desired orientation is 10 deg.

As can be seen from the results, both DEAs can turn to the desired orientation by the proposed framework, which implies that the developed soft robot is capable of achieving the steering task. The experimental results also verify that the motion selection and control framework is effective in the steering task and orientation regulation.

2) Point-to-Point Movement: In the second experiment, a point-to-point movement task is set to the soft robot, where there is a steering angle between the y-axis and the line connecting the target location and the origin. Both the steering phase and the forward-moving phase are implemented in this

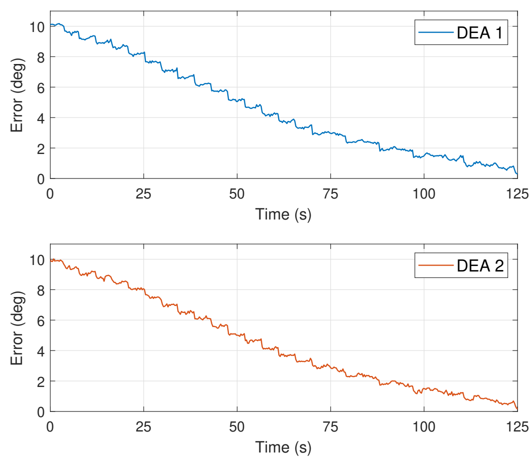


Fig. 17. Experimental results of steering task

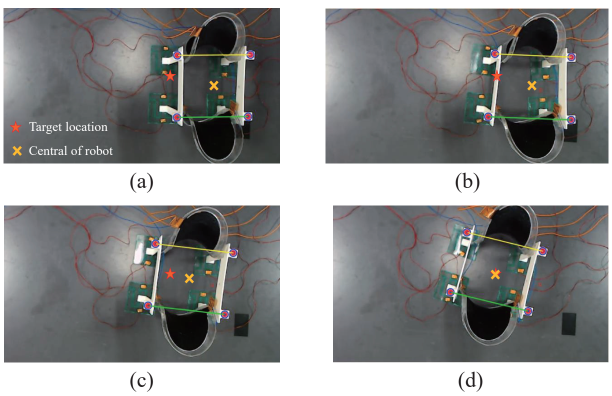


Fig. 18. Sequential view during the point-to-point movement task: (a) at the origin; (b) walking locomotion (tort gait); (c) turning locomotion; (d) at the target location.

experiment, where the steering phase will automatically switch to the forward-moving phase when the steering angle error is within the preset acceptable range.

Here, the steering angle is selected as 15 deg and the distance-to-target is set as 100 mm in the experiments. The sequential images during the point-to-point movement task and the experimental results of the point-to-point movement are depicted in Fig. 18 and Fig. 19, respectively.

As can be observed in both figures, it is obvious that the soft robot can reach the target location accurately. During the point-to-point movement task, the soft robot first aligns its orientation with a proper sequence of the different locomotion generated by the proposed motion selection module, hence, it can head towards the target location. After that, the soft robot moves straightly towards the target location by the forward walking locomotion. The success of the point-to-point movement task proves the effectiveness of the proposed control scheme in this soft mobile robot.

VI. Conclusion

In this paper, a soft myriapod robot that is capable of realizing the planar movements is designed based on two

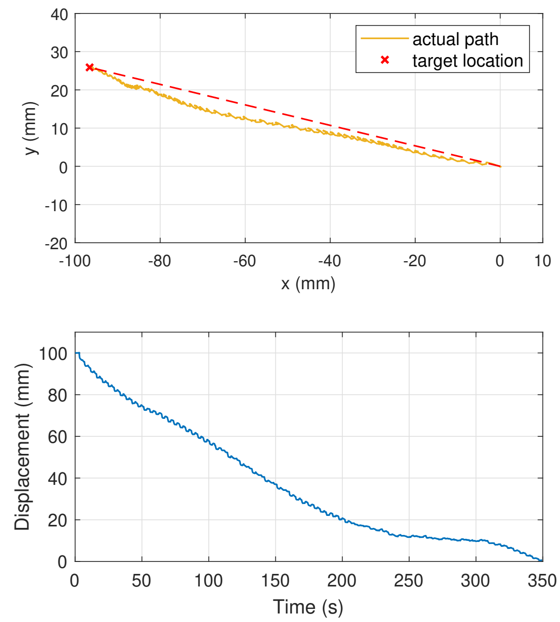


Fig. 19. Experimental results of point-to-point movement task

DEAs in antagonistic pairs and electroadhesion-based robot feet. To achieve the desired movement tasks, a motion control scheme inspired by the biological motor control system is proposed. The proposed control scheme is in the hierarchical architecture which consists of the motion control module and motion selection module. Each module of the proposed control scheme is presented and discussed in detail. Significantly, a mushroom-body-inspired controller with an error-based learning mechanism is designed for the soft actuator motion control, and a neural network-based RL algorithm (i.e., deep Q-learning) is designed for the motion selection. Finally, both numerical study and experiments are carried out, and which results verify and validate that the proposed motion control scheme is effective and the designed soft robot is able to reach the target location accurately and achieve the desired movement tasks successfully.

Besides that, it is worth mentioning that the proposed neuro-inspired hierarchical motor control scheme is based on several assumptions regarding the function and structure of the centipede brain, which may not be entirely accurate. Therefore, one possible future research direction would be to enhance the bio-inspired control approach with new findings in biology. Another future work will be the use of multiple sections of the centipede robot instead of just one section, which has the potential to improve the robot's locomotion performance.

References

[1] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, Jul. 2001.

[2] J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky, "Fast and robust: Hexapedal robots via shape deposition manufacturing," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 869–882, Oct. 2002.

[3] P. Birkmeyer, K. Peterson, and R. S. Fearing, "DASH: A dynamic 16g hexapedal robot," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. St. Louis, MO, USA: IEEE, Oct. 2009, pp. 2683–2689.

[4] S. Aoi, T. Tanaka, S. Fujiki, T. Funato, K. Senda, and K. Tsuchiya, "Advantage of straight walk instability in turning maneuver of multilegged locomotion: a robotics approach," *Scientific Reports*, vol. 6, no. 1, p. 30199, Jul. 2016.

[5] K. Osuka, T. Kinugasa, R. Hayashi, K. Yoshida, D. Owaki, and A. Ishiguro, "Centipede type robot i-centipot: From machine to creatures," *Journal of Robotics and Mechatronics*, vol. 31, no. 5, pp. 723–726, Oct. 2019.

[6] D. Koh, J. Yang, and S. Kim, "Centipede robot for uneven terrain exploration: Design and experiment of the flexible biomimetic robot mechanism," in *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, Sep. 2010, pp. 877–881.

[7] K. L. Hoffman and R. J. Wood, "Turning gaits and optimal undulatory gaits for a modular centipede-inspired millirobot," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Jun. 2012, pp. 1052–1059.

[8] C. Tang, B. Du, S. Jiang, Q. Shao, X. Dong, X.-J. Liu, and H. Zhao, "A pipeline inspection robot for navigating tubular environments in the sub-centimeter scale," *Science Robotics*, vol. 7, no. 66, p. eabm8597, May 2022.

[9] G. L. et al, "Self-powered soft robot in the mariana trench," *Nature*, vol. 591, pp. 66–71, 2021.

[10] Y. Chen, H. Zhao, J. Mao, P. Chirarattananon, E. F. Helbling, N.-s. P. Hyun, D. R. Clarke, and R. J. Wood, "Controlled flight of a microrobot powered by soft artificial muscles," *Nature*, vol. 575, no. 7782, pp. 324–329, Nov. 2019.

[11] G.-Y. Gu, U. Gupta, J. Zhu, L.-M. Zhu, and X.-Y. Zhu, "Feedforward deformation control of a dielectric elastomer actuator based on a nonlinear dynamic model," *Applied Physics Letters*, vol. 107, no. 4, p. 042907, Jul. 2015.

[12] G. Rizzello, D. Naso, A. York, and S. Seelecke, "Modeling, identification, and control of a dielectric electro-active polymer positioning system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 632–643, Mar. 2015.

[13] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, "Dynamic simulation of articulated soft robots," *Nature Communications*, vol. 11, no. 1, p. 2233, 2020.

[14] W. Barbara, "Robots with insect brains," *Science*, vol. 368, no. 6488, pp. 244–245, 2020.

[15] R. A. Campbell and G. C. Turner, "The mushroom body," *Current Biology*, vol. 20, no. 1, pp. R11–R12, Jan. 2010.

[16] S. M. Farris, "Are mushroom bodies cerebellum-like structures?" *Arthropod Structure & Development*, vol. 40, no. 4, pp. 368–379, Jul. 2011.

[17] R. Strauss, "The central complex and the genetic dissection of locomotor behaviour," *Current Opinion in Neurobiology*, vol. 12, no. 6, pp. 633–638, Dec. 2002.

[18] E. Arena, P. Arena, R. Strauss, and L. Patané, "Motor-Skill learning in an insect inspired neuro-computational control system," *Frontiers in Neurorobotics*, vol. 11, 2017.

[19] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 281–291, Jun. 2019.

[20] J. Zhang, J. Chen, W. Wu, and H. Qiao, "A cerebellum-inspired prediction and correction model for motion control of a musculoskeletal robot," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2022.

[21] R. Pelrine, R. Kornbluh, Q. Pei, and J. Joseph, "High-speed electrically actuated elastomers with strain greater than 100%," *Science*, vol. 287, no. 5454, pp. 836–839, 2000.

[22] J. Cao, W. Liang, Y. Wang, J. Zhu, and Q. Ren, "Control of a soft inchworm robot with environment adaptation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3809–3818, 2019.

[23] H. Wang, A. Yamamoto, and T. Higuchi, "Electrostatic-motor-driven electroadhesive robot," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 914–919.

[24] W. Liang, J. Cao, Q. Ren, and J.-X. Xu, "Control of dielectric elastomer soft actuators using antagonistic pairs," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2862–2872, 2019.

[25] N. J. Strausfeld and F. Hirth, "Deep homology of arthropod central complex and vertebrate basal ganglia," *Science*, vol. 340, no. 6129, pp. 157–161, Apr. 2013.

[26] T. Zhang, R. Wang, S. Wang, Y. Wang, L. Cheng, and G. Zheng, "Autonomous skill learning of water polo ball heading for a robotic fish: Curriculum and verification," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2022.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.