Snake Robot with Tactile Perception Navigates on Large-scale Challenging Terrain

Shuo Jiang, Adarsh Salagame, Alireza Ramezani, Lawson L.S. Wong

Abstract— **Along with the advancement of robot skin technology, there has been notable progress in the development of snake robots featuring body-surface tactile perception. In this study, we proposed a locomotion control framework for snake robots that integrates tactile perception to augment their adaptability to various terrains. Our approach embraces a hierarchical reinforcement learning (HRL) architecture, wherein the high-level orchestrates global navigation strategies while the low-level uses curriculum learning for local navigation maneuvers. Due to the significant computational demands of collision detection in whole-body tactile sensing, the efficiency of the simulator is severely compromised. Thus a distributed training pattern to mitigate the efficiency reduction was adopted. We evaluated the navigation performance of the snake robot in complex large-scale cave exploration with challenging terrains to exhibit improvements in motion efficiency, evidencing the efficacy of tactile perception in terrain-adaptive locomotion.**

I. INTRODUCTION

With the recent advancements in the field of bionic robots, snake robots have drawn increasing attention. These robots emulate the body structure of snakes, comprising sequentially interconnected joints. This unique body configuration affords them the capability to execute distinctive motions. Furthermore, owing to their slender body shape, they excel in accessing narrow environments that prove challenging for other types of robots. Consequently, snake robots have demonstrated commendable performance in a range of specialized scenarios, including underwater environments [1], earthquake rescue [2], etc.

Snake robots primarily rely on undulating motions to generate anisotropic friction on the contact surface for propulsion. In contrast to legged or wheeled robots, this mode of motion yields a plethora of ground contacts that pervade the entire body, presenting challenges in dynamics analysis. However, these intricate ground contacts serve as informative conduits for capturing various terrain characteristics, including surface roughness and slope. Early control strategies [3], [4], [5], [6], [7], [8], [9], [10], [11] for shape-shifting robots employed dynamic modeling and feedback control, often leveraging serpenoid curves [12] or backbone curves [13] to design gait patterns. These approaches allowed for the generation of diverse movements, including sidewinding, undulation, or lateral rolling. It is important to note that these control methodologies were predominantly validated for path tracking on flat terrain. Among these strategies, the Central Pattern Generator (CPG) [14] emerged as a comprehensive approach for gait generation and transition. By outputting sinusoidal curves with distinct parameters, CPG could generate a variety of snake robot gaits. Subsequently, the concept of segmented control was introduced [15]. The body of the snake robot can be divided into multiple segments according to its functionality. Employing different gaits in each segment unveiled the potential for the discovery of more complex gaits, such as the C-pedal wave and crawler gait [16].

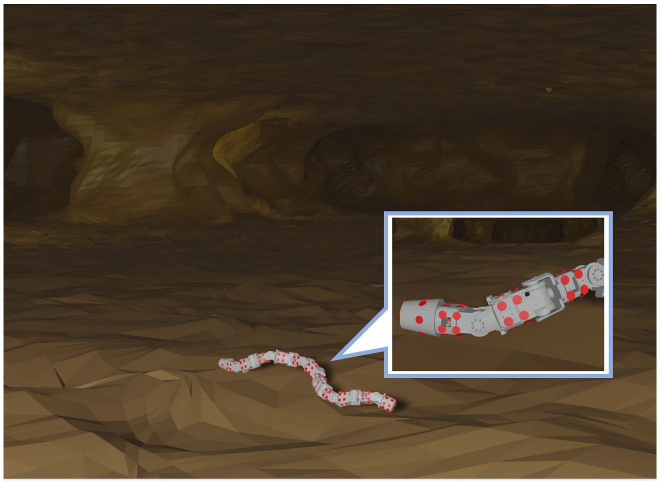


Fig. 1: Snake robot with whole-body tactile sensors performing cave navigation.

Deploying snake robots in more complex scenarios, such as those with obstacles or uneven terrain, makes dynamic modeling less feasible. Consequently, model-free control methods, such as reinforcement learning (RL), have garnered more extensive discussion. Additionally, adapting to complex environments relies on more comprehensive sensory patterns. Thus, some snake robots integrated various types of sensors, such as head cameras [17] and body tactile sensors [18], [15], [19], [20]. Leveraging whole-body tactile sensors, snake robots can perceive the precise location and force of each contact, adjusting their body shape to avoid damage or to generate additional propulsion. Several designs for applying whole-body tactile sensing on snake robots have been proposed. However, discussions regarding the integration of tactile information into control loops are less common. Key solutions include lateral inhibition [21] or lateral hump [22], relying solely on localized body curvature to adapt to contact, with less consideration for the resultant whole-body motion. When we view all surface tactile sensors as a unified entity, different gaits can generate distinct contact patterns, as shown in Fig. 2. These contact patterns encapsulate substantial information about terrain and body movement,

This work was supported by NSF Grant #2142519.

Northeastern University, Boston, MA 02115, USA {shuo.jiang, salagame.a, a.ramezani}@northeastern.edu ; lsw@ccs.neu.edu

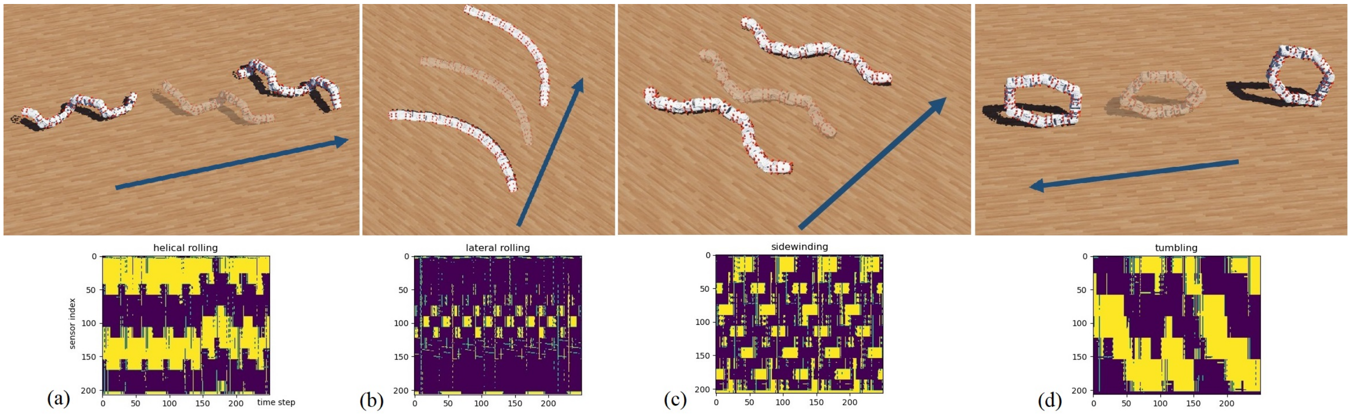


Fig. 2: (a) helical rolling; (b) lateral rolling; (c) sidewinding; (d) tumbling gaits and their tactile patterns.

and can be used to enhance environmental perception and motion control.

In this work, we propose a hierarchical reinforcement learning (HRL) approach to control snake robots for large-scale path tracking in complex terrains, while incorporating whole-body tactile sensing into the control loop to achieve terrain adaptability. Inspired by the tactile patterns in Fig. 2, we use a computer-vision-style signal processing scheme for tactile signal interpretation. Concurrently, we use curriculum learning to facilitate the expansion from small-scale solutions to large-scale scenarios, enhancing task generalization capabilities. We rely on multi-agent reinforcement learning (MARL) to provide local information exclusively to the control loop, while generating coordinated behavior under the guidance of a centralized critic. Lastly, to address the efficiency issues associated with robot simulators featuring numerous tactile sensors, we designed a distributed RL framework harnessing cluster computing for acceleration.

II. BACKGROUND

A. Markov Decision Process (MDP)

An MDP is a 4-tuple $M = \langle S, A, P, R\rangle$ where S is the set of states, A is the set of actions, $P\left(s_{t+1}|s_t, a_t\right)$ is the transition probability that action a in state s at time t that will lead to state s at time $t+1$, $R\left(a_t, s_t\right)$ is the distribution of reward when taking action a_t in state s_t. A policy $\pi\left(a_t|s_t\right)$ is defined as the probability distribution of choosing action a_t given state s_t. The learning goal is to find a policy π^* that maximizes the accumulated reward in given horizon T, $\pi^* = \underset{\pi}{\text{argmax}} \underset{a_t, s_t \sim \pi}{\mathbb{E}} \left[\sum_{t=0}^{T-1} \gamma^t \cdot R\left(a_t, s_t\right)\right]$, where γ is discount factor. RL algorithms are common choices to solve MDP problems.

B. Central Pattern Generators (CPGs)

CPG is a neural circuit in the vertebrate spinal cord that generates coordinated rhythmic output signals to control robot locomotion. By tuning its parameters, CPG can output sinusoidal waves on multiple channels. CPG-based control methods have been successfully applied to many kinds of robots, such as multi-legged robots [23], [24], [25] or snake robots [26], [14], [27]. Usually, to improve the terrain adaptability of CPG, optimization algorithms are often applied to adjust CPG parameters in real-time. As multiple CPG structures have been proposed, we adopted the structure in [14]. The dynamics of CPG are shown in Equation 1-3.

$$\begin{aligned} \dot{\varphi} &= \omega + \mathbf{A} \cdot \varphi + \mathbf{B} \cdot \theta \\ \ddot{r} &= a \cdot \left[\frac{a}{4}\left(R - r\right) - \dot{r}\right] \\ x &= r \cdot sin\left(\varphi\right) + \delta \end{aligned} \tag{1}$$

$$\mathbf{A} = \begin{bmatrix} -\mu_1 & \mu_1 & & & \\ \mu_2 & -2\mu_2 & \mu_2 & & \\ & & \ddots & & \\ & & \mu_{n-1} & -2\mu_{n-1} & \mu_{n-1} \\ & & & \mu_n & -\mu_n \end{bmatrix} \tag{2}$$

$$\mathbf{B} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & \ddots & & \\ & & \ddots & 1 & \\ & & & -1 & 1 \\ & & & & -1 \end{bmatrix} \tag{3}$$

$\varphi \in \mathbb{R}^n$ and $r \in \mathbb{R}^n$ are internal states of CPG, n is the number of output channels, typically the number of robot joints. a and μ_i are hyperparameters that control the convergence rate. $R \in \mathbb{R}^n$, $\omega \in \mathbb{R}^n$, $\theta \in \mathbb{R}^{n-1}$, $\delta \in \mathbb{R}^n$ are inputs that control the desired amplitude, frequency, phase shift and offset. $x \in \mathbb{R}^n$ is the output sinusoidal waves of n channels.

C. Robot Description

The Crater Observing Bio-inspired Rolling Articulator (COBRA), shown in Fig. 3, is designed for space exploration [28]. It consists of 11 actuated joints. The head module of the robot contains the onboard computing system, a radio antenna for communicating with a lunar orbiter, and an inertial measurement unit (IMU) for navigation. At the tail end, there is an interchangeable payload module containing a neutron spectrometer to detect water ice. The rest of the system consists of identical 1-DoF joint modules inter-placed perpendicularly containing a joint actuator and a battery. 207 virtual tactile sensors measuring normal pressure forces at 50 Hz are evenly placed throughout the robot body.

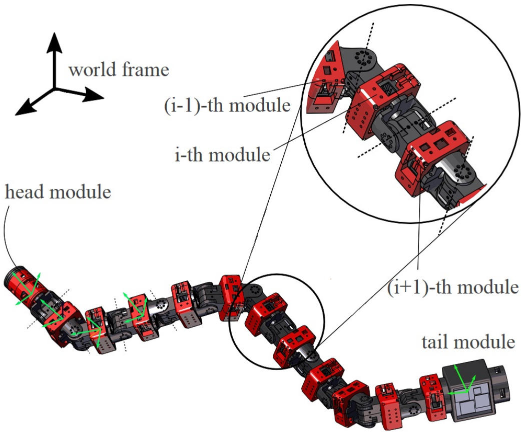


Fig. 3: COBRA robot body structure.

Global Navigation
Local Navigation
Gait Control

Fig. 4: Three-layer hierarchical control scheme.

III. HIERARCHICAL CONTROL SCHEME

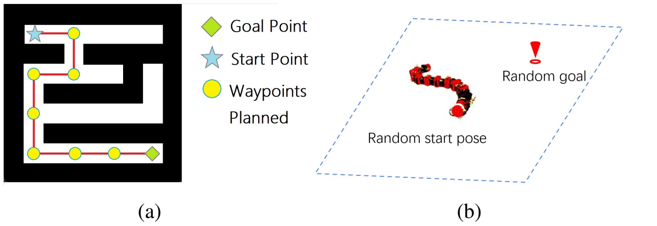
The problem addressed in this paper is robot navigation in complex terrains, namely, moving from any starting position to any target location on a map. To achieve this goal, we introduce a hierarchical control scheme (Fig. 4). At the highest level of global navigation, we use tree search (A*) algorithm to plan efficient paths, which are then segmented into a series of contiguous waypoints (Fig. 5a). At the middle level, local navigation, we use RL to train the robot to adjust its gait to navigate from one waypoint to the next (Fig. 5b). We integrate tactile perception information into the RL control loop to achieve real-time terrain adaptability. At the lowest level, we use PID controllers to actuate the robot's joints to execute the desired gaits.

A. Tactile-adaptive Local Navigation

Given the relatively straightforward solution of global navigation and gait controller, we omit their implementation details. The novelty of this study lies in the development of a new reinforcement-learning paradigm at the local-navigation level to govern the locomotion of the snake robot. The key problem rests in effectively exploiting the whole-body tactile sensing information to regulate the robot's gait for enhanced terrain adaptability. Our control scheme design adheres to four guiding principles: (1) Individual joint control; (2) using a pre-trained gait library built from curriculum learning; (3) joint gaits solely depend on local tactile signals; (4) the application of Centralized Training and Decentralized Execution (CTDE) to mitigate partial observability and improve learning efficiency.

B. Curriculum Learning

Snake robots adopt distinct gaits for efficient locomotion on various terrains. For instance, sidewinding is often used on slopes, while lateral rolling is preferable on smoother surfaces. Inspired by this, we train agents across a spectrum of randomly generated distinctive curriculum terrains (Fig. 6). Each agent contains a Central Pattern Generator (CPG) module, with the actor's outputs tuning the parameters of the corresponding CPG module. By CPG parameter adjustment, the agents generate optimal gaits pertinent to their curriculum terrains. The training terrains are generated by Perlin noise of size $16m \times 16m$, where the robot learns to navigate to reach a random goal pose from any start location in each episode. The corresponding MDP is defined as:



(a) (b)

Fig. 5: (a) Global navigation: plan a path from start position to goal position and segment the path by waypoints; (b) Local navigation: using RL to steer robot from a waypoint to the next.

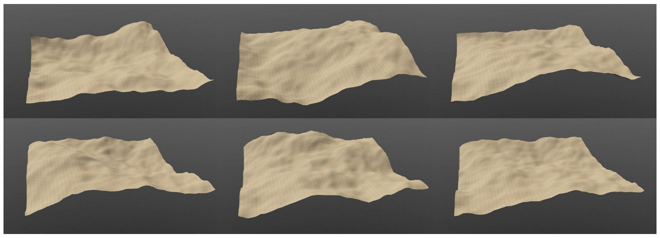


Fig. 6: Randomly generated curriculum terrains.

State space: The state space includes the robot state part and tactile readings part. The robot state part consists of the joint positions $\mathbb{R}^n$, IMU readings $\mathbb{R}^3$, spatial translation between robot frame and goal pose frame $\mathbb{R}^3$ and relative rotation parameterized by axis-angle system $\mathbb{R}^4$, i.e., 21 dimensions in total. We only use ego-centric observations from the robot, so a motion capture system is not required as in [29], [30], [31], which makes our system more practical in outdoor environments.

Action space: The action space outputs the CPG parameters, including the desired amplitude $R \in \mathbb{R}^n$, frequency $\omega \in \mathbb{R}^n$, phase shift $\theta \in \mathbb{R}^{n-1}$ and offset $\delta \in \mathbb{R}^n$.

Reward: We encourage the robot to reach the goal as soon as possible. The reward consists of the following terms:

$$\begin{aligned} r_1 &= \frac{1}{0.1 + d_t} \\ r_2 &= d_{t-1} - d_t \end{aligned} \tag{4}$$

where d_t is the distance between the robot frame and the waypoint frame. r_1 encourages getting closer to the goal and r_2 encourages higher velocities. r_1 and r_2 work in a complementary fashion, with $r_1 \to 0$ when the robot is far away from the goal and $r_2 \to 0$ when the robot is near the goal. We use Soft Actor Critic [32] as the backbone RL algorithm.

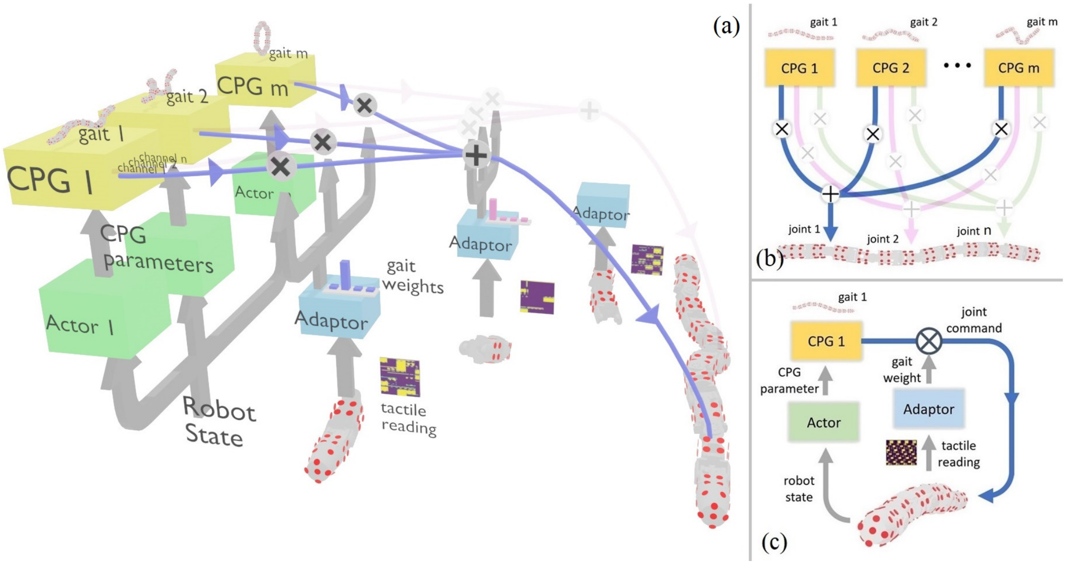


Fig. 7: (a) Diagram of reinforcement learning; (b) top view: each CPG module generates a particular gait pattern, consisting of n channels to guides joint motions. The joint command is a linear combination of different gaits; (c) side view: each joint has its own Adaptor and the Adaptors control the gait mixing factors given the local tactile readings

C. Local Navigation Control Scheme

Through curriculum learning as discussed earlier, we obtain a set of agents adapted to various types of terrains, achieving specific gaits by modulating the parameters of the CPG modules. The actors of all acquired agents constitute a gait library, as illustrated in the left side of Fig. 7(a), represented by the yellow and green boxes. Importantly, the training process of these gait libraries do not involve tactile information. Our experimental findings reveal that incorporating tactile information simply by adding it to the state space of a single agent does not yield effective terrain-adaptive gaits. Hence, we devised an approach to incorporate the tactile information as in Fig. 7(a), during a second phase of training (after the first phase of curriculum learning).

For each joint, we introduce an Adaptor, which takes as input the localized tactile information from adjacent links on the body, recognizes terrain features, and subsequently selects a gait output from the library in a one-hot manner. We use SAC with discrete action space to train the Adaptors, keeping the weights and biases of the Actors fixed during this training phase. We train on multiple new terrains not in the curriculum to improve the robot's terrain adaptation capabilities. In this second phase, the state space is the recent tactile readings gathered from the past one second, and the action space is the one-hot gait selection signal, and the reward is unchanged. Since the basic gaits were already learned during the curriculum learning phase, there was no need to learn gaits from scratch in this phase.

Each CPG module outputs a target joint value $\mathbf{q}_i \in \mathbb{R}^n$, $i \in \{1, \cdots m\}$, where m is the number of CPGs and n is the number of joints (channels). For each joint $j \in \{1, \cdots, n\}$, its target joint value shall be chosen as the j-th channel from one of the candidate joint values from m CPG outputs. This choice is determined by the Adaptors and becomes the final target joint value to execute (Fig. 7(b)).

Num. sensors	0	50	100	150	200
Gazebo	2.28	0.27	0.09	0.05	0.02
Mujoco	110.3	31.79	24.37	19.92	12.37
Webots	42.3	2.31	1.08	0.69	0.33
PyBullet	59.4	49.8	34.8	NaN	NaN

TABLE I: Real Time Factor (RTF) comparison among popular simulators. NaN represents unstable computation.

This formulation of localized Adaptors relies on the assumption that gait adjustments are locally dependent on tactile signals, with limited reliance on distant tactile signals. For instance, the motion of a robot's head exhibits negligible correlation with the tactile feedback at its tail. Such framework draws inspiration from the Centralized Training and Decentralized Execution (CTDE) learning paradigm [33], [34] within the context of multi-agent reinforcement learning (MARL). In this analogy, akin to our Adaptors, each agent exclusively bases its decision-making process on a subset of the global observation. This configuration eliminates the redundant inter-dependencies among agents and reduces model dimensions without degrading task performance.

An intriguing observation is that when Adaptors use a softmax-based output instead of a one-hot one, the weighted mixture of gaits from the library as the final gait did not yield effective performance. The Adaptors will converge toward the average of all gaits in the library, completely neglecting the tactile information. Introducing entropy as an additional loss term could circumvent this averaging tendency but simultaneously introduces computational instability. Hence we used SAC with discrete action space to output a hardmax (one-hot) gait selection.

D. Distributed Learning

Due to the introduction of tactile sensors, simulation becomes slow and does not scale well to the substantial amount of experience required in RL. Table I illustrates the

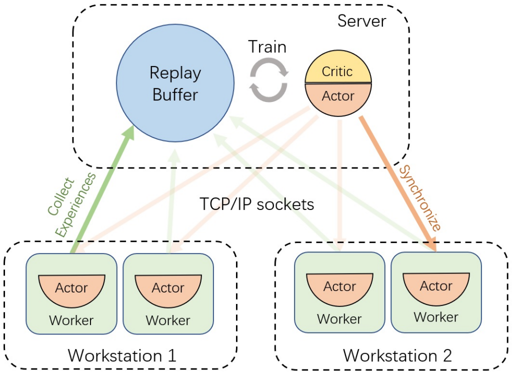


Fig. 8: Distributed reinforcement learning framework.

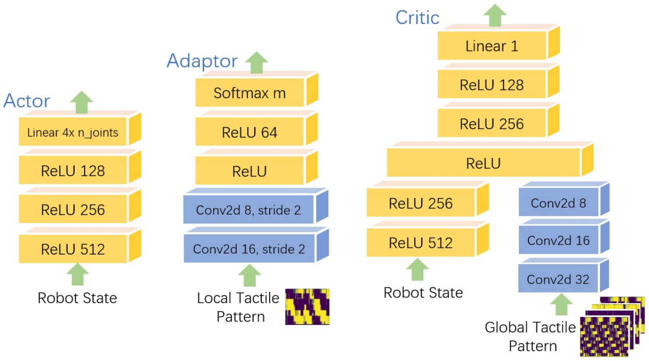


Fig. 9: Neural network architectures.

operational efficiency of several commonly used robot simulators concerning various numbers of tactile sensors. It can be observed that as the number of sensors increases, there is a noticeable decline in the simulator's efficiency, as manifested by the maximum real-time acceleration achievable by the simulator, denoted as the Real-Time Factor (RTF). As the bottleneck lies in the simulation side for large amount of collision detection, we developed a distributed RL framework deployable across multiple workstations (Fig. 8) to mitigate the situation. One of these workstations serves as a server, with an agent comprising a critic and an actor (gait library and Adaptors), along with a centralized replay buffer to store experiences. The other workstations run multiple simulator instances (workers), each instance containing only one agent interacting with the environment. The experiences gained by the workers are transmitted to the server via the TCP/IP protocol, and agent training (on GPU) is exclusively conducted at the server end. The server periodically synchronizes the actors to each worker. Notably, each Adaptor only receives a local tactile pattern, recorded from two adjacent links of a joint, while the critic receives global tactile patterns from all sensors across the entire body. The neural networks architectures we use are shown in Fig. 9.

IV. EXPERIMENTS

We tested the terrain adaptability of snake robot locomotion in a randomly generated cave, as shown in Fig. 10. The dimensions of the cave are 155m×102m. The uneven surface of the cave presents challenges for the robot to move. The task involves autonomous navigation of the robot from any initial position to any specified target point, using terrain-adaptive locomotion. We divided the cave into 4m×4m blocks, and the high level controller plans paths based on this grid (Fig. 10). To show the generalization ability, we generate multiple random cave layouts as shown in Fig. 11.

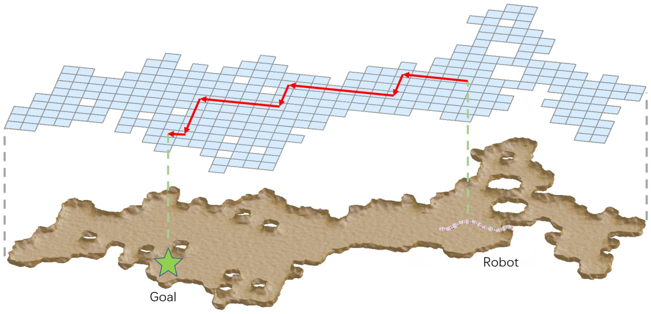


Fig. 10: Randomly generated cave as the test domain. Global navigation plans a path from current robot location to the goal (red arrows), and local navigation controls robot gaits to traverse between neighborhood blocks.

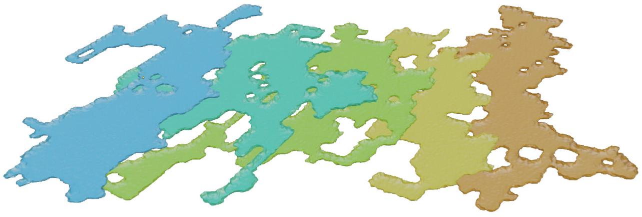


Fig. 11: Randomly generated cave layouts.

A. Curriculum Results

The training curves for the two phases of our designed algorithms are shown in Fig. 12. The left shows the results of 6 curriculum learning on different terrains, during which the robot learns basic gaits without the use of tactile perception. Following the first phase, on the right are the results of our terrain adaptation method (decSAC) trained on 6 terrains beyond the curriculum learning. It can be observed that at the beginning of the second phase, due to change in terrains, the gaits learned in the first phase are not readily adaptable to the new environment (sudden drop compared with the end of the left). However, after training, our algorithm demonstrated performance similar to curriculum learning on various new terrains. Additionally, we observed that there is little difference in the final performance between centralized and decentralized Adaptors (SAC vs decSAC), thus demonstrating the feasibility of training using MARL. For the method that does not use tactile information but relies solely on domain randomization (DR), it can be observed that learning has performance bottlenecks. Furthermore, we can see that directly incorporating tactile information as part of the state space (Tac) yields ineffective results. All the results are averaged from 10 independent trials.

Analysis of terrain adaptability can be referenced in TABLE II, where M1-M6 represent the 6 models in the curriculum training, and T1-T6 correspond to the matching training terrains for M1-M6. As observed, the diagonal shape

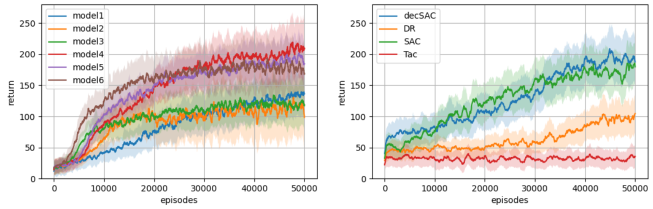


Fig. 12: Training curves: (left) curriculum training phase; (right) tactile-adaptation phase.

	T1	T2	T3	T4	T5	T6	T7	T8
M1	227.8 ± 6.1	75.3 ± 7.3	59.5 ± 5.6	90.6 ± 9.0	94.2 ± 5.8	76.4 ± 4.6	103.2 ± 4.6	88.1 ± 4.6
M2	124.9 ± 7.6	206.3 ± 3.8	78.5 ± 6.7	84.5 ± 4.0	101.2 ± 6.4	82.8 ± 8.8	62.5 ± 6.4	77.8 ± 6.4
M3	139.4 ± 4.1	64.3 ± 5.5	163.0 ± 5.4	103.2 ± 7.2	96.3 ± 6.7	70.4 ± 4.4	90.3 ± 4.4	82.8 ± 4.4
M4	108.6 ± 5.7	98.7 ± 8.4	59.8 ± 7.7	153.0 ± 4.0	82.4 ± 6.4	96.4 ± 5.5	83.1 ± 5.5	101.0 ± 5.5
M5	72.8 ± 6.3	76.5 ± 4.5	40.1 ± 7.2	78.4 ± 4.9	159.1 ± 7.0	85.1 ± 8.5	43.8 ± 8.5	56.6 ± 8.5
M6	96.3 ± 4.3	83.6 ± 6.5	67.4 ± 7.0	96.7 ± 6.2	71.4 ± 5.9	154.8 ± 4.1	72.3 ± 5.9	89.2 ± 5.9
DR	116.3 ± 7.2	102.9 ± 10.3	73.4 ± 6.4	120.2 ± 8.2	107.5 ± 4.6	140.3 ± 5.3	98.9 ± 7.0	158.2 ± 7.1
Ours	210.5 ± 13.3	230.8 ± 8.7	101.6 ± 8.5	172.6 ± 6.7	169.8 ± 9.9	152.2 ± 4.6	127.4 ± 6.7	235.0 ± 8.8

TABLE II: Model-terrain generalization analysis (return with standard deviation).

in the table indicates that M1-M6 only perform well in their respective training scenarios but are hard to adapt to untrained environments. T7 and T8 are two entirely new test environments beyond the two training phases. It can be seen that neither M1-M6 nor DR can perform well in the new environments, whereas our approach is capable of extracting terrain characteristics from tactile information and adopting adaptive gaits.

B. *Cave Navigation Performance*

We compared the results of several baselines in navigating through the five caves (Fig. 11) using RL and the comparison of their runtime is shown in Fig.13. The action space of method ”RJ” is the robot’s target joint angles [17], while the action space of method ”CPG” consists of parameters for the CPG modules [27]. The ”DR” method introduces Domain Randomization on top of CPG. The baselines in the figure did not utilize tactile information. It can be observed that our method achieved the most efficient navigation results. We found that similar to the Tac results in Fig. 12, directly incorporating tactile information into the state space, regardless of using RJ, CPG, or DR in the action space, failed to complete the navigation task within a reasonable timeframe, and therefore, the results are not depicted in the figure. We speculate that the reason might be the inherent difficulty of simultaneously learning both gait and terrain adaptability from scratch. In contrast, our approach, through curriculum learning, divides the training into two phases, each focusing on learning gait and terrain adaptability, respectively. This approach simplifies the problem by decoupling the two tasks.

The centroid motion trajectory of the robot in one of the caves is shown in Fig. 14, and it can be observed that the centroid motion trajectory closely aligns with the path planned by the high-level controller. By observing the robot’s motion at a closer distance, we found that when tactile information is not utilized, i.e., RL directly determines the parameters of the CPG module based on the robot’s state, the terrain adaptability is compromised. As shown in Fig. 15, when the robot uses the sidewinding gait on an uphill without the slope information, it is prone to turning over. We provide a detailed demonstration of this comparison in the supplementary video, showcasing the motion of various baselines within the caves.

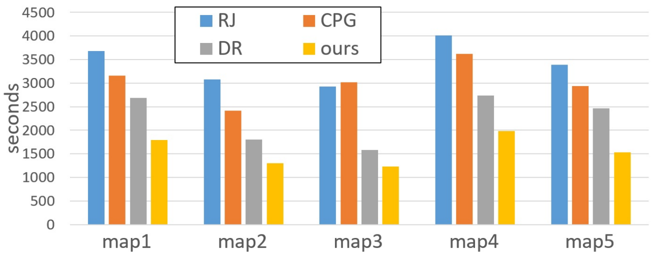


Fig. 13: Baseline Comparisons in the 5 test caves

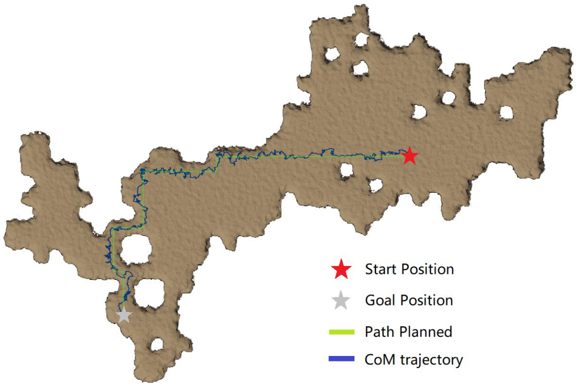


Fig. 14: Center of Mass (CoM) trajectory.

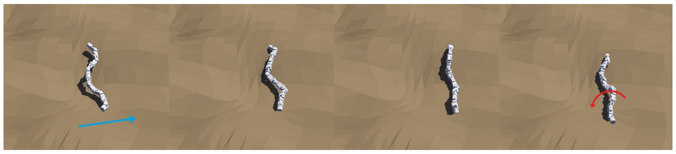


Fig. 15: Tactile-less CPG controller has difficulty traversing the terrain with sidewinding gait, the robot falls from uphill climbing.

V. CONCLUSION

In this paper, we proposed a novel hierarchical reinforcement learning control scheme to address the navigation problem of snake robots equipped with whole-body tactile perception in complex terrains. By incorporating tactile information, snake robots can perceive environment characteristics and adjust their gaits accordingly to achieve terrain adaptability. Validation experiments across various terrains demonstrated superior performance of our approach compared to traditional RL solutions. Future work will focus on sim-to-real transfer and conducting validation on real robot platforms.

References

[1] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl, "Innovation in underwater robots: Biologically inspired swimming snake robots," *IEEE robotics & automation magazine*, vol. 23, no. 1, pp. 44–62, 2016.

[2] J. Whitman, N. Zevallos, M. Travers, and H. Choset, "Snake robot urban search after the 2017 mexico city earthquake," in *2018 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2018, pp. 1–6.

[3] E. Sihite, P. Ghanem, A. Salagame, and A. Ramezani, "Unsteady aerodynamic modeling of Aerobat using lifting line theory and Wagner's function," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 10 493–10 500, iSSN: 2153-0866.

[4] E. Sihite, P. Dangol, and A. Ramezani, "Unilateral Ground Contact Force Regulations in Thruster-Assisted Legged Locomotion," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2021, pp. 389–395, iSSN: 2159-6255.

[5] A. Ramezani, P. Dangol, E. Sihite, A. Lessieur, and P. Kelly, "Generative Design of NUâ€™s Husky Carbon, A Morpho-Functional, Legged Robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 4040–4046, iSSN: 2577-087X.

[6] A. Lessieur, E. Sihite, P. Dangol, A. Singhal, and A. Ramezani, "Mechanical design and fabrication of a kinetic sculpture with application to bioinspired drone design," in *Unmanned Systems Technology XXIII*, vol. 11758. SPIE, Apr. 2021, pp. 21–27.

[7] A. C. B. de Oliveira and A. Ramezani, "Thruster-assisted Center Manifold Shaping in Bipedal Legged Locomotion," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2020, pp. 508–513, iSSN: 2159-6255.

[8] J. W. Grizzle, A. Ramezani, B. Buss, B. Griï¬ƒn, K. A. Hamed, and K. S. Galloway, "Progress on Controlling MARLO, an ATRIAS-series 3D Underactuated Bipedal Robot."

[9] E. Sihite, A. Kalantari, R. Nemovi, A. Ramezani, and M. Gharib, "Multi-Modal Mobility Morphobot (M4) with appendage repurposing for locomotion plasticity enhancement," *Nature Communications*, vol. 14, no. 1, p. 3323, June 2023, number: 1 Publisher: Nature Publishing Group.

[10] E. Sihite, A. Lessieur, P. Dangol, A. Singhal, and A. Ramezani, "Orientation stabilization in a bioinspired bat-robot using integrated mechanical intelligence and control," in *Unmanned Systems Technology XXIII*, vol. 11758. SPIE, Apr. 2021, pp. 12–20.

[11] A. Ramezani, "Towards biomimicry of a bat-style perching maneuver on structures: the manipulation of inertial dynamics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 7015–7021, iSSN: 2577-087X.

[12] E. Rezapour, K. Y. Pettersen, P. Liljebäck, J. T. Gravdahl, and E. Kelasidi, "Path following control of planar snake robots using virtual holonomic constraints: theory and experiments," *Robotics and biomimetics*, vol. 1, no. 1, pp. 1–15, 2014.

[13] B. A. Elsayed, T. Takemori, M. Tanaka, and F. Matsuno, "Mobile manipulation using a snake robot in a helical gait," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2600–2611, 2021.

[14] Z. Bing, L. Cheng, K. Huang, M. Zhou, and A. Knoll, "Cpg-based control of smooth transition for body shape and locomotion speed of a snake-like robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4146–4153.

[15] F. Sanfilippo, Ø. Stavdahl, G. Marafioti, A. A. Transeth, and P. Liljebäck, "Virtual functional segmentation of snake robots for perception-driven obstacle-aided locomotion?" in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 1845–1851.

[16] T. Takemori, M. Tanaka, and F. Matsuno, "Hoop-passing motion for a snake robot to realize motion transition across different environments," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1696–1711, 2021.

[17] Z. Bing, C. Lemke, F. O. Morin, Z. Jiang, L. Cheng, K. Huang, and A. Knoll, "Perception-action coupling target tracking control for a snake robot via reinforcement learning," *Frontiers in Neurorobotics*, vol. 14, p. 591128, 2020.

[18] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Snake robot locomotion in environments with obstacles," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 6, pp. 1158–1169, 2011.

[19] T. Kamegawa, T. Akiyama, Y. Suzuki, T. Kishutani, and A. Gofuku, "Three-dimensional reflexive behavior by a snake robot with full circumference pressure sensors," in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2020, pp. 897–902.

[20] D. Ramesh, Q. Fu, and C. S. Li, "A snake robot with contact force sensing for studying locomotion in complex 3-d terrain," in *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA*, 2022, pp. 23–27.

[21] T. Kamegawa, R. Kuroki, M. Travers, and H. Choset, "Proposal of earli for the snake robot's obstacle aided locomotion," in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2012, pp. 1–6.

[22] W. Zhen, C. Gong, and H. Choset, "Modeling rolling gaits of a snake robot," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3741–3746.

[23] G. Bellegarda and A. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 547–12 554, 2022.

[24] N. D. Kent, D. Neiman, M. Travers, and T. M. Howard, "Improved performance of cpg parameter inference for path-following control of legged robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 963–11 970.

[25] G. Bellegarda and A. Ijspeert, "Visual cpg-rl: Learning central pattern generators for visually-guided quadruped navigation," *arXiv preprint arXiv:2212.14400*, 2022.

[26] X. Liu, R. Gasoto, Z. Jiang, C. Onal, and J. Fu, "Learning to locomote with artificial neural-network and cpg-based control in a soft snake robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7758–7765.

[27] X. Liu, C. Onal, and J. Fu, "Learning contact-aware cpg-based locomotion in a soft snake robot," *arXiv preprint arXiv:2105.04608*, 2021.

[28] "https://www.nasa.gov/feature/northeastern-university-slithers-to-the-top-with-big-idea-alternative-rover-concept."

[29] A. A. Transeth, R. I. Leine, C. Glocker, and K. Y. Pettersen, "3-d snake robot motion: nonsmooth modeling, simulations, and experiments," *IEEE transactions on robotics*, vol. 24, no. 2, pp. 361–376, 2008.

[30] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Experimental investigation of obstacle-aided locomotion with a snake robot," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 792–800, 2011.

[31] S. Hasanzadeh and A. A. Tootoonchi, "Ground adaptive and optimized locomotion of snake robot moving with a novel gait," *Autonomous Robots*, vol. 28, pp. 457–470, 2010.

[32] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[33] X. Lyu, Y. Xiao, B. Daley, and C. Amato, "Contrasting centralized and decentralized critics in multi-agent reinforcement learning," 2021.

[34] S. Jiang and C. Amato, "Multi-agent reinforcement learning with directed exploration and selective memory reuse," in *Proceedings of the 36th annual ACM symposium on applied computing*, 2021, pp. 777–784.