ORIGINAL RESEARCH
published: 20 October 2020
doi: 10.3389/fnbot.2020.591128

Perception-Action Coupling Target Tracking Control for a Snake Robot via Reinforcement Learning

Zhenshan Bing[1*], *Christian Lemke*[2], *Fabric O. Morin*[1], *Zhuangyi Jiang*[1], *Long Cheng*[3†], *Kai Huang*[3] and *Alois Knoll*[1]

[1] *Department of Informatics, Technical University of Munich, Munich, Germany,* [2] *Department of Informatics, Ludwig Maximilian University of Munich, Munich, Germany,* [3] *School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China*

Visual-guided locomotion for snake-like robots is a challenging task, since it involves not only the complex body undulation with many joints, but also a joint pipeline that connects the vision and the locomotion. Meanwhile, it is usually difficult to jointly coordinate these two separate sub-tasks as this requires time-consuming and trial-and-error tuning. In this paper, we introduce a novel approach for solving target tracking tasks for a snake-like robot as a whole using a model-free reinforcement learning (RL) algorithm. This RL-based controller directly maps the visual observations to the joint positions of the snake-like robot in an end-to-end fashion instead of dividing the process into a series of sub-tasks. With a novel customized reward function, our RL controller is trained in a dynamically changing track scenario. The controller is evaluated in four different tracking scenarios and the results show excellent adaptive locomotion ability to the unpredictable behavior of the target. Meanwhile, the results also prove that the RL-based controller outperforms the traditional model-based controller in terms of tracking accuracy.

Keywords: snake robot, target tracking, reinforcement learning, motion planning, visual perception

OPEN ACCESS

Edited by:
Fei Chen,
Italian Institute of Technology (IIT), Italy

Reviewed by:
Xingdong Li,
Northeast Forestry University, China
Xiao Huang,
Beijing Institute of Technology, China

Correspondence:
Zhenshan Bing
bing@in.tum.de

†*Present address:*
Long Cheng,
College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China

Received: *03 August 2020*
Accepted: *15 September 2020*
Published: *20 October 2020*

Citation:
Bing Z, Lemke C, Morin FO, Jiang Z, Cheng L, Huang K and Knoll A (2020) Perception-Action Coupling Target Tracking Control for a Snake Robot via Reinforcement Learning. Front. Neurorobot. 14:591128. doi: 10.3389/fnbot.2020.591128

1. INTRODUCTION

Inspired by real snakes, snake-like robots are designed as a class of hyper-redundant mechanisms in order to achieve the agility and adaptability of their biological counterparts. Their long and narrow bodies with many degrees of freedom (DOF) enable them to perform diverse tasks that could never be carried out by other kinds of mobile robots, such as search and rescue in disaster scenes (Evan, 2017), complex teleoperation in space (Walker and Hannan, 1999), and even minimally invasive cardiac surgery (Webster et al., 2006). However, this high level of flexibility also corresponds to a complex control task involving the internal regulation of body joints and external interaction with the ground, in which model-based methods usually fail to control the robots adaptively in a dynamically changing environment.

Vision-guided locomotion, as one of the essential skills for moving in changeable scenarios, is a must-have capability for snake-like robots, to ensure that they can be deployed in an unattended environment by human operators. With the help of the visual information, the snake-like robots can solve more complex and realistic tasks, such as target tracking and obstacle avoidance. Especially in field operations, such as disaster rescue tasks and surveillance tasks, the target tracking capability can greatly improve the performances of the snake-like robots. Yet this kind of locomotion control still remains a challenging task for snake-like robots, since it involves not only the locomotion but also the target information obtained by cameras. For controlling the locomotion, different types of methods have been studied including the sinusoid-based methods (Hirose, 1993), CPG-based methods (Bing et al., 2017), and the dynamics-based methods (Miller, 1988). However, none of these methods can be used directly to perform vision-based tracking tasks. Such tracking tasks require the robot to be agile and adapt to their target trajectories with unpredictable changes in velocity or direction of travel, which is extremely challenging for traditional model-based methods. Our proposal is completely different as it tackles the object tracking and camera control pipeline in an end-to-end manner, based on reinforcement learning.

Strategies based on Reinforcement Learning (RL) are promising solutions for performing target tracking for a snake-like robot. This is because a RL-trained controller can take the visual image directly as the input, while simultaneously fully exploring the locomotion capabilities compared with model-based methods. This is particularly suitable for snake-like robots with redundant degrees of freedom. Although RL-based methods have been adopted to control mobile vehicle platforms (Morbidi and Mariottini, 2013; Yun et al., 2018; Luo et al., 2019), the effectiveness of such methods for generating agile steering motions for snake-like robots has, nevertheless, not yet been studied extensively, especially when interacting with the environment. The reasons are 2-fold. First, the steering control for snake-like robots is complex, especially when it comes to the sudden change in velocity or direction of travel, as this requires the coordination of bodies with redundant degrees of freedom from one moving pattern to another in a short time. Second, when traditional methods are used on mobile platforms, target tracking is usually divided into tracking and control sub-tasks, which makes it difficult to tune the pipeline jointly, especially considering the aforementioned motion barrier for snake-like robots. To cope with this hard-to-predict tracking and movement complexity, the RL-based control strategies need to map the visual inputs to the joint space directly, in order to perform the corresponding motions, and must operate with adequately defined reward functions to train a policy successfully. Hence, the design of a target tracking controller for snake-like robots based on RL is challenging.

To design a target tracking controller for snake-like robots, this paper proposes a RL-based strategy. Our main contributions to the literature are summarized as follows. First, we offer a novel alternative to solving the target tracking task for a snake-like robot via reinforcement learning. The learned policy directly transforms the external and internal observations to a sophisticated motion pattern for performing perception-action coupling tasks. It is worth to note that this RL-based method can be applied in different types of snake-like robots instead of designed solely for the one used in this work. Second, we propose a customized reward function that takes contiguous distances into calculation. With this reward design, the learned locomotion successfully solves adaptive target tracking tasks, and, more surprisingly it also learns the ability to keep the target within its visual field, even though this behavior is not specifically rewarded. Third, we propose a tracking accuracy metric that takes both the distance and direction into consideration. Based on this metric, we demonstrate that the learned locomotion outperforms the model-based locomotion in terms of tracking accuracy.

2. RELATED WORK

As our work is related to the perception-driven locomotion of snake-like robots and perception-driven algorithms based on reinforcement learning, we briefly review the state-of-the-art research on both aspects in the following.

2.1. Vision-Based Snake-Like Locomotion

Trajectory or target tracking of snake-like robots is important and operators usually control their locomotion by indicating the expected direction of its head module (Kamegawa et al., 2004; Fukushima et al., 2012; Yamada et al., 2013; Tanaka and Tanaka, 2015). Under the velocity constraints, which prevent the body from slipping sideways, trajectory tracking locomotion control of snake-like robots has been investigated (Matsuno and Mogi, 2000; Prautsch et al., 2000; Transeth et al., 2007; Ishikawa et al., 2009; Tanaka and Matsuno, 2014). Liljeback proposed a straight line path-following controller of a planar snake-like robot under the Line-of-Sight (LOS) guidance law, but the robot's head could not track the desired trajectory (Liljeback et al., 2011). Matsuno derived a dynamic model to avoid the singular configuration of the robot body. Using this control law, their snake robot can accomplish trajectory tracking of the head module without converging to a singular configuration (Matsuno and Sato, 2005). However, their results were only tested on a sinusoid-like track and this dynamics-based method may not adapt to unknown scenarios with changing dynamics. Similar ideas can also be found in Tanaka et al. (2015) and Huang et al. (2017). Xiao performed autonomous locomotion in a known scene and the positions of the snake-like robot and the obstacles were acquired from external web cameras (Xiao et al., 2015). This method, in fact, is a trade-off idea since they could not use the built-in camera in the snake-like robot due to its undulating locomotion pattern. In fact, there are few research efforts about the onboard vision-based locomotion control of snake-like robots, since the undulation of the body cannot provide a stable base for vision sensors. Bing et al. (2019a) proposed an end-to-end target tracking for snake-like robots based on spiking neural network. However, the network controller only outputs the steering signals and the locomotion itself is further generated with model-based methods. The robot "IRS Souryu" equipped with a ToF camera and 3D range sensors performs real-time localization and mapping tasks in a rubble environment (Ohno et al., 2006). A semi-auto snake-like robot with a B/W camera and a structured light sensor was investigated to perform a localization task of a pole, navigate and then climb it (Ponte et al., 2014). A slithering gait that specially designed for snake-like robots to perform target tracking tasks is introduced in Bing et al. (2017). A detailed survey about perception-driven and obstacle-aided locomotion for snake robots can be found in Sanfilippo et al. (2017).

2.2. RL-Based Tracking

As a principle approach to temporal decision-making problems, RL-based approaches have been used for solving visual object tracking tasks that aim at finding the target position in contiguous frames and whereby steering the locomotion of an mobile agent. Gaskett designed a mobile robot that can perform visual servoing and wandering behaviors through a Q-learning method (Gaskett et al., 2000; Garcia-Aracil et al., 2011). The work clearly demonstrated that a direct mapping from image space to actuator command using RL is a promising method. Similar work is also given in Asada et al. (1996), Takahashi et al. (1999), Busquets et al. (2002), and Hafner and Riedmiller (2003). Miljkovic presented a novel intelligent visual servo controller for a robot manipulator using RL (Miljković et al., 2013). Based on their control scheme, the performance of the robot is improved and is able to adapt to the changing environment. In the recent ViZDoom platform for visual reinforcement learning, they provided two basic scenarios: a basic move-and-shoot bot and a maze-navigation bot. Yun developed a reinforcement learning scheme to utilize labeled video sequences for training their action-driven tracker (Yun et al., 2018). In Ding et al. (2018), a partial RL based tracking algorithm was proposed to achieve adaptive control of a wheeled mobile robotic system working in highly complex and unpredictable environment. The controller required less calculation time than other optimization technologies and exhibited higher accuracy at the same time. As far as we are aware, to date, there have been no studies that employ reinforcement algorithm to control snake-like robots for performing vision-based locomotion, except for one of our previous research that used RL to design energy-efficient gaits (Bing et al., 2019b).

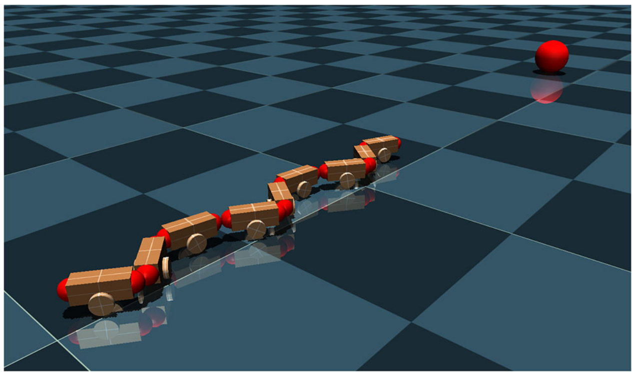
FIGURE 1 | Target tracking scene for a snake-like robot.

3. MODELS AND TASKS

This section first introduces the models of the snake-like robot. Then, the target tracking task is presented, together with the target tracking metric for evaluating the performances of different algorithms.

3.1. Models

The target tracking scene is modeled and simulated in MuJoCo (Todorov et al., 2012), in which a red ball is used as the target and a snake-like robot is the tracker (see **Figure 1**). The easily detectable sphere has a radius of 0.2 m and is placed in front of the snake at the distance of 4 m. The robot is inspired by the *ACM* snake-like robot (Hirose, 1993), which uses eight joints and nine identical modules. A head camera is used as the visual system for the snake-like robot, which is positioned in the center

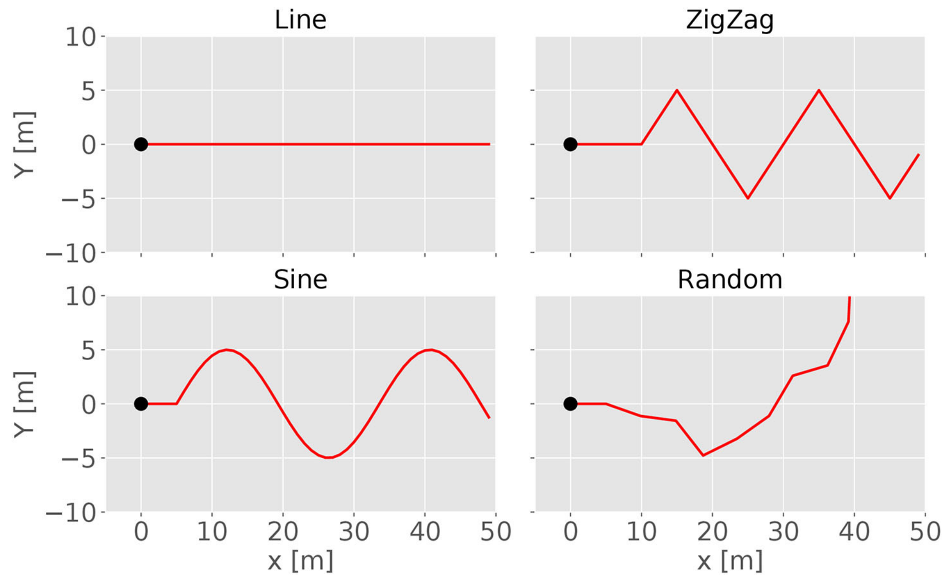

FIGURE 2 | Each of these four diagrams visualizes one of the testing tracks. The tracks define the trajectory of the target. The black dots indicate the start positions of the track. Only a limited length is displayed because the tracks are continuing. The first three figures show the *line*, *zigzag*, and *sine* track, respectively. The last sub-figure gives an example of a *random* track.

of the first head module. Due to the camera position and view volume, the ground area in front of the robot is clipped during rendering. The purpose of this camera is to obtain information of the moving target. More details about the model of the robot can be found in Bing et al. (2019b).

3.2. Tasks Description

In our target tracking task, the snake-like robot has to follow a target while maintaining a certain distance from it. The target moves with a specified velocity $v_t = 0.3\ m/s$ on the trajectories of random tracks with constrained conditions. The random-tracks consists of short straight forward sections linked by abrupt random direction changes with angles between -60° and 60°. A random seed is used to generate arbitrarily random tracks during the training process.

For evaluating the performances of the controller, we also design four predefined tracks for testing as shown in **Figure 2**. The *line* track is used to test a simple forward locomotion and is therefore the easiest. For testing the steering behavior, the *wave* track offers a continuous curve where the robot has to alternately change its steering direction. A modified sinus wave defines the *wave* track. The *zigzag* and *random* track scenarios test the robot's capability to handle abrupt directional changes. The *zigzag* track is defined by alternating abrupt left and right turns of 60°. The *random* track consists of short straight forward sections linked by abrupt random direction changes with angles between -60° and 60°. Heess et al. (2017) described that starting with easier tasks for training supports a faster learning process in RL. Thus, all the tracks have a short straight segment at the beginning, which enhances the learning process because it is easier to move straight forward at the beginning of a training process.

3.3. Tracking Metrics

As illustrated in **Figure 3**, the location of the target at time t can be represented as $\mathbf{X}_T(t) = (x_T(t), y_T(t))$ in the global frame of reference $\mathcal{R}_\mathcal{G}$. Similarly, let the position of the snake-like robot at time t be denoted by $\mathbf{X}_F(t) = (x_F(t), y_F(t))$. For simplicity, we discretize the time with steps of $\delta_t = 0.01\ ms$ and use the notation n to refer to the n^{th} time step. Let $d(n) = ||\mathbf{X}_T(n) - \mathbf{X}_F(n)||_2$ be the distance between the target and the robot at time-slot n, where $||\cdot||_2$ denotes the L_2 norm. In addition, $\theta_T = \arctan\frac{y_T(t)}{x_T(t)}$ represents the global angle of the target and $\theta_F = \arctan\frac{y_F(t)}{x_F(t)}$ is the angle of the head module in $\mathcal{R}_\mathcal{G}$. Then, the absolute relative angle between the head and the target ϕ_t can be calculated as

$$\phi_t = |\theta_T - \theta_F| \tag{1}$$

An expected tracking is to minimize the distance difference from $d(n)$ to desired distance d_t and aims at the target in the middle of the field of view (FoV) to greatest extent. Thus, the Averaged Tracking Error (*ATE*) is defined as

$$ATE = \sum_{i=0}^{N}\left(\frac{|d(n) - d_t|}{d_t} \times \frac{\phi_t(n)}{\phi_{t_{max}}}\right) \times \frac{1}{N}, \tag{2}$$

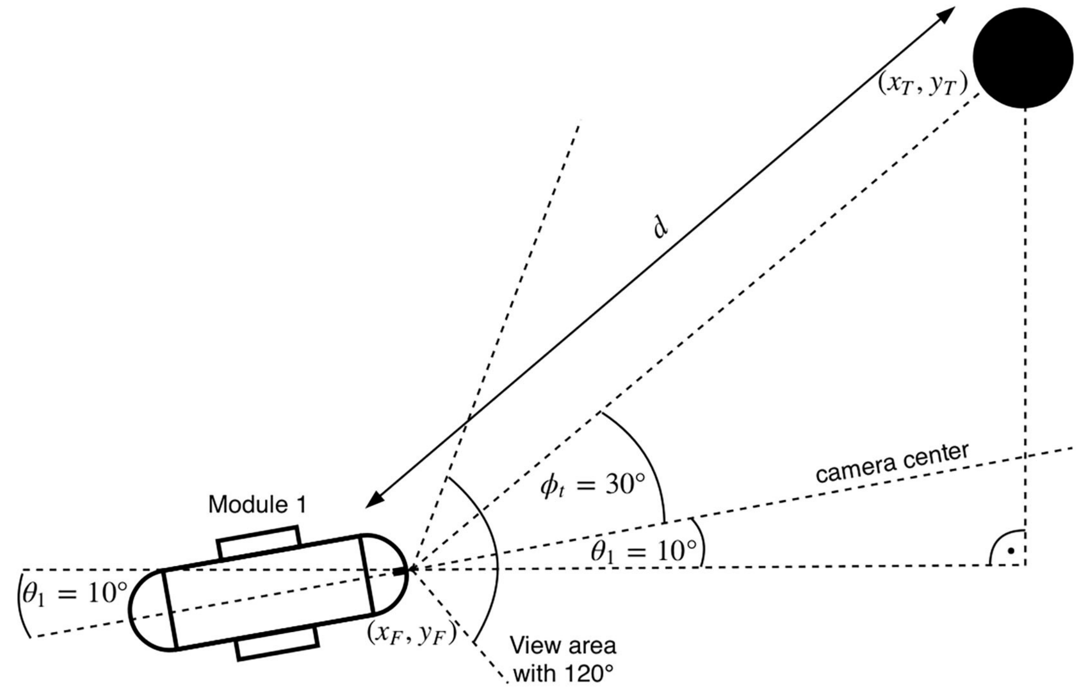

FIGURE 3 | An overview of the used angles between the camera and the target. It shows the first module and the target with their corresponding relative angles from a top view perspective. Here, the head to target angle ϕ_t has a value of 30°. The global angle of the head θ_F is 10°. The target position is at (x_T, y_T) and has a Euclidean distance d to the camera position (x_F, y_F).

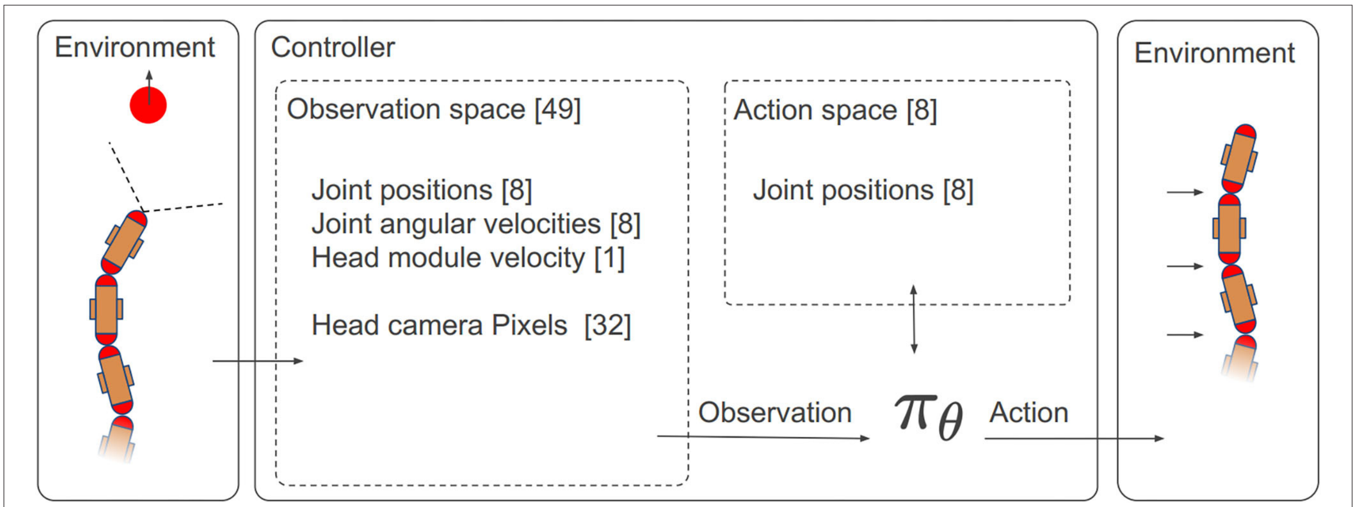

FIGURE 4 | The overall architecture of the RL controller with the details of the observation space and the action space.

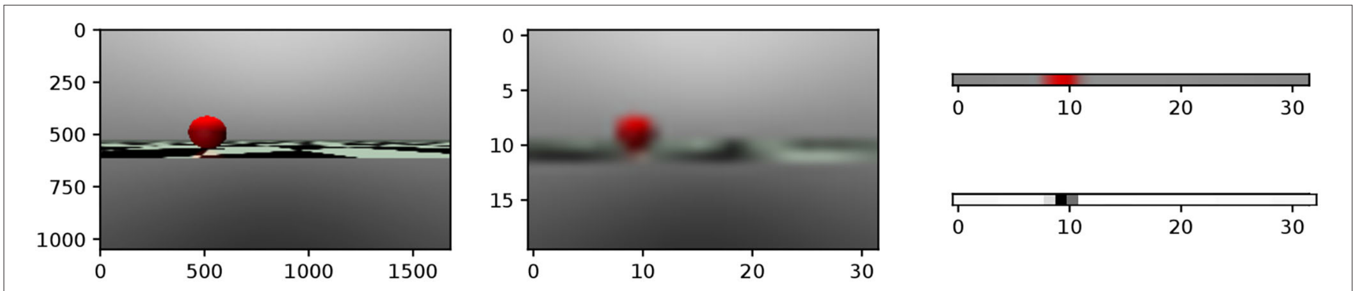

FIGURE 5 | This figure shows the visual information of the target at the distance of 4.0 m with 30° on the left. The left figure is the original image with $1,685 \times 1,050$ from the sensor. The middle figure is rendered with 32×20 pixels. The right array of pixels from the 10th row is used to estimate the relative position and distance of the target.

TABLE 1 | The observation space $\mathbf{o}_i^t$ of the controller.

Symbols	**Descriptions**
α_{1-8}	Relative joint angular positions
$\dot{\alpha}_{1-8}$	Relative joint angular velocity
v_1	Absolute head module linear velocity
$p_{10,1-32}$	Pixel 1–32 of the 10th row of the camera image

where N is the amount of the time steps. $|d(n) - d_t|$ calculates the absolute distance error in the normal direction. $\frac{1}{d_t}$ is used to calculate the error ratio against the desired target. $\frac{\phi_t(n)}{\phi_{t_{max}}}$ indicates the target's deviation in the FoV, where $\phi_{t_{max}} = 60°$.

It should be noted that the tracking metric *ATE* is not used the reward function for our RL controller, since the reward signal only depicts a desired behavior that keeps a distance with the target while the tracking metric specifically measures the tracking accuracy. For RL tasks, it is more realistic to simply use an intuitive reward instead of regularizing it with a specific metric.

4. BASELINE EXAMPLE

This section briefly explains the start-of-the-art and widely used method for controlling the locomotion of snake-like robots, which is the model-based *gait equation* controller.

The *gait equation* method is a kinematic locomotion controller that describes the gaits using a sinusoid-like wave. This method was first proposed as the *serpentine curve* (Hirose, 1993) by Hirose who gained inspiration from real snakes. In this work, an undulation gait equation developed in Tesch et al. (2009) is used for the purpose of comparison. The *gait equation* controller is modeled as

$$\alpha(m,t) = \left(\frac{m}{M}x + y\right) \times A \times \sin(\omega t + \lambda m) + C\,. \tag{3}$$

$\alpha(m,t)$ presents the joint angle value at time t, where m is the joint index and M is the joint amount. λ and ω are the spatial and temporal frequency of the movement, respectively. The spatial frequency represents the cycle numbers of the wave and the temporal frequency represents the traveling speed of the wave. A is the serpentine amplitude and x and y are the constants for shaping the body curve. C is the amplitude bias for steering the direction of a slithering locomotion.

The target tracking locomotion for the *gait equation* controller is divided into two sub-tasks, namely, lateral localization and speed control. Similar ideas can be also found in Bing et al. (2017) and Bing et al. (2019a). In the FoV of the robot, the target will be identified as a group of red pixels. For the lateral localization control, the moment of the red pixels is calculated and then used as the control target for a proportional integral (PI) controller, since it indicates the relative position of the target in the FoV of the robot. For the speed control, the number of the red pixels are counted to represent the distance from the robot to the target. In order to have a more accurate estimation, the visual image is rendered with a higher resolution 128 × 80 × 3. This is because a higher resolution can generate more amounts of red pixels for the same target and then result in a more accurate control performance. Due to the page limit, the implementation details of the *gait equation* controller will not be further explained. To ensure a fair comparison, we make many attempts to find the best control parameters to optimize the performance. But only the best tracking results are selected as the baseline example for further usage.

5. PROPOSED RL-BASED CONTROLLER

This section presents the details of the proposed RL-based controller, including the construction of the network and the training configuration and results.

5.1. Reinforcement Learning Setup

The most important components of a RL controller are the observation space, the action space, and the reward function. The overall architecture is shown in **Figure 4**.

5.1.1. Observation Space

The snake-like robot solely use a RGB vision sensor to track the target. Due to the undulation of the locomotion, the rendered image from the robot keeps shifting in the horizontal direction. In order to reduce unnecessary dimensions and enhance the computing efficiency, the following steps describe the image processing pipeline:

1. The image is directly rendered with 32 × 20 × 3 pixels. The middle figure in **Figure 5** shows an example of the rendered RBG image with 32 × 20 × 3 resolution.
2. The 10th row is extracted from the rendered image, since this line contains the pixels at which the target is located, as shown at the top right of **Figure 5**.
3. The color space is then transformed from RGB to gray with values in the range of $[0, 1]$ based on the intensity of the red pixel, as shown at the bottom right of **Figure 5**.

The resulting 32 pixels $p_{10,1-32}$ contain information about the target's relative position. Furthermore, its distance can also be estimated by the gray value intensities of the target's corresponding pixels. This value increases when the target is closer and decreases when the target is further away. These changes are caused by the illumination of the bright background. The further away the target is, the smaller it is rendered, resulting in a lower pixel count.

The controller also has to control the locomotion of the snake-like robot. To be able to propel forward, it performs a lateral

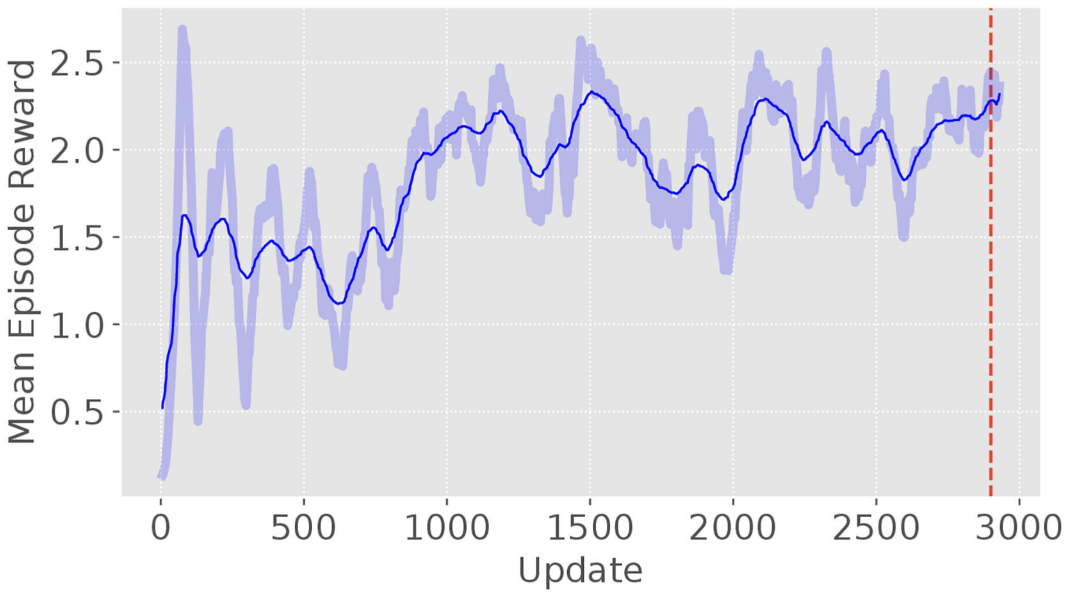

FIGURE 6 | The learning curve of the autonomous target tracking model. It is trained with 3 million time steps with 1,000 time steps per episode and 1,024 time steps per update.

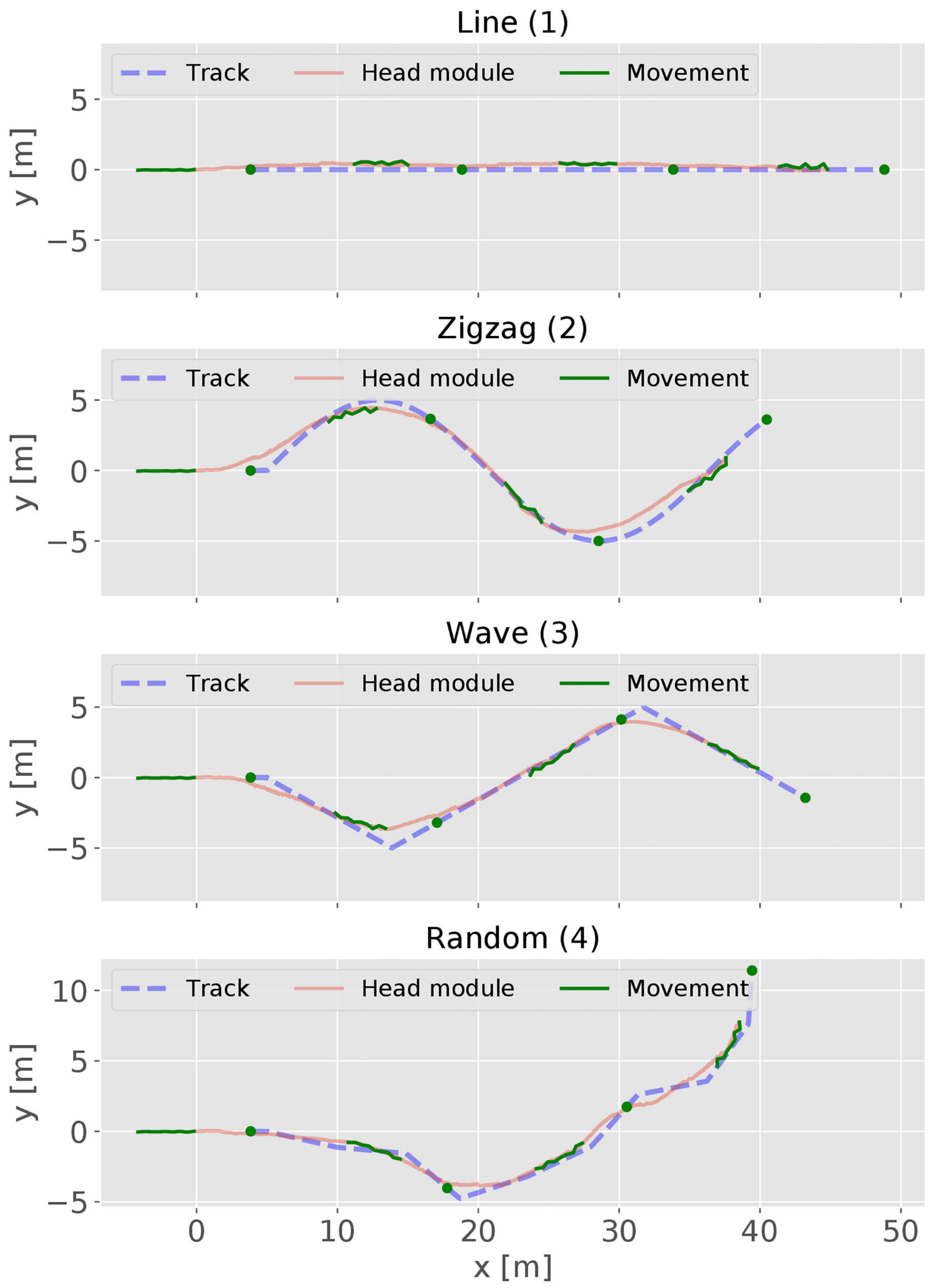

FIGURE 7 | The trajectories of the snake-like robot and the testing tracks. In addition, the body curve of the snake-like robot and the target position are added and captured every 1,000 time step.

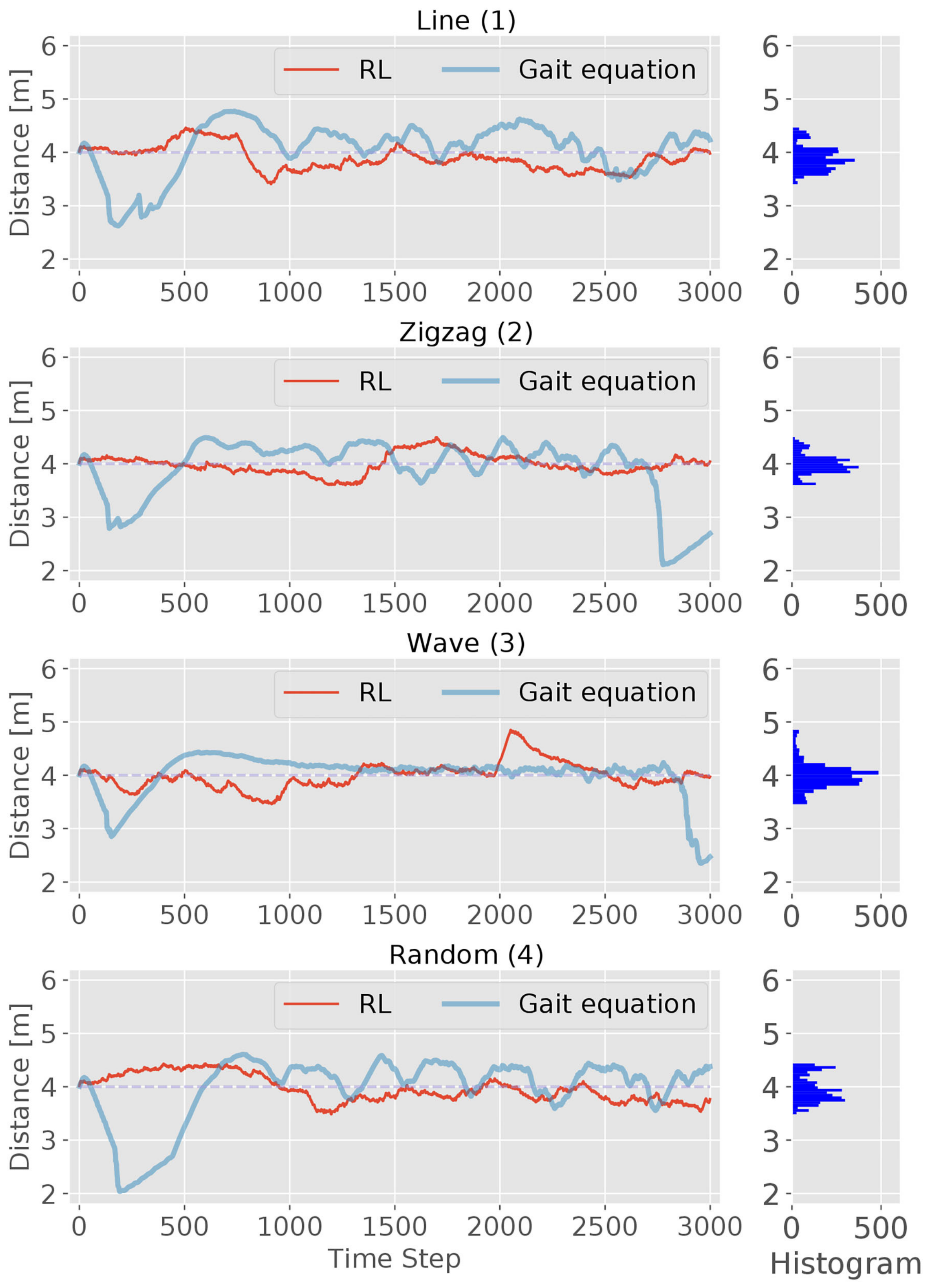

FIGURE 8 | The performance of the RL and *gait equation* controller in maintaining a certain distance from the target is shown in this figure. Each diagram represents one episode run on a track with 3,000 time steps. The target distance is set to 4.0 m and the range limits are set at 2.0 and 6.0 m. The histogram on the right of each diagram shows the distance distribution of the RL controller. Overall the distance varies in an adequate range around the target distance.

undulation motion pattern with its joints. At each time step, it receives the current joint angular positions α_{1-8} and the angular joint velocities $\dot{\alpha}_{1-8}$ to learn the locomotion and represent the proprioceptive awareness of the robot. In addition to these parameters, the controller also receives the global head module velocity along the moving direction v_1. It has been observed that this helps it to estimate its global velocity. This is required to control the velocity of the robot. In summary, an overall 49-DOF observation space is used in this work, as shown in **Table 1**.

5.1.2. Action Space

The action space $\mathbf{a}_i^t$ of the RL controller corresponds to the eight joint positions of the robot, which linearly map finite continuous values in the range of $[-1.5, 1.5]$ to $[-90^\circ,\ 90^\circ]$.

5.1.3. Reward Function

In the target tracking task, the snake-like robot follows a moving target while maintaining a specified distance d_t from it. Meanwhile, the robot should also try to maintain the target in the center of its FoV. Therefore, the distance-keeping and lateral localization in the FoV are the two criteria to find a successful behavior.

Thus, a distance-keeping reward is designed. Let $d_r = 2\ m$ define the distance radius from $t_d = 4\ m$ to its maximum distance $d_{max} = 6\ m$ and minimum distance $d_{min} = 2\ m$. In this approach, the reward represents the distance change between the head module and the target. The less distance changed toward the target distance d_t, the higher the reward. Similarly, the lower reward is received by increasing or decreasing the distance from the target distance d_t. The reward function is defined as follows:

$$\begin{aligned} r_d &= (1 - \frac{\left|d_t - d_{\text{after}}\right|}{d_r}) - (1 - \frac{\left|d_t - d_{\text{before}}\right|}{d_r}) \\ &= \frac{\left|d_t - d_{\text{before}}\right| - \left|d_t - d_{\text{after}}\right|}{d_r} \end{aligned}$$

Here, d_{before} defines the distance before the action, whereby the distance after the action is defined as d_{after}. Note that the distance change can also be denoted as velocity, since the measurements are time dependent. The term $|d_t - d|$ calculates the absolute distance difference between d_t and the current head position. The resulting normalization $\frac{|d_t - d|}{d_r}$ is 0, if $d = d_t$ and 1 if $d = d_t \pm d_r$. This effect is inverted by $1 - \frac{|d_t - d|}{d_r}$. As a result, the maximum reward of 1 is achieved if $d = d_t$.

In the task of autonomous target tracking, it is important to maintain vision of the target in order to react to its movement changes. It is worth noting, different from the evaluation metric defined in (2), that the reward function does not explicitly reward that behavior. The agent has to learn independently that it must observe the target's position in order to follow it.

5.2. Training

In order to map the input (observation $\mathbf{o}_i^t$) and the output (action $\mathbf{a}_i^t$), a fully connected 2-hidden-layer neural network is constructed as an approximator to the policy π_θ. The input layer and the output layer share the same dimensions as the observation space $\mathbf{o}_i^t$ and action space $\mathbf{a}_i^t$. The proximal policy optimization (PPO) algorithm adapted from is used to train the network, since PPO performs better on continuous action space tasks while being much simpler to implement and tune (Schulman et al., 2017). We train our policy network on a computer with an i7-9750H CPU and a Nvidia RTX 2070 GPU.

The model is trained by using the random track with a changing random seed for every episode. Therefore, a variety of tracks are generated and the model will not overfit to a specific track. This is necessary because it has been observed that the model tended to overfit while training on unvarying trajectories. As result for overfitting, the controller was not be able to adapt to other trajectory patterns. Based on the learning curve, a total of 3 million time steps (about 3,000 updates) were used for training (see **Figure 6**). The training process will terminate itself either when the target is out of view or reaching the end of the total time-steps. The mean reward gradually increases and levels up at around 2.4 with some fluctuation. This is because performance of the controller varies from the randomly changing track for each episode. The model at update 2,900 was selected for the further usage.

6. RESULTS AND DISCUSSIONS

This section will first describe the performance of the gaits generated by the RL controller in testing tracks. Then, we compare our gaits to the gaits generated from traditional model-based method in terms of tracking accuracy. Finally, we will give the limitation of the simulated results.

6.1. Results

The performance of the RL controller was tested on four different tracks (see **Figure 2**). For evaluation, the episode length is set to $3,000$ time steps. The trajectories of the head module of robot during the evaluation are shown using red solid lines in **Figure 7**, together with its corresponding track pattern (blue dash lines). In addition, the body curves are plotted every $1,000$ time step with the target position at that time using green lines and dots. For all four tracks, the RL controller was able to successfully follow the target. By comparing the trace of the snake-like robot and the track, a variation is observed in which the trajectories are not matched to each other. In some sections, they go in parallel or cross each other. This indicates that the snake-like robot is not heading directly for the target's position. In some cases, the head module's trace takes a shortcut in the curves of the target's track. However, the trace of the head module is maintaining a visible minimum distance. We can thus conclude that the controller performs a

TABLE 2 | The statistics for the head to target distances (unit: meters).

Track	Mean	Std	Min	Max
Line	3.99	0.18	3.60	4.49
Random	3.96	0.24	3.48	4.43
Wave	3.99	0.24	3.46	4.85
Zigzag	3.88	0.22	3.40	4.46

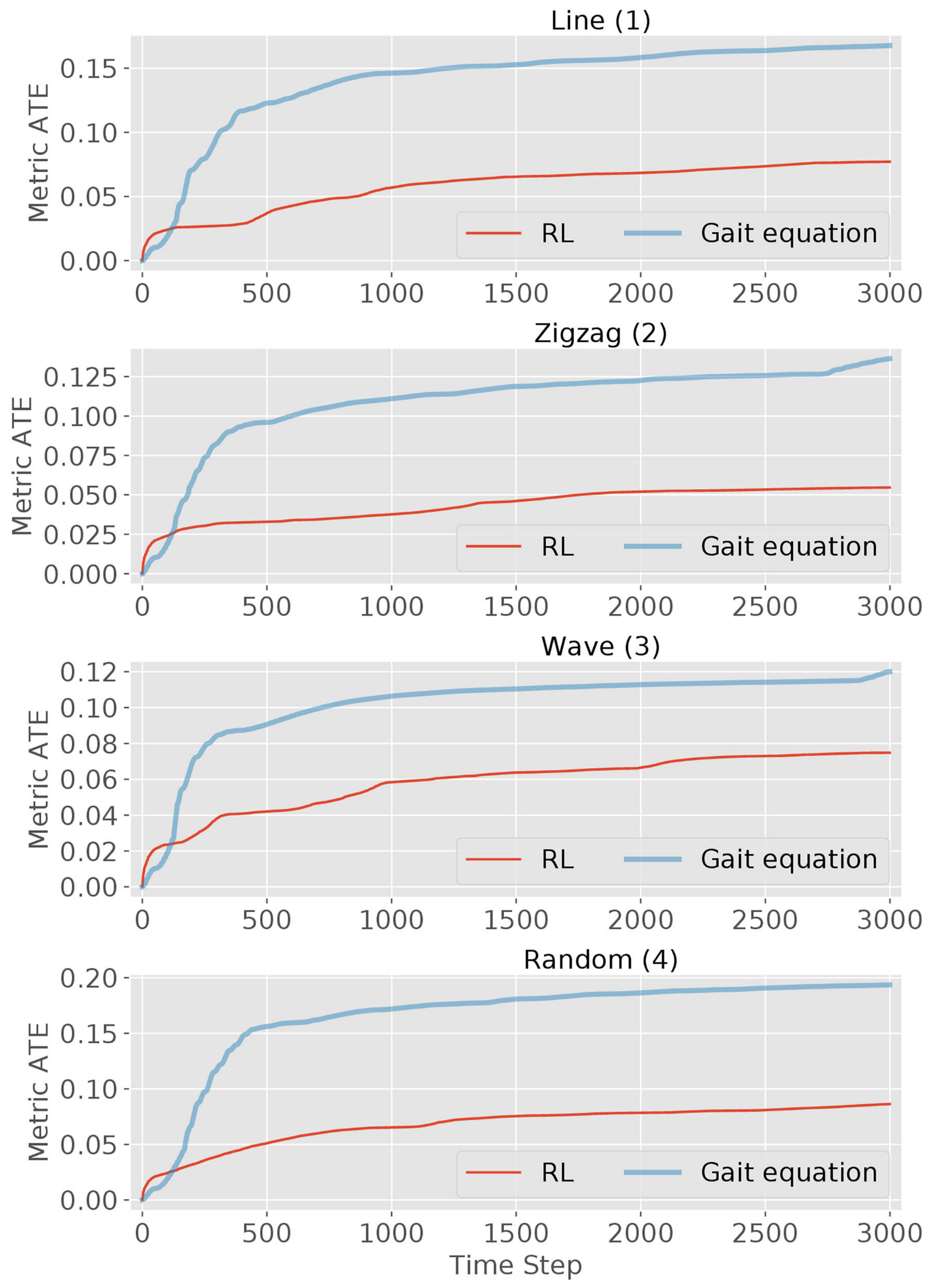

FIGURE 9 | The tracking metric curves of the RL controller and *gait equation* controller over the elapsed time (3,000 timestep in total) in four scenes. The tracking metric is defined in (2).

successful path-following behavior. Besides, the controller had to maintain a certain distance to the target. The red lines and the histogram in **Figure 8** show the distance distribution of runs on all tracks. In all runs, the distance varies around the target of 4.0 m which is measured at the center of the head. The oscillation movement of the head causes constant minimal distance changes. **Table 2** shows the statistics of the runs. As result, the controller was able to maintain the distance from the target with an adequate variance.

6.2. Comparisons

In order to evaluate the performances of the RL controller against the model-based method, the *gait equation* controller (section 4) was also executed on those four testing tracks.

We first compare the performance of the tracking accuracy. **Figure 8** shows the traces of the distance between the target and the robot for both controllers. In general, we can observe that the RL controller has a better tracking accuracy than the *gait equation* controller. For the RL controller, it exhibits better performances at the beginning process on all different tracks and keeps the distance very close to the desired value. Then with the accumulated error, all these four figures reach a relatively large error at some point, but then correct its direction to the right course. For the RL controller, the lag of tracking is much more smaller. After the starting of the movement, RL-based controller also exhibits better tracking accuracy. For the *gait equation* controller, it deviates most at the beginning of the movement for all four tracks. This is because the controller needs to response to the changing visual information and the effect will only show after the error has been accumulated for a period of time.

The second performance indicator is the tracking metric *ATE* defined in (2). The *ATE* is plotted over time steps in **Figure 9**. It can be observed that for both controllers, the metric curve gets higher due to the accumulated error with time passing by. But for all four scenes, the RL controller outperforms the *gait equation* controller: by around 50% in the simple *line* and *wave* scene and by 70% in the other difficult scenes. In conclusion, the RL controller outperform the *gait equation* controller both in terms of distance tracking accuracy and the averaged tracking error.

Since our training will terminate itself once the robot lose sight of the target, there will be no target recovery behavior obtained during training, such as retaining. In fact, the target recovery behavior is associated with some memory-like function that can predict the motion tendency of a moving object.

6.3. Limitations

It is worth noting that, our RL-based controller is demonstrated by simulations now and has not been support with results of physical snake-like robot yet. In order to ensure the validity of the simulated results, we first try to close the simulation-to-reality gap by setting simulation parameters with real-world properties (e.g., dimension, density, friction, etc.). Second, all the methods implemented in simulation can also be produced in a real-world setup.

For prototype experiments, the main challenge is how to train the RL controller in a real-world setup, which usually requires millions of episodes. Different from a robotic arm that can be set to its initial condition easily, there is no good way to reset the training scene for mobile robots in real world. Some algorithms (Fu et al., 2017; Hwangbo et al., 2019) may directly transfer the learned policy from simulation and implement it in real-world scenario. But this is out of the scope of this paper.

7. CONCLUSION

Performing target-tracking tasks for snake-like robots is a challenging task, since it not only involves designing agile locomotion patterns for the robot, but also overcoming difficulties to obtain stable visual information due to the inherent undulatory motions. In this paper, we try to solve this complex perception-to-action control task by using reinforcement learning, which directly maps the vision space to the joint space and reduces the computational complexity of dealing with object tracking and robot motion control in separate components. In our test scenarios, the learned gait shows much better tracking performances than the model-based method. Our work contributes to designing sophisticated and efficient moving patterns for perception-driven tasks with a snake-like robot.

Our future work will aim at performing tracking tasks with more complex visual information. For instance, the perception of the visual information can be replaced with more sophisticated technologies. To improve the adaptability of our RL controller, we will further investigate locomotion skills for more challenging scenarios, such as in a obstacle surrounding environment and the capability to recover tracking when the target runs out of the visual field of the robot.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

ZB, CL, and AK brought up the core concept. ZB and CL conducted the experiments and analyzed the results. ZB, CL, ZJ, LC, and KH wrote the paper. All authors contributed to the article and approved the submitted version.

FUNDING

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and the Specific Grant Agreement No. 945539 (Human Brain Project SGA3). Meanwhile, it was also supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program.

REFERENCES

Asada, M., Noda, S., Tawaratsumida, S., and Hosoda, K. (1996). Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Mach. Learn*. 23, 279–303.

Bing, Z., Cheng, L., Chen, G., Röhrbein, F., Huang, K., and Knoll, A. (2017). Towards autonomous locomotion: CPG-based control of smooth 3D slithering gait transition of a snake-like robot. *Bioinspir. Biomim*. 12:035001. doi: 10.1088/1748-3190/aa644c

Bing, Z., Jiang, Z., Cheng, L., Cai, C., Huang, K., and Knoll, A. (2019a). "End to end learning of a multi-layered SNN based on R-STDP for a target tracking snake-like robot," in *2019 International Conference on Robotics and Automation (ICRA)*, 9645–9651. doi: 10.1109/ICRA.2019.8793774

Bing, Z., Lemke, C., Jiang, Z., Huang, K., and Knoll, A. (2019b). "Energy-efficient slithering gait exploration for a snake-like robot based on reinforcement learning," in *The 28th International Joint Conference on Artificial Intelligence (IJCAI-19)* (Macao: IJCAI). doi: 10.24963/ijcai.2019/785

Busquets, D., de Mántaras, R. L., De, R. L. O., Antaras, M., Sierra, C., and Dietterich, T. G. (2002). "Reinforcement learning for landmark-based robot navigation," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (Bologna), 841–843. doi: 10.1145/544862.544938

Ding, L., Li, S., Gao, H., Chen, C., and Deng, Z. (2018). Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems. *IEEE Trans. Syst. Man Cybern. Syst*. 50, 2512–2523. doi: 10.1109/TSMC.2018.2819191

Evan, A. (2017). *What CMU's Snake Robot Team Learned While Searching for Mexican Earthquake Survivors*. Available online at: https://spectrum.ieee.org/automaton/robotics/industrial-robots/cmu-snake-robot-mexico-earthquake

Fu, J., Luo, K., and Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv* 1710.11248.

Fukushima, H., Satomura, S., Kawai, T., Tanaka, M., Kamegawa, T., and Matsuno, F. (2012). Modeling and control of a snake-like robot using the screw-drive mechanism. *IEEE Trans. Robot*. 28, 541–554. doi: 10.1109/TRO.2012.2183050

Garcia-Aracil, N., Perez-Vidal, C., Sabater, J. M., Morales, R., and Badesa, F. J. (2011). Robust and cooperative image-based visual servoing system using a redundant architecture. *Sensors* 11, 11885–11900. doi: 10.3390/s111211885

Gaskett, C., Fletcher, L., and Zelinsky, A. (2000). "Reinforcement learning for a vision based mobile robot," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)* (*Cat. No.00CH37113)*, Vol. 1 (Takamatsu), 403–409. doi: 10.1109/IROS.2000.894638

Hafner, R., and Riedmiller, M. (2003). "Reinforcement learning on an omnidirectional mobile robot," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (*Cat. No.03CH37453)*, Vol. 1 (Las Vegas, NV), 418–423.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., et al. (2017). Emergence of locomotion behaviours in rich environments. *arXiv* 1707.02286.

Hirose, S. (1993). *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*, Vol. 1093. Oxford: Oxford University Press.

Huang, K., Shao, K., Zhen, S., and Sun, H. (2017). A novel approach for modeling and tracking control of a passive-wheel snake robot. *Adv. Mech. Eng*. 9:1687814017693944. doi: 10.1177/1687814017693944

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

Ishikawa, M., Morin, P., and Samson, C. (2009). "Tracking control of the trident snake robot with the transverse function approach," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly With 2009 28th Chinese Control Conference* (Shanghai), 4137–4143. doi: 10.1109/CDC.2009.5399958

Kamegawa, T., Yarnasaki, T., Igarashi, H., and Matsuno, F. (2004). "Development of the snake-like rescue robot "kohga"," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, Vol. 5 (New Orleans, LA), 5081–5086. doi: 10.1109/ROBOT.2004.1302523

Liljeback, P., Pettersen, K. Y., Stavdahl, A, and Gravdahl, J. T. (2011). Controllability and stability analysis of planar snake robot locomotion. *IEEE Trans. Autom. Control* 56, 1365–1380. doi: 10.1109/TAC.2010.2088830

Luo, W., Sun, P., Zhong, F., Liu, W., Zhang, T., and Wang, Y. (2019). End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell*. 1, 1317–1332. doi: 10.1109/TPAMI.2019.2899570

Matsuno, F., and Mogi, K. (2000). "Redundancy controllable system and control of snake robots based on kinematic model," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, Vol. 5 (Sydney, NSW), 4791–4796. doi: 10.1109/CDC.2001.914686

Matsuno, F., and Sato, H. (2005). "Trajectory tracking control of snake robots based on dynamic model," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (Barcelona), 3029–3034. doi: 10.1109/ROBOT.2005.1570575

Miljkovicć, Z., Mitić, M., Lazarević, M., and Babić, B. (2013). Neural network reinforcement learning for visual control of robot manipulators. *Expert Syst. Appl*. 40, 1721–1736. doi: 10.1016/j.eswa.2012.09.010

Miller, G. S. (1988). The motion dynamics of snakes and worms. *ACM Siggraph Comput. Graph*. 22, 169–173. doi: 10.1145/378456.378508

Morbidi, F., and Mariottini, G. L. (2013). Active target tracking and cooperative localization for teams of aerial vehicles. *IEEE Trans. Control Syst. Technol*. 21, 1694–1707. doi: 10.1109/TCST.2012.2221092

Ohno, K., Nomura, T., and Tadokoro, S. (2006). "Real-time robot trajectory estimation and 3d map construction using 3D camera," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing), 5279–5285. doi: 10.1109/IROS.2006.282027

Ponte, H., Queenan, M., Gong, C., Mertz, C., Travers, M., Enner, F., et al. (2014). "Visual sensing for developing autonomous behavior in snake robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 2779–2784. doi: 10.1109/ICRA.2014.6907257

Prautsch, P., Mita, T., and Iwasaki, T. (2000). Analysis and control of a gait of snake robot. *IEEJ Trans. Ind. Appl*. 120, 372–381. doi: 10.1541/ieejias.120.372

Sanfilippo, F., Azpiazu, J., Marafioti, G., Transeth, A. A., Stavdahl, A., and Liljebäck, P. (2017). Perception-driven obstacle-aided locomotion for snake robots: the state of the art, challenges and possibilities. *Appl. Sci*. 7:336. doi: 10.3390/app7040336

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv* 1707.06347.

Takahashi, Y., Takeda, M., and Asada, M. (1999). "Continuous valued q-learning for vision-guided behavior acquisition," in *Proceedings. 1999 IEEE/SICE/RSJ. International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI'99 (Cat. No.99TH8480)* (Taipei), 255–260. doi: 10.1109/MFI.1999.815999

Tanaka, M., and Matsuno, F. (2014). Control of snake robots with switching constraints: trajectory tracking with moving obstacle. *Adv. Robot*. 28, 415–429. doi: 10.1080/01691864.2013.867285

Tanaka, M., and Tanaka, K. (2015). Control of a snake robot for ascending and descending steps. *IEEE Trans. Robot*. 31, 511–520. doi: 10.1109/TRO.2015.2400655

Tanaka, M., Tanaka, K., and Matsuno, F. (2015). Approximate path-tracking control of snake robot joints with switching constraints. *IEEE/ASME Trans. Mechatron*. 20, 1633–1641. doi: 10.1109/TMECH.2014.2367657

Tesch, M., Lipkin, K., Brown, I., Hatton, R., Peck, A., Rembisz, J., et al. (2009). Parameterized and scripted gaits for modular snake robots. *Adv. Robot*. 23, 1131–1158. doi: 10.1163/156855309X452566

Todorov, E., Erez, T., and Tassa, Y. (2012). "Mujoco: a physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. doi: 10.1109/IROS.2012.6386109

Transeth, A. A., van de Wouw, N., Pavlov, A., Hespanha, J. P., and Pettersen, K. Y. (2007). "Tracking control for snake robot joints," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA), 3539–3546. doi: 10.1109/IROS.2007.4399174

Walker, I. D., and Hannan, M. W. (1999). "A novel 'elephant's trunk' robot," in *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)* (Atlanta, GA: IEEE), 410–415. doi: 10.1109/AIM.1999.803204

Webster, R. J., Okamura, A. M., and Cowan, N. J. (2006). "Toward active cannulas: miniature snake-like surgical robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing), 2857–2863. doi: 10.1109/IROS.2006.282073

Xiao, X., Cappo, E., Zhen, W., Dai, J., Sun, K., Gong, C., et al. (2015). "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle), 3735–3740. doi: 10.1109/ICRA.2015.7139718

Yamada, H., Takaoka, S., and Hirose, S. (2013). A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism). *Adv. Robot*. 27, 47–60. doi: 10.1080/01691864.2013.752318

Yun, S., Choi, J., Yoo, Y., Yun, K., and Choi, J. Y. (2018). Action-driven visual object tracking with deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst*. 29, 2239–2252. doi: 10.1109/TNNLS.2018.2801826

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