IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Goal-Oriented Control Strategies for Soft Growing Robots

Wentao Huang, Pengchun Li, Ziyi Zhang, Zuankai Wang, Dekai Zhou, and Longqiu Li

***Abstract*—Soft growing robots, as highly mobile pneumatic membrane robots, are limited in control performance due to their soft structure and nonlinear mechanical properties, especially under dynamic conditions. Therefore, developing reliable control strategies for the robot is essential. This study proposes a dual-thread, goal-oriented control strategy for soft growing robot that combines planning and control. By integrating graph convolutional networks with deep reinforcement learning, the global path planning method is better suited to the self-growing behaviors of soft robots, leading to improvements in both computational efficiency and accuracy compared to inverse kinematics planning methods. Motion control reduces the adverse effects of deformation errors caused by its own low stiffness or by disturbances in the external environment. This strategy effectively combines reinforcement learning-based global planning with a multiple closed-loop motion control system, addressing the issues of low precision and reliability under dynamic conditions. Experimental results demonstrate that the robot achieves a tracking accuracy of 11.83 mm within a 5-meter range and successfully tracks and approaches a non-cooperative dynamic target. These results highlight the significant potential of the proposed approach in applications such as target capture and dynamic manipulation.**

***Index Terms*—Modeling, control, and learning for soft robots, motion and path planning, reinforcement learning.**

I. Introduction

SOFT growing robots provide key benefits for inspection and exploration in uncharted environments, such as flexible movement, exceptional adaptability, and high safety [1], [2], [3], [4], [5]. Since their creation, soft growing robots have been designed for tasks that involve soft contact and adaptive deformation. However, their soft mechanical systems possess theoretically infinite degrees of freedom and exhibit complex motion behaviors. This complexity hinders the ability to achieve reliable motion intelligence under dynamic conditions. Therefore, it is essential to develop intelligent systems specifically for these robots to enhance their application in dynamic environments.

A variety of theoretical models and control methods are currently available for continuum soft robots [6], [7], [8], [9], [10]. Della et al. has conducted extensive research on their modeling and control, including the use of Inertial Measurement Units (IMUs) for pose estimation [11], the development of reduced-order models that capture the theoretically infinite-dimensional kinematic properties of soft robots [12], and investigations into model-based control addressing system uncertainties and input saturation [13], yielding promising results. Hawkes et al. [14], [15] proposed a vision system integrated into the tip of a soft growing robot that monitors the robot's field of view in real time. Coad et al. [16], [17], [18], [19] introduced various motion system designs for soft growing robots, making significant contributions to their motion flexibility, reliability, and accuracy. Selvaggio et al. [20] and Joseph D et al. [21] have focused on growth path planning with obstacle avoidance, which achieves robust path planning through robot-environment interaction and greatly exploits the low inertia and compliant of soft robot body. Connor et al. [22] proposed an approach for controlling soft growing robots based on leveraging real-time position and orientation measurements. In a statically constrained environment, the approach enables soft robots to be controlled with an accuracy of about 4.5 mm. Yuki et al. [23] proposed an efficient path planning method for soft growing robots with constraints on bending angles. However, in the aforementioned work, the robot system demonstrates low adaptability to dynamic environments.

To address the challenges in real-time environments, Emanuela et al. [3] and A Sadeghi et al. [24] developed strategies to respond to stimuli such as light, temperature, humidity, and gravity, based on the multi-sensory capabilities and tendency-based behaviors of plant. Enrico et al. [25] developed a sensory-driven grasping strategy for continuum robots without model assumptions. Reactive strategies endows the robot with the ability to navigate and explore in real time but lacks the ability to optimize movement when interacting with specific objects. In order to avoid the dynamic accuracy defects caused by the global planning and the inefficient exploration of the global optimum by feedback control, this study focuses on the problem of real-time tracking of dynamic targets by soft growing robots and proposes a control strategy combining planning with feedback control.

Reinforcement learning, as a nonlinear self-modeling method, has demonstrated powerful capabilities in control and planning [26], [27], [28], [29]. Zhou et al. [30] and Nazeer et al. [31]

Received 3 March 2025; accepted 3 August 2025. Date of publication 13 August 2025; date of current version 20 August 2025. This article was recommended for publication by Associate Editor E. Falotico and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFE0209900, and in part by the National Natural Science Foundation of China under Grant 52125505, and Grant T2388101. *(Wentao Huang and Pengchun Li are co-first authors.)(Corresponding authors: Dekai Zhou; Longqiu Li.)*

Wentao Huang, Dekai Zhou, and Longqiu Li are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China, and also with Zhengzhou Research Institute, Harbin Institute of Technology, Zhengzhou 450003, China (e-mail: dekaizhou@hit.edu.cn; longqiuli@hit.edu.cn).

Pengchun Li and Ziyi Zhang are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China.

Zuankai Wang is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong SAR 99077, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3598668, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3598668

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

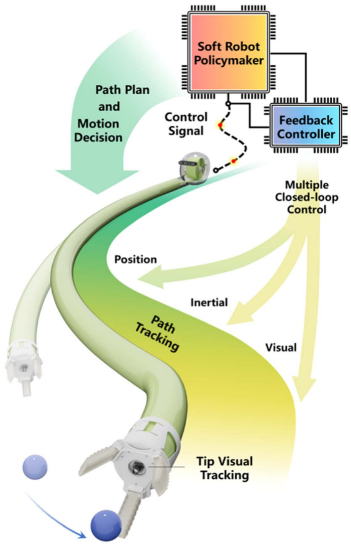

Fig. 1. Control strategies for soft growing robot.

applied reinforcement learning to the motion control of continuum soft robots, achieving improved control accuracy and planning results. Research related to reinforcement learning also plays a significant role in motion planning for multi-joint snake robots [32] and in the multi-stage dynamic control of soft robotic hands [33], [34].

This work proposes a dual-threaded decision-making system comprising two parts: robot motion control and global planning. The system can collect environmental data in real time as feedback and autonomously drive the robot to track a non-cooperative dynamic target. The planning method can take full advantage of the growth behavior of the robot to quickly generate high-precision and low-loss growth paths. In motion control, multiple real-time feedback control systems are designed to effectively correct motion errors caused by factors such as insufficient stiffness or external friction. Finally, dynamic target tracking experiments are designed to validate the accuracy of both the method and the system.

II. Control System Design for Soft Growing Robot

A. Robot System Design

Fig. 1 shows the goal-oriented control strategies for soft growing robots proposed in this letter. The control system consists of a Policymaker and a Motion controller. The policymaker plans the robot's motion path by synthesizing state information as well as delivers motion commands, and the motion controller completes the closed-loop control of the robot's motion based on the signals returned from the robot's sensors to ensure the accuracy of the motion. At the same time, the motion controller forms a comprehensive feedback by integrating multiple visual and posture signals to assist the optimization of the policymaker's motion strategy, so that it can make real-time corrections to the path.

Fig. 2(a) shows the structure and composition of the robot. Robot system contains three parts: Base, Main Body and Tip. Base stores the robot body and is equipped with a base camera. Body is a multi-jointed soft robot composed of repetitive single-jointed components. Tip is a tip-loading device that is connected to the robot, equipping with a tip camera. The robot's body is a cylindrical film structure. When working, air pressure is generated inside the base and the body by an external air source, and the body expands into a long cylindrical shape. The internal pressure is controlled by a PID controller to maintain the pressure at around $+15$kPa. The driver module and other components are arranged on the inner surface of the body. When the air pressure is sufficient, the tip of the robot will be subjected to sufficient force to drive the robot to grow forward, while the speed of the growth is limited by the rotation speed of the shaft in the base. Each joint contains three motors, each of which can drive a drive wire. The three motors and their drive wires are evenly distributed on the circular cross-section of the main body, separated from each other by 1/3 of the circumference. With this distribution, the robot can turn 360° using motor drive. As shown in Fig. 2(a), the drive wires are named A, B, and C, separately. When C is driven alone, the robot can move in the direction of the intersection between R1 and R3. When A and C are driven together, the robot can move within the R1 area. Similarly, driving any two motors can enable the robot to move in any area of R1, R2 and R3, achieving omnidirectional movement of the robot. Each joint contains an inertial measurement unit (IMU) and control chips, which is used to control the motor and provide feedback. The robot body is composed of the same joint structure mentioned above, with each joint having a length of 1m, meaning that the drive motors are spaced 1m apart

B. Control System Components

As shown in Fig. 2(b), the control system can be divided into four parts, which are soft robot, the visual representation, the dynamic path planning, and the robot motion control. In terms of hardware, the IMUs and the control chips are connected to the robot bus, which is connected to the host computer, using the CAN bus communication protocol. Joint detector are installed inside the base to provide feedback on the number of joints that the robot has grown. The ungrown body is stored in a roll on the shaft of the motor inside the base, so the growing speed can be controlled by the motor. The motor speed is controlled by PID controller based on a reference speed. Software consists of visual representation, dynamic programming (in host device) and motion control (in slave device).

C. Control Method

The PCC model is particularly suitable for robots with high flexibility, bendable structures, and no rigid joints [11]. The soft growing robot in this work uses pneumatic membranes as its body and utilizes wire drives to achieve joint movement, which is characteristic of this type of robot [1]. Therefore, this work uses the PCC model to model kinematics. Unlike traditional PCC, the robot in this work possesses growing characteristics, and growing variables need to be considered in modeling. As shown in Fig. 2(c), the joint steering of the robot can be represented by the steering angle θ_n and the deflection angle φ_n. The coordinate system O_{n+1} can be regarded as formed by O_n after a displacement and three rotation transformations, and the displacement can be expressed as:

$$P_{n+1}^{n} = \begin{bmatrix} & I_3 & \begin{matrix} \cos\varphi_n \cdot r \cdot (1-\cos\theta_n) \\ \sin\varphi_n \cdot r \cdot (1-\cos\theta_n) \\ l - r\cdot\theta_n + r\cdot\sin\theta_n \end{matrix} \\ 0 & 0 \quad 0 & 1 \end{bmatrix} \tag{1}$$

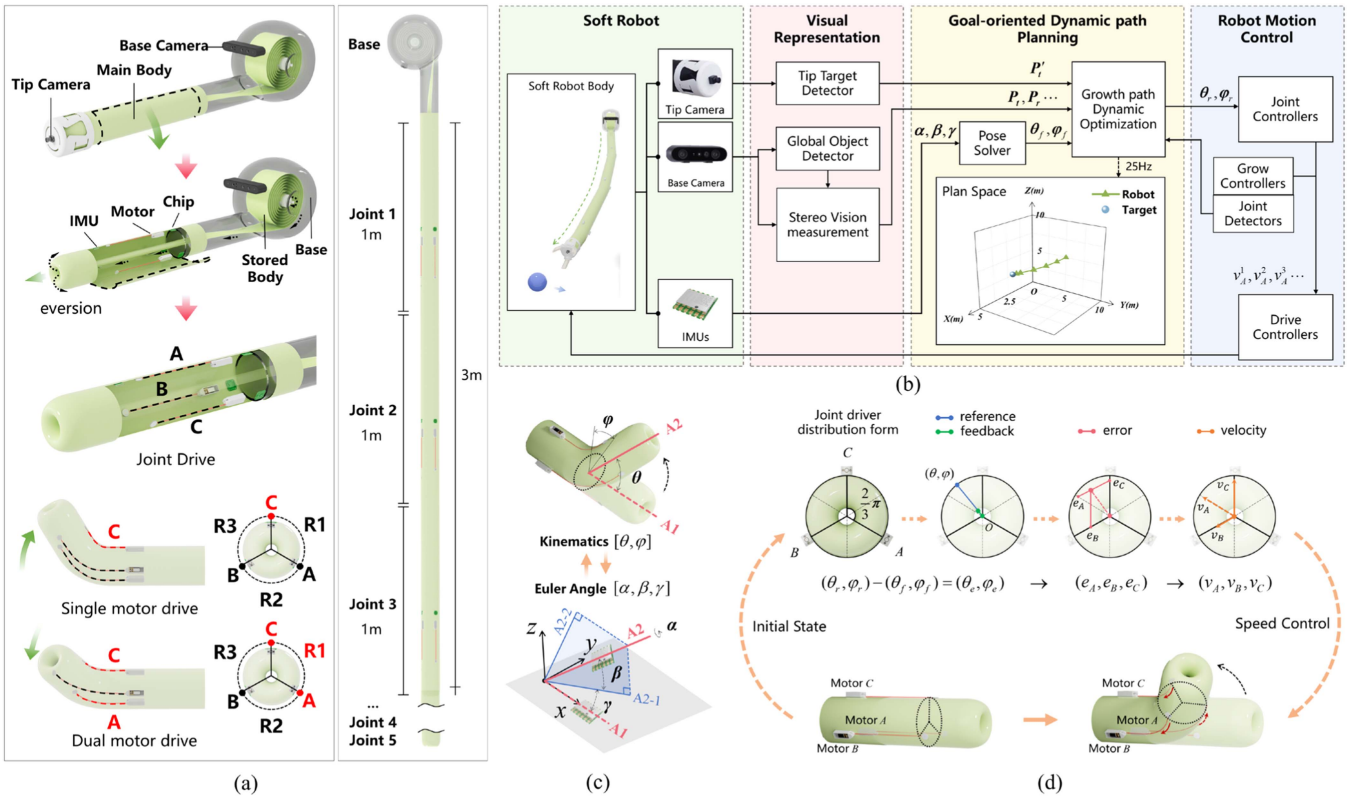

Fig. 2. Soft robot system. (a) Soft robot mechanical structure. (b) Control system design. (c) Pose solver for transforming sensor data into kinematic data. (d) Control allocator for converting joint control quantities into motor control quantities.

where l denotes the total length of a joint. r denotes the cross-sectional radius of the robot body after inflation. Then the rotation matrix from O_1 to O_n can be expressed as:

$$R^n_{n+1} = R(Z_n, \varphi_n)R(Y_n, \theta_n)R(Z_n, -\varphi_n) \tag{2}$$

$$T^n_{n+1} = P^n_{n+1}R^n_{n+1} \tag{3}$$

$$T^n_1 = T^1_2 \cdots T^{n-2}_{n-1}T^{n-1}_n \begin{bmatrix} & I_3 & & 0 \\ & & & 0 \\ & & & l_t + \frac{d}{dt}l_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where l_t with its derivative can represent the value of tip extension and the velocity. Due to the repeatability of joints, length changes only occur at the tip joints, so the overall length change can be expressed by the number of fully grown joints n and the tip length l_t.

According to the kinematics, the control system generates references for each joint as θ_n and φ_n, while the data fed back from the IMU are Euler angles, therefore, a pose solver is needed to convert the two to each other. As shown in Fig. 2(c), the Euler angles α, β, γ denote the angles of rotation of the robot joints around the x, y, and z axis, respectively, while the kinematic parameters θ, φ denote the steering and deflection angles of the soft robot. According to the geometrical, the expression of θ, φ calculated from α, β, γ can be expressed as:

$$\begin{cases} \varphi = \varphi', \beta \geq 0, \gamma \geq 0 \\ \varphi = \pi - \varphi', \beta < 0, \gamma \geq 0 \\ \varphi = \pi + \varphi', \beta < 0, \gamma < 0 \\ \varphi = 2\pi - \varphi', \beta \geq 0, \gamma < 0 \end{cases}, \varphi' = \arctan\left(\frac{\cos\beta \cdot \sin\gamma}{\sin\beta + \tau}\right)$$

$$\theta = \arccos\left(\cos(\beta) \cdot \cos(\gamma)\right) \tag{5}$$

where β, γ denote the Euler angles, θ, φ denote the kinematic parameters, and τ is a minima preventing the denominator from being 0. Since the soft robots do not have the body twisting around the x-axis during the modeling process, α is not involved in the calculation. This solver allows for the real-time conversion of sensor feedback data into kinematic data, forming a feedback that participates in control calculations.

As shown in Fig. 2(d), the cross-section of the robot body taken at the position of the drive motors is the steering plane of the robot. As in the front view of the robot in Fig. 2(d), the reference of the control (blue) and the current feedback of the robot (green) can be represented in polar coordinates on the steering plane of the robot, and the reference minus the feedback is the current error (light red). According to the arrangement position of the drive wire, the robot joint steering plane is divided into three regions by the drive wires A, B and C. By calculating the coordinate values of the projection points e_A, e_B and e_C of the error coordinate point (light red) in the directions of A, B and C, the distance between the current and target positions of the robot joints in the three drive directions can be solved. This distance is the input error E of the steering motor controller, and these three values E_A, E_B, and E_C can be expressed respectively as: (6) to (8) shown at the bottom of the next page

By inputting the values of E_A, E_B, and E_C into the controller, the controller will output control signals to drive the robot in the direction of the reducing error, and eventually the robot's head will point in the reference direction.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Similarly, when performing tip vision control, it can be achieved if the error is replaced with the vision error. In the case of multi-joint control, single-joint control methods can be equivalently applied to multi-joint control due to the repetitive structure and joint detector.

III. Growing Path Plan for Soft Growing Robot

Motion control methods are accurate and fast, but they are unable to plan the robot's motion globally, thus often resulting in local optimization. To compensate for this limitations, this work establishes the GAT-RL Growing algorithm for soft growing robots based on PPO algorithm. The algorithm can operate the robot to move close to the target by outputting control commands. Compared with the IK-based methods, this method incorporates graph data representation of the environmental state and the growth properties to match the mechanical behavior of soft growing robots.

A. Planning Method for Soft Growing Robot

As shown in Agent part of Fig. 3(a), the Actor network and Critic network of RL Agent are built with similar structure, both of them contain two layers of Graph Attention and two layers of tanH activation function. And then the Actor network outputs the action value via two liner layers, and the Critic network outputs the state value from the liner layer with one output dimension. The last layer of the Actor network uses the tanH activation function layer. The value domain of TanH is [-11], so it can effectively control the size of the value when outputting. Accordingly, the target networks are included in the Agent, and the soft update is used to update the networks to stabilize the training process.

Graph State Space: In the state space, in order to achieve efficient goal orientation as well as complete motion feedback, this work represents robot joints as well as targets uniformly as nodes of graph data. In the node features, the kinematic and positional information of the object is added. The nodes are connected by dynamic edges. The node features as well as the information of the edges will change according to different states, fully describing the relationship between the robot's joints and the target in each growth phase, giving the decision maker a sufficient information base. As shown in the State Graph part of Fig. 3(a), the feature can be represented as:

$$f_J = (\theta_i, \varphi_i, x_i, y_i, z_i) \tag{9}$$

$$f_T = (0, 0, x_t, y_t, z_t) \tag{10}$$

where θ, φ denote kinematic parameters, x, y, z denote joint end coordinates, i represents the i-th joint, and t represents the target.

The number of edges in the graph depends on the growing state, and changes dynamically according to the growth of the robot. For example, if the robot currently grows n joints, and there are n-1 edges between joints and n edges between joints and targets, there will be a total of $(n\text{-}1)+n$ edges in the graph. Fig. 3(a) shows the number of edges for $n = 5$.

The above analysis then leads to the construction of the feature matrix F and the adjacency matrix A' of the graph. The state space of a fully grown five-joint robot can be represented as:

$$S = \left\{ F = \begin{bmatrix} \theta_1 & \varphi_1 & x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_n & \varphi_n & x_n & y_n & z_n \\ 0 & 0 & x_t & y_t & z_t \end{bmatrix}, \right.$$

$$\left. A' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \right\} \tag{11}$$

Action Space: The Action space can also be considered as the robot's motion commands. Unlike other soft robots, soft growing robot achieves body extension by the eversion of the tip, while the joints will maintain their position without displacement. Based on such characteristics, this work introduces the growing mark g. The activation or deactivation of g in the program determines the growing action of the robot. At the same time, in order to take advantage of joint fixation, the motion commands in this work will not cover the whole joints of the robot, and the joints that receive the motion commands will gradually move forward as the robot grows,. The active joints will always remain at the tip. The advantage of this approach is that it can minimize the impact of the soft robot's low stiffness or external forces on its motion. When a robot grows to a length exceeding 2 or 3 meters (two or three joints), the drive generated at the root position is difficult to transmit to the tip of the robot. Even if transmission is possible, it is difficult to ensure accuracy, which can cause vibration, drive failure, and other problems, resulting in the loss of some or even all of the system's functionality. In addition, this approach can reduce control dimensions and improve control stability. As shown in the Action part of Fig. 3(a), the action space of Agent can be represented as:

$$A_l = [g, v_l^g, v_l^\theta \cdots, v_l^\varphi \cdots] \tag{12}$$

Where A_l represents the Action of the Agent at the l-th step, and g represents the self-growing mark. v_l^g, $v_l^\theta \cdots$, $v_l^\varphi \cdots$ represent the growth velocity, the steering angular velocity and the deflection angular velocity. The value of both angular velocity is determined by the number of active joints. Due to the value

$$E_A = \begin{cases} d_1, -\frac{\sqrt{3}}{4}Y + \frac{3}{4}X \geq 0 \\ -d_1, -\frac{\sqrt{3}}{4}Y + \frac{3}{4}X < 0 \end{cases}, d_1 = \sqrt{\left(-\frac{\sqrt{3}}{4}Y + \frac{3}{4}X\right)^2 + \left(\frac{1}{4}Y - \frac{\sqrt{3}}{4}X\right)^2} \tag{6}$$

$$E_B = \begin{cases} d_2, \frac{\sqrt{3}}{4}Y + \frac{3}{4}X < 0 \\ -d_2, \frac{\sqrt{3}}{4}Y + \frac{3}{4}X \geq 0 \end{cases}, d_2 = \sqrt{\left(\frac{\sqrt{3}}{4}Y + \frac{3}{4}X\right)^2 + \left(\frac{1}{4}Y + \frac{\sqrt{3}}{4}X\right)^2} \tag{7}$$

$$E_C = Y \tag{8}$$

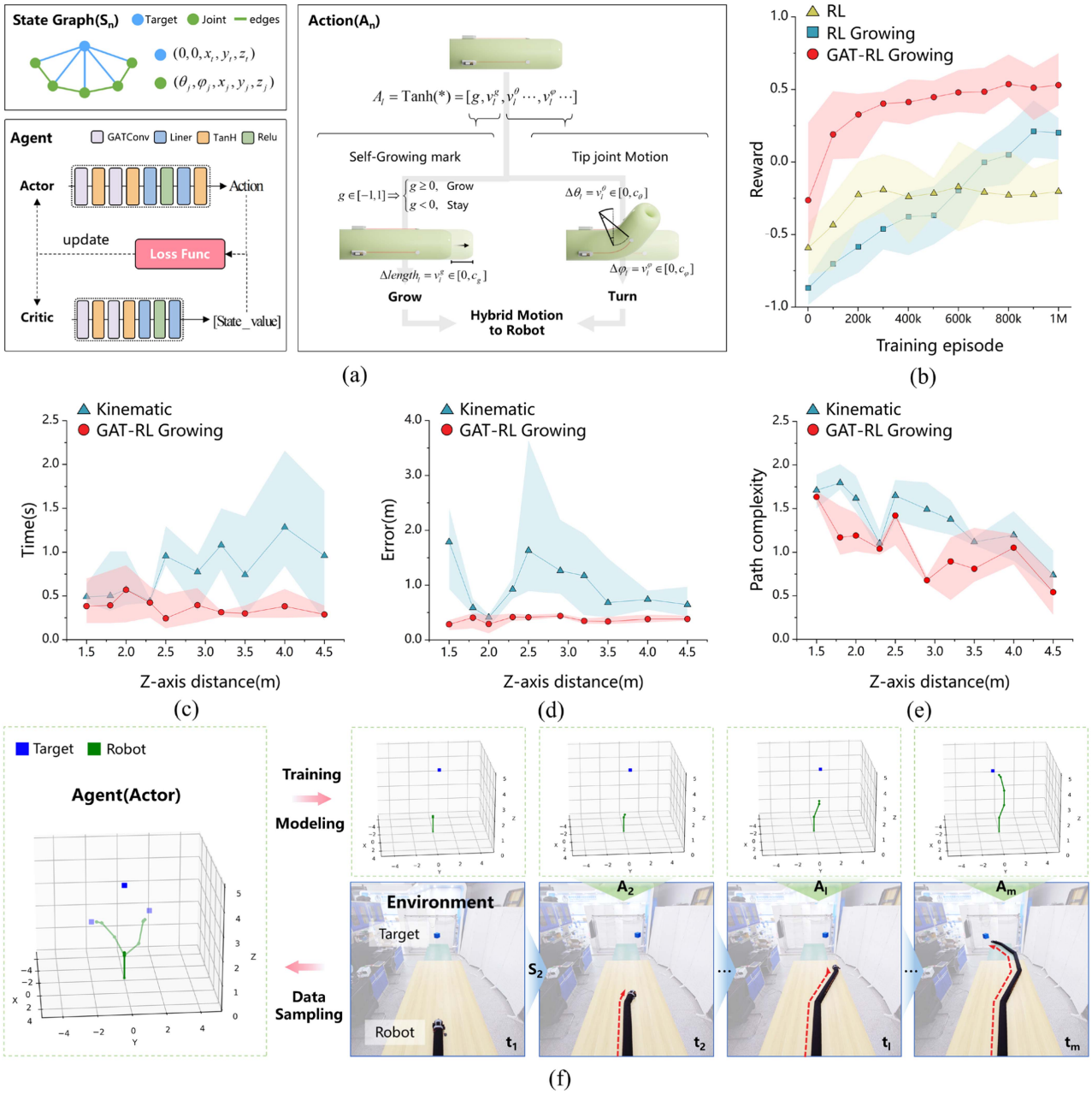

Fig. 3. Path Plan Algorithm for Soft Growing Robot. (a) Algorithm Frame. (b) Comparison results between RL, RL Growing and GAT-RL Growing. (c)(d)(e) Comparative results between GAT-RL Growing methods and Kinematics methods. (f) Data-based interaction of algorithms with real-world environments.

domain of the tanH, the value of g will be in the interval [-11], therefore, the growth action of the robot will be set to grow forward when $g>0$, and stop growing when $g<0$. Then, in the initial state, the robot will take a probability of 0.5:0.5 in the judgment of whether to grow or not. By introducing g to control growth, the impact of network output errors on growth motion can be mitigated, allowing the robot to remain stationary when growth is not required. Similarly, the values ofv_l^g, v_l^θ, v_l^φ are in the interval [-11] and can be deflated to adjust their value domains to the correct position, for example:

$$v_l^g = (vt_l^g + 1) \cdot c_g \tag{13}$$

Where vt_l^g denotes the value output directly from the network, while v_l^g denotes the value after performing the value domain adjustment, and c_g denotes the maximum value that can be accepted, so the new value domain is [0, c_g], as expected. Similarly, v_l^θ and v_l^φ can be adjusted by the same way. Since the controlled object is the parameters of each joint, the change in length is represented as the number of fully grown joints in the feedback. The growth speed generated by the planner is used to calculate the optimal path, while the actual growth speed of the robot is controlled by the growth motor.

Reward Function: The reward function used in this work can be expressed as:

$$R = \lambda \cdot (-1 + r_{\Delta d} - r_p) \tag{14}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

where R denotes the reward at a certain moment, λ is the adjustable parameter, $r_{\Delta d}$ is Potential-Based Reward, and r_p is the Energy Dissipation Reward.

Potential-Based Reward$r_{\Delta d}$:

$$r_{\Delta d} = \Delta d \cdot (\cos\beta + 1) \tag{15}$$

Where Δd denotes the variation in the distance between the robot position and the target position, the larger the value of Δd, the stronger the tendency of the robot to make a transfer to a state close to the target. β denotes the angle between two lines connecting the robot and the target in two neighboring moments, and the larger the value of $\cos\beta$, the stronger the tendency of the robot to transfer to a state toward the target.

Energy Dissipation Rewardr_p:

$$r_p = \eta_1 \cdot |A_\theta| + \eta_2 \cdot n \tag{16}$$

Where A_θ and n denote the vectors consisting of the steering angle of the joints and the number of joints that have been grown, respectively, so that these two values are added as penalties to the reward function, which can reduce the unnecessary steering and growth of the robot, and optimize the path complexity and energy dissipation. After reaching the target point or reaching the limit of the number of optimization steps, a Done Reward is generated and the Episode is ended.

Done RewardR_{done}:

$$R_{\text{done}} = 1 - \lambda \cdot r_p,\ if goal \tag{17}$$

$$R_{\text{done}} = -1 - \lambda \cdot r_p,\ if Reach the limit \tag{18}$$

Where 1 and -1 are used as the limiting values of the gain, when the target is not reached but the limit of the optimization step is reached, the reward will be close to -1, and when the target is reached, the reward will be close to 1, which can reduce the difficulty of training and speed up the convergence.

Loss Function: The loss function represents the updating direction of the parameters of the Agent network. The loss function mainly contains estimated state value loss $\mathcal{L}(\boldsymbol{w})$ and action value target $J(\boldsymbol{\theta})$, which correspond to the parameter optimization of Critic network and Actor network, respectively. Meanwhile, in order to encourage the Agent to take more aggressive strategies and avoid falling into local optimality, differential entropy h of the strategy distribution sampled by action A is introduced to enhance the uncertainty of its output. Therefore, the loss function in this work contains three parts, which can be expressed as:

$$\mathcal{L}_{total} = k_1 \cdot \mathcal{L}(\boldsymbol{w}) - k_2 \cdot J(\boldsymbol{\theta}) - k_3 \cdot h \tag{19}$$

$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}\left[\left(R - \hat{v}(S, \boldsymbol{w})\right)^2\right] \tag{20}$$

$$J(\boldsymbol{\theta}) = \mathbb{E}_{s,a\sim\pi_{\boldsymbol{\theta}}}\left[\mathbb{E}\left[v - \hat{v}\right] \cdot \pi(a|s, \boldsymbol{\theta})\right] \tag{21}$$

where $\mathcal{L}(\boldsymbol{w})$ is the mean square error between the actual reward and the state value estimate, $J(\boldsymbol{\theta})$ is the mathematical expectation of the action value, and k_1, k_2, and k_3 are the coefficients.

B. Growing Path Plan

By tracking dynamic targets during training, the reward performance of the agent will eventually converge. Fig. 3(b) shows the training data of three methods. RL indicates that deep reinforcement learning is directly used to fit the inverse kinematics model of the soft robot, RL Growing indicates that growth markers are added to reinforcement learning to realize growing planning, and GAT-RL Growing indicates that graph convolution and growing planning are simultaneously combined into reinforcement learning. It can be seen that the RL method is difficult to achieve convergence, and the RL Growing method has slower convergence and worse long-term rewards. While the GAT-RL Growing method can achieve fast convergence with higher long-term rewards.

Fig. 3(c), (d), and (e) represent the performance comparison between GAT-RL Growing and inverse kinematics planning methods in terms of computational efficiency, error, and path complexity, respectively. Ten dynamic points are randomly selected in the robot's workspace, and the two methods are utilized to plan multiple times and compare the results. The horizontal axis represents the initial distance of the 10 position points from the robot base in the Z-axis direction, which is also the initial robot orientation. In addition, since soft robot steering causes not only deformation errors but also control errors, the path complexity in Fig. 3(e) is measured as the modulus of the vector consisting of the robot steering angles. From the figure, it can be seen that GAT-RL Growing has the advantage of accuracy and stability in the comparison, which confirms the effectiveness of the method in this study.

As shown in Fig. 3(f), in a time step t_l, the Agent will collect a variety of data from the environment and convert it into graph data S_l, which is input into the Agent network. Agent will use this as the initial state to plan many times, and finally select the planning result with the highest reward R as the output A_l. The output will be converted into contorl commands to be transmiited to the robot. The robot motion may be not synchronized with the Agent planning, but the latest data is acquired. Compared to synchronous planning, this approach prevents the Agent from falling into a local optimum that leads to failure, while enabling continuous optimization of the generated paths. The control system continuously loops the execution of the above process at each time step, then the robot continuously moves in the direction of the target. After several time steps, the robot will eventually get closer to the target, at same time, the distance between the robot tip and the target to determine whether the Stop sign is true, if it is true, then stop, otherwise continue to execute until it reaches the target.

IV. Experiments

In order to verify the correctness of the method, target tracking experiments are designed in this work. As shown in Fig. 4(a), this work builds an experimental site for testing the tracking performance of a soft growing robot. The position data in the experiment was measured by a motion capture system.

Before performing the dynamic target test, the tracking accuracy of the robot system was measured. A total of 50 tracking sessions with different target initial positions were conducted in the experiment, and the final experimental results were obtained as shown in Fig. 4(b), (c). In Fig. 4(b), the relative distance between the robot and the target over time is shown. The distance at 0s shows the initial distance between the robot and the target. In all experiments, the target initial with dimensions of 4.8 m $\times$ 1.2 m. In 25% of the experiments, we changed the initial orientation of the robot to test robots' robustness to this. In Fig. 4(c), the relative position between the robot and the target after the robot reached the target in all experiments is shown. In the figure, L_t denotes the distance from the center to the edge of the target and L_h denotes the distance from the robot tip to the position capture point. Together, they form an unreachable area

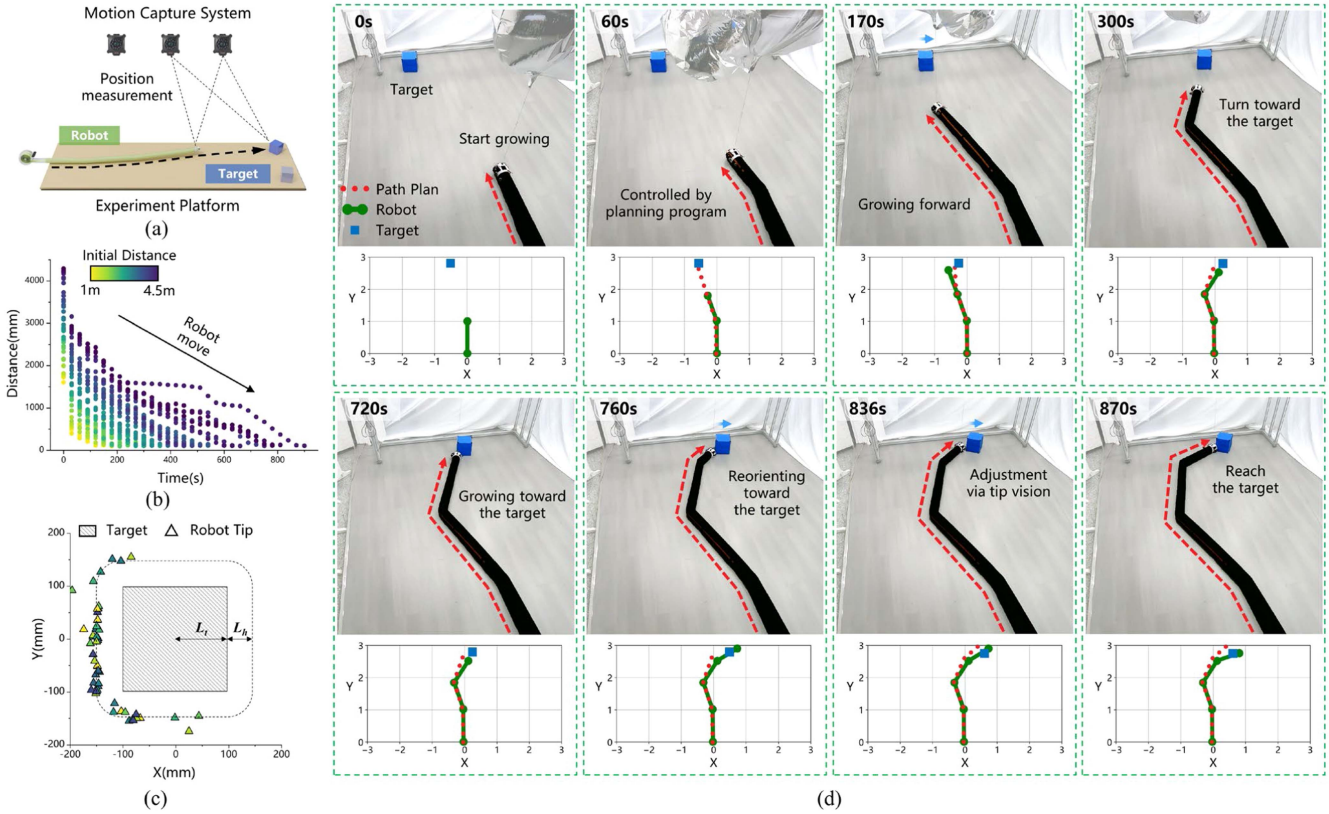

Fig. 4. Experimental Methods and Results. (a) Experimental site. (b) Distance variation results for accuracy experiments. (c) Relative distance results for accuracy experiments. (d) The process of a tracking experiment on a dynamic target.

around the target. In all experiments, the robot tip reached the target. After removing the unreachable area, the mean error of the experiments was about 11.83 mm.

As shown in Fig. 4(c), there are individual experiments with large errors. The reason for this situation is that when the robot is very close to the target, the target undergoes rapid displacement. The robot was able to reach the target, but there was a deviation.

Fig. 4(d) illustrates a dynamic target tracking experiment. At 60s, controlled by the planning program, the robot completed the first turn and determined the forward direction. At 170 s, the robot moves forward, and the target also moves. At 300 s, the target stops moving, while the robot turns toward the target.At 760 s, The target moves again, and the robot turns its tip joint toward the target.At 870 s, the robot successfully reached the target position.

The data in Fig. 4(d) are derived from the sensing feedback. The red dotted line indicates the planning path, the green line indicates the robot pose, and the blue square indicates the target position. At 870 s, it can be seen that there is a small difference between the path and the pose, mainly due to the fact that when the distance is far away, the base vision has a lower accuracy. At this point, the robot relies on the tip vision to make precise adjustments. The experiments proved the effectiveness of the proposed method.

V. Discussion

Learning-based models exhibit inherent inaccuracies, mitigated partially by closed-loop control but still reliant on precise environmental feedback. Our visual localization for non-cooperative targets performs well generally, but fails under darkness, strong light, or physical occlusion, necessitating alternative feedback mechanisms. Sim-to-real transfer remains challenging. To address sensor noise (vision, IMU), only basic filters were employed; future work could reduce errors further via circuit and transmission optimization. Joint fixation minimizes movement errors from external forces (e.g., friction) by localizing actuation to the robot tip. Asynchronous path optimization and real-time data updates mitigate execution latency. Future enhancements include incorporating force sensors to model contact forces within planning, and predicting robot states (e.g., pose, deformation) for global state estimation. Experiments were conducted indoors without physical obstacles or large-scale 3D motion, constrained by the robots' low stiffness; small-scale 3D motion was achieved (see supplementary movies). While the controller incorporates 3D capabilities, adapting to complex unstructured/outdoor environments requires future work on enhanced environmental perception, semantic understanding, robot-environment interaction modeling, optimization of visual hardware (e.g., filter plates), and novel detection methods.

VI. Conclusion

This study proposes a dual-thread control strategy—combining global planning and motion control—to enhance the dynamic target tracking performance of soft growing robots. In terms of global planning, the integration of graph convolutional networks with fully connected neural networks improves the performance of deep reinforcement learning in growing path planning for soft robots. Compared to Inverse kinematics

planning methods, the proposed approach offers improvements in accuracy, computational efficiency and path complexity. In motion control, the control system reduces the motion error of the soft robot with the feedback data streams. Experimental results demonstrate that the robot achieves a tracking accuracy of 11.83 mm within a 5-meter range. Moreover, the robot is capable of tracking and approaching non-cooperative dynamic targets, proving the effectiveness of the proposed control strategy in dynamic control tasks. In summary, this work outlines the principles of the control strategy and the construction of the control system, evaluates the target tracking performance of the soft robot, and provides a theoretical foundation for the practical application of soft growing robots.

REFERENCES

[1] P. Li, Y. Zhang, G. Zhang, D. Zhou, and L. Li, "A bioinspired soft robot combining the growth adaptability of vine plants with a coordinated control system," *Research*, vol. 2021, 2021, Art. no. 9843859.

[2] S. Wang, R. Zhang, D. A. Haggerty, N. D. Naclerio, and E. W. Hawkes, "A dexterous tip-extending robot with variable-length shape-locking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9035–9041.

[3] E. Del Dottore, A. Mondini, N. Rowe, and B. Mazzolai, "A growing soft robot with climbing plant-inspired adaptive behaviors for navigation in unstructured environments," *Sci. Robot.*, vol. 9, no. 86, 2024, Art. no. eadi5908.

[4] L. H. Blumenschein, M. M. Coad, D. A. Haggerty, A. M. Okamura, and E. W. Hawkes, "Design, modeling, control, and application of everting vine robots," *Front. Robot. AI*, vol. 7, Nov. 2020, Art. no. 548266.

[5] C. Girerd, A. Alvarez, E. W. Hawkes, and T. K. Morimoto, "Material scrunching enables working channels in miniaturized vine-inspired robots," *IEEE Trans. Robot.*, vol. 40, pp. 2166–2180, 2024.

[6] F. Campisano et al., "Closed-loop control of soft continuum manipulators under tip follower actuation," *Int. J. Robot. Res.*, vol. 40, no. 6/7, pp. 923–938, Jun. 2021.

[7] D. A. Haggerty et al., "Control of soft robots with inertial dynamics," *Sci. Robot.*, vol. 8, no. 81, Aug. 2023, Art. no. eadd6864.

[8] Z. Hawks, C. Frazelle, K. E. Green, and I. D. Walker, "Motion planning for a continuum robotic mobile lamp: Defining and navigating the configuration space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2559–2566.

[9] L. Xun, G. Zheng, and A. Kruszewski, "Cosserat-rod-based dynamic modeling of soft slender robot interacting with environment," *IEEE Trans. Robot.*, vol. 40, pp. 2811–2830, 2024.

[10] Y. Ma et al., "Magnetic continuum robot for intelligent manipulation in medical applications," *SmartBot*, vol. 1, no. 2, Jun. 2025, Art. no. e12011.

[11] F. Stella, C. D. Santina, and J. Hughes, "Soft robot shape estimation with IMUs leveraging PCC kinematics for drift filtering," *IEEE Robot. Automat. Lett.*, vol. 9, no. 2, pp. 1945–1952, Feb. 2024.

[12] P. Pustina, D. Calzolari, A. Albu-Schäffer, A. D. Luca, and C. D. Santina, "Nonlinear modes as a tool for comparing the mathematical structure of dynamic models of soft robots," in *Proc. IEEE 7th Int. Conf. Soft Robot.*, 2024, pp. 779–785.

[13] X. Shao et al., "Model-based control for soft robots with system uncertainties and input saturation," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7435–7444, Jul. 2024.

[14] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Sci. Robot.*, vol. 2, no. 8, Jul. 2017, Art. no. eaan3028.

[15] S.-G. Jeong et al., "A tip mount for transporting sensors and tools using soft growing robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8781–8788.

[16] M. M. Coad et al., "Vine robots," *IEEE Robot. Automat. Mag.*, vol. 27, no. 3, pp. 120–132, Sep. 2020.

[17] N. G. Kim, D. Seo, S. Park, and J.-H. Ryu, "Self-retractable soft growing robots for reliable and fast retraction while preserving their inherent advantages," *IEEE Robot. Automat. Lett.*, vol. 9, no. 2, pp. 1082–1089, Feb. 2024.

[18] T. Takahashi et al., "Eversion robotic mechanism with hydraulic skeletonto realize steering function," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5413–5420, Jul. 2021.

[19] D. A. Haggerty, N. D. Naclerio, and E. W. Hawkes, "Hybrid vine robot with internal steering-reeling mechanism enhances system-level capabilities," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5437–5444, Jul. 2021.

[20] M. Selvaggio, L. A. Ramirez, N. D. Naclerio, B. Siciliano, and E. W. Hawkes, "An obstacle-interaction planning method for navigation of actuated vine robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3227–3233.

[21] J. D. Greer, L. H. Blumenschein, R. Alterovitz, E. W. Hawkes, and A. M. Okamura, "Robust navigation of a soft growing robot by exploiting contact with the environment," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1724–1738, Dec. 2020.

[22] C. Watson, R. Obregon, and T. K. Morimoto, "Closed-loop position control for growing robots via online Jacobian corrections," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6820–6827, Oct. 2021.

[23] Y. Satake and H. Ishii, "Path planning method with constant bending angle constraint for soft growing robot using heat welding mechanism," *IEEE Robot. Automat. Lett.*, vol. 8, no. 5, pp. 2836–2843, May 2023.

[24] A. Sadeghi et al., "A plant-inspired robot with soft differential bending capabilities," *Bioinspiration Biomimetics*, vol. 12, no. 1, Dec. 2016, Art. no. 015001.

[25] E. Donato, Y. T. Ansari, C. Laschi, and E. Falotico, "To enabling plant-like movement capabilities in continuum arms," in *Proc. I-RIM Conf.*, 2022, pp. 1–4.

[26] X. Li, Z. Serlin, G. Yang, and C. Belta, "A formal methods approach to interpretable reinforcement learning for robotic planning," *Sci. Robot.*, vol. 4, no. 37, Dec. 2019, Art. no. eaay6276.

[27] I. Masmitja et al., "Dynamic robotic tracking of underwater targets using reinforcement learning," *Sci. Robot.*, vol. 8, no. 80, Jul. 2023, Art. no. eade7811.

[28] P. R. Wurman et al., "Outracing champion Gran Turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[29] Y. Song, A. Romero, M. Muller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Sci. Robot.*, vol. 8, no. 82, Sep. 2023, Art. no. eadg1462.

[30] K. Zhou et al., "A cable-actuated soft manipulator for dexterous grasping based on deep reinforcement learning," *Adv. Intell. Syst.*, vol. 6, no. 10, Oct. 2024, Art. no. 2400112.

[31] M. S. Nazeer, C. Laschi, and E. Falotico, "RL-based adaptive controller for high precision reaching in a soft robot arm," *IEEE Trans. Robot.*, vol. 40, pp. 2498–2512, 2024.

[32] X. Liu, C. D. Onal, and J. Fu, "Reinforcement learning of CPG-regulated locomotion controller for a soft snake robot," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3382–3401, Oct. 2023.

[33] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 124–134, Feb. 2019.

[34] F. Liu, F. Sun, B. Fang, X. Li, S. Sun, and H. Liu, "Hybrid robotic grasping with a soft multimodal gripper and a deep multistage learning scheme," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2379–2399, Jun. 2023.