SCIENCE CHINA
Information Sciences

• MOOP •

November 2025, Vol. 68, Iss. 11, 214201:1–214201:3
https://doi.org/10.1007/s11432-025-4544-x

A novel sim-to-real reinforcement learning algorithm for soft growing robot navigation

Haoran WU[1], Fuchun SUN[2*], Zengxin KANG[1], Lin TANG[1] & Zhongyi CHU[1*]

[1]*School of Instrument and Optoelectronics Engineering, Beihang University, Beijing 100191, China*
[2]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

Received 13 March 2025/Revised 28 May 2025/Accepted 10 July 2025/Published online 8 September 2025

Citation Wu H R, Sun F C, Kang Z X, et al. A novel sim-to-real reinforcement learning algorithm for soft growing robot navigation. Sci China Inf Sci, 2025, 68(11): 214201, https://doi.org/10.1007/s11432-025-4544-x

Navigating confined and restricted environments poses significant challenges for conventional rigid robots. In contrast, soft growing robots [1], which draw inspiration from plant growth mechanisms, offer a biologically inspired solution by elongating through material eversion. This unique growth-based locomotion significantly reduces friction during movement. Furthermore, the robot's inherent compliance enables it to navigate through environmental interaction [2], thereby enhancing its capability to confined spaces.

Effective navigation in constrained environments requires soft growing robots to incorporate steering mechanisms, typically achieved through passive (e.g., pre-bent structures) or active (e.g., pneumatic artificial muscles, PAMs) methods [3]. Existing studies on passive steering primarily focus on modeling the behavior of pre-bent robots or developing motion planning strategies, often leveraging environmental interactions to assist navigation. Although these strategies are mechanically simple and exploit the robot's compliance to maneuver in tight spaces, they are inherently irreversible and provide limited capability for correcting motion errors during operation. Active steering via PAMs, when integrated with model-based controllers such as visual servoing or model predictive control, can achieve high positioning accuracy at the end effector. However, these approaches often struggle to manage obstacle avoidance along the entire length of the robot body. Model-free reinforcement learning methods, such as those employing deep Q-networks (DQN) [4], remain largely conceptual in this field. They encounter substantial challenges during training due to structural parameter uncertainties, inaccurate state estimation, environmental disturbances, and prolonged training durations. Sim-to-real reinforcement learning [5] presents a promising alternative by reducing training costs while enhancing safety, reliability, and real-time performance. This approach has already shown considerable advantages in soft robot control tasks.

To address these issues, this study proposes a hybrid motion planning framework for soft growing robots by integrating passive pre-bending and active PAM-based steering, guided by a sim-to-real reinforcement learning algorithm. The proposed method trains control policies in a Unity-based simulation environment and subsequently transfers them to real-world robotic systems. A dual-layer deep deterministic policy gradient (DDPG) architecture is employed: the first layer learns an environmental interaction policy to determine the optimal pre-bending configuration, while the second layer actively controls the PAMs to compensate for sensor noise and manufacturing deviations, thereby minimizing motion errors.

Integration of soft growing robot prototype. As illustrated in Figure 1(a), the soft growing robot prototype combines pre-bending and PAM actuation to enable steering. The system comprises a drive-storage unit, a polyethylene film-based body, a steering module, and a tip-guiding mechanism. Pre-bending is implemented by folding the polyethylene film into a Z-shape and anchoring one end to introduce a controlled length differential. A PAM is constructed by reinforcing a section of the film with nylon thread; when inflated, it contracts laterally, enabling directional bending. An HWP906P gyroscope mounted at the tip continuously provides heading angle feedback, while a base-mounted encoder tracks the robot's extension length. The Kalman filter is used to fuse multi-sensor data, enabling accurate estimation of the robot's motion trajectory.

$$\begin{aligned}
&\hat{\mathbf{x}}_{i+1/i} = \mathbf{A}_i\hat{\mathbf{x}}_i + \mathbf{B}_i\mathbf{u}_i,\\
&\mathbf{K}_i = \mathbf{P}_{i+1/i}{\mathbf{H}_i}^{\mathrm{T}}\left(\mathbf{H}_i\mathbf{P}_{i+1/i}{\mathbf{H}_i}^{\mathrm{T}} + \mathbf{R}_i\right)^{-1},\\
&\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_{i+1/i} + \mathbf{K}_i\left(\mathbf{Z}_i - \mathbf{H}_i\hat{\mathbf{x}}_{i+1/i}\right),
\end{aligned} \tag{1}$$

where $\mathbf{K}$ is the Kalman filter gain, $\mathbf{H}$ is the measurement matrix, $\mathbf{P}$ is the covariance matrix of state variables, and $\mathbf{R}$ is the covariance of measurement noise. The matrices in (1) are defined as follows:

$$\begin{aligned}
\hat{\mathbf{x}} &= \begin{bmatrix} p_x & p_y & \theta_{\mathrm{end}} \end{bmatrix},\\
\mathbf{A} &= \begin{bmatrix} 1 & 0 & v\cdot\Delta t\cdot\cos\left(\hat{x}_3\right)\\ 0 & 1 & v\cdot\Delta t\cdot\sin\left(\hat{x}_3\right)\\ 0 & 0 & 1 \end{bmatrix},\\
\mathbf{B} &= \begin{bmatrix} \Delta t\cdot\cos\left(\hat{x}_3\right) & \Delta t\cdot\sin\left(\hat{x}_3\right) & 0 \end{bmatrix}^{\mathrm{T}},
\end{aligned}$$

* Corresponding author (email: fcsun@tsinghua.edu.cn, chuzy@buaa.edu.cn)

© Science China Press 2025

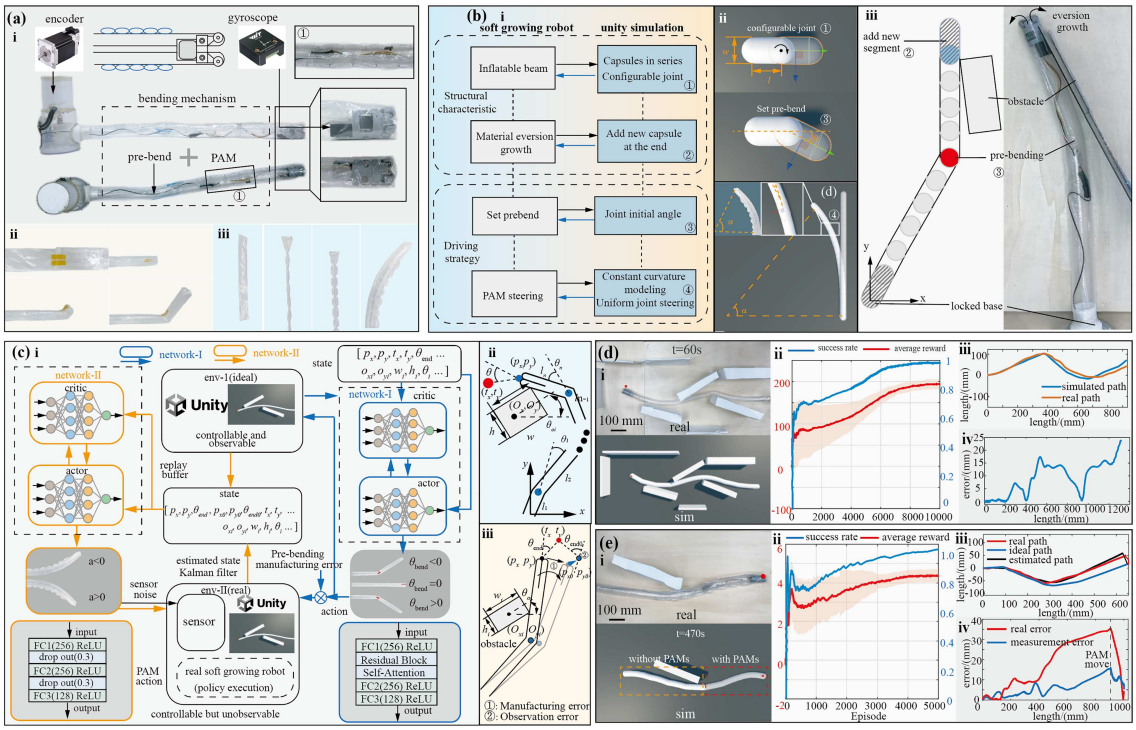

Figure 1 (Color online) (a) Prototype of the soft growing robot, including its bending and PAM-based actuation structure; (b) sim-to-real reinforcement learning framework for soft growing robot control; (c) dual-layer DDPG reinforcement learning framework: network architecture and robot state vector; (d) validation of the pre-bending action selection network, training performance of the policy, and analysis of the resulting robot trajectory and tracking error; (e) validation of the error compensation network for real-time motion correction.

where, p_x and p_y represent the planar position of the end, v is the linear velocity, Δt is the discrete time step, and θ_{end} denotes the orientation angle of the end.

Sim-to-real framework of soft growing robot. The sim-to-real transfer framework for the soft growing robot is illustrated in Figure 1(b). The structural characteristics and actuation strategies of the robot are analyzed and simplified. To balance simulation fidelity and real-time performance, the soft growing robot—modeled as a flexible inflatable beam—is approximated using a series of connected capsule segments in simulation. Growth is modeled by progressively attaching new capsules at the tip to emulate eversion-driven extension.

Adjacent segments are connected via configurable joints with virtual springs and dampers to emulate the elastic behavior of soft beams. Since the robot operates within a 2D plane, rotations around the X and Z axes are constrained, allowing only Y-axis rotation. Pre-bending is realized by setting initial joint angles, whereas PAM-based steering is approximated using a constant-curvature model that applies uniform rotations across all joints. To enhance transferability, domain randomization is introduced in the Unity environment to account for variations in dynamics and perception during sim-to-real deployment.

Dual-layer DDPG network. As illustrated in Figure 1(c), a dual-layer DDPG network is employed for motion planning of the soft growing robot. The first-level network learns interaction strategies with the environment and determines the pre-bending configuration to guide the robot toward the target. Based on the output of the first-level network, the second-level network further accounts for manufacturing inaccuracies and sensor noise, and controls the PAMs to compensate for resulting trajectory errors. The state vector used by the pre-bending action selection network is defined as follows:

$$x = [\underbrace{p_x \quad p_y \quad \theta_{\text{end}}}_{x_{\text{robot}}} \quad \underbrace{t_x \quad t_y}_{x_{\text{target}}} \quad \underbrace{O_{xi} \quad O_{yi} \quad w_i \quad h_i \quad \theta_i}_{x_{\text{obstacle}}}], \tag{2}$$

where p_x and p_y denote the tip coordinates of the soft growing robot. θ_{end} is the angle between the robots growth direction and the line to the target point (t_x, t_y). All obstacles are rectangular, with centroid coordinates (O_x, O_y), dimensions w_i and h_i, and orientation angle θ_i.

Furthermore, the action of the soft growing robot determines whether to set the pre-bending or the angle of pre-bending.

$$a_t = \theta_{\text{bend}}, \begin{cases} \theta_{\text{bend}} < 0, \text{ turn left}, \\ \theta_{\text{bend}} = 0, \text{ straight}, \\ \theta_{\text{bend}} > 0, \text{ turn right}. \end{cases} \tag{3}$$

The reward comprises two components: the target-approaching reward r_{target} and the turning behavior reward r_{action}, weighted by w_{target} and w_{action}, respectively.

$$r = w_{\text{action}} \cdot r_{\text{action}} + w_{\text{target}} \cdot r_{\text{target}}, \tag{4}$$

$$r_{\text{target}} = \mathrm{e}^{-10d}, \quad d = \sqrt{(t_x - p_x)^2 + (t_y - p_y)^2},$$
$$r_{\text{action}} = \begin{cases} \dfrac{20}{|\theta_t - \theta_{\text{bend}}| + 1}, & \text{if } |\theta_t - \theta_{\text{bend}}| < \theta_t, \\ -|\theta_{\text{bend}}|, & \text{if } |\theta_t - \theta_{\text{bend}}| \geqslant \theta_t, \end{cases} \tag{5}$$

where d denotes the Euclidean distance between the robot tip and the target point, θ_t is the angle between the growth direction and the target direction, and θ_{bend} is constrained within $(-20^\circ, 20^\circ)$ to prevent excessive turning.

Although the first-level network is trained in an ideal environment, real-world deployment often results in trajectory deviations caused by pre-bending imperfections and sensor noise. To address these issues, a second-level perception and error correction network is introduced and trained across both ideal and quasi-realistic environments. The ideal environment assumes full observability and no noise, whereas the quasi-realistic environment incorporates uncertainties, sensor noise, and partial observability. The input state vector for the second-level network is defined as

$$x = [\underbrace{p_x\ p_y\ \theta_{\text{end}}}_{x_{\text{ideal}}}\ \underbrace{p_{x0}\ p_{y0}\ \theta_{\text{end0}}}_{x_{\text{real}}}\ \underbrace{t_x\ t_y}_{x_{\text{target}}}\ \underbrace{O_{xi}\ O_{yi}\ w_i\ h_i\ \theta_i}_{x_{\text{obs}}}], \quad (6)$$

where p_x, p_y, and θ_{end} denote the ideal coordinates and orientation of the soft growing robot's tip. p_{x0}, p_{y0}, and θ_{end0} represent the actual coordinates and orientation of the soft growing robot's tip. The action of the second-layer network is the movement of PAMs.

$$a_{t2} = \theta_{\text{spam}},\ \begin{cases} \theta_{\text{spam}} > 0,\ \text{turn right}, \\ \theta_{\text{spam}} < 0,\ \text{turn left}. \end{cases} \quad (7)$$

The reinforcement learning reward is defined by the gap between the actual (d_{t0}) and expected (d_t) end-effector distances resulting from PAM actuation.

$$\begin{cases} r = |d_t - d_{t0}|, d_{t0} < d_t, d_{t0} = \sqrt{(t_x - p_{x0})^2 + (t_y - p_{y0})^2}, \\ r = -|d_t - d_{t0}|, d_{t0} > d_t, d_t = \sqrt{(t_x - p_x)^2 + (t_y - p_y)^2}. \end{cases} \quad (8)$$

Both Network I and Network II comprise three fully connected layers activated by ReLU functions. Network I incorporates residual blocks and a self-attention mechanism to enhance feature extraction and ensure training stability. Network II incorporates two dropout layers to mitigate overfitting during training. The DDPG framework utilizes prioritized experience replay to enhance sample efficiency and employs decaying Ornstein-Uhlenbeck noise to encourage effective exploration during training.

Experimental results. The proposed dual-layer DDPG framework was trained in simulation and validated through real-world experiments, with results compared to simulation. As shown in Figures 1(d) and (e), the learning curves of the action selection and error correction networks initially exhibited fluctuations in average reward and success rate, followed by convergence to stable performance. The agent eventually learned an effective policy, achieving a success rate over 95%. Under identical experimental conditions, DDPG outperformed soft actor-critic (SAC) and proximal policy optimization (PPO) in terms of convergence speed, control precision, and task success rate, demonstrating its superiority in high-precision continuous control tasks with high-dimensional state spaces and low-dimensional action spaces.

To validate the action selection network, an iterative correction mechanism was employed to compensate for fabrication-induced pre-bending errors. The robot executed motion commands from Network I while interacting with the environment, enabling effective navigation in confined spaces. The observed end-effector trajectory and overall robot shape closely matched Unity-based simulation results. At a total extension of 1300 mm, the terminal tracking error remained below 25 mm, with a standard deviation of 5.13 mm.

For the error correction network, pre-bending paths from the trained action network were used as input. The actual motion was reconstructed via encoder-gyroscope fusion and compared to the ideal trajectory. The resulting deviation was processed by the correction network, which actuated the PAMs to reduce tracking errors. Experiments showed that cumulative deviations were reduced from under 40 mm (std: 6.03 mm) to below 8 mm (std: 1.47 mm) through active compensation.

Conclusion and limitations. This work proposes a sim-to-real reinforcement learning framework for motion planning in soft growing robots by integrating pre-bending structures and PAM actuation. A dual-layer DDPG network is designed: the first layer learns environmental interaction strategies and determines pre-bending, while the second layer compensates trajectory deviations caused by sensor noise and fabrication errors via PAM control. Sim-to-real transfer is realized via a Unity-based simulation environment. Experimental results validate the effectiveness of the proposed approach, demonstrating a success rate exceeding 95% and a significant reduction in tracking errors. Despite its effectiveness, the current approach is limited to static planar environments, and the integration of PAMs increases internal pressure, slightly compromising flexibility. Future work will focus on co-growing PAMs with the robot body to maintain compliance, expand to 3D and dynamic environments, and enhance autonomous perception and decision-making capabilities, ultimately moving toward fully intelligent and adaptable soft growing robots.

Acknowledgements This work was supported by Guoqiang Institute, Tsinghua University (Grant No. 2020GQG0006) and National Natural Science Foundation of China (Grant No. 52375006).

Supporting information Videos and other supplemental documents. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Qin K, Tang W, Zong H, et al. *Parthenocissus*-inspired soft climbing robots. Sci Adv, 2025, 11: eadt9284
2 Greer J D, Blumenschein L H, Alterovitz R, et al. Robust navigation of a soft growing robot by exploiting contact with the environment. Int J Robot Res, 2020, 39: 1724–1738
3 Kübler A M, du Pasquier C, Low A, et al. A comparison of pneumatic actuators for soft growing vine robots. Soft Robot, 2024, 11: 857–868
4 El-Hussieny H, Hameed I A. Obstacle-aware navigation of soft growing robots via deep reinforcement learning. IEEE Access, 2020, 12: 38192–38201
5 Shakya A K, Pillai G, Chakrabarty S. Reinforcement learning algorithms: a brief survey. Expert Syst Appl, 2023, 231: 120495