Open-Loop Motion Control of a Hydraulic Soft Robotic Arm Using Deep Reinforcement Learning

Yunce Zhang[1], Tao Wang[1,3,4,5(✉)], Ning Tan[6], and Shiqiang Zhu[1,2]

[1] Ocean College, Zhejiang University, Zhoushan 316000, People's Republic of China
twang001@zju.edu.cn
[2] Zhejiang Lab, Hangzhou 310058, People's Republic of China
[3] State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou, People's Republic of China
[4] Engineering Research Center of Oceanic Sensing Technology and Equipment, Ministry of Education, Zhoushan 316000, People's Republic of China
[5] Key Laboratory of Ocean Observation-Imaging Testbed of Zhejiang Province, Zhoushan 316000, People's Republic of China
[6] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, People's Republic of China

Abstract. Soft robotic arms are of great interests in recent years, but it is challenging to perform effective control due to their strongly nonlinear characteristics. This work develops a model-free open-loop control method for a hydraulic soft robotic arm in spatial motion. A control policy based on reinforcement learning technique is proposed by using Deep Deterministic Policy Gradient. The kinematic model of the soft robotic arm is employed instead of physical prototype to train the control policy. A complete training framework is established through the Reinforcement Learning Toolbox and Deep Learning Toolbox in Matlab software. To make the control policy fast converge and avoid falling into local optimum, the reward is shaped by combining the position error and the action together. A series of simulations are implemented and the results verify the effectiveness of the control policy. It is also shown that the proposed control policy can achieve both of good stability and tracking performance simultaneously.

Keywords: Motion control · Reinforcement Learning · Soft robot · Hydraulic

1 Introduction

In recent years, soft robots have received extensive attention from researchers due to inherent compliance, environmental adaptability, lower inertia and safe human-machine interaction [1]. Inspired by nature and composed of low-modulus

X.-J. Liu et al. (Eds.): ICIRA 2021, LNAI 13013, pp. 302–312, 2021.
https://doi.org/10.1007/978-3-030-89095-7_30

materials, soft robots can deform continuously like flexible structures in biological systems [2]. Features aforementioned make soft robots have obvious advantages and promising prospect in the application of flexible grasping, surgery, rehabilitation, and bionic locomotion [3,4].

Conventional rigid robotic arms have been extensively employed in tasks such as grasping, assembling and handling. But the limited DOF and the possibility of harm to humans restrict them from working in an unstructured environment or human-machine interaction scenes [5]. Compared with the rigid robotic arms, soft robotic arms have the advantages of lightweight, flexibility and safety, so they could be used in an unstructured environment or human-machine interaction scene and perform well. With the development of the soft robotic arms, an amount of actuation methods have been applied, like hydraulic actuation [6], shape memory alloy actuation [7], pneumatic artificial muscles actuation [8], and cable-driven actuation [9]. In the aforementioned actuation technique, hydraulic actuation is widely applied and has got lots of studies due to their conformability [10]. However, modeling and controlling of the hydraulic soft robotic arms are challenging and difficult because of the strong nonlinearity between hydraulic pressure and elastic deformation [11].

Researchers have paid much effort on motion control of soft robotic arms by using both model based and model-free methods [12]. The premise of using model-based control approaches is to establish a mathematical model of the controlled object, an accurate model or a reasonably simplified model is the guarantee of the good control performance. Xie et al. develop the kinematic model of the soft robotic arm by using the piecewise constant-curvature (PCC) assumption to predict the position of its tip position [5]. Ohta et al. develop the kinematic model of the robotic arm by using DH parameters and carry out simulation and experimental results for closed-loop position control based on the kinematic model [13]. Yang et al. build a direct kinematic model from the sensor data to the deformation and an inverse kinematic model used to calculate the actuation of SMA coils base on given planned deformation [14]. In order to achieve more precise control performance, some studies pay attention to the dynamic model and achieved great progress. Renda et al. develop a dynamic model of a soft continuum robotic arm by using a rigorous geometrically exact method [15]. Tutcu et al. combine a kinematic model with a quasi-static equilibrium solution for more accurate modeling of the end effector of a soft continuum robot [16]. In addition to these, some novel methods are derived, like Chen et al. using force balance of the ending plate to build the model [17]. Tang et al. propose a model based online learning and adaptive control algorithm for the wearable soft robot [18].

For multi-segments hydraulic soft robotic arms, model-based control is difficult to achieve real-time and high accuracy without additional restrictions due to the complexity and imprecision of the mathematical model, and model-free methods offer the possibility of good control performance. Li et al. use adaptive Kalman filter to achieve path tracking for a continuum robot [19]. Melingui et al. develop two controllers based on a distal supervised learning scheme and

an adaptive neural to control CBHA's kinematics and dynamics [20]. With the development of machine learning techniques, reinforcement learning has been widely used in robotic control [21], model-free reinforcement learning has obvious advantages in soft robotic arms control tasks. Ma et al. propose a reinforcement learning method based on the Deep Deterministic Policy Gradient (DDPG) algorithm to solve position control problem [22]. Shahid et al. develop a control policy parameterized by a neural network and learned using modern Proximal Policy Optimization (PPO) algorithm [23]. Satheeshbabu et al. present an open loop position control policy based on deep reinforcement learning and use Deep-Q Learning with experience replay [24]. Although some efforts have been paid on using reinforcement learning to control soft robotic arms, most of the current studies focus on planar motions, or spatial motions with less control inputs in limited environments or in small action space.

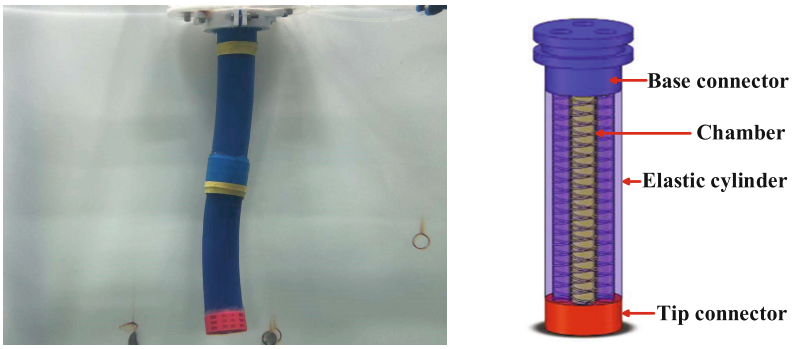
In this paper, we investigate the motion control of a double-segment hydraulic soft robotic arm, which has six control inputs and a large state-action space. To achieve open-loop motion control, a model-free control policy based on deep reinforcement learning (RL) is proposed by using Deep Deterministic Policy Gradient (DDPG) algorithm. The kinematic model [5] of the soft robotic arm is employed instead of physical prototype to train the control policy. A complete training framework is established through the Reinforcement Learning Toolbox and Deep Learning Toolbox in Matlab software. To make the control policy fast converge and avoid falling into local optimum, the reward is shaped by combining the position and the action together. A control policy with excellent performance was obtained via parameter optimization and reward function optimization. A series of simulations are implemented to evaluate the control policy, the effectiveness and good tracking performance of the control policy are verified in simulations.

The remainder of this paper is structured as follows. Section 2 describes the architecture of the system. Section 3 introduces the training framework and configurations, and show the results of simulations. Section 4 presents conclusions and future works.

2 System Description

2.1 Hydraulic Soft Robotic Arm

The studied hydraulic soft robotic arm is as shown in Fig. 1. It is totally made of soft materials and composed of an elastic cylinder, two connectors and three chambers with double-helical fiber reinforcement for each segment. Table 1 gives the key parameters of the arms. Each segment of the soft robotic arms is independent and can be quickly assembled and disassembled through the connector. In the current design, the arms can be extended to three segments, but considering the overall length of the arms and the existing experimental conditions, a two-segments arm is employed to carry out the work.



(a) (b)

Fig. 1. (a) Two-segments hydraulic soft robotic arm prototype. (b) Schematic of the one-segment hydraulic soft robotic arm.

Table 1. Key parameters of the hydraulic soft robotic arm.

Parameters	Value
Length of elastic cylinder	140 mm
External diameter of elastic cylinder	50 mm
Height of base connector	40 mm
Height of tip connector	20 mm
External length of chamber	170 mm
Internal length of chamber	140 mm
External diameter of chamber	15 mm
Internal diameter of chamber	10 mm
The distance between the center of the chamber and the elastic cylinder	14 mm
Maximum pressure of chamber	300 kpa

2.2 Markov Decision Process Modeling

Markov Decision Process (MDP) formally defines the reinforcement learning problem, using reinforcement learning on robots requires it to be abstracted and represented as an MDP. A MDP is based on the integration of each interactive object, composed of agent and environment, and its elements include state, action and reward. The motion control task is modeled into a continuous-state, continuous-action MDP. Assuming the simplest form of representation, the RL-based motion control task of the hydraulic soft robotic arm is abstracted as follows:

State(s): State is the condition of the agent described by the environment. In the soft robotic arm motion control task, the state consists of two parts, which are the current state of the soft robotic arm and the action at the previous

moment. More specifically, the current state of the soft robotic arm is error between the soft robotic arm and the target position in the direction of each coordinate axis.

Action(s): Action is the collection of actions which the agent could take, called action space. Agents based on DDPG can output continuous actions. Considering that it is difficult to establish an accuracy dynamic model from the pressure of chambers to the position of the tip of the soft robotic arm, the forward kinematic model from the length of chambers to the position of the tip of the soft robotic arm is used to train the agent in the simulation. Therefor, actions that the soft robotic arm could take is the increment of each chamber length, the upper limit and the lower limit of each increment are +1 mm and –1 mm respectively. According to the maximum pressure of the chambers, the upper bound of the length of the chambers is 200 mm. This setting can make the soft robotic arm reach the target position smoothly and quickly.

Reward(s): The reward is a quantitative indicator used to judge each action of the agent and guide the robot to complete tasks. In our task, in order to make the robotic arm move to the target position quickly and stably, the Euclidean distance between arm's tip position and target position and the action at the previous moment are used as the basis for formulating rewards. Actions that move the manipulator away from the target and are not conducive to the stability of the robot will be subject to greater penalties. On the contrary, actions that bring the robot closer to the target and approach stability will be rewarded. This can speed up the training process of the policy and contribute to the steady-state performance of the soft robotic arm. The reward structure is shown as follows:

$$r = \begin{cases} -0.001 err_d - 0.05 \sum |a_i| - 0.0003(|err_x| + |err_y| + |err_z|), err_d > \varepsilon \\ 500 - 0.05 \sum |a_i|, err_d \leq \varepsilon \end{cases} \quad (1)$$

where the $\varepsilon = 5\,\text{mm}$ is the target threshold, the err_d is the Euclidean distance between arm's tip position and target position, the err_x, err_y and err_z is the distance between arm's tip position and target position between the tip position of the arm and the target position on each coordinate axis. When the agent reaches the target within ε, the training episode is done. The reward is to penalize actions that are not conducive to completing the task and make the soft robotic arm reach the target position in the shortest path.

2.3 Deep Deterministic Policy Gradient Framework

DDPG is a model-free reinforcement learning method that can be extended to continuous action control [25]. We use an actor-critic framework on DDPG to make the policy stable. Convolutional neural network is used to approximate the optimal policy function μ and Q function, namely the policy network and the Q network, and the deep learning method is used to train the above neural

network. DDPG needs to learn Q network while learning policy network. The implementation and training method of the Q function refers to the DQN [26]. The value iteration update of the Q function follows the Bellman equation and is defined as:

$$Q_t^{\mu}(s_t, a_t) = Q_t^{\mu}(s_t, a_t) + \alpha(r_t + \gamma \max_a Q^{\mu}(s_{t+1}, a) - Q_t^{\mu}(s_t, a_t)) \tag{2}$$

where the s_t is the state at time step t, a_t is the action at time t, s_{t+1} is the state after taking action a_t, r_t is the reward value about a_t, α is the learning rate, and γ is the discount rate.

In the continuous action spaces training process, exploration is important to find potential better policies, so we add random noise for the action to transit the action from a deterministic process to a random process, and then sample the action from this random process and send it to the environment for execution. The above policy is called the behavior policy, which is represented by β. Ornstein-Uhlenbeck process is used to generate random noise as shown is Eq. 3.

$$\partial n = \Phi(\eta - n) + \sigma W \tag{3}$$

where the η is the mean, the Φ is the decay rate, the σ is the variance, the W is the Wiener process. The process of training policy network is to find the optimal solution of policy network parameters, and the stochastic gradient ascent method is used to train the network. The Eq. 4 is used to judge the performance of a policy, and the optimal policy is defined by Eq. 5. The whole algorithm framework is as shown in Algorithm 1.

$$\begin{aligned} J_{\beta}(\mu) &= \int_S \rho^{\beta}(s) Q^{\mu}(s, a) \, ds \\ &= E_{s \sim \rho^{\beta}}[Q^{\mu}(s, a)] \end{aligned} \tag{4}$$

where the s is the state, the ρ^{β} is the distribution function of the state.

$$\mu = \arg\max_{\mu} J(\mu) \tag{5}$$

3 Training and Simulations

3.1 Training Setup

A high-performance computer consists of a 10900X CPU and an RTX2080Ti GPU is used to train the control policy and validate the effectiveness of policy in simulations. The policy training framework is deployed in Matlab by using Reinforcement Learning Toolbox and Deep Learning Toolbox, and trained by using a critic network and an actor network. The critic network has two hidden layers with 400 neurons and the learning rate is $1e^{-3}$. The actor network has four hidden layers with 400 neurons and the learning rate is $1e^{-4}$. The outputs of

Algorithm 1. DDPG framework

Input: max training episodes M, max steps of each episode T, discount factor γ, target smooth factor τ, replay buffer $\boldsymbol{R}$
Output: control policy μ
Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$
Initialize target network Q' and μ' with weights $Q' \leftarrow Q$, $\mu' \leftarrow \mu$
Initialize Replay Buffer: $\boldsymbol{R}$

for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}$ based on current policy and exploration noise
    Excute action a_t and observe reward r_t and new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer $\boldsymbol{R}$
    Sample a random minibatch of N transition (s_i, a_i, r_i, s_{i+1}) from $\boldsymbol{R}$
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic network by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor network using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

the actor network are bounded between −1 and +1 with a tanhLayer followed a ScalingLayer. Other training parameters are set as follows, the maximum number of training episodes is set as 100000, the maximum number of steps per episode is set as 200, the discount factor is set as 0.99, the minibatch size is set as 256, the target smooth factor is set as 0.001, and the experience reply buffer is set as $1e^8$. A simulation model is built based on the kinematic model of the soft robotic arm [5] and connected with the agent in Simulink.

3.2 Position Control Results

The position control simulations and trajectory tracking simulations are implemented to validate the effectiveness and dynamic performance of the presented control policy. As for position control, we select a series of points in the workspace of the soft robotic arm as target points to test the steady-state performance of the control policy. The results are as shown in Fig. 2, the simulation step size is set to 0.01 s. The control policy transfers the soft robotic arm from the initial

state to the target state with few steps, and the steady-state error is controlled within 3 mm. Simulation results revealed the effectiveness and stability of the control policy.

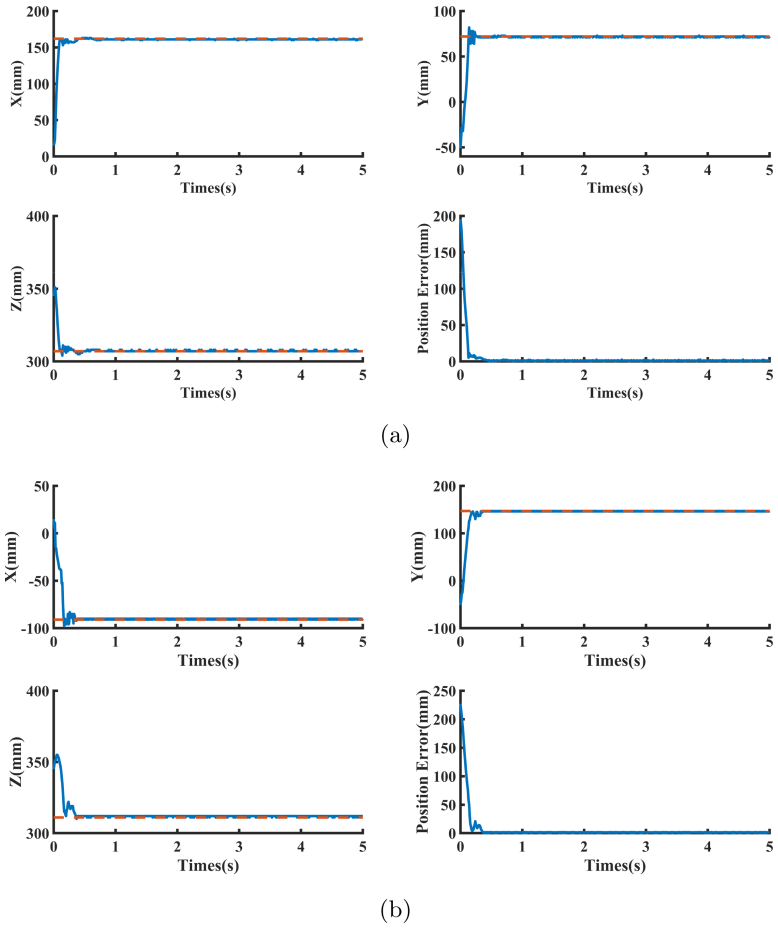
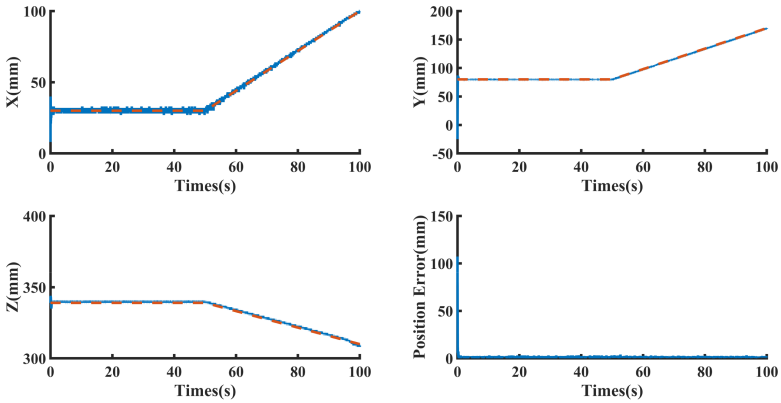


Fig. 2. (a) Position control with the target point (162, 72, 307). (b) Position control with the target point (–91, 147, 311).

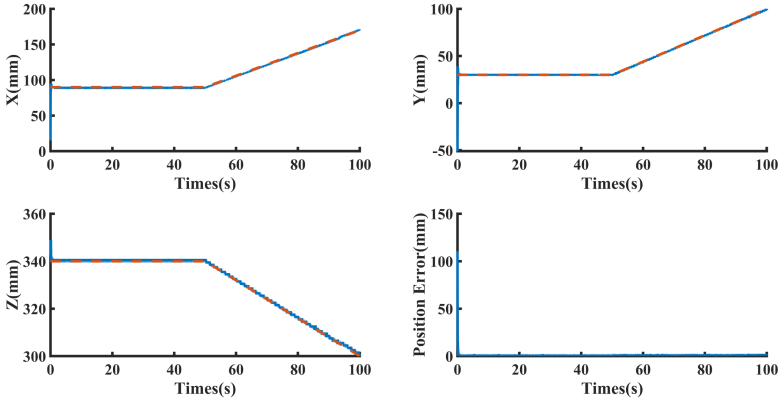
3.3 Trajectory Tracking Results

The dynamic response of the control policy is the key factor that determines the dynamic performance of the system. As shown in Fig. 3, We select some trajectories according to the workspace of the soft robotic arm to verify the dynamic performance of the control policy. The soft robotic arm begins moving along the target trajectory from 50 s, the policy control soft robotic arm to

quickly follow with the target trajectory, and the dynamic error is controlled within 5 mm during the whole movement. Simulation results proved the rapid dynamic response of the control policy, and the soft robotic arm based on this control policy has good tracking performance.



(a)



(b)

Fig. 3. (a) Trajectory tracking control with the target trajectory from the point (30, 80, 339) to the point (100, 170, 310). (b) Trajectory tracking control with the target trajectory from the point (90, 30, 340) to the point (170, 100, 300).

4 Conclusion and Future Work

Focusing on the motion control of hydraulic soft robotic arm, this paper implements the kinematic model of the soft robotic arm in simulations and develops a model-free control policy based on deep reinforcement learning. The Reinforcement Learning Toolbox and Deep Learning Toolbox are used to deploy the policy

training framework, and the Deep Deterministic Policy Gradient (DDPG) algorithm is used to train the policy. The simulations experiments show the effectiveness, robustness and good dynamic performance in motion control of the proposed control policy. After experimental verification, this article is a good attempt of applying reinforcement learning to the motion control of a hydraulic soft robotic arm with highly nonlinear characteristics.

In future work, further improvement and optimization of the proposed control policy will be studied, and the policy will be deployed into the physical prototype control system.

References

1. Lee, C., et al.: Soft robot review. Int. J. Control Autom. Syst. **15**(1), 3–15 (2016). https://doi.org/10.1007/s12555-016-0462-3
2. Wang, T., Zhang, Y., Chen, Z., Zhu, S.: Parameter identification and model-based nonlinear robust control of fluidic soft bending actuators. IEEE/ASME Trans. Mech. **24**(3), 1346–1355 (2019). https://doi.org/10.1109/TMECH.2019.2909099
3. Wang, Y., et al.: A biorobotic adhesive disc for underwater hitchhiking inspired by the remora suckerfish. Sci. Robot. **2**(10) (September 2017). https://doi.org/10.1126/scirobotics.aan8072
4. Choi, C., Schwarting, W., DelPreto, J., Rus, D.: Learning object grasping for soft robot hands. IEEE Robot. Autom. Lett. **3**(3), 2370–2377 (2018). https://doi.org/10.1109/LRA.2018.2810544
5. Xie, Q., Wang, T., Yao, S., Zhu, Z., Tan, N., Zhu, S.: Design and modeling of a hydraulic soft actuator with three degrees of freedom. Smart Mater. Struct. **29**(12) (2020). https://doi.org/10.1088/1361-665X/abc26e
6. Gong, Z., et al.: A soft manipulator for efficient delicate grasping in shallow water: modeling, control, and real-world experiments. Int. J. Robot. Res. 027836492091720 (July 2020). https://doi.org/10.1177/0278364920917203
7. Laschi, C., Cianchetti, M., Mazzolai, B., Margheri, L., Follador, M., Dario, P.: Soft robot arm inspired by the octopus. Adv. Robot. **26**(7), 709–727 (2012). https://doi.org/10.1163/156855312X626343
8. Grissom, M.D., et al.: Design and experimental testing of the OctArm soft robot manipulator. In: Unmanned Systems Technology VIII, vol. 6230, p. 62301F. International Society for Optics and Photonics (May 2006). https://doi.org/10.1117/12.665321
9. Xu, F., Wang, H., Au, K.W.S., Chen, W., Miao, Y.: Underwater dynamic modeling for a cable-driven soft robot arm. IEEE/ASME Trans. Mechatron. **23**(6), 2726–2738 (2018). https://doi.org/10.1109/TMECH.2018.2872972
10. Shengda, Y., Wang, T., Zhu, S.: Research on energy consumption of fiber-reinforced fluidic soft actuators. Smart Mater. Struct. **30**(2) (2021). https://doi.org/10.1088/1361-665X/abd7e6
11. Wang, T.: A computationally efficient dynamical model of fluidic soft actuators and its experimental verification p. 8 (2019)
12. George Thuruthel, T., Ansari, Y., Falotico, E., Laschi, C.: Control strategies for soft robotic manipulators: a survey. Soft Robot. **5**(2), 149–163 (2018). https://doi.org/10.1089/soro.2017.0007

13. Ohta, P., et al.: Design of a lightweight soft robotic arm using pneumatic artificial muscles and inflatable sleeves. Soft Robot. **5**(2), 204–215 (2018). https://doi.org/10.1089/soro.2017.0044
14. Yang, H., Xu, M., Li, W., Zhang, S.: Design and implementation of a soft robotic arm driven by SMA coils. IEEE Trans. Industr. Electron. **66**(8), 6108–6116 (2019). https://doi.org/10.1109/TIE.2018.2872005
15. Renda, F., Giorelli, M., Calisti, M., Cianchetti, M., Laschi, C.: Dynamic model of a multibending soft robot arm driven by cables. IEEE Trans. Robot. **30**(5), 1109–1122 (2014). https://doi.org/10.1109/TRO.2014.2325992
16. Tutcu, C., Baydere, B.A., Talas, S.K., Samur, E.: Quasi-static modeling of a novel growing soft-continuum robot. Int. J. Robot. Res. **40**(1), 86–98 (2021). https://doi.org/10.1177/0278364919893438
17. Chen, X., Guo, Y., Duanmu, D., Zhou, J., Zhang, W., Wang, Z.: Design and modeling of an extensible soft robotic arm. IEEE Robot. Autom. Lett. **4**(4), 4208–4215 (2019). https://doi.org/10.1109/LRA.2019.2929994
18. Tang, Z.Q., Heung, H.L., Tong, K.Y., Li, Z.: Model-based online learning and adaptive control for a "human-wearable soft robot" integrated system. Int. J. Robot. Res. **40**(1), 256–276 (2021). https://doi.org/10.1177/0278364919873379
19. Li, M., Kang, R., Branson, D.T., Dai, J.S.: Model-free control for continuum robots based on an adaptive Kalman filter **23**(1), 12 (2018)
20. Melingui, A., Lakhal, O., Daachi, B., Mbede, J.B., Merzouki, R.: Adaptive neural network control of a compact bionic handling arm. IEEE/ASME Trans. Mech. **20**(6), 2862–2875 (2015). https://doi.org/10.1109/TMECH.2015.2396114
21. Wang, H., et al.: Deep reinforcement learning: a survey. Front. Inf. Technol. Electron. Eng. (1), 1–19 (2020). https://doi.org/10.1631/FITEE.1900533
22. Ma, R., et al.: Position control of an underwater biomimetic vehicle-manipulator system via reinforcement learning. In: 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), pp. 573–578 (November 2020). https://doi.org/10.1109/DDCLS49620.2020.9275206
23. Shahid, A.A., Roveda, L., Piga, D., Braghin, F.: Learning continuous control actions for robotic grasping with reinforcement learning. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4066–4072 (October 2020). https://doi.org/10.1109/SMC42975.2020.9282951
24. Satheeshbabu, S., Uppalapati, N.K., Chowdhary, G., Krishnan, G.: Open loop position control of soft continuum arm using deep reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 5133–5139 (May 2019). https://doi.org/10.1109/ICRA.2019.8793653
25. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs, stat] (September 2015)
26. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015). https://doi.org/10.1038/nature14236