International Journal of Intelligent Robotics and Applications
https://doi.org/10.1007/s41315-021-00194-z

REGULAR PAPER

Prediction model-based learning adaptive control for underwater grasping of a soft manipulator

Hui Yang[1] · Jiaqi Liu[1] · Xi Fang[1] · Xingyu Chen[2] · Zheyuan Gong[1] · Shiqiang Wang[1] · Shihan Kong[2] · Junzhi Yu[2,3] · Li Wen[1]

Received: 27 February 2021 / Accepted: 23 June 2021

Abstract
Soft robotic manipulators have promising features for performing non-destructive underwater tasks. Nevertheless, soft robotic systems are sensitive to the inherent nonlinearity of soft materials, the underwater flow current disturbance, payload, etc. In this paper, we propose a prediction model-based guided reinforcement learning adaptive controller (GRLMAC) for a soft manipulator to perform spatial underwater grasping tasks. In the GRLMAC, a feed-forward prediction model (FPM) is established for describing the length/pressure hysteresis of a chamber in the soft manipulator. Then, the online adjustment for FPM is achieved by reinforcement learning. Introducing the human experience into the reinforcement learning method, we can choose an appropriate adjustment action for the FPM from the action space without the offline training phase, allowing online adjusting the inflation pressure. To demonstrate the effectiveness of the controller, we tested the soft manipulator in the pumped flow current and different gripping loads. The results show that GRLMAC acquires promising accuracy, robustness, and adaptivity. We envision that the soft manipulator with online learning would endow future underwater robotic manipulation under natural turbulent conditions.

Keywords Non-destructive underwater tasks · Guided reinforcement learning · Prediction model · Soft manipulator · Underwater environment

1 Introduction

The ocean covers more than seventy percent of our planet; however, more than eighty percent of our ocean is unobserved and unexplored. This uncharted part of our planet offers huge potential for the industrial sectors, as well as for disruptive and exploration-driven scientific discoveries. Soft robots are compliant, lightweight, and multifunctional, and have nice environmental adaptability and safety. Compared with the existing rigid robots, soft robots have many advantages in a diverse range of underwater applications, such as manipulation in coral reefs, cleaning coast and offshore pollutants, collecting marine

✉ Li Wen
alex.wenli@gmail.com

Hui Yang
405377205@qq.com

Jiaqi Liu
jiaqiliu_buaa@163.com

Xi Fang
fx1120132692@126.com

Xingyu Chen
chenxingyu2015@ia.ac.cn

Zheyuan Gong
zheyuangong@163.com

Shiqiang Wang
wangsq@buaa.edu.cn

Shihan Kong
kongshihan2016@ia.ac.cn

Junzhi Yu
junzhi.yu@ia.ac.cn

[1] School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

[2] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[3] Beijing Innovation Center for Engineering Science and Advanced Technology, Peking University, Beijing, China

Published online: 03 September 2021

biological samples, monitoring underwater structures, and so on (Palli et al. 2017; Zhang et al. 2018a; Xie et al. 2020; Zhuo et al. 2020). However, developing agile, dexterous, and reliable underwater soft robots faces substantial challenges in structural design, actuation, modeling, and control. Soft manipulators have been applied for performing underwater grasping tasks in very recent studies owing to their outstanding environment adaptability and safety (Teeples et al. 2018; Liu et al. 2020; Kurumaya et al. 2018; Mura et al. 2018; Xu et al. 2018). Soft manipulators generally have strong nonlinearities (e.g., asymmetric hysteresis, creep, and so on) due to the characteristics of the materials used as actuators and structures (Hosovsky et al. 2016; Shiva et al. 2016; Stilli et al. 2017; Pawlowski et al. 2019; Thérien and Plante 2016). Furthermore, a soft manipulator mounted on a vehicle for underwater grasping tasks suffers from the effects of ocean currents, water pressure, load change, and disturbances caused by the movement of the vehicle (Zhang et al. 2018b). Efficiently controlling the soft manipulator for underwater tasks remains meaningful and challenging work.

In previous studies, the common control approaches of manipulators can be divided into model-based controllers and model-free controllers (Zhang et al. 2016). Model-based controllers are derived based on physical or semi-physical models of `manipulators (Best et al. 2016; Trivedi and Rahn 2014; Robinson et al. 2014; Li et al. 2018; Chen et al. 2020). The control performance is relevant to the accuracy of the model. Compared with model-based controllers, the model-free controllers require no model information from soft manipulators but require control structures based on real-time accurate feedback data (Vikas et al. 2015; George et al. 2018; Li et al. 2017; Jiang et al. 2020; Bruder et al. 2002, 2020).

Recently, researchers start to apply machine learning methods to model-based controllers for improving the robustness of the soft manipulator. For common dynamic control problems of soft manipulators, Thuruthel et al., proposed a model-based learning method for closed-loop predictive control of a soft robotic manipulator (George et al. 2019). The feedforward dynamic model was established via a recurrent neural network, and then a closed-loop control policy was derived by trajectory optimization and supervised learning based on the dynamic model. Fang et al. (2019) proposed a vision-based online learning kinematic controller for performing precise robotic tasks by local Gaussian process regression, which did not need physical model information of the manipulator and camera parameters. To improve the position control accuracy of soft manipulators, Hofer et al. (2019) presented a norm-optimal iterative learning control algorithm for a soft robotic arm and applied this method for adjusting the output of a PID controller to improve the robustness of the manipulation system . To improve model-based control methods with a low tolerance for external environments, Ho et al. (2018) used a localized online learning-based control to update the inverse model of a redundant two-segment soft robot, which makes the system adapt to the unknown external disturbance. However, machine learning controllers applied for soft manipulators usually require an offline pre-training process, and the trained model cannot be online updated for practical scenes. Furthermore, the training results are likely to get stuck at a locally optimal value (Liu et al. 2017). Therefore, the online training process is essential for the underwater tasks of soft manipulators regarding the time-varying water current disturbance.

In our previous work, we have integrated an opposite-bending-and-stretching structure (OBSS) soft manipulator on a remotely operated vehicle (ROV) system (as shown in Fig. 1) and accomplished harvesting tasks by manual control (Gong et al. 2018, 2019, 2020). However, the soft manipulator has an obvious hysteresis and a low rigidity, which leads to the movement of the manipulator is easily affected by the external disturbance. Therefore, to further improve robustness and adaptivity for autonomous delicate grasping in the aquatic environment, we propose a learning adaptive controller based on the temporal difference reinforcement learning method. In this controller, we design an action selection guidance strategy based on the human experience, thus compared with the above-mentioned controllers, the controller has a good online learning ability and control performance, and doesn't need the offline training process. By using the proposed controller, the predictive output of a feedforward prediction model (chamber length vs. pneumatic pressure) can be adjusted online, which endows the soft manipulator with robustness while encountering underwater disturbances (external loads and stable flow). For abbreviation, we name this controller as a prediction model-based guided reinforcement learning adaptive controller (GRLMAC). Then, we test and validate the effectiveness of GRLMAC on simulation and experiment platforms by carrying on static reaching tasks, dynamic trajectory tracking tasks, and grasping tasks.

This paper is organized as follows. Section 2 introduces the inverse kinematics modeling process of the OBSS soft manipulator briefly. Section 3 introduces the feed-forward prediction model briefly. Then, we design the guided reinforcement learning policy to modify the prediction model output and proposed a prediction model-based guided reinforcement learning adaptive controller (GRLMAC). Section 4 establishes the simulation platform in the MATLAB environment. And then control performance, learning efficiency, and robustness of GRLMAC for different external loads and time-varying disturbance are analyzed. Section 5 gives the physical experimental platform and then conducts some experiment tasks to further verify the performance of GRLMAC. Section 6 draws conclusions.

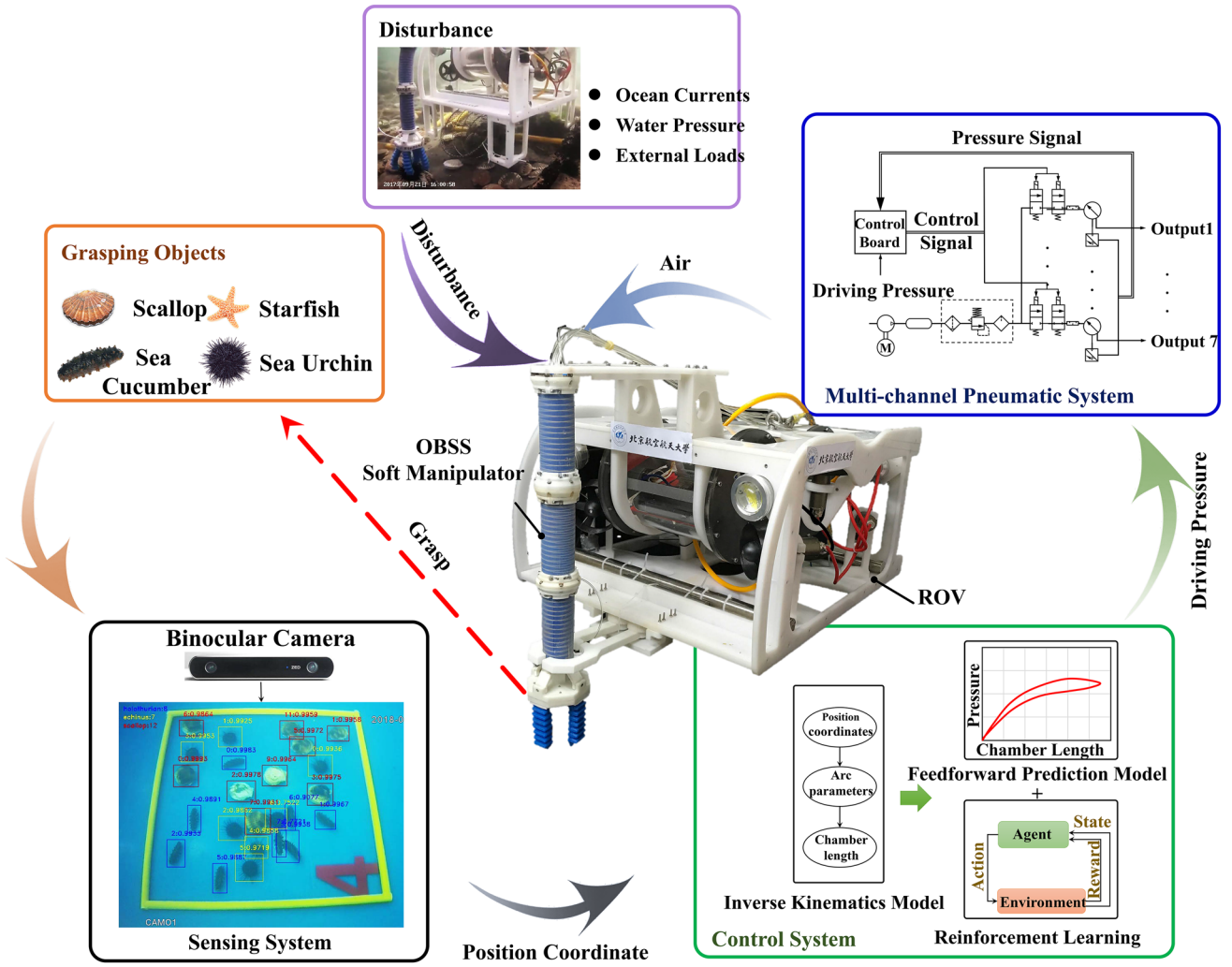


Fig. 1 The underwater grasping system consists of the OBSS soft manipulator and an ROV. The OBSS soft manipulator system contains a sensing system (a binocular camera), a control system, and a multi-channel pneumatic system

2 Inverse kinematics model

The physical prototype and space coordinate systems of the OBSS soft manipulator are shown in Fig. 2. The soft manipulator consists of two bending segments, one extending segment, and one soft gripper. All of the parts are fabricated with silicon rubber. Each bending segment with 2-DOFs has three actuated chambers and the extending chamber with 1-DOF has one actuated chamber. The two bending segments assembled with an offset angle of 180° have the same radius and initial length, and always keep equal bending angles and sigmoidal opposing curvatures during manipulation so that the orientation of the soft gripper is always kept vertically downward.

Based on the characteristics of the OBSS soft manipulator, we have established its kinematics model in our previous work (Gong et al. 2019). In this section, we will introduce the modeling process briefly, and the notations are summarized in Table 1.

The constraint conditions for kinematics modeling are determined as follows

$$\begin{cases} \theta_1 = \theta_2, \quad \phi_2 = \phi_1 + \pi \\ \lambda_1 = \lambda_2, \quad r_1 = r_2 \\ l_{1j} = l_{2j} \quad (j = 1,\ 2,\ 3) \end{cases} \tag{1}$$

where θ_i ($i = 1, 2$ represents the ith bending segment) is the bending angle, ϕ_i is the deflection angle, λ_i is the radius of center curvature, r_i is the distance from the cross-sectional center to the center of a chamber, and l_{ij} is the length of the jth chamber in the ith bending segment. Based on the above conditions, relative to the base coordinate system

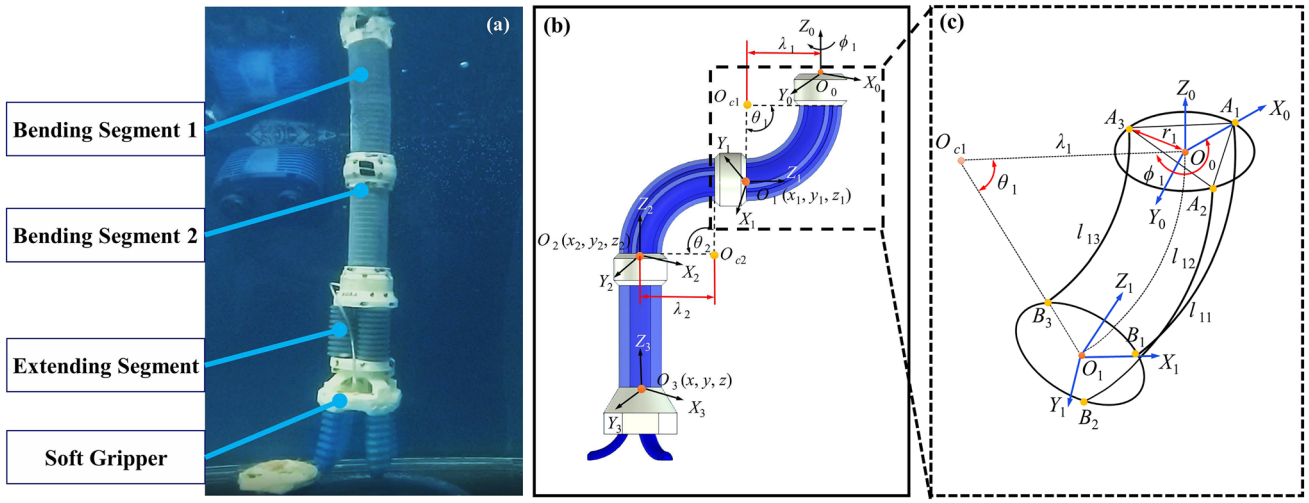


Fig. 2 The OBSS soft manipulator. **a** The physical prototype of the OBSS soft manipulator which is consisted of four parts including bending segment 1, bending segment 2, extending segment, and soft gripper. **b** Space coordinate systems of the OBSS soft manipulator, where O_0-$X_0Y_0Z_0$ is the base coordinate system; O_1-$X_1Y_1Z_1$, O_2-$X_2Y_2Z_2$, and O_3-$X_3Y_3Z_3$ are moving coordinate systems for each segment of the soft manipulator. **c** Geometric relationship of bending segment 1

Table 1 Notation and definitions

Symbol	Unit	Definition
θ_i	rad	Bending angle of the ith bending segment
ϕ_i	rad	Deflection angle of the ith bending segment
λ_i	mm	Radius of center curvature of the ith bending segment
r_i	mm	Radius of the ith bending segment
l_{ij}	mm	Length of the jth actuated chamber in the ith bending segment
l_{dij}	mm	Desired length of the jth actuated chamber in the ith bending segment
l_e	mm	Length of the actuated chamber in the extending segment
l_{de}	mm	Desired length of the actuated chamber in the extending segment
O_1 (x_1, y_1, z_1)	mm	The center coordinates of O_1-$X_1Y_1Z_1$ relative to O_0-$X_0Y_0Z_0$
O_2 (x_2, y_2, z_2)	mm	The center coordinates of O_2-$X_2Y_2Z_2$ relative to O_0-$X_0Y_0Z_0$
$O_3(x, y, z)$	mm	The center coordinates of O_3-$X_3Y_3Z_3$ relative to O_0-$X_0Y_0Z_0$
$u_p(t)$	bar	Predictive driving pressure of the chamber
κ	bar	Correction coefficient for $u_p(t)$
$u_a(t)$	bar	Actual driving pressure of the chamber
$F_{r_m, \alpha_m, \beta_m}[l](t)$	–	Improved unparallel Prandtl- Ishlinskii operator
ω_m	–	Weight coefficient for $F_{r_m, \alpha_m, \beta_m}[l](t)$ in the mth dead zone
r_m	–	Boundary threshold value of the mth dead zone
α_m	–	Tilt coefficient of the pressurization edge in the mth dead zone
β_m	–	Tilt coefficient of the depressurization edge in the mth dead zone
p_n	–	Weight coefficient of the polynomial portion P[l](t)

O_0−$X_0Y_0Z_0$, the end center coordinates of each bending segment are expressed as Eq. (2).

$$x_1=\frac{x_2}{2}=\frac{x}{2},\quad y_1=\frac{y_2}{2}=\frac{y}{2},\quad |z_1|=\frac{|z_2|}{2}=\frac{|z|-l_e}{2} \tag{2}$$

where O_1 (x_1, y_1, z_1) is the end center coordinates of bending segment 1, O_2 (x_2, y_2, z_2) is the end center coordinates of bending segment 2, O_3 (x, y, z) is the end center coordinates of the soft gripper, and l_e is the length of the extending segment. Then, the deflection angle ϕ_1 can be calculated by

$$\phi_1=\tan^{-1}\left(\frac{y}{x}\right) \tag{3}$$

Based on the geometric relationship described in Fig. 2, the bending angle θ_1 can be obtained by setting the value of z_1

$$\theta_1 = \pi - 2\sin^{-1}\left(\frac{z_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}}\right) \tag{4}$$

Then, the radius of curvature λ_1 is

$$\lambda_1 = \sqrt{\frac{x_1^2 + y_1^2 + z_1^2}{2(1 - \cos\theta_1)}} \tag{5}$$

Based on Eqs. (3)–(5), the length of the jth chamber in bending segment 1 can be obtained

$$\begin{cases} l_{11} = \theta_1(\lambda_1 - r_1\cos\phi_1) \\ l_{12} = \theta_1\left(\lambda_1 - r_1\cos\left(\frac{2\pi}{3} - \phi_1\right)\right) \\ l_{13} = \theta_1\left(\lambda_1 - r_1\cos\left(\frac{4\pi}{3} - \phi_1\right)\right) \end{cases} \tag{6}$$

And then, the length of the jth chamber in bending segment 2 can be obtained from Eq. (1), and the length of the extending segment l_e can be obtained from Eq. (2). If the results are not satisfied with the length requirement for each chamber, we modify the value of z_1, and then calculate the chamber length again.

3 Guided reinforcement learning model-based adaptive controller

3.1 Hysteresis model

For measuring the relationship between pressure and length for a chamber in the bending segment or the extending segment, we conducted an isotonic test in the water and non-loaded condition. From the measurement results (as shown in Fig. 3), we found that the actuated chamber in each segment of the soft manipulator has an obvious unsymmetric hysteresis. To describe the phenomenon, in this paper, the extended unparallel Prandtl- Ishlinskii (EUPI) model is adopted and expressed as Eq. (7). This model is an effective method to describe the hysteresis of artificial muscles (Hao et al. 2017; Sun et al. 2017).

$$\begin{cases} u_{\mathrm{p}}(t) = \Gamma_{\mathrm{UPI}}[l](t) + \mathrm{P}[l](t) \\ \Gamma_{\mathrm{UPI}}[l](t) = \sum_{m=1}^{N_r} \omega_m F_{r_m,\,\alpha_m,\,\beta_m}[l](t) \\ F_{r_m,\,\alpha_m,\,\beta_m}[l](t) = \max\{\alpha_m(l(t) - r_m), \\ \qquad\qquad \min\{\beta_m(l(t) + r_m),\ F_{r_m,\,\alpha_m,\,\beta_m}[l](t-1)\}\} \\ \mathrm{P}[l](t) = p_1 l(t)^3 + p_2 l(t)^2 + p_3 l(t) \end{cases} \tag{7}$$

where $u_{\mathrm{p}}(t)$ represents the driving pressure for the chamber predicted by the EUPI model. The EUPI model consists of unparallel PI (UPI) portion $\Gamma_{\mathrm{UPI}}[l](t)$ and polynomial portion $\mathrm{P}[l](t)$. $l(t)$ is the chamber length of the soft manipulator, $\omega_m > 0$ ($m = 1, 2, \ldots, N_r$, N_r is the total number of dead

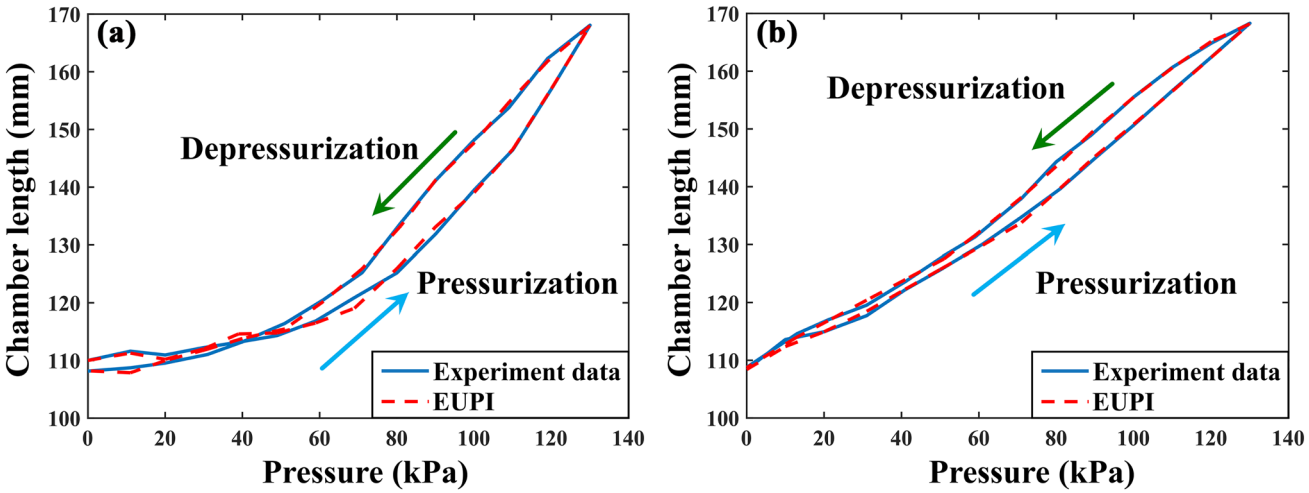


Fig. 3 Pressure-length hysteresis curves and corresponding fitting curves for chambers in the OBSS soft manipulator in the underwater environment and non-load condition. **a** The hysteresis curve and the fitting curve for the bending segment. **b** The hysteresis curve and the fitting curve for the extending segment

zones) is a weight coefficient, $F_{r_m,\alpha_m,\beta_m}[l](t)$ is the unparallel PI operator, α_m and β_m are tilt coefficients of pressurization edge and depressurization edge respectively, $r_m(r_m \geq 0)$ is the boundary threshold value of the mth dead zone, and p_n ($n=1, 2, 3$) is the weight coefficient of the polynomial portion. α_m, β_m, ω_m, and p_n are identified by particle swarm optimization (PSO) algorithm. The EUPI fitting curves are shown in Fig. 3.

3.2 Guided reinforcement learning policy

Based on the kinematic model of the soft manipulator and the EUPI model of the chamber, we establish a soft manipulator motion control system. In this control system, based on the target position, we first calculate each chamber's desired length through the kinematic model. Then, we take the desired length into the EUPI and calculate the predictive driving pressure of each chamber, that is, the EUPI model is treated as a feed-forward prediction model (FPM) in our work.

Nevertheless, the EUPI model is identified in a specific condition (no external load), so it has a poor universality. Its predictive performance is easily affected by changes in external conditions. To address the problem, we use a correction coefficient $\kappa(t)$ to modify the predictive driving pressure $u_p(t)$ and the actual driving pressure for a chamber in the soft manipulator are expressed as

$$u_a(t) = u_p(t) + \kappa(t) \tag{8}$$

where $\kappa(t)$ is represented as follows

$$\kappa(t) = \kappa(t-1) + \Delta\kappa(s_1(t)) \tag{9}$$

where $\Delta\kappa(s_1(t))$ is an adjustment function which is about $s_1(t) = l_d\ (t+1) - l\ (t)$ (l_d is the desired chamber length). In this paper, $\Delta\kappa(s_1(t))$ is set as an exponential function expressed in Eq. (10) to ensure it is bounded.

$$\Delta\kappa(t) = \left(p_1' e^{\left(p_2' - \frac{p_3'}{|s_1(t)|}\right)} + p_4'\right)\mathrm{sgn}(s_1(t))$$
$$\mathrm{sgn}(s_1(t)) = \begin{cases} -1 & s_1(t) < 0 \\ 0 & s_1(t) = 0 \\ 1 & s_1(t) > 0 \end{cases} \tag{10}$$

In (10), p_1', p_2', p_3' and $p_4' > 0$.

To improve the flexibility and stability for adjusting $\kappa(t)$, we need the above parameters in $\Delta\kappa(s_1(t))$ are also related to $s_1(t)$. To this end, in this section, we design an online learning strategy to determined p_1', p_2', p_3' and p_4' by using the Sarsa learning algorithm because of its high learning rate without the knowledge of the environment model (such as the state transition probabilities) comparing with dynamic programming and Monte Carlo (Sutton and Barto 1998; Sutton 1988; Kirkpatrick and Valasek 2009; Kirkpatrick et al. 2013). The detailed design procedure is described as follows.

For the soft manipulator system, we set state variables $s_1(t)$ and $s_2(t+1) = l_d(t+1) - l(t+1)$ belong to a state-space $\boldsymbol{S} = \{s | -\infty < s < \infty\}$. Based on the displacement range of the actuated chamber described in Fig. 3, the state space $\boldsymbol{S}$ could be divided into the following seven continuous intervals.

$$\mathbf{S} = \{\mathbf{S}_1,\ \mathbf{S}_2,\ \mathbf{S}_3,\ \mathbf{S}_4,\ \mathbf{S}_5,\ \mathbf{S}_6,\ \mathbf{S}_7\}$$
$$\begin{cases} \mathbf{S}_1 = \{s | -\infty < s < -50\};\ \mathbf{S}_2 = \{s | -50 \leq s < -0.1\}; \\ \mathbf{S}_3 = \{s | -0.1 \leq s < -0.00001\}; \\ \mathbf{S}_4 = \{s | -0.00001 \leq s \leq 0.00001\}; \\ \mathbf{S}_5 = \{s | 0.00001 < s \leq 0.1\}; \\ \mathbf{S}_6 = \{s | 0.1 < s \leq 50\};\ \mathbf{S}_7 = \{s | 50 < s < \infty\}. \end{cases} \tag{11}$$

Then, corresponding to each state interval, based on the soft manipulator's driving performance, we set an action space $\boldsymbol{A}$ which contains four actions and is expressed in Eq. (12).

$$\mathbf{A} = \{a_1,\ a_2,\ a_3,\ a_4\}$$
$$\begin{cases} a_1 :\ [p_1'\ p_2'\ p_3'\ p_4'] = [10^3\ \ 1\ \ 50\ \ 10^2], \\ a_2 :\ [p_1'\ p_2'\ p_3'\ p_4'] = [10^2\ \ \frac{0.1}{50}\ \ 0.1\ \ 10], \\ a_3 :\ [p_1'\ p_2'\ p_3'\ p_4'] = [10\ \ \frac{0.00001}{0.1}\ \ 0.00001\ \ 1], \\ a_4 :\ [p_1'\ p_2'\ p_3'\ p_4'] = [1\ \ 1\ \ 0.00001\ \ 0] \end{cases} \tag{12}$$

Based on Eq. (12), at the current state $s_1(t)$, we can select an action $a(k)$ from $\boldsymbol{A}$ to determine the parameters in the $\Delta\kappa(s_1(t))$. After that, the driving pressure $u_a(t)$ is calculated by Eqs. (8)–(10), and then we execute $u_a(t)$ and obtain the state variable $s_2(t+1)$. To evaluate the selected action $a(t)$, by considering the plausibility and validity of the selected action at state $s_1(t)$, we design a reward matrix $\boldsymbol{R}$. In our work, we set the allowable range of $s_2(t+1)$ as $[-0.1\ 0.1]$, hence the reward matrix $\boldsymbol{R}$ is described as (3).

$$\mathbf{R} = \begin{bmatrix} a(t) \in \mathbf{A} / s_2(t+1) \in \mathbf{S} & a_1 & a_2 & a_3 & a_4 \\ \mathbf{S}_1 & 0 & 0 & 0 & 0 \\ \mathbf{S}_2 & 0 & 0 & 0 & 0 \\ \mathbf{S}_3 & 1 & 1 & 1 & 1 \\ \mathbf{S}_4 & 10 & 10 & 10 & 10 \\ \mathbf{S}_5 & 1 & 1 & 1 & 1 \\ \mathbf{S}_6 & 0 & 0 & 0 & 0 \\ \mathbf{S}_7 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

According to Eq. (13), the current reward $r(t) = 0$ means that state $s_2(t+1)$ generated by action $a(t)$ is a bad or impossible state, $r(t) = 1$ means that state $s_2(t+1)$ is a reasonable, but not the best state, and $r(t) = 10$ means that state $s_2(t+1)$ is

the best state under action $a(t)$. Therefore, to make $s_2(t+1)$ is the best state, we need the soft manipulator system to learn to select the appropriate action from $\boldsymbol{A}$ at the current state $s_1(t)$.

To this end, based on the Sarsa algorithm (Gong et al. 2018, 2019, 2020; Hao et al. 2017), a state-action value matrix $\boldsymbol{Q}(\boldsymbol{S}, \boldsymbol{A}) \in \mathbf{R}^{7\times4}$ is designed and its recursive equation is defined as

$$\begin{aligned}&\mathbf{Q}(\mathbf{S}_1(t), a(t)) \leftarrow \mathbf{Q}(\mathbf{S}_1(t), a(t))\\&+\alpha[r(t)+\gamma\mathbf{Q}(\mathbf{S}_1(t+1), a(t+1))-\mathbf{Q}(\mathbf{S}_1(t), a(t))]\end{aligned} \tag{14}$$

where $\boldsymbol{S}_1(t)$ is the state interval in state-space $\boldsymbol{S}$ which is the state $s_1(t)$ belongs to, α is the learning rate, and $\gamma \in [0,1]$ is the discount factor. Then, based on $\boldsymbol{Q}$, we select the action $a(t)$ from the action space $\boldsymbol{A}$.

The ε-greedy policy is the basic and commonly used action selection strategy for reinforcement learning methods and contains the exploration phase and the exploitation phase (Sutton and Barto 1998). In the exploration phase, we arbitrarily choose an action from $\boldsymbol{A}$ with a small probability ε. In the exploitation phase, we choose the optimal action (the action corresponding to the maximum of $\boldsymbol{Q}$ at state $s_1(t) \in \boldsymbol{S}_k$) with a probability $1-\varepsilon$. For the soft manipulator system, the ε-greedy policy is represented as Eq. (15).

$$\begin{cases}\text{if rand}() < \varepsilon & a(t) \leftarrow \text{rand}_a(\mathbf{A}(1, \{a_1, a_2, a_3, a_4\}))\\ \text{else} & a(t) \leftarrow \max\limits_a(\mathbf{Q}(\mathbf{S}_1(t), \{a_1, a_2, a_3, a_4\}))\end{cases} \tag{15}$$

From Eq. (15), to converge to the optimal $\boldsymbol{Q}$, it always needs to take a large number of steps, and the optimal results can easily drop into the local optimum because of the diversity of the optional action.

We design an action selection guidance strategy to improve ε-greedy policy based on our knowledge of the actuation performance of the chamber and experience in soft manipulator operation, which is obtained from a large number of experiments in our early work (Gong et al. 2019). This method reduces the step cost and improves the convergence rate and the global convergence of the reinforcement learning method. Then, we can determine the corresponding action choice ranges for different state intervals in the state space $\boldsymbol{S}$. For example, when $s_1(t)$ belongs to $\boldsymbol{S}_1$, according to our experience, to make the actuated chamber reach the desired length rapidly, we need to adjust $\kappa(t)$ considerably; thus the action a_1 is the best option.

Then, based on the action selection guidance strategy, the ε-greedy policy can be rewritten as Eq. (16). The improved ε-greedy policy reduces the variety of action choices and endows reasonable choice ranges, which enable the learning method with a fast convergence performance and can be used for online adjusting $\kappa(t)$ in a real-time manner without an offline pre-training phase.

$$\begin{cases}\text{if rand}() < \varepsilon & a(t) \leftarrow \text{rand}_a(\mathbf{A}(1, \{a_1, a_2, a_3, a_4\}))\\ \text{else} & \text{if } s_1(t) \in \mathbf{S}_1, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_1, \{a_1\})\\ & \text{if } s_1(t) \in \mathbf{S}_2, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_2, \{a_1, a_2\})\\ & \text{if } s_1(t) \in \mathbf{S}_3, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_3, \{a_2, a_3\})\\ & \text{if } s_1(t) \in \mathbf{S}_4, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_4, \{a_3, a_4\})\\ & \text{if } s_1(t) \in \mathbf{S}_5, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_5, \{a_2, a_3\})\\ & \text{if } s_1(t) \in \mathbf{S}_6, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_6, \{a_1, a_2\})\\ & \text{if } s_1(t) \in \mathbf{S}_7, \quad a(t) \leftarrow \max\limits_a \mathbf{Q}(\mathbf{S}_7, \{a_1\})\end{cases} \tag{16}$$

According to Eqs. (8)–(16), we represent the prediction model-based guided reinforcement learning adaptive controller (GRLMAC) for a chamber, and the control schematic diagram for the OBSS soft manipulator system is depicted in Fig. 4. The corresponding control procedure is shown in Algorithm 1.

Algorithm 1 GRLMAC

Input: Desired length $l_{\mathrm{d}}(t)$ and length error $\Delta l(t)$ of an actuated chamber.
Output: Actual driving pressure $u_{\mathrm{a}}(t)$.

1. Initialize the state-action value function $\boldsymbol{Q}$ arbitrarily (random numbers).
2. Initialize the state $s_1(t)$ arbitrarily.
3. Measure the target position and calculate $l_{\mathrm{d}}(t)$ by the inverse kinematics model.
4. Select action $a(t)$ based on the improved ε-greedy policy.

$$\begin{cases}\text{if rand}() < \varepsilon & a(t) \leftarrow \text{rand}_a(\boldsymbol{A}(1, \{a_1, a_2, a_3, a_4\}))\\ \text{else} & \text{if } s_1(t) \in \boldsymbol{S}_1, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_1, \{a_1\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_2, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_2, \{a_1, a_2\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_3, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_3, \{a_2, a_3\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_4, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_4, \{a_3, a_4\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_5, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_5, \{a_2, a_3\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_6, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_6, \{a_1, a_2\})\\ & \text{if } s_1(t) \in \boldsymbol{S}_7, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_7, \{a_1\})\end{cases}$$

5. **repeat**
6. Take action $a(t)$ and obtain the correction coefficient $\kappa(t)$;
7. Based on $l_{\mathrm{d}}(t)$, calculate the predictive driving pressure $u_{\mathrm{p}}(t)$ using FPM;
8. Calculate the actual driving pressure $u_a(t)=u_p(t)+\kappa(t)$;
9. Execute $u_a(t)$, measure the distance between the soft manipulator end and the target $\Delta\boldsymbol{P}(t)$, calculate $\Delta l(t)$ using the inverse kinematics model, move to the next state $s_1(t+1)$, and receive the reward $r(t)$;
10. Choose the next action $a(t+1)$ at state $s_1(t+1)$, using the improved ε-greedy policy.

$$\begin{cases}\text{if rand}() < \varepsilon & a(t+1) \leftarrow \text{rand}_a(\boldsymbol{A}(1, \{a_1, a_2, a_3, a_4\}))\\ \text{else} & \text{if } s_1(t+1) \in \boldsymbol{S}_1, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_1, \{a_1\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_2, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_2, \{a_1, a_2\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_3, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_3, \{a_2, a_3\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_4, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_4, \{a_3, a_4\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_5, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_5, \{a_2, a_3\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_6, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_6, \{a_1, a_2\})\\ & \text{if } s_1(t+1) \in \boldsymbol{S}_7, \quad a(t) \leftarrow \max_a \boldsymbol{Q}(\boldsymbol{S}_7, \{a_1\})\end{cases}$$

11. Update the state-action value matrix $\boldsymbol{Q}(\boldsymbol{S}, \boldsymbol{A})$

$$\boldsymbol{Q}(\boldsymbol{S}_1(t), a(t)) \leftarrow \boldsymbol{Q}(\boldsymbol{S}_1(t), a(t)) + \alpha[r(t)+\gamma\boldsymbol{Q}(\boldsymbol{S}_1(t+1), a(t+1)) - \boldsymbol{Q}(\boldsymbol{S}_1(t), a(t))]$$

12. **until** the specified number of steps completed.

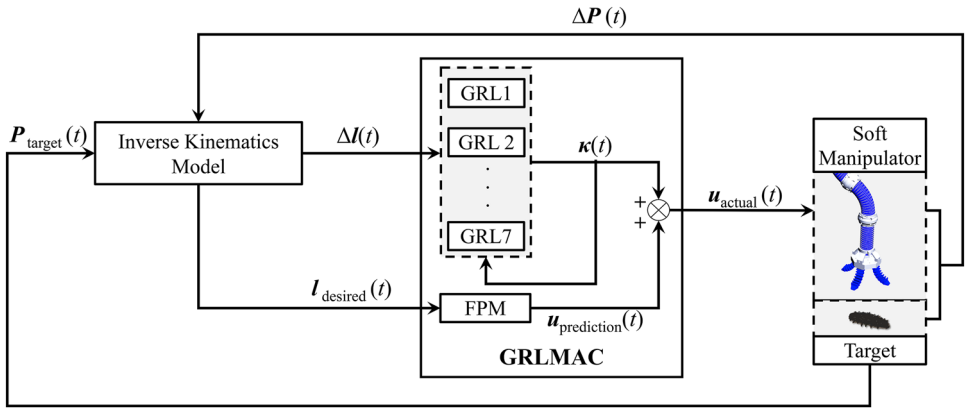


Fig. 4 Schematic diagram of GRLMAC for the soft manipulator. Based on the position of the target $\boldsymbol{P}_{\text{target}}(t)$ and the distance $\Delta\boldsymbol{P}(t)$, the desired chamber length $\boldsymbol{l}_{\text{desired}}(t)$ and the chamber length error $\Delta\boldsymbol{l}(t)$ can be obtained via the inverse kinematics model. The predictive driving pressure $\boldsymbol{u}_{\text{prediction}}(t)$ is calculated by taking $\boldsymbol{l}_{\text{desired}}(t)$ into the FPM and is online adjusted by the correction coefficient $\boldsymbol{\kappa}(t)$ which is obtained by taking $\Delta\boldsymbol{l}(t)$ and $\boldsymbol{\kappa}(t-1)$ into the GRL module. $\boldsymbol{u}_{\text{actual}}(t)$ is the actual driving pressure for the soft manipulator

In Fig. 4, $\Delta\mathbf{P}(t) = [\Delta x(t), \Delta y(t), \Delta z(t)]$ is the distance between the soft manipulator end and the target, $\mathbf{P}_{\text{target}}(t) = [x_t(t), y_t(t), z_t(t)]$ is the position of the target, $\mathbf{l}_{\text{desired}}(t) = [l_{d11}(t), l_{d12}(t), l_{d13}(t), l_{d21}(t), l_{d22}(t), l_{d23}(t), l_{de}(t)]$ and $\Delta\mathbf{l}(t) = [\Delta l_{11}(t), \Delta l_{12}(t), \Delta l_{13}(t), \Delta l_{21}(t), \Delta l_{22}(t), \Delta l_{23}(t), \Delta l_e(t)]$ are the desired length and the length error of chambers respectively, $\mathbf{u}_{\text{prediction}}(t) = [u_{p11}(t), u_{p12}(t), u_{p13}(t), u_{p21}(t), u_{p22}(t), u_{p23}(t), u_{pe}(t)]$ is the predictive driving pressure for the soft manipulator calculated by (7), $\boldsymbol{\kappa}(t) = [\kappa_{11}(t), \kappa_{12}(t), \kappa_{13}(t), \kappa_{21}(t), \kappa_{22}(t), \kappa_{23}(t), \kappa_e(t)]$ is the correction coefficient for $\boldsymbol{u}_{\text{prediction}}(t)$, and $\mathbf{u}_{\text{actual}}(t) = [u_{a11}(t), u_{a12}(t), u_{a13}(t), u_{a21}(t), u_{a22}(t), u_{a23}(t), u_{ae}(t)]$ is the actual driving pressure for the soft manipulator. Each chamber requires a corresponding GRLMAC for controlling its length variation.

3.3 Stability analysis

The actuated chamber of the OBSS soft manipulator as a controllable system should satisfy the following assumptions which are described in Bu et al. (2019):

A1: The input and the output of the soft manipulator control system are measurable and controllable. When disturbances are bounded, there is always one bounded input signal corresponding to a bounded desired output signal, which makes the actual system output signal equal to the desired one.

A2: The nonlinear system function has a continuous partial derivative with respect to the current input signal.

A3: The actuated chamber control system satisfies the generalized Lipschitz condition that exists a parameter $b > 0$ makes Eq. (17) be established.

$$\begin{aligned} &t_1 \neq t_2, \quad t_1,\ t_2 > 0 \\ &u_a(t_1) \neq u_a(t_2) \\ &\left|l(t_1+1) - l(t_2+1)\right| \leq b\left|u_a(t_1) - u_a(t_2)\right| \end{aligned} \tag{17}$$

Based on the assumptions **A2** and **A3**, when $|u_a(t)-u_a(t-1)|\neq 0$ there must be a $\psi(t)\in R$, so that $|l(t)-l(t-1)|=\psi(t)|u_a(t)-u_a(t-1)|$. Moreover, as shown in Fig. 3, the input pressure and the output length of the actuated chamber have the same monotonicity, which means that

$$\operatorname{sgn}\left(\Delta\, u_a(t)\right) = \operatorname{sgn}(\Delta\, l(t))$$

Moreover, the discrete-time state equation could be expressed as

$$\begin{cases} \mathbf{x}_{\text{st}}(t+1) = \mathbf{A}\mathbf{x}_{\text{st}}(t) + \mathbf{B}\Delta u_a(t) \\ \Delta l(t+1) = \mathbf{C}\mathbf{x}_{\text{st}}(t) + \mathbf{D}\Delta u_a(t) \end{cases} \tag{18}$$

where $\boldsymbol{x}_{\text{st}}(t)$ denotes the state variable, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are coefficient matrixes and expressed as follows

$$\mathbf{A} = \begin{bmatrix} -\dfrac{C_{ac}}{K_{ac}}, & 0 \\ 0\ , & -\dfrac{C_{ac}}{K_{ac}} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} -\dfrac{C_{ac}}{(K_{ac})^2} \\ 0 \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{D} = \begin{bmatrix} \dfrac{1}{K_{ac}} \\ 0 \end{bmatrix}, \quad \mathbf{x}_{\text{st}}(t) = \begin{bmatrix} K_{ac}\Delta l(t) \\ K_{ac}\Delta l(t) \end{bmatrix}$$

where C_{ac} and $K_{ac} > 0$.

For analyzing the stability of GRLMAC, we need to construct a common Lyapunov function that exists in each state interval and satisfies the following conditions

$$\begin{cases} \mathrm{V}(\mathbf{x}_{st}(t)) > 0 \text{ and } \mathrm{V}(0) = 0 \\ \Delta \mathrm{V}(\mathbf{x}_{st}(t)) = \mathrm{V}(\mathbf{x}_{st}(t+1)) - \mathrm{V}(\mathbf{x}_{st}(t)) < 0 \\ \text{when } \mathbf{x}_{st}(t) \to \infty,\ \mathrm{V}(\mathbf{x}_{st}(t)) \to \infty \end{cases} \tag{19}$$

Then, based on Eqs. (18) and (19), we consider the following Lyapunov candidate function

$$\mathrm{V}(\mathbf{x}_{\mathrm{st}}(t+1)) = \mathbf{x}_{\mathrm{st}}(t+1)^{\mathrm{T}}\mathbf{P}\mathbf{x}_{\mathrm{st}}(t+1) \tag{20}$$

where $\boldsymbol{P}$ is symmetric positive matrixes and represented as

$$\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A} - \mathbf{P} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Then, we obtain

$$\mathbf{P} = \begin{bmatrix} \frac{(K_{ac})^2}{(K_{ac})^2 - (C_{ac})^2} & 0 \\ 0 & \frac{(K_{ac})^2}{(K_{ac})^2 - (C_{ac})^2} \end{bmatrix}$$

Because $\boldsymbol{P}$ is a positive matrix, we have $K_{ac} > C_{ac}$. Then, the deviation of $\mathrm{V}(\boldsymbol{x}_{\mathrm{st}}(t+1), \boldsymbol{\alpha}(t))$ is

$$\begin{aligned} &\Delta V(\mathbf{x}_{\mathrm{st}}(t+1)) \\ &= \mathbf{x}_{\mathrm{st}}(t+1)^{\mathrm{T}}\mathbf{P}\mathbf{x}_{\mathrm{st}}(t+1) - \mathbf{x}_{\mathrm{st}}(t)^{\mathrm{T}}\mathbf{P}\mathbf{x}_{\mathrm{st}}(t) \\ &= \mathbf{x}_{\mathrm{st}}^{\mathrm{T}}(t)[\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A} - \mathbf{P}]\mathbf{x}_{\mathrm{st}}(t) + \mathbf{x}_{\mathrm{st}}^{\mathrm{T}}(t+1)\mathbf{P}\mathbf{B}\Delta u_a(t) \\ &\quad + \Delta u_a(t)\mathbf{B}^{\mathrm{T}}\mathbf{P}[\mathbf{x}_{\mathrm{st}}(t+1) - \mathbf{B}\Delta u_a(t)] \end{aligned} \tag{21}$$

Hence, for $\Delta V(\mathbf{x}_{\mathrm{st}}(t+1)) < 0$, we know that the value of $\Delta u_a(t)$ should be over $2K_{ac}$ times than $\Delta l(t)$. In our work, according to the driving performance of the actuated chamber (as shown in Fig. 3) and multiple parameter adjustment experiments, we design the action space $\boldsymbol{A}$ and determine its parameters so that the proposed controller satisfies the above stability conditions.

4 Simulation and results

For verifying the control performance of GRLMAC, some tracking tasks are performed under a time-varying disturbance and different external load conditions. For all the tasks, the simulation step size $h = 0.01$ s and the parameters of GRLMAC are set as follows: the learning rate $\alpha = 1$, the discount rate $\gamma = 0.8$, the initial value of $\boldsymbol{\kappa} =$ [1, 1, 1, 1, 1, 1, 1], and the value matrix $\boldsymbol{Q}(s, a)$ is initialized to a zeros matrix. All simulations are performed on a PC with an i7 CPU @ 2.70 GHz, 16 GB RAM, and MATLAB 2016b.

4.1 Static performance

To validate the static performance of GRLMAC, we formulated reaching tasks with different external conditions. The actual time consumption for each reaching task is 2 s. Figure 5 shows the tracking performance of GRLMAC for a target point (100, 100, 400) under different external conditions. For different external load conditions (Mload = 0 g and 200 g) without disturbance, GRLMAC maintains a short settling time (less than 0.2 s) and low steady-state distance, as shown in Table 2. Then, to demonstrate the robustness of GRLMAC for the external disturbance, we add a time-varying disturbance to the X direction (X dis = 5t). Compared with the results without disturbance, the static performance is non-significantly affected by the external disturbance (the variations in settling time and distance are about 0.02 s and 0.3 mm) as shown in Fig. 5 and Table 2. Therefore, GRLMAC may endow the system with strong robustness and fast online- adjustment ability for the external disturbance.

4.2 Dynamic performance

To validate the dynamic performance of GRLMAC, trajectory tracking tasks with different initial external loads and disturbances are formulated. The trajectory is set as (20sin(πt/1.25), 20cos(πt/1.25), 500). The time consumption for each task is 5 s. Figure 6 describes simulation results and shows that GRLMAC maintains a superior control performance for different external loads and the time-varying disturbance (the mean distance is less than 0.2 mm). By comparing indicators shown in Table 3, we can find that GRL can efficiently adjust predictive driving pressure $\boldsymbol{u}_{\mathrm{prediction}}(t)$ and ensure that the variation in distance is less than 0.02 mm, which illustrates that the proposed controller improves the robustness of the soft manipulator for the external disturbance.

5 Experiments and results

To further verify the performance of GRLMAC, experiments were conducted on a soft manipulator system. Figure 7 shows that the system is composed of a soft manipulator, a binocular camera (ZED, Stereolab, USA) which is used to measure the distance between the target and the gripper, a multi-channel pneumatic system, a vibration pump, and a PC.

In the multi-channel pneumatic system, eight proportional valves (ITV0030, SMC, Japan) are used for actuating the soft manipulator. Besides, a vibration pump is used for generating a constant flow disturbance for the manipulator in the X-direction. In this section, three kinds of experiment tasks including the static reaching task, the

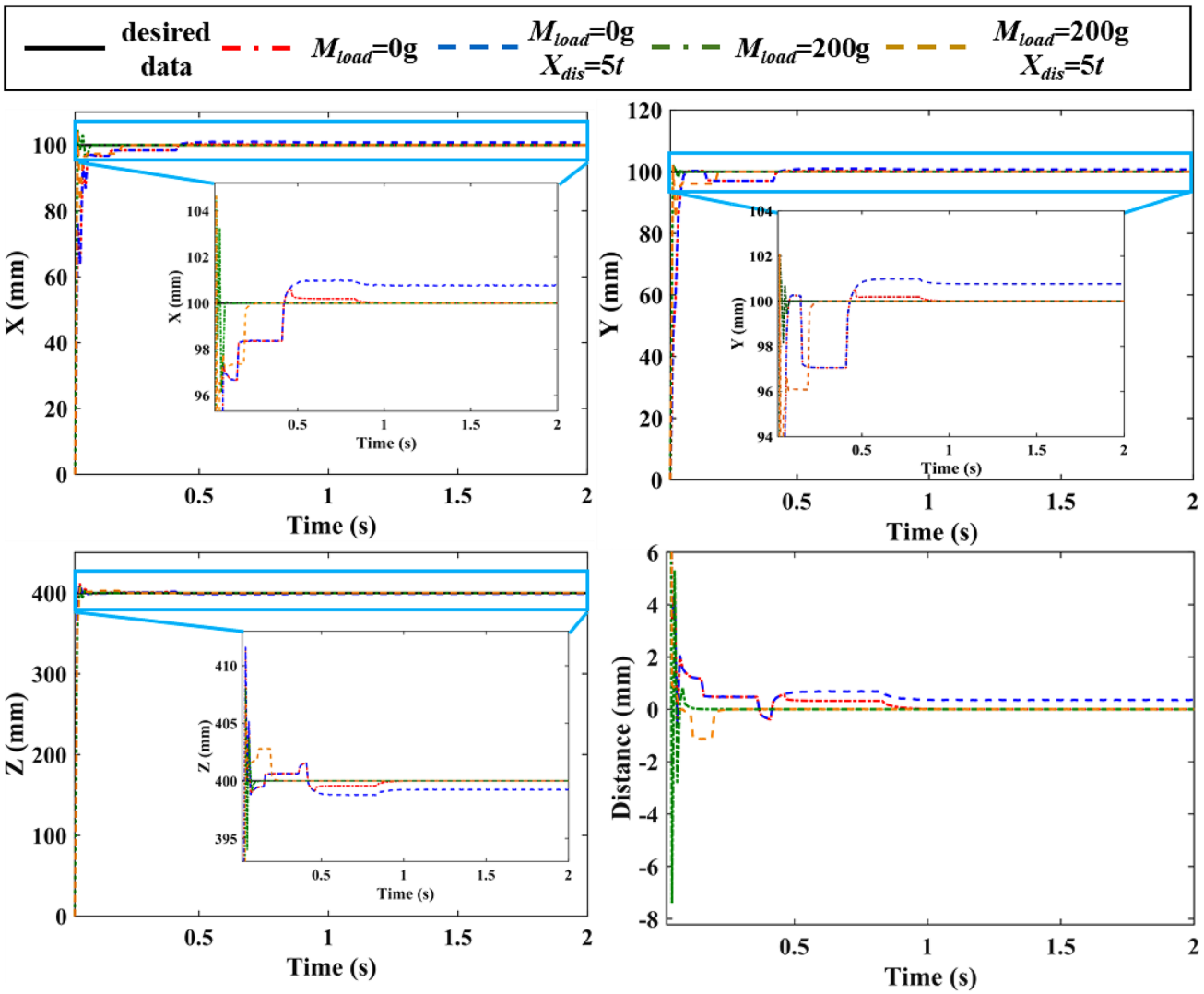


Fig. 5 Reaching tasks results under different external conditions

dynamic trajectory tracking task, and the grasping task are performed. All of the above tasks are accomplished in the water environment. Moreover, the parameters of GRLMAC are set as the same value as in Sect. 4. It should be noted that the soft manipulator is only controlled in the XY plane for the above experiment tasks. The reason is that it can avoid the problem of position detection caused by shading from the manipulator. Moreover, Because of the response time of the proportional valve, data transmission time, and airflow rate, each control step of the OBSS control system takes about 0.8–1.5 s. Hence to demonstrate the actual response performance of the proposed controller, in the following curve figures, the unit of the X-axis is the number of consuming steps.

Table 2 Indicators of static performance without or with disturbance (target point is (100, 100, 400))

External load (g)	Settling time (s)		Steady-state distance (mm)	
	No disturbance	Disturbance ($X_{dis}=5t$)	No disturbance	Disturbance ($X_{dis}=5t$)
0	0.16	0.16	0.135	0.439
200	0.09	0.11	1.03e-05	0.004

5.1 Static reaching task

In this section, we conducted reaching tasks for validating the static performance of GRLMAC. For the reaching task, the soft manipulator with an external load (0 g, 30 g, and 96 g) is controlled to move toward a target point under a water flow disturbance. As shown in Fig. 8, the control performance of GRLMAC is non-significantly affected by

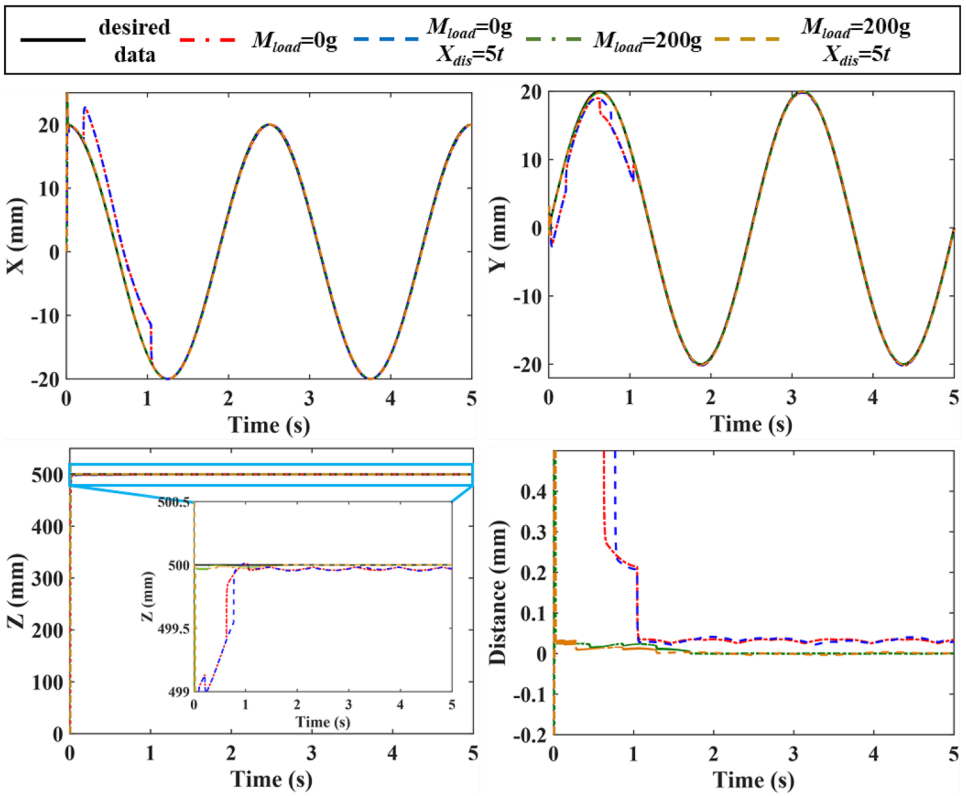


Fig. 6 Trajectory tracking tasks results under different external conditions

Table 3 Indicators of dynamic performance without or with disturbance (trajectory is (20sin(πt/1.25), 20cos(πt/1.25), 500))

External load (g)	Distance (mm)	
	No disturbance	Disturbance ($X_d = 5t$)
0	0.146	0.155
200	0.005	0.006

varying loads and flow disturbance, which demonstrates that the proposed controller has good robustness. Moreover, the settling step is less than 20 steps, and that means GRL has a high online learning efficiency by introducing the action selection guidance strategy. It is noteworthy that the data instability of position coordinates detected by the binocular camera has a significant impact on the stability and the accuracy of GRLMAC. Therefore, the

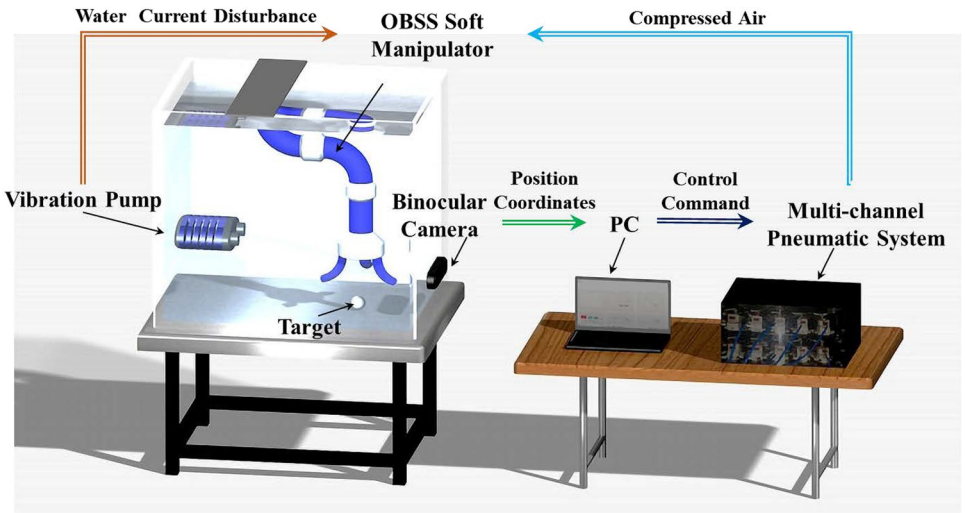


Fig. 7 The OBSS soft manipulator experiment system

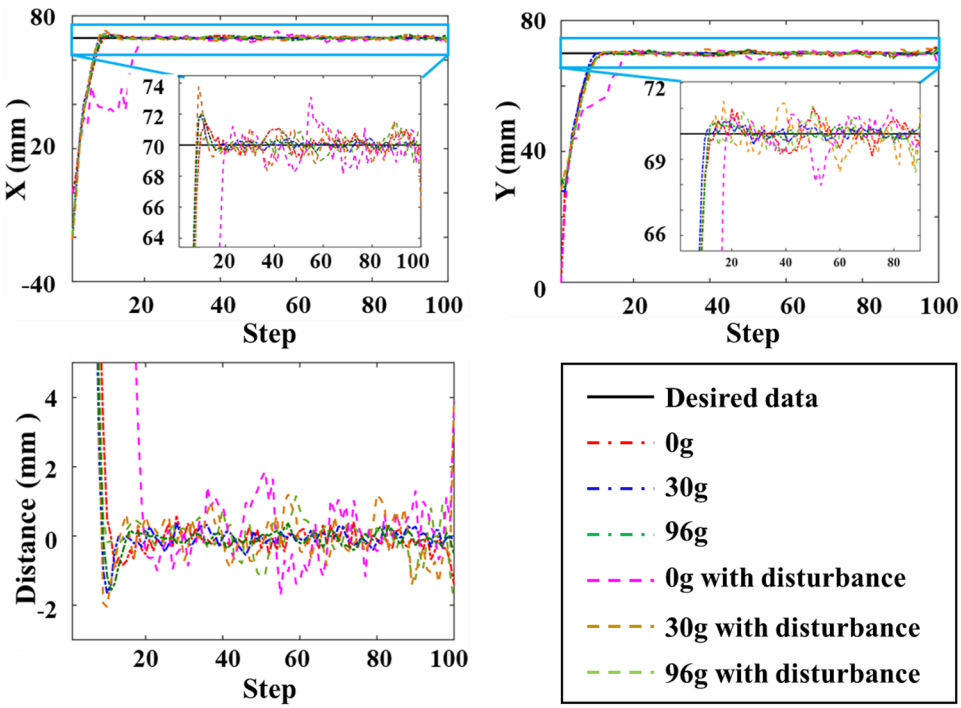


Fig. 8 Static reaching task results. For more details refer to Movie S1

Table 4 Indicators of static performance without or with disturbance

External load (g)	Settling step		Steady-state distance (mm)	
	No disturbance	Disturbance	No disturbance	Disturbance
0	16	20	0.485	1.273
30	12	17	0.304	1.021
96	16	17	0.252	0.735

steady-state error (shown in Table 4) is larger than the simulation results.

5.2 Dynamic trajectory tracking task

To verify the dynamic performance and the robustness of GRLMAC more systematically, we control the soft manipulator to track a square signal and a sin signal with or without a constant flow disturbance. Figures 9 and 10 illustrate the control performance of GRLMAC for trajectory tracking tasks. GRLMAC always has almost the same control performance for the soft manipulator, whether the flow disturbance exists or not (the change of error is about 1 mm, as shown in Table 5). This result validates the effectiveness and robustness of the proposed controller. However, the stability and the control accuracy of GRLMAC for the sine-wave signal are still affected by the measured data instability.

5.3 Grasping task

In this section, to validate grasping performance, we execute grasping tasks under a water flow disturbance. The objects (including a ping-pong ball, a sea cucumber, and a scallop) need to be grasped into a circle by the OBSS soft manipulator which is controlled via GRLMAC. The initial position of the soft gripper is set as shown in Fig. 11. For each grasp task, the soft manipulator takes 15 steps to move to the object. After reaching the object, the manipulator took 3 steps to complete the grasping-return-release the object. The task process takes 18 steps in total. As shown in Movie S3, the soft manipulator can autonomously grasp the objects with different sizes and weights into the circle under a flow disturbance based on the algorism of GRLMAC. The result demonstrates that the proposed controller has good robustness for the external disturbance. Moreover, the soft manipulator reaching the object only takes less than 15 steps, which illustrates that GRLMAC has a fast online learning ability.

Moreover, to demonstrate the characteristic of GRLMAC, the control performance comparison between GRLMAC and the other controllers mentioned in the introduction is provided in Table 6. According to comparison results, the proposed controller has a good online learning ability, which makes it has a better control performance than the offline learning controller.

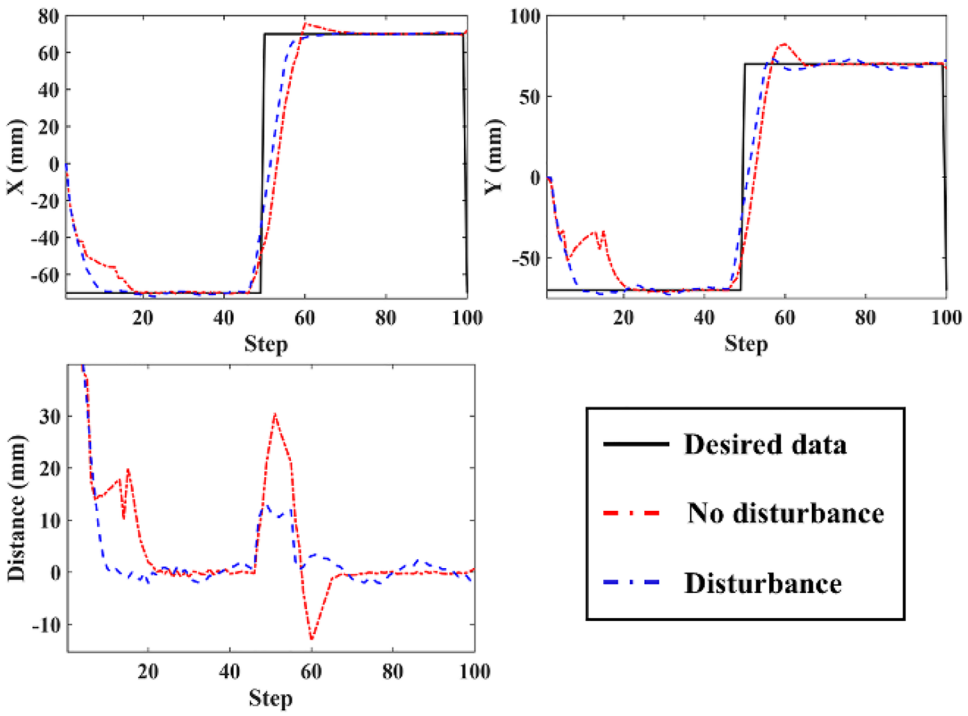


Fig. 9 Dynamic trajectory tracking task results for a square signal. For more details can refer to Movie S2

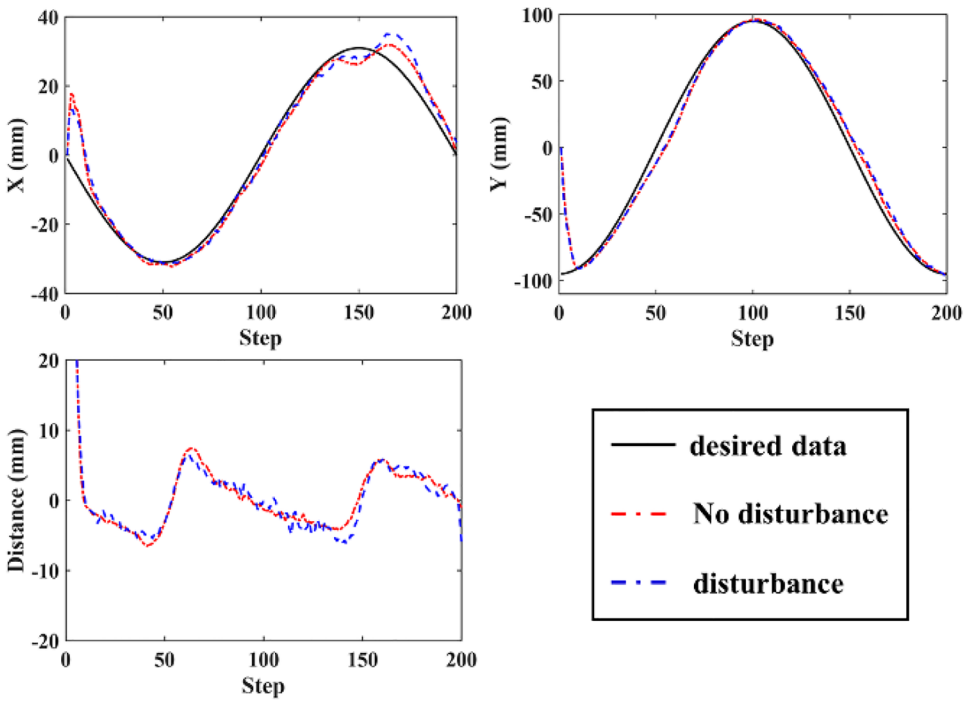


Fig. 10 Dynamic trajectory tracking task results for a sin signal. For more details refer to Movie S2

Table 5 Indicators of dynamic performance without or with disturbance

Trajectory	Distance (mm)	
	No disturbance	Disturbance
Square signal	1.315	2.579
Sin signal	3.474	4.383

By combining the OBSS soft manipulator with GRL-MAC, our soft manipulator can offer a promising option for high-performance and low-cost underwater manipulation systems for marine tasks. To validate the ability of the gripping tasks in a real-world underwater environment with influences of ocean current, water pressure, visibility, we constructed the OBSS soft manipulator with an ROV. We

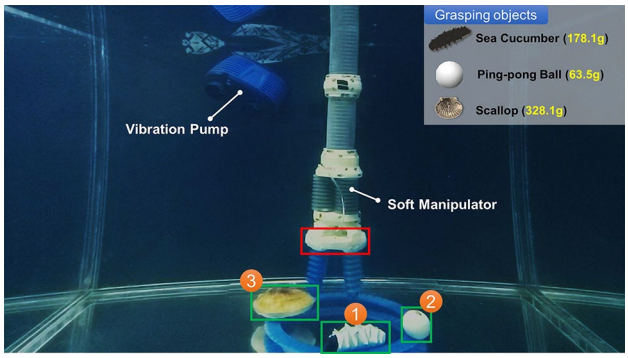


Fig. 11 Grasping task platform. Three kinds of objects (including sea cucumber, a ping-pong ball filled with bolts, and a scallop) need to be grasped into the circle in order. All the tasks are executed under a flow disturbance generated by a vibration pump

performed collecting seafood animals in the natural undersea environment through this robot (Fig. 12 and Movie S4). From the results, the low visibility in the offshore marine area and the strong and time-varying current in the open ocean increase the difficulty for grasping tasks. According to the above experiment results, we notice that the stability and the accuracy of the proposed controller are affected by measurement noise and non-stationary stochastic disturbance, which have effects on the action selection. The reason is that action space $\boldsymbol{A}$ is designed based on the length error of a chamber. Therefore, an online signal processing algorithm based on the optimal estimation theory is essential to obtain stable feedback data in the future.

6 Conclusion

In this study, a prediction model-based guided reinforcement learning adaptive controller (GRLMAC) is presented to control an OBSS soft manipulator, so that the soft manipulator can efficiently complete the grasping task in a water environment with external disturbances (e.g., currents, water pressure, external loads, etc.). In GRLMAC, an action selection guidance strategy based on the human experience is designed to direct the reinforcement learning method to choose an appropriate adjustment behavior for the FPM. This approach endows reinforcement learning with efficient online learning ability and avoids the offline training process. To verify the control performance of GRLMAC, both simulation and experiment platforms were established, and tracking and grasping tasks are conducted. Both simulation and experimental results show that the proposed controller has a good position control performance (the distance is about 1 mm for reaching tasks) and robustness (the error change is less than 1 mm) under different external loads and time-varying disturbance. Moreover, efficient online learning ability enables the manipulator to reach the target point just within a few steps (the settling step is about 20 steps), which is less time-consuming. The above results demonstrate the effectiveness of GRLMAC in the underwater grasping task. In the future study, we will analyze the effects of stochastic environmental disturbances for the grasping task, and then design a disturbance predictive policy and introduce it into ε-greedy policy to select the appropriate control action of the soft manipulator for water disturbances.

Table 6 Comparison of control performance

Signal	Controller	Load conditions	Settling time or step	Distance (mm)	Learning mode
Static signal	GRLMAC	96 g	16 steps	0.252	Online
	Ref. (George et al. 2018)	105 g	15.5 ± 3 s	22	Offline
Dynamic signal	GRLMAC	0 g	-	3.474	Online
	Ref. (Bruder et al. 2002)	0 g	-	4.42 ~1.63	Offine +Online

Fig. 12 Grasping task in the natural undersea environment. **a** Performing the grasping task in the offshore marine area with low visibility. **b** Performing the grasping task in the open-ocean with strong and time-varying. More details can refer to Movie S4

Acknowledgements Li Wen conceived the project. Hui Yang accomplished the control method design, simulations, grasping experiments, and analysis of data. Zheyuan Gong model the kinematics model of OBSS soft manipulator. Jiaqi Liu, Xi Fang, Shiqiang Wang, Xingyu Chen, and Shihan Kong established the underwater robot system and participated in the underwater grasping experiments. Li Wen and Hui Yang prepared the manuscript, and all authors provided feedback during subsequent revisions. The authors also thank sincerely the reviewers and editors for their very pertinent remarks that helped this article become clearer and more precise. This work was also supported by the National Science Foundation support projects, China (Grant No. 91848206, 92048302, 61822303, 61633004, 91848105), in part by the National Key R&D Program of China (Grant No. 18YFB1304600), and in part by the National Science Foundation support project, China (Grant No. 91848206, 62003014).

References

Best, C.M., Gillespie, M.T., Hyatt, P., Rupert, L., Sherrod, V., Killpack, M.D.: A new soft robot control method: using model predictive control for a pneumatically actuated humanoid. IEEE Robot. Autom. Mag. **23**(3), 75–84 (2016)

Bruder, D., Fu, X., Gillespie, R.B., Remy, C.D., Vasudevan, R.: Data-driven control of soft robots using koopman operator theory. IEEE Trans. Robot. (2020). https://doi.org/10.1109/TRO.2020.3038693

Bruder, D., Fu, X., Gillespie, R.B., Remy, C.D., Vasudevan, R.: Koopman-based control of a soft continuum manipulator under variable loading conditions https://arxiv.org/abs/2002.01407 (2020)

Bu, X.H., Yu, Q.X., Hou, Z.S., Qian, W.: Model free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems. IEEE Trans. Syst. Man Cybern. Syst. **49**(4), 677–686 (2019)

Chen, Z., Huang, F.H., Sun, W.C., Gu, J., Yao, B.: RBF neural network based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay. IEEE/ASME Trans. Mech. **25**(2), 906–918 (2020)

Fang, G., Wang, X.M., Wang, K., Lee, K.H., Ho, J.D.O., Fu, H.C., Fu, D.K.C., Kwok, K.W.: Vision-based online learning kinematic control for soft robots using local Gaussian process regression. IEEE Robot. Autom. Lett. **4**(2), 1194–1201 (2019)

George, T.T., Ansari, Y., Falotico, E., Laschi, C.: Control strategies for soft robotic manipulators: a survey. Soft Rob. **5**(2), 149–163 (2018)

George, T.T., Falotico, E., Renda, F., Laschi, C.: Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. IEEE Trans. Robot. **35**(1), 124–134 (2019)

Gong, Z.Y., Cheng, J.H., Chen, X.Y., Sun, W.G., Fang, X., Hu, K.N., Xie, Z.X., Wang, T.M., Wen, L.: A bio-inspired soft robotic arm: kinematic modeling and hydrodynamic experiments. J. Bionic. Eng. **15**(2), 204–219 (2018)

Gong, Z.Y., Chen, B.H., Liu, J.Q., Fang, X., Liu, Z.M., Wang, T.M., Wen, L.: An opposite-bending-and-extension soft robotic manipulator for delicate grasping in shallow water. Front. Robot. AI **6**, 26 (2019)

Gong, Z.Y., Fang, X., Chen, X.Y., Cheng, J.H., Xie, Z.X., Liu, J.Q., Chen, B.H., Yang, H., Kong, S.H., Hao, Y.F., Wang, T.M., Yu, J.Z., Wen, L.: A soft manipulator for efficient delicate grasping in shallow water: modeling, control, and real-world experiments. Int. J. Robot. Res. **40**(1), 449–469 (2020)

Hao, L.N., Yang, H., Sun, Z.Y., Xiang, C.Q., Xue, B.C.: Modeling and compensation control of asymmetric hysteresis in a pneumatic artificial muscle. J. Intel. Mat. Syst. Str. **28**(19), 2769–2780 (2017)

Ho, J.D.O., Lee, K.H., Tang, W.L., Hui, K.M., Althoefer, K., Lam, J., Kwok, K.W.: Localized online learning-based control of a soft redundant manipulator under variable loading. Adv. Robot. **32**(21), 1168–1183 (2018)

Hofer, M., Spannagl, L., D'Andrea, R.: Iterative learning control for fast and accurate position tracking with a soft robotic arm. https://arxiv.org/abs/1901.10187v3 (2019)

Hosovsky, A., Pitel, J., Zidek, K.: Analysis of hysteretic behavior of two-DOF soft robotic arm. MM Sci. J. **18**(1), 935–941 (2016)

Jiang, N.J., Zhang, S., Xu, J., Zhang, D.: Model-free control of flexible manipulator based on intrinsic design. IEEE/ASME Trans. Mech. (2020). https://doi.org/10.1109/TMECH.2020.3043772

Kirkpatrick, K., Valasek, J.: Reinforcement learning for characterizing hysteresis behavior of shape memory alloys. J. Aeros. Comp. Inf. Com. **6**(3), 227–238 (2009)

Kirkpatrick, K., Valasek, J., Haag, C.: Characterization and control of hysteretic dynamics using online reinforcement learning. J. Aerosp. Inf. Syst. **10**(6), 297–305 (2013)

Kurumaya, S., Phillips, B.T., Becker, K.P., Rosen, M.H., Gruber, D.F., Galloway, K.C., Suzumori, K., Wood, R.J.: A modular soft robotic wrist for underwater manipulation. Soft Rob. **5**(4), 399–409 (2018)

Li, S., Zhang, Y.N., Jin, L.: Kinematic Control of redundant manipulators using neural networks. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2243–2254 (2017)

Li, Z.J., Zhao, T., Chen, F., Hu, Y.B., Su, C.Y., Fukuda, T.: Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoid-like mobile manipulator. IEEE/ASME Trans. Mech. **23**(1), 121–131 (2018)

Liu, J.M., Xu, C., Yang, W.F., Sun, Y.Y., Zheng, W.W., Zhou, F.F.: Multiple similarly effective solutions exist for biomedical feature selection and classification problems. Sci Rep-UK **7**(10), 12830 (2017)

Liu, L.Q., Iacoponi, S., Laschi, C., Wen, L., Calisti, M.: Underwater mobile manipulation: a soft arm on a benthic legged robot. IEEE Robot. Autom. Mag. **27**(4), 12–26 (2020)

Mura, D., Barbarossa, M., Dinuzzi, G., Grioli, G., Caiti, A., Catalano, M.G.: A soft modular end effector for underwater manipulation: a gentle, adaptable grasp for the ocean depths. IEEE Robot. Autom. Mag. **25**(4), 45–56 (2018)

Palli, G., Moriello, L., Scarcia, U., Melchiorri, C.: An underwater robotic gripper with embedded force/torque wrist sensor. IFAC-PapersOnLine **50**(1), 11209–11214 (2017)

Pawlowski, B., Sun, J.F., Xu, J., Liu, Y.X., Zhao, J.G.: Modeling of soft robots actuated by twisted-and-coiled actuators. IEEE/ASME Trans. Mech. **24**(1), 5–15 (2019)

Robinson, R., Kothera, C., Wereley, N.: Control of a heavy-lift robotic manipulator with pneumatic artificial muscles. Actuators **3**(2), 41–65 (2014)

Shiva, A., Stilli, A., Noh, Y., Faragasso, A., Falco, I., De, G.G., Cianchetti, M., Menciassi, A., Althoefer, K., Wurdemann, H.A.: Tendon-based stiffening for a pneumatically actuated soft manipulator. IEEE Robot. Autom. Lett. **1**(2), 632–637 (2016)

Stilli, A., Wurdemann, H.A., Althoefer, K.: A novel concept for safe, stiffness-controllable robot links. Soft Rob. **4**(1), 16–22 (2017)

Sun, Z.Y., Song, B., Xi, N., Yang, R.G., Hao, L.N., Yang, Y.L., Chen, L.: Asymmetric hysteresis modeling and compensation approach for nanomanipulation system motion control considering working-range effect. IEEE Trans. Ind. Electron. **64**(7), 5513–5523 (2017)

Sutton, R.S.: Learning to predict by the methods of temporal difference. Mach. Learn. **3**(1), 9–44 (1988)

Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction, pp. 90–127. MIT Press, Cambridge (1998)

Teeples, B.C., Becker, K.P., Wood, R.J.: Soft curvature and contact force sensors for deep-sea grasping via soft optical waveguides. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain (2018)

Thérien, F., Plante, J.S.: Design and calibration of a soft multiple degree of freedom motion sensor system based on dielectric elastomers. Soft Rob. **3**(2), 45–53 (2016)

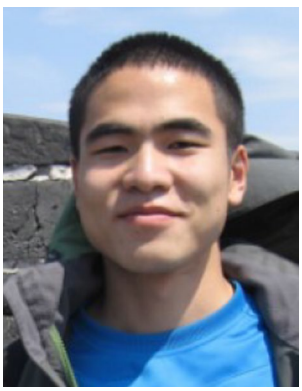
Trivedi, D., Rahn, C.D.: Model-based shape estimation for soft robotic manipulators: the planar case. J. Mech. Robot. **6**(2), 021005 (2014)
Vikas, V., Grover, P., Trimmer, B.: Model-free control framework for multi-limb soft robots. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany (2015)
Xie, Z.X., Domel, A.G., An, N., Green, C., Gong, Z.Y., Wang, T.M., Knubben, E.M., Weaver, J.C., Bertoldi, K., Wen, L.: Octopus arm-inspired tapered soft actuators with suckers for improved grasping. Soft Rob. **7**(5), 639–648 (2020)
Xu, F., Wang, H., Au, K.W.S., Chen, W.D., Miao, Y.Z.: Underwater dynamic modeling for a cable-driven soft robot arm. IEEE-ASME Trans. Mech. **23**(6), 2726–2738 (2018)
Zhang, Y.Y., Liu, J.K., He, W.: Vibration control for a nonlinear three-dimensional flexible manipulator trajectory tracking. Int. J. Control **89**(8), 1641–1663 (2016)
Zhang, J.J., Liu, W.D., Gao, L.E., Li, L., Li, Z.Y.: The master adaptive impedance control and slave adaptive neural network control in underwater manipulator uncertainty teleoperation. Ocean Eng. **165**(1), 465–479 (2018a)
Zhang, J.J., Liu, W.D., Gao, L.E., Zhang, Y.W., Tang, W.J.: Design, analysis and experiment of a tactile force sensor for underwater dexterous hand intelligent grasping. Sensors **18**(8), 2427 (2018b)
Zhuo, S.Y., Zhao, Z.G., Xie, Z.X., Hao, Y.F., Xu, Y.C., Zhao, T.Y., Li, H.J., Knubben, E.M., Wen, L., Jiang, L., Mingjie, L.M.J.: Complex multi-phase organohydrogels with programmable mechanics towards adaptive soft-matter machines. Sci. Adv. **6**(5), 1–10 (2020)

Hui Yang received the B.S. degree and M.S. degree in machinery design and manufacture from Liaoning Shihua University, Fushun, China in 2010 and 2013, respectively. He received a Ph.D. candidate at Northeastern University, Shenyang, China. His research interests include modeling and control the bionic manipulator actuated by artificial muscles. He is a student member of IEEE and the International Society of Bionic Engineering.

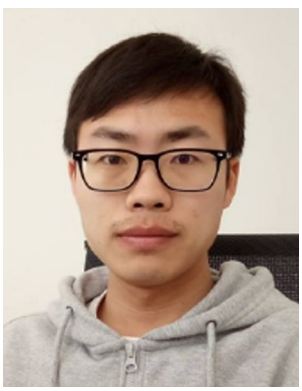
Jiaqi Liu received a B.S. degree in mechanical engineering from the School of Mechanical Engineering and Automation, Beihang University, Beijing, China, in 2018, where he is currently working toward the master's degree. His research interests include soft gripper, soft robotic arm, and stretchable sensor

Xi Fang received a B.E. degree in mechanical engineering from the School of Mechatronic Engineering, Beijing Institute of Technology, Beijing, China, in 2017. She received an M.E. degree in mechatronic engineering from Beihang University, Beijing, China, in 2020. Her research interests include soft robotics, bio-inspired robotics, and underwater robotics.

Xingyu Chen received a B.E. degree in electrical engineering and automation from the College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Chengdu, China, in 2015. He received a Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, deep learning, and underwater robotics.



Zheyuan Gong is currently a Ph.D. candidate at Toronto University, Toronto, Canada. His research interests include design, kinematic modeling, and control of the soft robotic arm.



Shiqiang Wang is currently a Ph.D. candidate at the School of Mechanical Engineering and Automation, Beihang University, Beijing, China. His research interests include modeling and control of the soft robotic arm.

Shihan Kong received a B.E. degree in automation from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2016. He is currently working toward a Ph.D. degree in control theory and control engineering at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include underwater robotics and underwater robotic vision.

Junzhi Yu received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2003. He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, IACAS. His research interests include intelligent robots, motion control, and intelligent mechatronic systems.

Li Wen received the B.E. degree in mechatronics engineering from the Beijing Institute of Technology, Beijing, China, in 2005, and the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China, in 2011. From 2011 to 2013, he was a Postdoctoral Fellow with George Lauder Laboratory, Harvard University. He is currently a Professor at Beihang University. His research interests include bio-inspired robotics, soft robotics, smart materials, and comparative biomechanics.