information

MDPI

Article

Position Control of Cable-Driven Robotic Soft Arm Based on Deep Reinforcement Learning

Qiuxuan Wu [1], Yueqin Gu [1], Yancheng Li [1], Botao Zhang [1], Sergey A. Chepinskiy [2,*], Jian Wang [1], Anton A. Zhilenkov [3], Aleksandr Y. Krasnov [2] and Sergei Chernyi [4]

[1] School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China; wuqx@hdu.edu.cn (Q.W.); gyq182060098@hdu.edu.cn (Y.G.); 172060045@hdu.edu.cn (Y.L.); billow@hdu.edu.cn (B.Z.); wangjian@hdu.edu.cn (J.W.)

[2] Faculty of Control Systems and Robotics, ITMO University, St. Petersburg 197101, Russia; krasnov.aleksander@gmail.com

[3] Department of Marine Electronics, State Marine Technical University, St. Petersburg 198262, Russia; zhilenkovanton@gmail.com

[4] Department of Integrated Information Security, Admiral Makarov State University of Maritime and Inland Shipping, St. Petersburg 198035, Russia; chernysg@gumrf.ru

* Correspondence: schepinskiy@itmo.ru

Received: 20 April 2020; Accepted: 4 June 2020; Published: 8 June 2020

Abstract: The cable-driven soft arm is mostly made of soft material; it is difficult to control because of the material characteristics, so the traditional robot arm modeling and control methods cannot be directly applied to the soft robot arm. In this paper, we combine the data-driven modeling method with the reinforcement learning control method to realize the position control task of robotic soft arm, the method of control strategy based on deep Q learning. In order to solve slow convergence and unstable effect in the process of simulation and migration when deep reinforcement learning is applied to the actual robot control task, a control strategy learning method is designed, which is based on the experimental data, to establish a simulation environment for control strategy training, and then applied to the real environment. Finally, it is proved by experiment that the method can effectively complete the control of the soft robot arm, which has better robustness than the traditional method.

Keywords: cable-driven; reinforcement learning; soft arm; data-driven; deep Q learning

1. Introduction

Compared with traditional rigid robots, soft robots are made of soft materials, which make them interact with the surrounding environment safely, so they are more suitable for the task of interaction with humans [1]. Their ability to flexibly deform and theoretically have an infinite degree of freedom allow them to perform better in confined space, but at the same time, it makes them difficult to model and control.

In recent years, many researchers have studied the control problem of soft arm. Wang et al. [2] used a model-based control method to analyze the dynamic model of a soft arm and design the shape controller based on the piecewise constant curvature. Phung et al. [3] used the experimental data to construct the kinematics model of the flexible multi-link manipulator, and used the neural network to approximate the inverse kinematics model of the system; a feedback control architecture was proposed to solve the vibration problem caused by the load change. Satheeshabu et al. [4] applied reinforcement learning to BR2 pneumatic soft manipulator position control, to verify the effectiveness of its control strategy. You et al. [5] proposed a model-free control method based on reinforcement

learning. The focus of research is on the learning of control strategies rather than the learning of physical models. The validity of the control strategy is verified in a two-dimensional planar multi-segment flexible manipulator. Jae In Kim et al. [6-14] first propose a new pneumatic vibration actuator that utilizes the nonlinear stiffness characteristics of a hyperelastic material, then present a reinforcement learning algorithm called adaptive soft actor-critic (ASAC), which provides efficient exploration strategy and is adaptive to various control tasks; it shows the possibility of reinforcement learning applied to various soft robots. T. G. Thuruthel et al. [15] present a data-driven approach to control the soft robotic manipulator; it not only has good system robustness, but also has good data efficiency. The work is carried out as a scientific project.

The above research focuses on the control method of the model-based soft robot arm, data-driven modeling method and reinforcement learning method of the flexible manipulator. The purpose of this paper is to combine data-driven modeling with reinforcement learning, to create an easy-to-implement and versatile control framework. This paper adopts the deep Q learning (DQN) algorithm [6]. Compared with the control strategy based on the mathematical model, the control strategy can be directly learned from experience, which effectively reduces the error introduced by the modeling process. However, in the learning process of using reinforcement learning to control the strategy, if it is directly carried out in the actual physical environment, not only will the driver execution process take a lot of time, but the resource consumption, such as the physical wear of the instrument, will also be increased. Therefore, it is very necessary to pre-train the control strategy in advance of the simulation environment.

So, based on the idea of being data-driven, this paper establishes a multi-layer perceptron (MLP) model through experimental data, so that reinforcement learning can be used as a pretraining for control strategies on MLP models. Compared with the actual robot training, simulation training has the advantages of fast execution speed, high efficiency in learning and not requiring one to participate in the training process manually [7]. Experiments show that this method can not only effectively achieve the expected goals, but is also very meaningful to reduce resource waste and improve training efficiency. Due to the lack of feedback of soft arm, firstly, vision is used as the feedback of the control system In the future, feedback will be implemented using the built-in sensor signal. Thus, the feasibility of the control method is proved.

2. System Architecture

2.1. Design of Cable-Driven Soft Arm

The soft arm used in this paper is shown in Figure 1. It is made of silica gel. Four cables are embedded in the soft arm. One end of the cable is connected to the servo connector, and the other end is connected with the fixed ring embedded in the soft arm. The arm is driven by four independent steering gears. When the steering gear rotates, the cable is contracted by the connector, to drive the fixing ring to make the soft arm move.

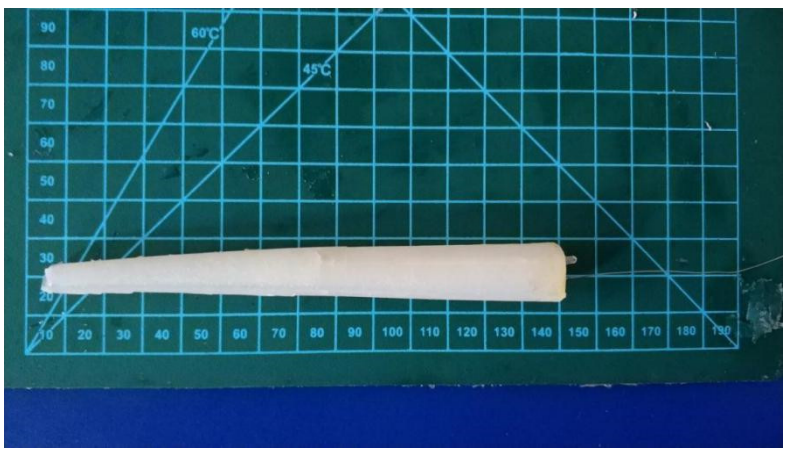


Figure 1. Design of cable-driven soft arm Table 1.

The pull wires are divided into two groups; one set of fixed rings is close to the end of the soft arm, and the other set of fixed rings is located in the middle of the soft arm, and the same set of fixed rings are symmetrically distributed in the center. The mass of the soft robot arm is 42.5 g, and the main parameters are shown in Table 1.

Table 1. Parameters of soft arm.

Subjects	Numbers	Unit
length	150	mm
Diameter (small)	6	mm
Diameter (big)	14	mm
Shore hardness	20	A°
density	1.1	g/cm^3
elongation	150	%
viscosity	2000	Pa·s

2.2. Overall Design of Control System

Due to the high dimensional continuous state space and nonlinear characteristics of the soft arm, many conventional control methods, such as PID, fuzzy control and synovial control, are not ideal. With the rapid development of artificial intelligence, model-independent reinforcement learning has opened up a new direction for robot control. Reinforcement learning learns control strategies directly from practice, without the need to control the precise mathematical model of the system, thus avoiding the modeling and parameter tuning process that relies on expert experience. For reinforcement learning, in order to improve its learning efficiency, most of the learning strategies are studied in the simulation environment, and then applied in the actual environment. However, it is very difficult to construct a virtual environment that is highly similar to the actual environment. It is not only necessary to build an environmental geometric model, but also to consider the dynamic model of the actual environment and other external disturbances, such as gravity or buoyancy.

Therefore, the soft arm control proposed in this paper consists of three steps. In the first step, the MLP model of the soft robot arm is established; firstly, the steering gear controlled is used to traverse the motion space of the entire soft robot arm in a fixed step size, and the soft arm shape corresponding to each motion is saved by using the camera, and then morphological characteristic data of the soft robot arm are extracted by image processing method. Finally, the MLP model of the flexible manipulator is obtained by controlling the steering gear parameters as the input and the soft arm shape characteristic data as the output training.

In order to extract the shape characteristics of the soft robot arm, firstly, six black feature points are marked on the soft robot arm, and then the soft robot arm image is acquired using a camera with a relatively fixed position. In order to eliminate the error caused by the camera, it is required to calibrate the camera, then perform image processing on the collected image, and after screening, 600 sets of available data are obtained. Finally, the MLP neural network is used to model the soft robot arm to obtain its simulation environment.

The second step is to complete the simulation training of DQN. Firstly, the initial position and the target position of the soft arm are randomly initialized in the range of motion space of the soft robot arm, and then the MLP model after training is used to forwardly propagate according to the input parameters of the current virtual steering gear, and the morphological characteristic data of the soft manipulator are obtained and provided to DQN. After DQN selects the action, it converts the action into steering gear control parameters and then feeds back to the MLP model and calculates the reward of the current action. The above process is repeated to realize DQN learning in the virtual environment.

The third step is the real environment experiment. The image data are collected by the camera, and then input into the DQN after feature extraction. Finally, the DQN directly outputs the motion and converts it into a steering gear control parameter to directly control the steering gear, and realize

the control of the soft robot arm. The overall process flow diagram of the system is shown in Figure 2.

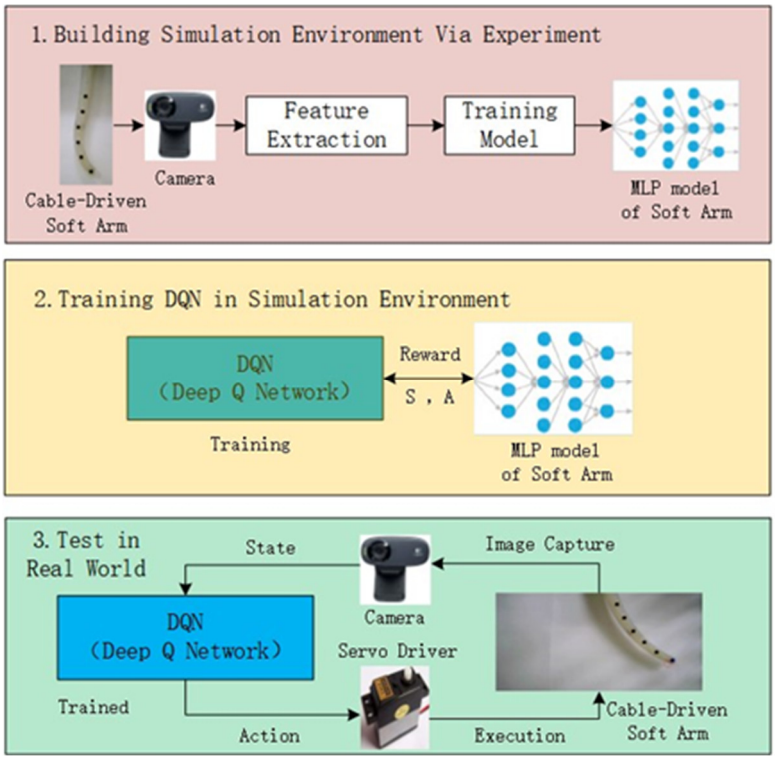


Figure 2. System overview.

3. MLP Model of Cable-Driven Soft Arm

Due to the nonlinear nature of the soft arm itself, modeling is difficult, and it is not easy to establish an accurate mathematical model. The multi-layer perceptron network can fit the nonlinear mapping function under the condition of nonlinear activation function. Therefore, the data-driven MLP neural network is used to model the soft arm [8]. Since the data are derived from the real soft arm, they can effectively reduce the error caused by neglecting some nonlinear factors in modeling by other mathematical methods.

The MLP model used in this paper is based on the python Keras library. The designed neural network model has four input neurons, which are the angle information of the servo 1 to the servo 4. In order to enable the network to fully learn the model features, seven hidden layers are set up, each layer containing 64 neurons; the number of output layers is the number of coordinates of the feature points of the soft arm, since a set of coordinates includes the abscissa and the vertical one coordinates the data into two dimensions, so the horizontal and vertical coordinates are flattened into one dimension of data as the output of the neural network, and the output layer has a total of 12 neurons. All neurons use tanh as the activation function, the mean square error function as the training loss function, and Adam as the optimizer.

Deep Q learning traditional Q learning is a dynamic programming method based on value iteration. The goal of the algorithm is to find an optimal strategy to maximize the total benefit when the agent makes decisions and moves accordingly. A decisive strategy can be defined as a Q function. The iterative update of the value of the Q function follows the Bellman equation and is given below:

$$Q_t(s_t, a_t) = r_t + \alpha\left(r_t + \gamma \max_a Q_{t+1}(s_{t+1}, a) - Q_t(s_t, a)\right) \quad (1)$$

where $Q(s,a)$ is Q function, r is the reward value, α is the learning rate, and γ is the attenuation coefficient.

Traditional Q learning uses a table to record value functions, but in practice, there are often a large number of states, or even a continuous state space. At this time, the traditional Q learning

method is difficult to learn effectively. In order to solve this problem, the traditional Q learning and deep learning are combined, and the deep neural network is used to approximate the Q function, so that Q learning can not only process the continuous state space, but also have generalization ability. It effectively improves the application range of traditional Q learning. So, this kind of algorithm is called deep Q learning. Algorithm 1 shows the DQN algorithm [9] framework used in this paper.

Algorithm 1. Event-Triggered Distributed Optimization Approach

```
For episode do
  reset start position and target position
  While not terminal do
  state, reward, terminal = ENV.observe
  action = DQN.chooseAction(state)
  ENV.executeAction(action)
  stateNext, reward, terminal = ENV.observe
  DQN.store(state, action, reward, stateNext)
  DQN.learn
  End while
  End for
```

The paper uses DQN as the soft arm control algorithm. The algorithm is based on the Markov decision process (MDP). If the next state of a discrete decision sequence is only related to the current state and has nothing to do with the past, then this sequence of cycles is called the Markov decision process. The Markov decision process is represented by four elements: state space, motion, state transition probability, and reward. The MDP element definitions for this task are given below.

3.1. State Design

In the soft robot arm control problem, the state space represents the set of system states in which the agent is located, mainly including the current state of the soft robot arm and the target position [10]. The current state includes two parts: the steering servo state and the current position attitude. Therefore, the state space of the system consists of the following three parts: the first part is the current soft arm shape parameter, which is composed of six coordinates' information; the second part is the current state of the steering gear driven by the soft arm, which consists of four servo parameters; the third part is the target position of the target soft arm, which is composed of two coordinates' information.

3.2. Action Design

An action represents a collection of actions that an agent can take, that is, an action space. In the soft arm problem, the original motion space is high-dimensionally continuous, such as the servo control frequency of the drive cable, and for the characteristics of the deep Q learning algorithm, the original motion space needs to be discretized. The soft robot arm is driven by four servos, and each servo can choose two actions: tighten or relax. However, the simultaneous execution of the same action on the steering gear in the symmetrical position must be mutually exclusive, so the steering gear is divided into two groups of defined actions according to the control position. The actions that can be taken by the soft arm steering gear are represented, so that group 1 is being pulled closer, group 1 is being relaxed, group 2 is being pulled closer, and group 2 is being relaxed.

3.3. Reward Design

In reinforcement learning, the reward function is defined as the effect of the environment feeding the agent to the action after taking some action [11], so the function design plays a very important role in the learning result. A good reward function speeds up the convergence of the intensive learning process, while a bad reward function slows down the learning speed and even

causes the reinforcement learning model to learn anything completely. The training end flag includes the following two types: one is that the position error of the soft arm is lower than the set threshold, and the other is that the current number of steps exceeds the limit of the allowable operation. In this experiment, when the soft robot arm reaches the maximum active boundary value, if the soft arm attempts to select an action that exceeds the specified limit position, the DQN will not perform any action and return the bonus value to a larger penalty value. This can effectively prevent the soft robot arm from exceeding the limited position during the movement (Algorithm 2).

Algorithm 2. Reward function

Input:	The end coordinates of the current soft arm X_c, the last round X_l and the target position X_t
Output:	The reward value R of the current state and the termination flag terminal of a single loop
1	for *sequence* do
2	for *episode* do
3	$Dis_c = \|X_c - X_t\|$, $Dis_l = \|X_l - X_t\|$, $Terminal = False$
4	else if $Dis_c < \epsilon$ then
5	$Terminal = True$, $R = 2$
6	else if $Dis_c - Dis_l > 0$ then
7	R = 1
8	else
9	$R = 0$
10	end for

4. Simulation and Experiment

This section will analyze the effects of DQN in soft arm control through simulation and experiment. It is hoped that the advantage of the soft-arm control based on reinforcement learning is verified by experiments: (1) DQN selects the optimal action sequence during control; (2) the system is robust to external disturbances. The hardware structure of the system is shown in Figure 3. It consists of five parts: soft arm, camera, computer, steering gear and servo controller. The computer device CPU used is Intel CORE i5 and the GPU is NVIDIA GTX1050.

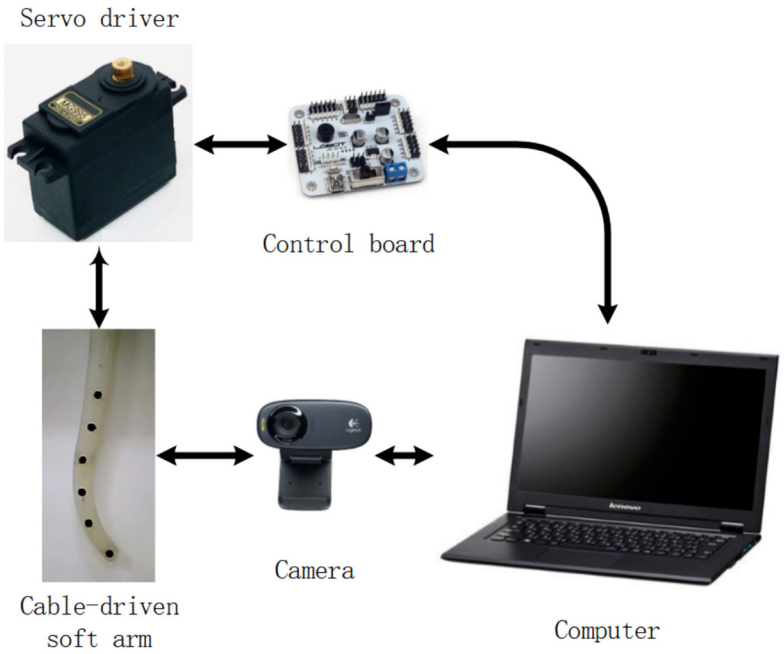


Figure 3. Hardware of system.

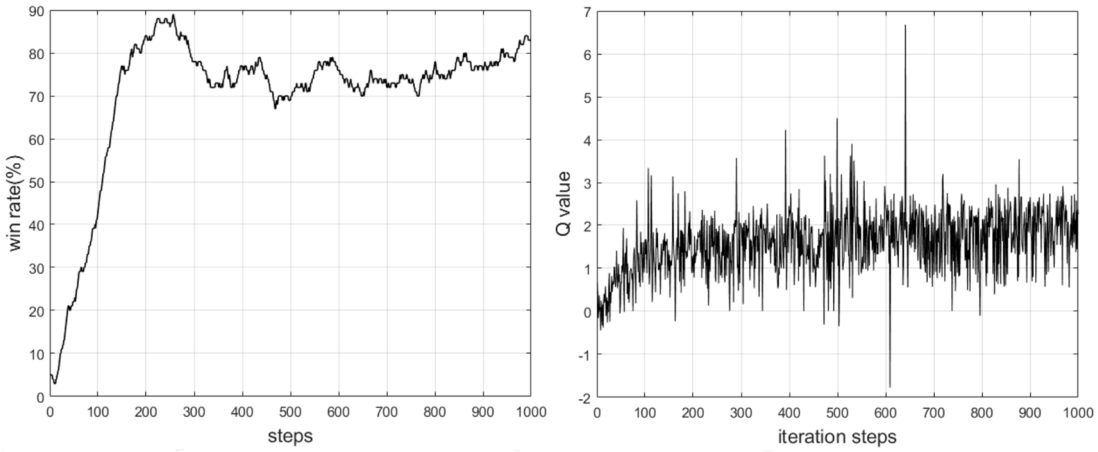
4.1. Train in Simulation Environment

The Q network in this experiment is a fully connected network containing seven hidden layers; each contains 50 neurons and a tanh activation function. In order to improve the robustness of the system, Gaussian noise is added as random interference in the x-direction and y-direction of the MLP model output of the soft robot arm during simulation training. Other parameters are shown in Table 2.

Table 2. Parameters of deep Q learning (DQN).

Subjects	Numbers
total steps	1000
Memory size	20,000
optimizer	RMSPropOptimizer
learning rate	0.0005
ε-greedy	0.4–0.9
Maximum steps in round	300

During each round of training, a starting point and an ending point are randomly selected from the training data set. The training is based on the greedy strategy. At the beginning, the random action is selected with a large probability, so that DQN enhances the exploration of the surrounding environment. The optimal action is beneficial to get rid of the local extremum and find the global optimal solution. The training results are shown in Figure 4 below.



(**a**) (**b**)

Figure 4. (**a**) Win rate; (**b**) The average Q value.

Among them, the success rate on Figure 4a is the ratio of the total number of rounds to the total number of rounds in the last 100 steps; Figure 4b shows the average Q value obtained per step and per round. It can be seen that as the number of learning increases, the success rate continues to increase, and the average Q value obtained per round is also increasing. This shows that DQN can effectively learn the control strategy of the soft robot arm in the simulation environment [12]. Next, the effectiveness and robustness of the control strategy will be verified on the actual physical prototype.

4.2. Test in Real World

Position control. Experiment: In the physical environment experiment, the control object is the soft robot arm used in MLP modeling. Compared with the simulation experiment, the control

strategy does not need any change. It only needs to convert the action signal into a control signal and send it to the servo controller to complete the actual control. Figure 5 is the experimental environment.

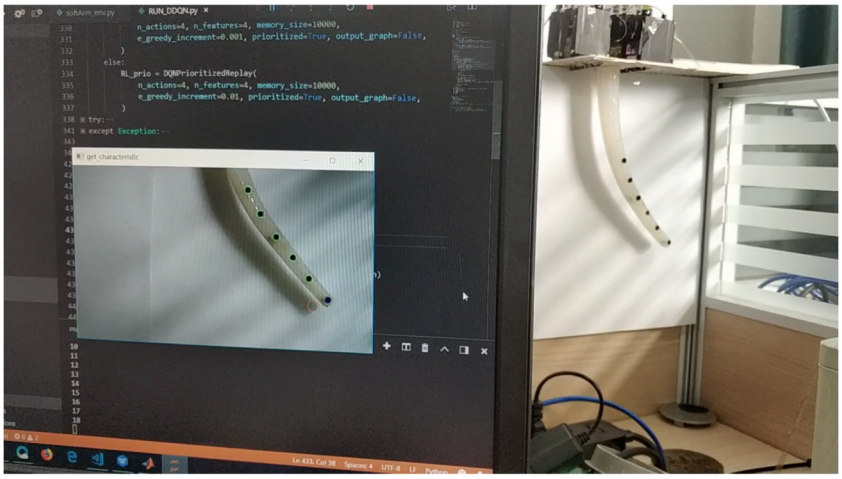
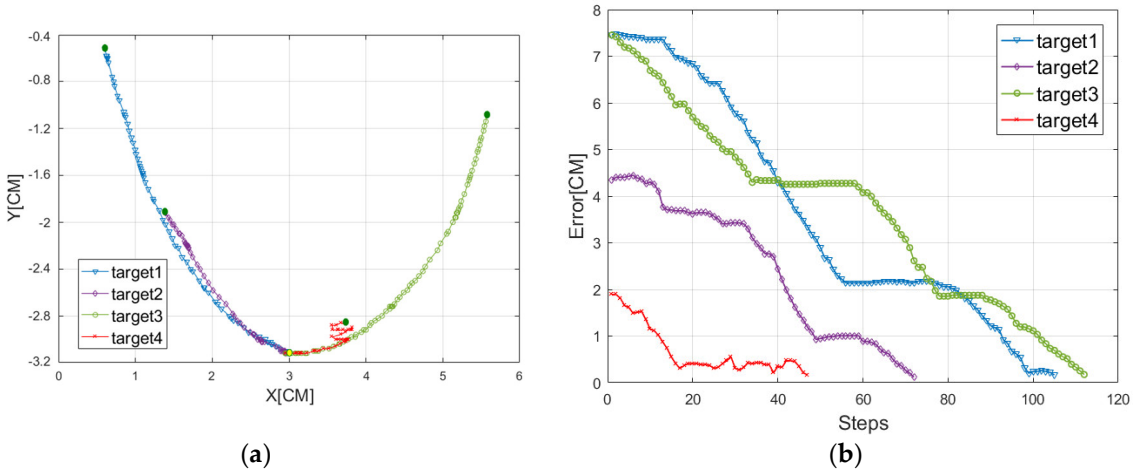


Figure 5. Experiment environment.

In the experiment, the starting point is fixedly set to drive the initial state of the steering gear, and then the target point is randomly selected in its motion space, and the condition for determining the suspension is the same as the simulation experiment. The experimental results are shown in Figure 6. Figure 6a is the end of the soft robot arm, and Figure 6b is the error of the end of the soft arm from the target. It can be concluded from the figure that the distance between the end of the soft robot arm and the desired target is continuously reduced as the action of the soft robot arm is performed, and finally the expected accuracy is achieved.



(**a**) (**b**)

Figure 6. The End of Soft arm: Motion Trajectory and Error.

Experiments demonstrate the effectiveness of the control strategy. The error curve in Figure 6b is stepped. Due to the soft arm being in a relaxed state, the relaxed pull wire is trying to shrink. First, we have to go through a process from relaxation to tightening, which leads to the soft arm being static during this time, but it is actually performing the action, and DQN can effectively control the stage.

Experiment with loads. Similar to the traditional robot arm for grabbing or placing tasks, some artificial perturbations have been added to verify the robustness of the control strategy. The load is copper-core wire hoisted at the end of the soft arm, which not only causes the soft arm to be subjected to a downward force, but the load can also periodically and laterally shaken during the movement of the soft arm, which increases the difficulty of control. Different from the traditional rigid arm, due to the lack of rigidity, the working space of the soft arm will decrease with the increase of the load,

which will be verified in the experiment. In the experiment, the weight of per unit load is about 5.2 g, and the length of the suspension wire is 110 mm.

The experimental results are shown in Figures 7–9. Due to the lack of rigidity, with the load increases from 1 unit to 3 units, the deformation of the manipulator becomes larger and larger. The end position of the manipulator deviates downward and outward relative to the no-load, which results in the success rate of 100 tests per unit load reduced by 10%, which is 87%, 75% and 66%. This shows that the effective working space of the arm decreases with the increase of load, but in a certain range, the algorithm can still control the flexible arm to reach the designated position, which shows that the algorithm has robustness.

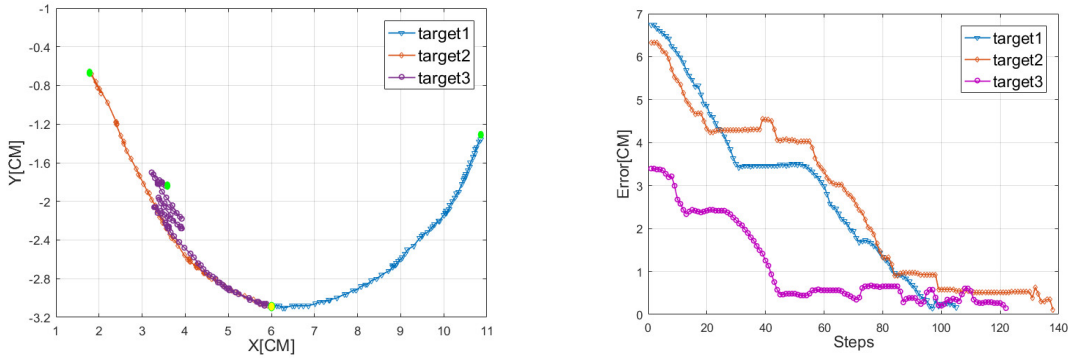


Figure 7. The End of Soft arm: Motion Trajectory and Error for Load = 1.

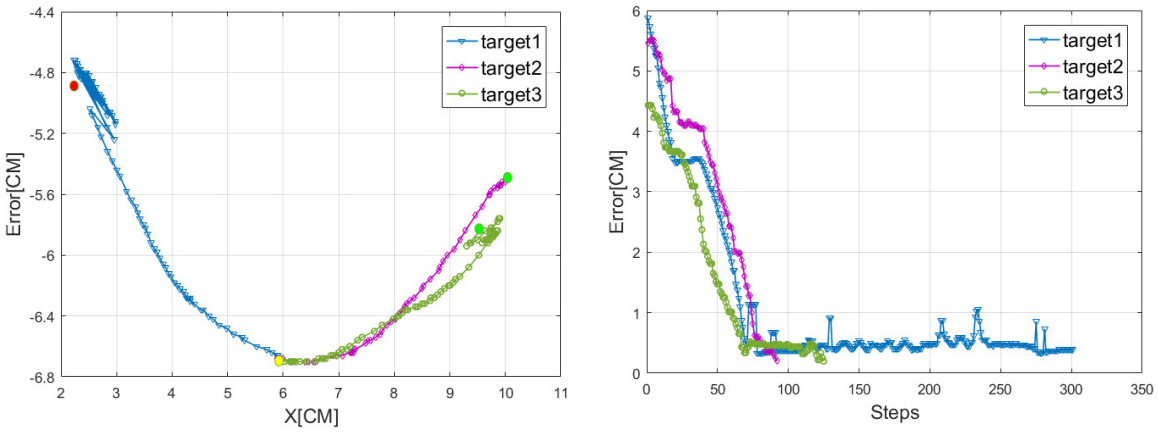


Figure 8. The End of Soft arm: Motion Trajectory and Error for Load = 2.

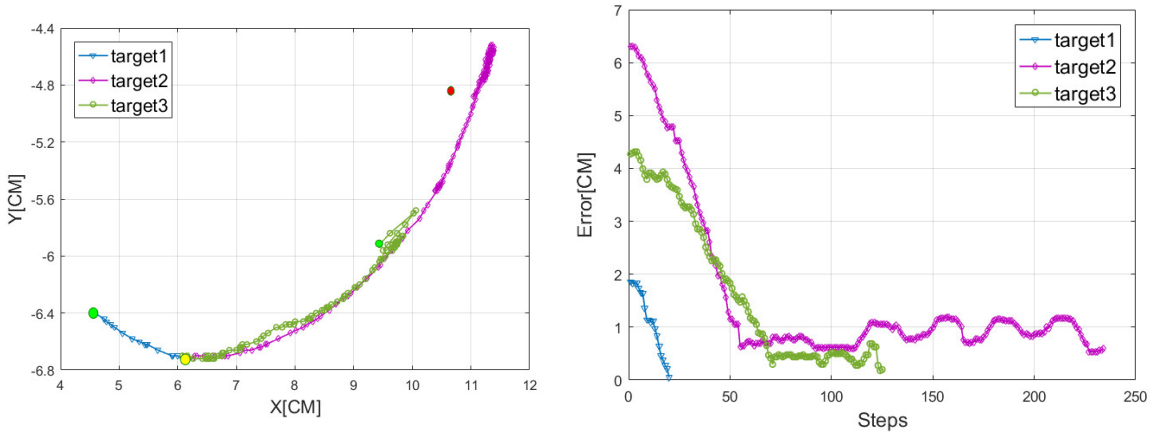


Figure 9. The End of Soft arm: Motion Trajectory and Error for Load = 3.

By observing the experimental process, it can be concluded that the failed part gradually decreases from above the workspace as the load increases. Experiments show that although the workspace of the system is decreased when carrying the load, the control strategy is still partially effective.

5. Conclusions

In this paper, the end position control of the soft robot arm is realized, based on the data driven model and deep Q learning. Firstly, the morphological characteristics data of the driving motor and the soft robot arm are obtained through an experiment, and then the MLP model is built according to the data, and the simulation model of the actual soft robot is obtained. Then, the deep Q learning algorithm is applied to train the control strategy of the soft robot arm in the simulation environment. Finally, the effectiveness and robustness of the algorithm are verified by real environment experiments. Compared to traditional methods, this method does not need complex robot dynamics modeling. Compared with the deep Q learning algorithm that directly uses images as input, data-driven modeling reduces the number of convergences of deep Q learning from tens of thousands of times to several thousand times, and is successfully applied to soft robot arm entities [13]. Future work includes considering the use of a confidence strategy gradient (DDPG) or A3C algorithm that is more suitable for robot continuous control, and then extending the control range from a two-dimensional plane to a three-dimensional space.

Author Contributions: Conceptualization, Q.W., Y.G., Y.L. and B.Z.; methodology, Q.W., Y.G. and B.Z.; generation of results, Q.W., Y.G., Y.L., B.Z. and A.Y.K.; validation, Q.W., S.A.C., J.W., A.A.Z. and Y.G.; formal analysis, Q.W., Y.G., Y.L. and A.A.Z.; resources, J.W.; writing—original draft preparation, Q.W., Y.G., J.W. and B.Z.; writing—review and editing, A.A.Z. and S.C.; project administration, B.Z., S.A.C.; funding acquisition, B.Z., J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This material was based upon the work supported by Key Projects of Science and Technology Plan of Zhejiang Province (2019C04018).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rus, D.; Tolley, M.T. Design, fabrication and control of soft robots. *Nature* **2015**, *521*, 467–475.
2. Anton, Z.; Oleg, K.; Sergei, C.; Anatoliy, N.; Sergei, S. Numerical Methods for Solving the Problem of Calibrating a Projective Stereo Pair Camera, Optimized for Implementation on FPGA. *Procedia Comput. Sci.* **2020**, *167*, 2229–2235, doi:10.1016/j.procs.2020.03.275.

3. Phung, A.S.; Malzahn, J.; Hoffmann, F.; Bertram, T. Data Based Kinematic Model of a Multi-Flexible-Link Robot Arm for Varying Payloads. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Phuket, Thailand, 7–11 December 2011; pp. 1255–1260.
4. Satheeshbabu, S.; Uppalapati, N.K.; Chowdhary, G.; Krishnan, G. Open Loop Position Control of Soft Continuum Arm Using Deep Reinforcement Learning. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
5. You, X.; Zhang, Y.; Chen, X.; Liu, X.; Wang, Z.; Jiang, H.; Chen, X. Model-Free Control for Soft Manipulators Based on Reinforcement Learning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2909–2915.
6. Ivanov, A.; Zhilenkov, A. The Use of IMU MEMS-Sensors for Designing of Motion Capture System for Control of Robotic Objects. In Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow/St. Petersburg, Russia, 29 January–1 February 2018.
7. Snderhauf, N.; Brock, O.; Scheirer, W.; Hadsell, R.; Fox, D.; Leitner, J.; Corke, P. The limits and potentials of deep learning for robotics. *Int. J. Robot. Res.* **2008**, *37*, 405–420.
8. Wang, H.; Yang, B.; Liu, Y.; Chen, W.; Liang, X.; Pfeifer, R. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Trans. Robot.* **2017**, *35*, 124–134.
9. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
10. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
11. Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; De Freitas, N. Dueling network architectures for deep reinforcement learning. *arXiv* **2015**, arXiv:1511.06581.
12. Zhilenkov, A.; Chernyi, S.; Sokolov, S.; Nyrkov, A. Intelligent autonomous navigation system for UAV in randomly changing environmental conditions. *J. Intell. Fuzzy Syst.* **2020**, 1–7, doi:10.3233/jifs-179741.
13. Zhang, F.; Leitner, J.; Milford, M.; Upcroft, B.; Corke, P. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv* **2015**, arXiv:1511.03791.
14. Kim, J.I.; Hong, M.; Lee, K.; Kim, D.; Park, Y.; Oh, S. Learning to Walk a Tripod Mobile Robot Using Nonlinear Soft Vibration Actuators With Entropy Adaptive Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2317–2324, doi:10.1109/LRA.2020.2970945.
15. Feng, N.; Wang, H.; Hu, F.; Gouda, M.; Gong, J.; Wang, F. A fiber-reinforced human-like soft robotic manipulator based on sEMG force estimation. *Eng. Appl. Artif. Intell.* **2019**, *86*, 56–67, doi:10.1016/j.engappai.2019.08.016.