applied sciences

MDPI

Article

Robotic Peg-in-Hole Assembly Strategy Research Based on Reinforcement Learning Algorithm

Shaodong Li [1], Xiaogang Yuan [1] and Jie Niu [2,3,4,*]

[1] Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, Guangxi University, Nanning 530004, China
[2] Hunan Province Key Laboratory of Intelligent Live Working Technology and Equipment (Robot), Changsha 410100, China
[3] Live Inspection and Intelligent Operation Technology State Grid Corporation Laboratory, Changsha 410100, China
[4] State Grid Corporation of China, Beijing 100031, China
* Correspondence: niu.jie.robot@hotmail.com

Abstract: To improve the robotic assembly effects in unstructured environments, a reinforcement learning (RL) algorithm is introduced to realize a variable admittance control. In this article, the mechanisms of a peg-in-hole assembly task and admittance model are first analyzed to guide the control strategy and experimental parameters design. Then, the admittance parameter identification process is defined as the Markov decision process (MDP) problem and solved with the RL algorithm. Furthermore, a fuzzy reward system is established to evaluate the action–state value to solve the complex reward establishment problem, where the fuzzy reward includes a process reward and a failure punishment. Finally, four sets of experiments are carried out, including assembly experiments based on the position control, fuzzy control, and RL algorithm. The necessity of compliance control is demonstrated in the first experiment. The advantages of the proposed algorithms are validated by comparing them with different experimental results. Moreover, the generalization ability of the RL algorithm is tested in the last two experiments. The results indicate that the proposed RL algorithm effectively improves the robotic compliance assembly ability.

Keywords: reinforcement learning; robot compliance control; peg-in-hole assembly; admittance controller

Citation: Li, S.; Yuan, X.; Niu, J. Robotic Peg-in-Hole Assembly Strategy Research Based on Reinforcement Learning Algorithm. *Appl. Sci.* **2022**, *12*, 11149. https://doi.org/10.3390/app122111149

Academic Editor: Alessandro Gasparetto

Received: 14 August 2022
Accepted: 31 October 2022
Published: 3 November 2022

1. Introduction

Recently, research on the complex tasks performed by robots has become a hot topic. However, for the peg-in-hole assembly task, most research is implemented through teaching or simple programming, and is limited to structured environments [1]. In fact, when encountering an unstructured environment, the assembly success rate, efficiency, and generalization ability of traditional methods cannot meet expectations.

In general, there is a small clearance between the peg and the hole in the assembly task, leading to frequent contact. Even a few motions can result in excessive contact force, which can lead to assembly failure because of jamming and wedging [2]. Therefore, compliance control is necessary to avoid the above cases and improve the assembly effects [3]. Compliance control can give robots compliance ability, and is divided into passive and active compliance. In passive compliance control, the robot relies on energy storage elements, such as springs and damping, to generate natural compliance [4]. Passive compliance has a low control complexity, while system stiffness is poor due to utilizing energy storage elements. In active compliance control, the robot actively adapts to the environment based on a control strategy through the use of force sensor perception [5], and has better flexibility than passive compliance. Thus, active compliance is more widely researched and applied. Shimizu et al. [6,7] proposed an effective design approach for an admittance control system for assembling generic polyhedral parts despite various uncertainties. Nicky et al. [8]

proposed a nested admittance/impedance control strategy, which could allow to adjust for large misalignment errors between parts that needed be assembled. However, the above research is based on constant parameters, which is not suitable for unstructured environments. Fortunately, variable admittance control can solve this problem.

Variable admittance control can understand and adapt to unstructured environments, thereby, the success rate, efficiency, and generalization ability of the assembly can be improved. Wu et al. [9] proposed a variable admittance control based on an iterative learning algorithm for manufacturing the assembly line. Wu et al. [10] proposed a variable admittance control that could effectively improve the phenomenon of reverse acceleration mutation and contact bounce of robots in compliance assembly. In [11], the proposed controller had a variable stiffness to make the peg-in-hole task easier, but could not meet expectations in high-precision tasks. Nevertheless, the effect of the above approaches depended on the model's accuracy, with the still many limitations in practical applications making it difficult to build a precise model.

In addition to the above methods, reinforcement learning (RL), an important branch of machine learning, has become a promising method for solving unstructured tasks [12]. A variable impedance controller is adopted based on the fuzzy Q-learning algorithm to yield compliant behavior from the robot during the hole insertion process [13]. Nevertheless, the above approach outputs the discrete actions by discretizing the action space, which inevitably affects the learning effect of the algorithm. Deep reinforcement learning (DRL) [14] can solve the continuous-state input problem, and is widely used in high-dimensional state–action spaces. The impedance policies for small peg-in-hole tasks are learned using the DRL algorithm [15]. The robot learning the skill of precise insertion is realized with the DQN, but only implemented on 4-DOF [16]. A general learning-based algorithm, DDPG, is proposed to perform different assembly tasks well, but it requires some prior knowledge [17]. We noticed that the DRL improved the assembly effect of the robot in unstructured environments, yet led to massive data requirements during the training process. Many recent methods focus on the above problems [18–20]. Related work [18] has demonstrated that the action space reformulation can effectively reduce data requirements. Since the admittance controller can adapt to environmental contact forces, the RL method with an admittance action space can obtain the dynamic relationship between force and motion to realize continuous assembly tasks [19,20]. Thus, we define the admittance parameter identification process as a MDP problem and solve it with the SARSA and DDPG algorithms, respectively. This article proposes methods of admittance parameter identification using the SARSA and DDPG algorithms to realize variable admittance control to improve the assembly effects. Furthermore, a fuzzy reward system is established to evaluate the action–state value to solve the complex reward establishment problem. Finally, in a peg-in-hole assembly task, the proposed methods are verified and the results of different methods are compared.

The remainder of the article is organized as follows: Section 2 describes the mechanism of the compliance assembly and proposes variable admittance control methods based on the SARSA and DDPG algorithms. Section 3 validates the effectiveness of the proposed methods and discusses the experimental results. Section 4 summarizes the results of the current work and discusses future research directions.

2. Materials and Methods

This section describes the mechanism analysis of compliance assembly, and proposes the RL algorithm to realize variable admittance control.

2.1. Mechanism Analysis of Compliance Assembly

2.1.1. Peg-in-Hole Assembly Task

In general, the process of the peg-in-hole assembly mainly includes three phases: the approach, search, and insertion [2], where the insertion phase is the most important stage in the assembly process. Meanwhile, the circular peg is the most common assembly object.

Therefore, this article focuses on solving the insertion phase problem of the circular peg assembly task.

As shown in Figure 1, the insertion phase can be divided into one-point and two-point contact according to the contact state. In two-contact states, jamming or wedging can easily occur due to lateral and angular errors, even though the motion is small. Thus, to avoid the object's surface being scratched through jamming or wedging, admittance control was used to realize the compliance control in the robotic assembly. Furthermore, to describe the motion of the peg, the coordinate system O-XYZ was established, as shown in Figure 1. Obviously, in the circular peg assembly task, only 5 motion dimensions were required, including the translation along the *X*-axis, *Y*-axis, and *Z*-axis and the rotation around the *X*-axis and *Y*-axis. In the insertion process, we were always eager to reduce contact force by translating or rotating the peg.

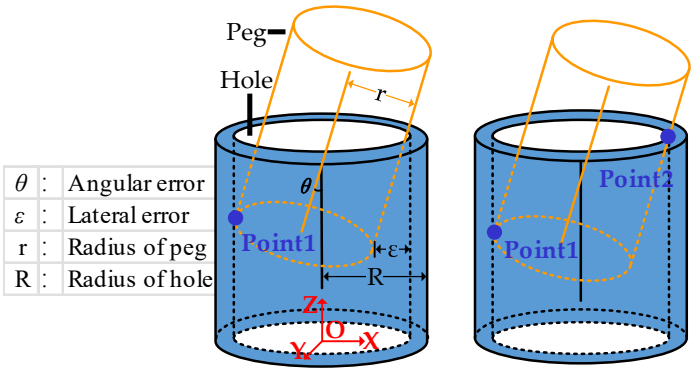

Figure 1. Contact states of peg-in-hole assembly in insertion phase.

2.1.2. Admittance Model

The admittance model was used to establish the dynamic relationship between the motion and the force. Based on the admittance model, the robot could adjust the position and pose according to external forces, avoiding excessive contact that could potentially cause jamming or wedging between the peg and hole. To facilitate the description, a one-dimensional admittance model was described in Figure 2, and could be extended to a multidimensional model for robot control using a similar method. The admittance controller could be established as follows:

$$\frac{\Delta X(s)}{\Delta F(s)} = \frac{1}{M_d s^2 + B_d s + K_d} \tag{1}$$

where M_d, B_d, and K_d represent, respectively, the virtual inertia, virtual damping, and virtual stiffness of the robot, ΔF is a deviation between the environmental force and the force expected, and ΔX is the position increment. To realize the robot control, the second-order difference expression of admittance controller could be obtained after the discretization of Equation (4) with the bilinear transformation, and could be expressed as follows:

$$X(k) = \frac{1}{A}[T^2F(k) + 2T^2F(k-1) + T^2F(k-2) - BX(k-1) - CX(k-2)] \tag{2}$$

where $A = 4M_d + 2B_dT + K_dT^2$, $B = -8M_d + 2K_dT^2$, $C = 4M_d - 2B_dT + K_dT^2$, and T represent the sampling periods. Through the simulation analysis of the admittance controller, we determined the value range of the admittance model parameters.

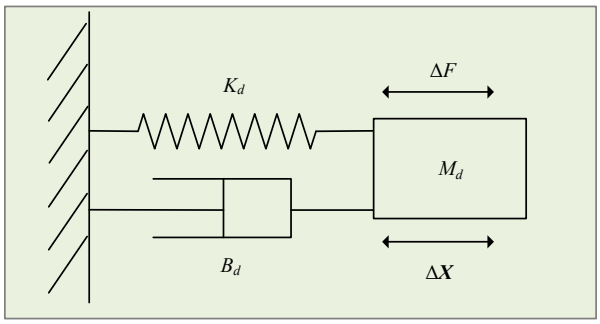

Figure 2. One-dimensional admittance model.

2.2. Variable Admittance Control Based on RL Algorithm

2.2.1. MDP Model for Variable Admittance Control

In the peg-in-hole assembly task using compliance control, the output of the admittance controller was the increment in the robotic pose, which highly depended on the model's parameters. Therefore, the assembly effects could be increased by actively adjusting the model parameters to adapt to changes in the current environment. We formulated the parameter identification for the variable admittance control as a MDP problem, including the current environment state s_t, the agent action a_t, reward r_t, and the next environment state s_{t+1}. The RL process started with an interaction between an agent and a given environment. In each time step, the agent performed an action a_t based on the current state s_t and policy π; then, the agent reached a new state s_{t+1} and obtained a reward r_t in the form of feedback. The RL process is shown in Figure 3. The reward could help the agent to find an optimal set of action methods to maximize the cumulative reward for the total assembly process. In the admittance parameter identification process, the discrete or continuous parameters did not affect the continuous control of the robot. Thus, both the SARSA and DDPG algorithms could be used to solve this MDP problem, and the interaction process could be described such as that in Figure 3. SARSA is a value-based method and has less design complexity, but requires the problem to be discrete. DDPG is a policy-based method, and introduces a neural network to handle continuous and more complex types of problems, while the design is more complex. Therefore, this work first utilized SARSA learning to realize the discrete admittance parameter identification. Then, the DDPG algorithm was used to realize the continuous admittance parameter identification to further improve the assembly effect.

2.2.2. Admittance Parameter Identification via SARSA

The discrete parameter identification did not affect the controller's stability and safety under the appropriate parameters set; therefore, SARSA learning could be used in this task. In each step of the insertion phase, the state and allowable range of motion were greatly different under different depths. Thus, the insertion depth h was used as the environment state s_t, and the stiffness parameters were used as the action output a_t. Moreover, the ε-greed policy was adopted as the action selection strategy, and the qualification trace function was introduced to improve the algorithm learning efficiency. The qualification trace function could be expressed as:

$$e_t(s,a) = \begin{cases} \lambda e_{t-1}(s,a) + \omega, & s = s_t \text{ and } a = a_t \\ \lambda e_{t-1}(s,a), & s \neq s_t \text{ and } a \neq a_t \end{cases} \quad (3)$$

where λ represents the decay factor, ω represents the positive weight, and $s = s_t$ and $a = a_t$ represent the selected state–action. After the input state s_t and output action a_t were determined, the qualification trace function $e_t(s,a)$ was updated, where the selected state–action increased the corresponding weight, and the others decayed proportionally. Then, the state–action value $Q_t(s,a)$ update could be expressed as:

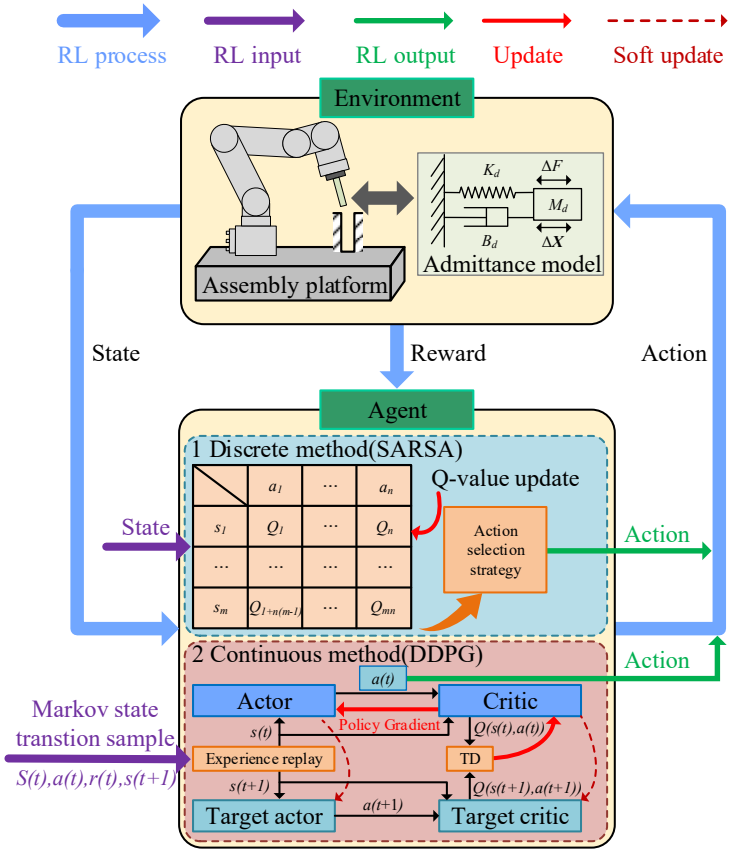

Figure 3. Variable stiffness control strategy based on RL algorithm.

$$\begin{cases} Q_{t+1}(s,a) \leftarrow \begin{cases} Q_t(s_t,a_t) + \delta_t e_t(s,a), & s = s_t \text{ and } a = a_t \\ Q_t(s,a) e_t(s,a), & s \neq s_t \text{ and } a \neq a_t \end{cases} \\ \delta_t = \alpha[r_t + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)] \end{cases} \tag{4}$$

where δ_t represents the time difference error for the selected state–action, α represents the learning rate, γ represents the discount factor, and r_t represents the reward. In the learning process, we were eager to obtain the optimal strategy to minimize the adjustment steps and maximize the reward. The reward function depended on the process reward R_t and failure punishment R_p, and could be expressed as follows:

$$r_t = R_t + R_p \tag{5}$$

where R_t and R_p are both negative. In the assembly task, when there was excessive contact force or too many adjustment steps, the task failed and was punished. In the insertion process, the process reward R_t highly depended on the insertion depth h and the displacement in each step Δh, which had a nonlinear relationship. Thus, a fuzzy reward method was established to evaluate the state–action value. At the same insertion depth h, the bigger the displacement in each step Δh, the higher the process reward R_t. When the displacement in each step Δh was the same, the greater the insertion depth h at the previous moment, the higher the process reward R_t. The training process based on the SARSA algorithm is shown in Figure 4.

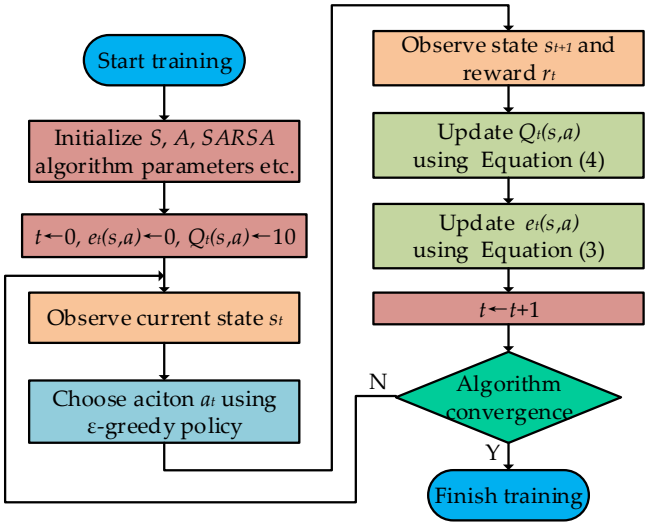

Figure 4. The training process of SARSA algorithm.

2.2.3. Admittance Parameter Identification via DDPG

For the admittance parameter identification, using the SARSA algorithm required discrete states and action spaces, which would affect the performance of the learning policy. In contrast to the SARSA algorithm, the DDPG can solve more complex and high-dimensional problems. Obviously, the contact force/torque, peg pose, and insertion depth are very important pieces of information in the assembly task, which can reflect on the current state. Thus, the environment state s_t was set to an 8-dimensional vector in the DDPG-based method, and could be defined as follows:

$$s_t = [F_x, F_y, F_z, M_x, M_y, \theta_x, \theta_y, h] \tag{6}$$

where F_x, F_y, F_z and M_x, M_y are the force and torque data from the sensor, θ is the Euler angle to describe the pose, h is the insertion depth, and the subscripts x, y, and z denote the axis of the robot's base coordinates. We noticed that there was only a need for 5-dimensional motion control in the circular peg assembly task, where the displacement and rotation along the X-axis and Y-axis were similar. Therefore, the output action a_t was set to a 3-dimensional vector defined as:

$$a_t = [K_1, K_2, K_3] \tag{7}$$

where K_1 is the stiffness parameter to control the displacement along the X-axis and Y-axis, K_2 is the stiffness parameter to control the displacement along the Z-axis, and K_3 is the stiffness parameter to control the rotation around the X-axis and Y-axis. Meanwhile, to increase the experience diversity and randomness of the training process, the output action had Gaussian noise added in the early training process. DDPG algorithm used the actor and critic network to represent the DPG and Q function. The actor network was used to map the environment states to output the actions using the deterministic policy. The critic network was used to estimate the value of the current policy, and was optimized by minimizing the loss function:

$$L(\theta^Q) = \frac{1}{N}\sum_i \left(r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{Q'}) - Q(s_i, a_i|\theta^Q)\right)^2 \tag{8}$$

where Q and Q' are the critic and target critic network, μ is the actor network, and θ^Q, $\theta^{Q'}$, and θ^μ are the network parameters of the critic's work, target critic network, and actor

network, respectively. Additionally, we updated the actor network using the experience policy gradient:

$$\nabla\theta^{\mu} J = \frac{1}{N}\sum_{i} \nabla_a Q(s_i, a_i|\theta^{Q}) \nabla_{\theta^{\mu}} \mu(s_i|\theta^{\mu}) \tag{9}$$

Additionally, we updated the target network:

$$\theta' = \tau\theta + (1-\tau)\theta' \tag{10}$$

We noticed that the aim of the DDPG algorithm was also to maximize the cumulative reward for the assembly task so that it would use the same reward function as the SARSA algorithm. The training process based on the DDPG algorithm is shown in Figure 5.

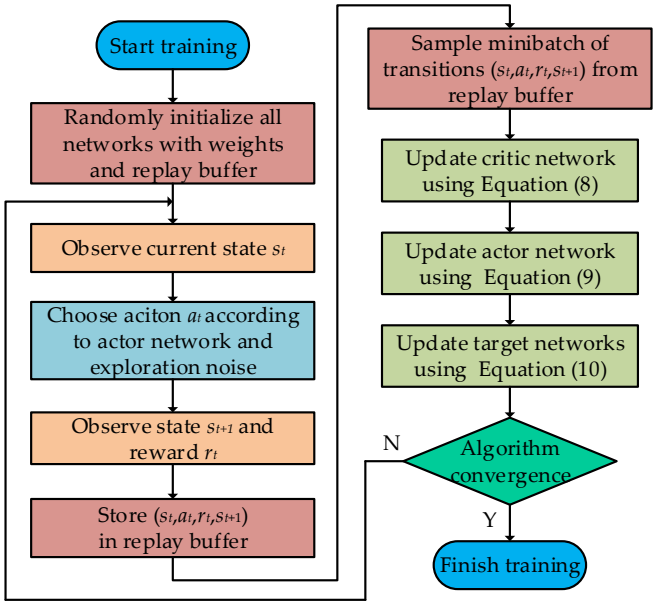

Figure 5. The training process of DDPG algorithm.

3. Results and Discussion

This section describes the experiment setup, as well as the results and discussions of the different control methods.

3.1. Experiment Setup

The experimental platform is shown in Figure 6, where a M4313M2B force/torque sensor was mounted on the end effector of manipulator UR10 and used to collect raw force/torque data from the environment. Notice that the raw force/torque data were processed with a gravity compensation algorithm to obtain the real contact force and torque. Furthermore, the peg was mounted on the force sensor, and the hole was fixed on the experimental table. The diameter and length of the peg were 10 mm and 40 mm, respectively. The diameter of the hole was 13 mm.

To verify the validity of the variable stiffness admittance control based on the RL algorithm, a series of experiments for the peg-in-hole assembly was carried out. In the first experiment, the assembly task was realized by using position control. Then, the compliance assembly experiment using fuzzy control was implemented in the second experiment. Finally, compliance assembly experiments based on the SARSA and DDPG algorithms were performed separately. In all experiments, the targets of the assembly tasks were to push the peg into the expected position into a hole. Before the assembly, the peg was positioned randomly near the hole, where the rotation angle θ was set from 5 to 10 degrees (the angle errors caused by positioning were usually less than 5 degrees). In the assembly process, when the depth was larger than 30 mm and the force value was less than 2 N, the task

was realized successfully. Obviously, the insertion depth could be used to indicate the assembly success rate. Meanwhile, we also focused on the assembly efficiency, where fewer adjustment steps meant a higher assembly efficiency. Furthermore, the cumulative reward could reflect the assembly effect well. Therefore, the reward, steps, and depth were selected to evaluate algorithm performance in this article.

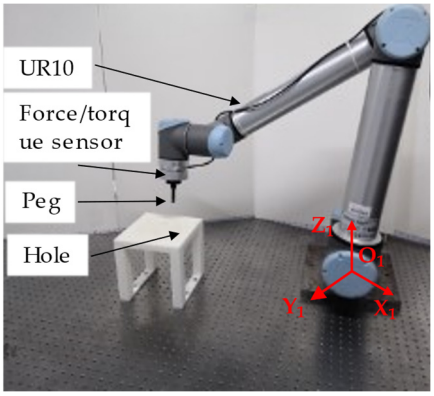

Figure 6. Experimental platform.

Furthermore, the action selection during the initial training stage had strong randomness in the reinforcement learning. Thus, the velocity and acceleration of the robot were limited (0.001 m/s^2) to ensure the assembly's safety in the experiment. Additionally, the contact force threshold was set to 40 N. Moreover, in each adjustment step, the maximum translation offset along the X_1-axis and Y_1-axis was set to 0.001 m, and the maximum rotation degree was set to 5 degrees.

3.2. Assembly Experiment using Position Control

To analyze the assembly effect based on the admittance model, the movement along the X_1-axis was directly controlled by the position of the SARSA output, and the others were driven by the constant admittance controller. The environment states could be represented as S = {5, 10, 15, 22, 26, 28, 30} based on experimental results and obtained experience. The action could be set to A_1 = {−0.001, −0.0005, 0, 0.0005, 0.001} by analyzing the assembly mechanism. Moreover, the other experiment parameters needed are listed in Table 1.

Table 1. Experiment parameters.

Parameter Name	Value
Virtual inertia M_d	0.0001
Virtual damping B_d	0.02
Virtual stiffness K_d	1
Learning rate α	0.95
Discount factor γ	0.95
Probability factor ε	0.05
Degradation factor λ	0.95
Max step	100

The experimental results from the position control are shown in Figure 7. The curves of Figure 7a,c have severe oscillation. The reward curve and depth curve have no convergence trend when the episodes increased. This meant that the success rate was very low. In Figure 7b, an assembly failure always occurred in the first three steps, because of the excessive force value. The main reason was that the high randomness could not be avoided in the initial action selection. The magnitude of movement then could not be chosen appropriately. Thus, an excessive force value would occur easily during the assembly movement. All in all, position control would increase the sensitivity of the contact force,

and decrease the compliance of the robot. Only a small movement would lead to a large contact force. Therefore, it was difficult to apply the position control in the assembly task.

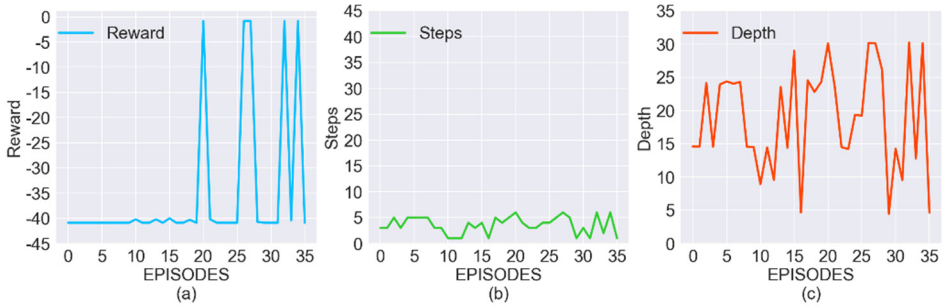

Figure 7. The results of assembly experiments using position control. (**a**) Cumulative reward in each episode; (**b**) Adjustment steps in each episode; (**c**) Insertion depth in each episode.

3.3. Compliance Assembly Experiment Using Fuzzy Control

Like the RL algorithm, the fuzzy control method did not require the accurate modeling of the environment. Therefore, the compliance assembly experiment using the fuzzy control was implemented as a comparison for the RL algorithm. In the experiment, the depth and maximum force were selected as the fuzzy controller input, and the stiffness value of the admittance model was selected as the fuzzy controller output. Each crisp input value of the fuzzy set was divided into five ranges, *VB*, *B*, *N*, *G*, and *VG*, denoting very bad, bad, normal, good, and very good, respectively. The experimental results are shown in Figure 8; the variable parameter algorithm based on the fuzzy control had good stability. The reward, steps, and depth curves remained stable throughout the experimental process, where the adjustment step was approximately 16, the cumulative reward was approximately -10, and the success rate was 100%. Obviously, compared with the experimental results of the position control, the insertion could be performed well with the compliance control. It was concluded that this method with the proper fuzzy model could realize acceptable experimental results. However, the most suitable model that realized a better effect was difficult to detect. Moreover, we noticed that the assembly effect of this fuzzy set decreased when we slightly changed the initial position and angle of the peg.

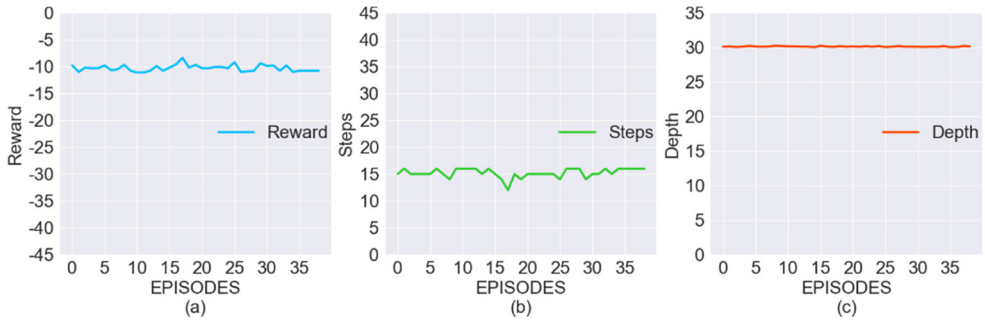

Figure 8. The results of compliance assembly experiment using fuzzy control. (**a**) Cumulative reward in each episode; (**b**) Adjustment steps in each episode; (**c**) Insertion depth in each episode.

3.4. Compliance Assembly Experiment Using SARSA

In the variable stiffness experiment using SARSA, the action was set to $A = \{0.1, 1, 1.5, 2, 4\}$ based on the mechanism analysis and simulation results. Additionally, the other experimental parameters were the same as the first experiment's. As shown in Figure 9, the reward, steps, and depth of the compliance assembly process converged gradually with the training steps increasing. It could also be found that the variable stiffness strategy began to converge in the sixth episode, and could be applied in complex conditions. Nevertheless, the peg-in-hole assembly failed in the fourteenth and twenty-second step due to the strong randomness in the early stages of training. We noticed that the latter assembly would be affected by an

unsuccessful experience, and the strategy convergence could also be realized. The adjustment step was approximately seven, the cumulative reward was approximately −2.8, and the success rate was 100%. Compared with the position and fuzzy controls, the assembly efficiency reached the maximum under the 100% success rate, and the overall performances were superior to the above two methods. Moreover, unlike the above experiments using position and fuzzy controls, this method had a robustness against changes in the initial position and angle of the peg.

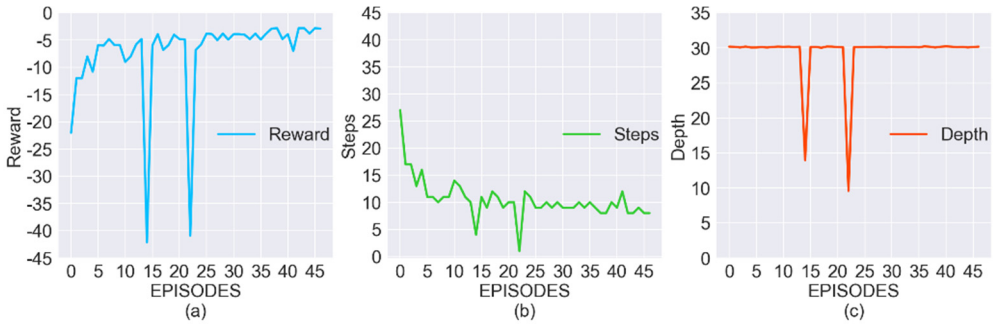

Figure 9. Online training results of variable stiffness compliance assembly using SARSA. (**a**) Cumulative reward in each episode; (**b**) Adjustment steps in each episode; (**c**) Insertion depth in each episode.

3.5. Compliance Assembly Experiment Using DDPG

To search the optimal policies, the DDPG algorithm explored all possible actions, which required a substantial amount of experimental data, and would probably generate some risky actions in realistic environments. Thus, we realized the assembly experiment based on the variable admittance control using DDPG in the simulation environment, where the simulation platform (as shown in Figure 10) was consistent with the realistic platform. Moreover, the gravity compensation algorithm for the simulation environment was verified to ensure that the force/torque data used were realistic. As shown in Figure 11, the compensated force data were approximately −0.001 in the no-contact state. Obviously, the gravity compensation algorithm was effective, and the real contact force/torque could be obtained. The stiffness value of the network output was set to 0.1~4. The primary training parameters of the networks were decided through experimental testing, as described in Table 2. Additionally, the other experimental parameters were the same as in the former assembly experiment.

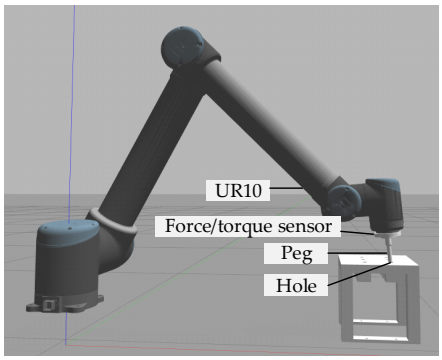

Figure 10. Simulation platform.

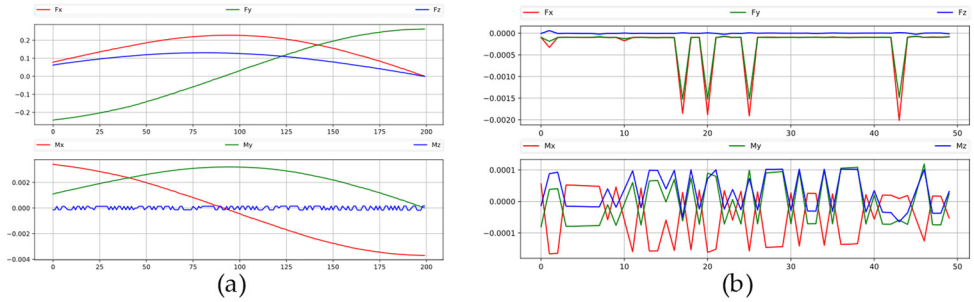

Figure 11. The results of the gravity compensation. (**a**) Uncompensated force data; (**b**) Compensated force data.

Table 2. Training parameters.

Parameter Name	Value
Actor learning rate	0.005
Critic learning rate	0.005
Target update rate	1
Batch size	50
Memory pool size	300
Reward discount	0.95

The online training results of the variable stiffness compliance assembly using DDPG are shown in Figure 12. There were approximately 14 episodes needed to fill the memory pool, so the curves oscillated in the early training episodes. Then, the models in the DDPG began to learn and converge gradually. Additionally, the three curves finally converged approximately at the twenty-fifth episode, where the reward and steps stabilized at approximately -0.2 and 7, respectively. Meanwhile, the assembly task of all episodes was successful. We noticed that three curves approximated a straight line after convergence due to fewer disturbance in the simulation environment. Compared with the compliance assembly experiment using SARSA, it could be found that the agent using the DDPG could learn a better assembly policy with a higher cumulative reward in each episode. Meanwhile, this method also had robustness against changes in the initial position and angle of the peg.

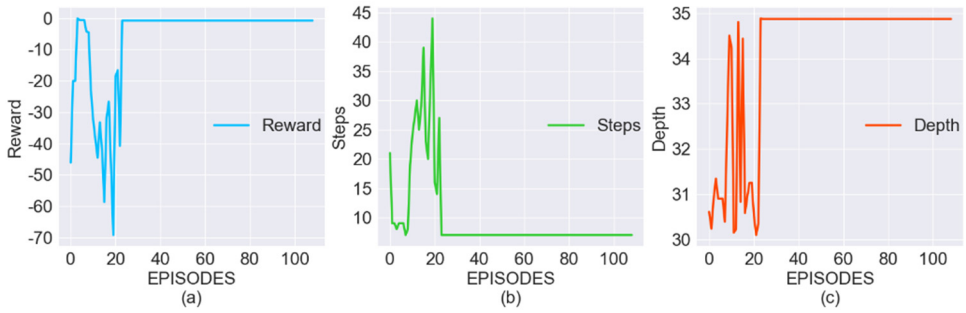

Figure 12. Online training results of variable stiffness compliance assembly using DDPG. (**a**) Cumulative reward in each episode; (**b**) Adjustment steps in each episode; (**c**) Insertion depth in each episode.

For the quantitative analysis of the proposed variable admittance control method based on the RL algorithm, the results of the assembly experiments with different methods were compared, as shown in Table 3. The proposed methods in this article had fewer convergence steps and adjustment steps under similar success rates, as well as better comprehensive performance than other methods. In summary, the above experimental results verified the improvement of the assembly effects well for the robotic assembly task using the variable admittance control based on the RL algorithm.

Table 3. Results for peg-in-hole assembly with different methods.

Method	Peg-Hole Clearance	Converge Steps	Adjustment Steps	Success Rate
Fuzzy Q-learning [13]	1 mm	160	26	100%
PPO using asymmetric impedance matrices [15]	2.5 mm	-	-	79–94%
Fuzzy SARSA (ours)	3 mm	23	7–8	100%
Fuzzy DDPG (ours)	3 mm	25	7–8	100%

4. Conclusions

In this article, the admittance parameter identification process was defined as a MDP problem, which was solved by using the RL algorithm. A fuzzy reward method was established to solve the complex reward establishment problem. Furthermore, a series of experiments was carried out, including assembly experiments based on the position control, fuzzy control, and RL algorithm. In the first experiment, severe oscillations were revealed by the resulting curves of the position control. Therefore, the position control could not be directly used here, which further verified the necessity for the compliant control. By comparing the results of the fuzzy control and RL algorithm, the assembly effect of the RL algorithm was found to be better than the fuzzy control, where the results of the DDPG were best, and the reward and steps stabilized at approximately -0.2 and 7, respectively. Moreover, the generalization ability of the RL algorithm was tested by changing the initial position and angle of the peg in the third and fourth experiments. The results indicated that the RL algorithm effectively improved the robotic compliance assembly.

In future works, model migration between the simulation and reality is a major challenge for autonomous and complex robot operations, and it is also our next research topic.

Author Contributions: Conceptualization, S.L. and X.Y.; methodology, S.L. and X.Y.; software, X.Y.; validation, S.L., X.Y. and J.N.; formal analysis, J.N.; investigation, S.L.; resources, S.L. and J.N.; data curation, S.L. and X.Y.; writing—original draft preparation, X.Y.; writing—review and editing, S.L.; visualization, X.Y.; supervision, S.L. and J.N.; project administration, S.L. and X.Y.; funding acquisition, S.L. and J.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by State Grid Hunan EHV Transmission Line Company opening program, "Research on robot peg-in-hole assembly strategy based on multiple sensors" (project number 2021KZD2002), in part by Innovation Project of Guangxi Graduate Education (project number YCSW2022014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Yang, X.T.; Wang, Z.Y.; Li, F.M.; Song, R. Robot phased guided assembly based on process modeling. *Comput. Inregrated Manuf. Syst.* **2021**, *27*, 2321–2330.
2. Liu, K.X. *Research on Robotic Assembly Theory of Circular-Rectangular Compound Peg in Hole*; Harbin Institute of Technology: Harbin, China, 2021.
3. Beltran-Hernandez, C.C.; Petit, D.; Ramirez-Alpizar, I.G.; Harada, K. Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach. *Appl. Sci.* **2020**, *10*, 6923. [CrossRef]
4. Kilikevicius, S.; Baksys, B. Dynamic analysis of vibratory insertion process. *Assem. Autom.* **2011**, *31*, 275–283. [CrossRef]
5. Kim, Y.L.; Song, H.C.; Song, J.B. Hole detection algorithm for chamferless square peg-in-hole based on shape recognition using F/T sensor. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 425–432. [CrossRef]
6. Shimizu, M.; Kosuge, K. Designing robot admittance for polyhedral parts assembly taking into account grasping uncertainty. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, Canada, 2–6 August 2005.

7. Shimizu, M.; Kosuge, K. An admittance design approach to dynamic assembly of polyhedral parts with uncertainty. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA), Orlando, FL, USA, 15–19 May 2006.
8. Mol, N.; Smisek, J.; Babuska, R.; Schiele, A. Nested compliant admittance control for robotic mechanical assembly of misaligned and tightly toleranced parts. In Proceedings of the 2016 IEEE International Conference on Systems Man and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016.
9. Wu, H.; Li, M. Iterative Learning Algorithm Design for Variable Admittance Control Tuning of a Robotic Lift Assistant System. *SAE Int. J. Engines* **2017**, *10*, 203–208. [CrossRef]
10. Wu, C.; Shen, Y.; Li, G.; Li, P.; Tian, W. Compliance Auxiliary Assembly of Large Aircraft Components Based on Variable Admittance Control. In Proceedings of the International Conference on Intelligent Robotics and Applications, Yantai, China, 22–25 October 2021; Springer: Cham, Switzerland, 2021; pp. 469–479.
11. Tsumugiwa, T.; Yokogawa, R.; Hara, K. Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002.
12. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [CrossRef]
13. Zou, P.; Zhu, Q.; Wu, J.; Xiong, R. Learning-based Optimization Algorithms Combining Force Control Strategies for Peg-in-Hole Assembly. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
14. Chen, J.P.; Zheng, M.H. A Survey of Robot Manipulation Behavior Research Based on Deep Reinforcement Learning. *Robot* **2022**, *44*, 236–256.
15. Kozlovsky, S.; Newman, E.; Zacksenhouse, M. Reinforcement Learning of Impedance Policies for Peg-in-Hole Tasks: Role of Asymmetric Matrices. *IEEE Robot. Autom. Lett.* **2022**, *7*, 10898–10905. [CrossRef]
16. Wu, X.P.; Zhang, D.P.; Qin, F.B.; Xu, D. Deep Reinforcement Learning of Robotic Precision Insertion Skill Accelerated by Demonstrations. In Proceedings of the 15th IEEE International Conference on Automation Science and Engineering (IEEE CASE), Vancouver, BC, Canada, 22–26 August 2019.
17. Hou, Z.M.; Dong, H.M.; Zhang, K.G.; Gao, Q.; Chen, K.; Xu, J. Knowledge-Driven Deep Deterministic Policy Gradient for Robotic Multiple Peg-in-Hole Assembly Tasks. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018.
18. Martín-Martín, R.; Lee, M.A.; Gardner, R.; Savarese, S.; Bohg, J.; Garg, A. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, China, 3–8 November 2019.
19. Beltran-Hernandez, C.C.; Petit, D.; Ramirez-Alpizar, I.G.; Nishi, T.; Kikuchi, S.; Matsubara, T.; Harada, K. Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5709–5716. [CrossRef]
20. Zhang, X.; Sun, L.; Kuang, Z.; Tomizuka, M. Learning variable impedance control via inverse reinforcement learning for force-related tasks. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2225–2232. [CrossRef]