2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Learning-based Optimization Algorithms Combining Force Control Strategies for Peg-in-Hole Assembly

Peng Zou, Qiuguo Zhu, Jun Wu, and Rong Xiong

***Abstract*—In this paper, an approach for automatic peg-in-hole assembly is proposed. The task is divided into two main steps: searching phase and inserting phase. First, a multilayer perceptron network is designed to address the hole search problem and a hybrid force position controller is introduced to ensure a safe and stable interaction with the external environment. Then, for the inserting phase, 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. This approach is a practical and general approach to solve complex peg-in-hole assembly problems by taking advantage of both learning-based algorithms and force control strategies, which can greatly improve the efficiency and safety of the industrial manufacturing process without identifying the unknown contact model and tuning tedious parameters. Finally, the peg-in-hole experimental results for an industrial robot verified the effectiveness and robustness of the proposed approach.**

I. Introduction

Industrial robots have been widely used in practical peg-in-hole assembly work in industrial manufacturing with the progress of robotic technology. The contact force between the peg and the hole determines the performance of the peg-in-hole task and the assembly quality of products. To control the contact force well, force control strategies are widely introduced to generate compliant behavior between the robot and the external environment. There are two main types of force control strategy: passive force control and active force control. In passive force control, the robot can adjust its motion trajectory automatically as the result of the design of specific elements installed in the robot, such as remote center compliance[1]. Clearly, it is necessary to design a suitable passive compliant device for each specific task, which is not universal and uncontrollable. Because of this disadvantage of passive force control, many experts focus on active force control to find a general solution for the peg-in-hole task. In active force control, force controllers are designed according to external force information to achieve the robot's compliant action. The most common method to obtain force information is to install a force/torque sensor at the joint or end of the robot. However, because of the high price of the sensor and the inconvenience of installation in some special circumstances, sensorless assembly technology has also been studied. For example,Deluca[2] and Lee[3] designed a sensorless assembly system based on the estimation of the external contact force according to the joint current of the robot. However, this sensorless method generally has poor accuracy; it cannot be applied on some occasions that require high accuracy. Active force control strategies contain two common categories: hybrid force/position control and impedance control. Impedance control[4] was proposed by Hogan in 1985 and is widely used in compliant assembly tasks. Impedance control makes the interaction between the robot and the environment satisfy an ideal relationship by adjusting the impedance parameters, which can also be classified into two main groups: position-based impedance control and force-based impedance control. Position-based impedance control does not need to change the control system of the robot and only requires the design of an outer force controller so that it is easily applied in a realistic production process.

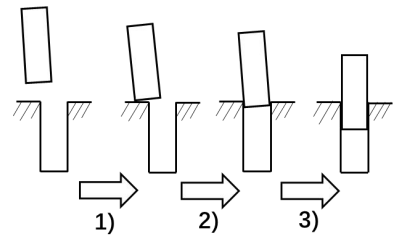


Fig. 1 Schematic diagram of the peg-in-hole assembly process.

There have been many successful cases of the peg-in-hole assembly task using active force control theory. Wu[5] proposed a force/position hybrid control method to complete the assembly task for large aviation parts. This method first determines whether the corresponding direction adopts force control or position control through a selection matrix S. The force control direction is a proportion controller based on the robot end-effector speed control loop, which is also a special case of the impedance controller. This method has a fast response speed and good tracking performance, and has been applied in an actual production process. Zhang[6] applied position-based impedance control to the peg-in-hole task and verified that the controller can effectively avoid jamming through jamming model analysis.

The traditional active force control strategy has some limitations, such as the lack of adaptability and robustness to a new environment. Once the environment changes or is disturbed, the traditional active force control method cannot complete the assembly task well. To enhance the robustness of the low-level force controller in an uncertain environment, some scholars have used a parameter optimization algorithm to optimize the controller parameters to achieve adaptive control. Petrovic[7] proposed a set of identification methods based on a neural network and adaptive fuzzy theory, which can identify the contact state and output the optimal control

*This work was supported by the National Nature Science Foundation of China (Grant No. NSFC: 51405430) and National Defense Innovation (Grant No.1716311XJ00100501).

Peng Zou, Qiuguo Zhu, Jun Wu, Jianxiang Jin, and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Institute of Cyber-System and Control, Zhejiang University, Hangzhou, P.R. China. Qiuguo Zhu is the corresponding author (e-mail: qgzhu@zju.edu.cn).

parameters of the force controller in real time. Hou[8] proposed an evolutionary optimization method based on a learning algorithm to optimize the control parameters of the underlying impedance controller. This method does not require prior information about the environment, and the optimal parameters can be selected by a small number of assembly experiments.

Additionally, to enhance the adaptability and learning ability of the robot, some learning-based methods have been gradually applied to the peg-in-hole task. There are two main types of algorithms: imitation learning and reinforcement learning. Imitation learning learns assembly skills from human demonstration trajectories, which consist of three principal phases: sensing, encoding, and reproducing[9]. Abu-Dakka[10] and Kramberger[11] proposed a complete method that combines dynamic movement primitives (DMPs) to learn from human assembly demonstrations and capture the trajectories of pegs with force-torque profiles. Moreover, the differential equation of DMPs has been improved to adapt to uncertainty in the desired position and obstacle avoidance[12]. Tang[13] and Fei[14] used Gaussian mixture regression to predict the velocities in a manner similar to a human in response to wrench signals. Then, the output velocities were executed through a low-level controller (impedance controller) to realize the peg-in-hole insertion phase. To construct a heavy-weight component assembly process, Wan [15] proposed a complete methodology through learning assembly skills from human demonstrations and compensating for the large deformation with Gaussian process regression.

Unlike the imitation learning method, reinforcement learning achieves the assembly strategy through an interaction with the environment and does not require contact model information. This has better adaptability to the new state and environment. As an intelligent learning method, reinforcement learning is very suitable for guiding robots to learn skills from the environment to solve complex problems. Andrew Barto[16] used a value function-based reinforcement learning algorithm to learn a discrete admittance assembly strategy based on force feedback information. Nuttin[17] modeled the peg-in-hole problem as a sequence decision-making problem and used an actor-critic algorithm to optimize the assembly time. The simulation results verified the effectiveness of the algorithm. The IBM Research Institute of Japan[18] used a deep reinforcement learning algorithm to realize the high precision peg-in-hole assembly task under the condition of the limitation of robot positioning accuracy and sensor accuracy. It used a deep recurrent neural network to estimate the value function and then used the learned value function to select an optimal assembly action. The selected action was realized by the underlying force controller, which ensured the safety and efficiency of the reinforcement learning algorithm. This provides a new idea for reinforcement learning applied to the peg-in-hole task; that is, reinforcement learning is used as an upper-level optimization method to improve the performance of the underlying force controller.

Although the development of peg-in-hole assembly technology has been very fast and various algorithms have been applied to the task successfully, most studies have only focused on active control strategies for the inserting phase and neglect the searching phase of the peg-in-hole task. The searching phase is an essential step for the subsequent inserting phase and can greatly impact the successful completion of the assembly task.

In this paper, a complete method for both the searching phase and inserting phase of the peg-in-hole task is proposed. Moreover, the proposed method combines the traditional active force control strategy and learning-based optimization algorithms so that it not only ensures the security and stability of the assembly task through a low-level force controller but also ensures the optimization and efficiency of the assembly task through upper-level learning-based optimization algorithms. The proposed method improves assembly efficiency greatly and enhances the robot's adaptability, in addition to its robustness to a new state or environment.

II. Problem Formulation

Generally, as shown in Fig. 1, the operation of the peg-in-hole assembly task mainly includes three phases: (i) approaching phase, (ii) searching phase, and (iii) inserting phase. The approaching phase typically makes use of some visual positioning technologies to obtain the target hole pose, and then the robot moves to this target pose. This phase is easily realized, so in this paper, the focus is on the latter two phases.

A. Searching Phase

Although industrial robots have achieved a good level of accuracy, it is difficult to set a peg and hole to a few tens of μm of precision using a position controller[18]. Visual servo is also impractical because of the limited resolution of cameras or internal parts that are occluded during assembly, for example, in the case of meshing gears and splines in transmission. Image-based boundary extraction techniques, visual servo tracking approaches, and blind search strategies based on a designed human-like searching path have been applied to locate holes and track the hole position. However, these methods only consider search trajectories and cannot ensure stable and safe contact between the robot and mating parts during the searching phase. In this paper, a multilayer perceptron (MLP) network is used to learn the hole location with respect to the peg position, and combined with a hybrid force/position controller to ensure safety and stability during the searching phase.

B. Inserting Phase

After the searching phase, the peg position error has been eliminated, but there still exists a small attitude error that may lead to jamming or wedging, so the assembly task will fail, or parts or the robot itself may be damaged. Therefore, it is necessary to adopt force control methods to compensate for the attitude error and control the interaction force between the peg and the hole during the inserting phase. In this paper, a position-based impedance controller is used to yield compliant behavior between the robot and the environment during the inserting phase, which is practical in a complex industrial environment and easier to realize in an industrial robot compared with other researcher's work. However, to achieve good performance, it is very important to select the appropriate impedance parameters. Because the contact model cannot be obtained during insertion, some model-based optimization algorithms cannot be used any more.In this paper

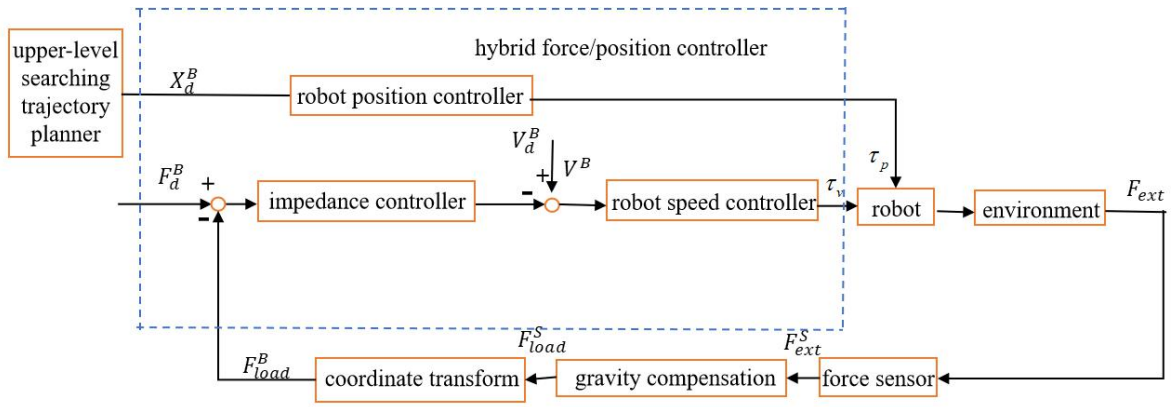


Fig. 2 Searching hole block diagram

a reinforcement learning algorithm is used to learn the optimal parameters for the impedance controller and enhance the insertion strategy's adaptability and robustness to the dynamic and complex inserting phase.

III. Proposed Method

A. Searching Phase

In this paper, the searching phase method is mainly divided into two components, as shown in Fig. 2: the upper-level searching trajectory planner, which is an MLP network used to output the next search action, and the low-level hybrid force/position controller, which is used to receive the upper-level command and generate a smooth interaction with the environment to ensure the safety and stability of the searching phase.

1) Low-level hybrid force position controller: The hybrid force position controller, shown in Fig. 2, is used to control the interaction force during the hole searching process. In the peg-in-hole task, force is regulated in the direction normal to the hole surface, whereas the position is regulated in the tangential directions. The errors of the three attitude directions are relatively small and have little influence on the searching phase, so just keep them the same. Note that the force control and position control of the hybrid controller do not work in the same cycle. Generally, the execution frequency of force control is about ten times that of position control; that is, position control only executes after force control has been steady, where the corresponding direction force has been within a certain target range. Position control is realized by the encapsulated position loop of the robot control system and only needs to receive the upper planner's command to guide the robot to move to the desired hole search point. The force control direction is divided into inner and outer controllers. The inner controller is the encapsulated robot end-effector speed controller, so only the outer controller needs to be designed. In this paper, the impedance controller is chosen as the outer force control method and its control law is

$$F_d - F = M_d(\dot{V}_d - \dot{V}) + B_d(V_d - V). \qquad (1)$$

The entire model of the force control direction based on impedance control is shown in Fig. 3. The system can be stabilized by choosing a suitable inertial matrix M and damping matrix B, and it can be proven that this control system can stabilize the contact force near the target value so that safety and stability during the searching phase is ensured.

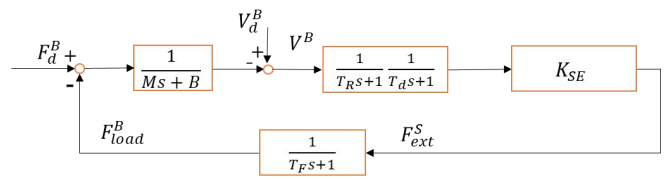


Fig. 3 Entire force control block diagram

2) Upper-level searching trajectory planner: This part is used to generate the hole search trajectory. In this paper, an MLP network is proposed to search the hole. The MLP network is used to train a four classifier. Its input is the contact force/torque information and the output is the four translation directions in which the robot should move in the next step. The network is shown in Fig. 4.

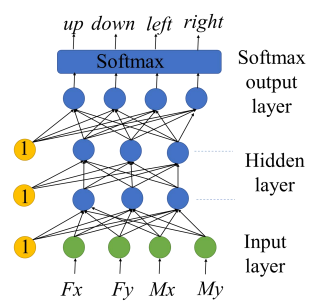


Fig. 4 MLP model for searching the hole

First, the input training data need to be collected. The data are collected by sending some commands to the robot and peg will move to the center of the hole with a predefined offset in the x and y directions. The peg is moved in increments of 0.2 mm in the x and y directions within the range of +/- 10 mm from the center of the hole. The previous impedance controller, shown in Fig. 3, is used to stabilize the interaction with the environment and ensure the quality of collected data. The collected data format is as follows: $\{Fx,Fy,Fz,Mx,My,Mz,Px,Py,Pz\}$, where Fx,Fy,Fz,Mx,My,Mz are the forces and moments measured by a wrist force sensor

expressed in the base frame, and *Px,Py,Pz* are the peg positions with respect to the hole center. In this paper, only *Px,Py* are used to label each entry in the dataset for four actions: move left, move down, move right, and move up. The label rules are as follows:

If *Py*<*Px* and *Py*<-*Px*, then move up.

If *Py*>=*Px* and *Py*>= -*Px*, then move down.

If *Py*>-*Px* and *Py*<*Px*, then move left.

If *Py*>*Px* and *Py*<-*Px*, then move right.

MLPClassifier in scikit-learn is used to train the proposed network. The input is [*Fx,Fy,Mx,My*] and the output is the four labels described above. The network is composed of two hidden layers of size [100,50], the solver is Adam, and activation is ReLU.

After training, an MLP model was obtained and its evaluation index is shown in Table 1. The average accuracy of the four labels was 79%, which indicates that the method can meet the requirements of the hole searching task to a certain extent and the accuracy can be improved higher by some more complicated network structures .

Table 1 Evaluation index of the trained MLP model

label	precision	recall	F1 score
0	0.70	0.80	0.74
1	0.88	0.91	0.90
2	0.77	0.74	0.75
3	0.83	0.71	0.76

B. Inserting Phase

After the searching phase, the position error between the peg and the hole has been eliminated, but there still exists an attitude error that may lead to jamming or wedging problems. Hence, an impedance controller is adopted, shown in Fig. 3, to compensate for this attitude error and control the interaction force during the inserting phase, which is common and practical for position-controlled industrial manipulators. However, because of the complex and dynamic industrial environment, in addition to severe noise interference, the fixed constant parameters cannot meet the requirements of dynamic interaction with the environment, and have low efficiency. Therefore, in this paper, a variable impedance controller is proposed to improve the adaptive ability and robustness to a complex industrial assembly environment based on the fuzzy Q-learning algorithm, which is a widely used RL algorithm to deal with continuous-state and real-world problems without any prior knowledge of the contact model during the inserting phase.

1) Fuzzy Q-learning algorithm theory: The inserting phase can be regarded as a Markov decision process. At each time *t*, the algorithm first obtains the current state of the assembly, and then the assembly action is performed indirectly by outputting the damping parameters of the impedance controller. Then the assembly system reaches a new state and a reward value is observed. The execution action in each state is chosen by the policy of the reinforcement learning algorithm, which is continuously optimized according to the received reward value. The beginning of the inserting phase to the completion of the assembly task can be regarded as an episode of reinforcement learning, and the goal of the algorithm is to maximize the total reward of the entire episode. The state of reinforcement learning is defined as $X = \{V,F_e\}$, where V denotes the actual Cartesian velocity and F_e denotes the external force measured by the wrist force sensor. Instead of using a discrete state, fuzzy Q-learning uses a fuzzy set to represent a fuzzy state. The degree of belonging to a certain fuzzy state is determined by the premise strength $\emptyset_i$ of the corresponding fuzzy rules. Different from the standard fuzzy inference system, the output of fuzzy rules is clear; that is, the choice of action is selected from a discrete action set according to the policy of reinforcement learning. The selected action by each rule R_i contributes to a continuous global action U_t , which is the damping value provided to the impedance controller, according to the premise strength $\emptyset_i$ of that rule. Clearly, a fuzzy system is introduced to approximate the Q function, complete the discrete representation of the continuous input state, and integrate the output of the discrete action into continuous output linearly. A rule of the algorithm can be expressed as follows:

$$R_i: IF\ X\ is\ S_i\ THEN\ a_1\ with\ q(S_i,a_1)$$
$$\cdots \qquad \text{OR}\ \ a_m\ with\ q(S_i,a_m),$$

where S_i is the fuzzy state of rule R_i, $A = (a_1,\cdots a_m)$ are the m possible discrete actions of the rule, and $q(S_i,a_j)$ is the q-value that determines the probability of choosing action j of the rule. To explore all possible actions, policy π selects actions according to the ε-greedy exploration-exploitation strategy:

$$a_i^{'} = \begin{cases} \arg\max\limits_{a_j \in A} \left(q(S_i,a_j)\right) & \text{probability } \varepsilon \\ \text{choose a random selection} & \text{probability } 1-\varepsilon \end{cases} \tag{2}$$

The selected damping value is the global action U_t given by the aggregation of all n rules:

$$U_t(X_t) = \sum_{i=1}^{n} a_i^{'} \emptyset_i \tag{3}$$

The Q value of the current state action pair is

$$Q_t(X_t,U_t) = \sum_{i=1}^{n} q_t(S_i,a_j)\emptyset_i \tag{4}$$

The optimal action for the current state is given by

$$Q_t^*(X_t) = \sum_{i=1}^{n} (\max_{a_j \in A} q_t(S_i,a_j))\emptyset_i \tag{5}$$

The iterative updating formula of q values is as follows:

$$q_{t+1}(S_i,a_j) = q_t(S_i,a_j) + \beta\epsilon_{t+1} \tag{6}$$

where β is the learning rate, ϵ_{t+1} is the temporal difference error given by

$$\epsilon_{t+1} = r_{t+1} + \gamma Q_t^*(X_{t+1}) - Q_t(X_t,U_t), \tag{7}$$

where r_{t+1} is the reward received at time t+1 and γ is a discount factor that weights the effect of future rewards.

2) Reward design: The reward r_{t+1} should be designed according to the specific task. In the peg-in-hole task, the reward generally consists of two parts[19]:

$$r = r_1 + r_2 c \tag{8}$$

where r_1 is the positive reward at the end of each episode to reward the agent for completing the task successfully:

$$r_1 = 1 - \frac{k}{k_{max}}, \tag{9}$$

where k and k_{max} represent the assembly steps and maximum step, respectively. r_2 is the negative reward at each time step to punish the low assembly speed and large contact forces. In this paper, a fuzzy reward system is used to compute r_2 considering four factors: the current depth of pegs d_t^z, current translation offset δ_t^z in the z direction of each step, and corresponding force and moment. The damping parameters of six directions are optimized separately. The relationship between each direction parameter and its force/moment input of the fuzzy system is shown in Fig. 5.

fuzzy system input	correspond parameter	fuzzy system input	correspond parameter
F_y, M_x	B_d^{rx}	F_x, M_y	B_d^x
F_x, M_y	B_d^{ry}	F_y, M_x	B_d^y
$\max(F_x, F_y), M_z$	B_d^{rz}	$F_z, \max(M_x, M_y)$	B_d^z

Fig. 5 Relationship between the fuzzy logic input and six direction parameters

To simplify the membership function, a two-layer fuzzy system is designed, as shown in Fig. 6.

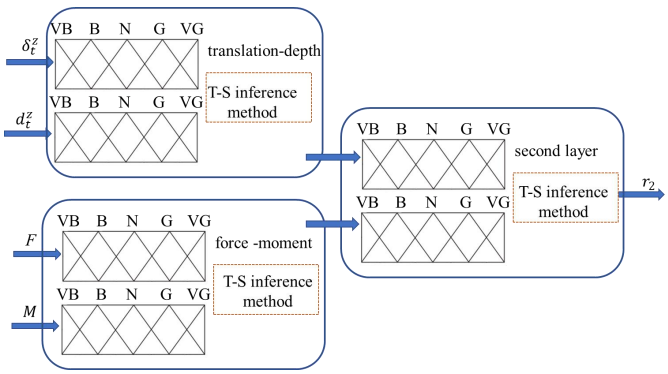


Fig. 6 Fuzzy reward system to compute r_2

Each input value of all the fuzzy sets is divided into a five triangular membership range: VB, B, N, G, and VG,which denotes very bad, bad, normal, good, and very good of the current action respectively.

The total number of fuzzy rules is 75 and each fuzzy set includes 25 rules. The logic rules table for three fuzzy logic systems are shown in Table 2, which is designed according to prior assembly knowledge and can easily be adjusted to meet flexible requirements, unlike complex implementations that use an accurate function. For instance, during the assembly process, d_t^z and δ_t^z contribute different importance weights to the final reward at different stages according to previous assembly experience.

Table 2 Fuzzy logic table

a) First layer with the inputs of translation and depth

zdzoutput δ_t^z \ d_t^z	NB	B	N	G	VG
NB	-1.0	-0.707	-0.5	-0.177	-0.177
B	-1.0	-0.5	-0.354	-0.125	-0.177
N	-0.5	-0.354	-0.25	-0.125	-0.177
G	-0.25	-0.354	-0.125	-0.177	-0.0625
VG	-0.177	-0.125	-0.177	-0.0625	-0.0625

b) First layer with the inputs of force and moment

mfoutput F \ M	VG	G	N	B	VB
VG	-0.0625	-0.0625	-0.125	-0.177	-0.177
G	-0.0625	-0.177	-0.177	-0.125	-0.177
N	-0.177	-0.177	-0.25	-0.5	-0.707
B	-0.125	-0.125	-0.125	-0.707	-1.0
VB	-0.177	-0.177	-0.5	-0.707	-1.0

c) Second layer with the output of r_2

reward $-r2$ \ $-r1$	VG	G	N	B	VB
VG	-0.11	-0.11	-0.16	-0.16	-0.23
G	-0.16	-0.16	-0.23	-0.23	-0.33
N	-0.23	-0.33	-0.33	-0.48	-0.48
B	-0.33	-0.33	-0.48	-0.69	-0.69
VB	-0.48	-0.48	-0.69	-1.0	-1.0

IV. Experimental Results

The proposed assembly method was verified using a 6-axis universal robot. A 6-axis ATI force-torque sensor and a Robotiq gripper were attached to the end effector of the robot. The architecture of the peg-in-hole assembly experimental platform is shown in Fig. 7. The material of the pegs and holes was aluminum and their clearance was 1 mm. A computer was used to communicate with the robot controller via the TCP/IP protocol, which was developed by a socket written in Python. The force/torque values were read by communicating with the ATI NetBox using Python scripts and the sensor was calibrated first. The methods proposed in this paper for searching and inserting were verified distinctly. First, a hole searching experiment was conducted and then an inserting experiment was conducted when the search phase was complete. Finally, an entire peg-in-hole process was used to verify the effectiveness and optimality of the proposed method for peg-in-hole assembly. The inertial parameters of the force controller during these three experiments are selected as constant values to ensure the stability of the system ,in which the translation inertial parameter as well as the rotation inertial parameter is set as 0.008 .

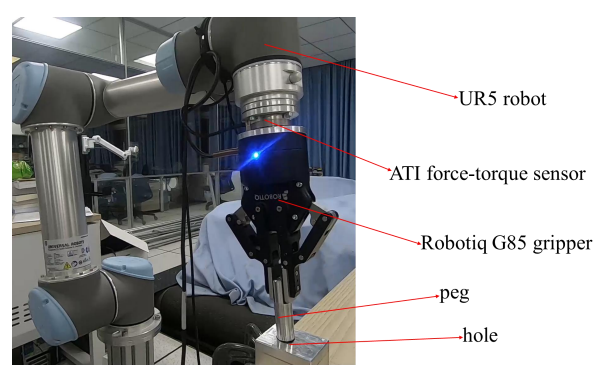
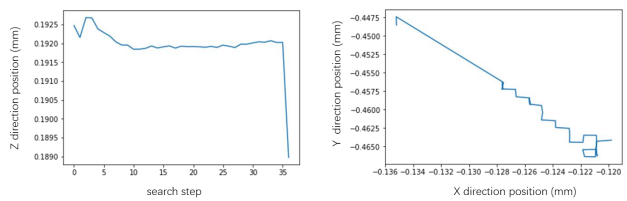


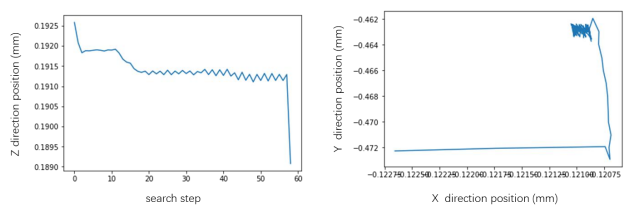
Fig. 7 Peg-in-hole experimental platform

A. Searching Phase

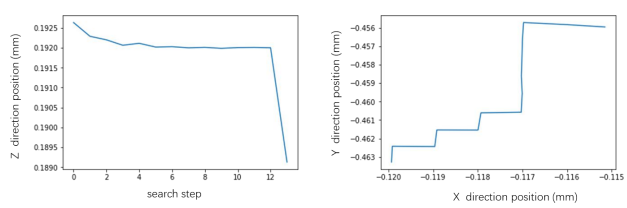
To verify the efficiency and robustness of the trained MLP model described in Section 3 for searching holes, a hole search experiment was conducted using four different direction errors in the position with respect to the hole center. The peg moved down along the z-axis of the robot base coordinate first until it made contact with the hole surface. Then the trained MLP model was used to learn the hole center position. The force/torque values in the x and y directions measured by the sensor were the input of the network, and the next search action was obtained in real time. The searching increment was set to 0.5 mm and the hybrid force controller described in Section 3 was adopted to stabilize the interaction between the robot and the environment. Hence, the robot executed the searching action from the MLP network's output until the force controller in the z direction was steady. The experimental results and snapshots are shown in Figs. 8 and 9, respectively.



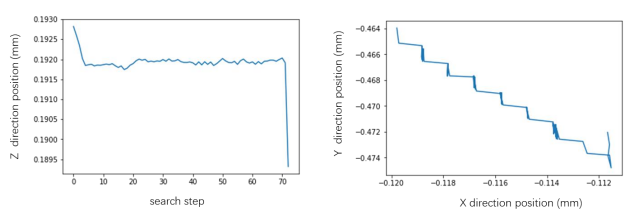
(a) Left-up with respect to the hole



(b) Left-down with respect to the hole



(c) Right-up with respect to the hole



(d) Right-down with respect to the hole

Fig. 8 Searching hole experiment with four different direction errors in the position with respect to the hole center.

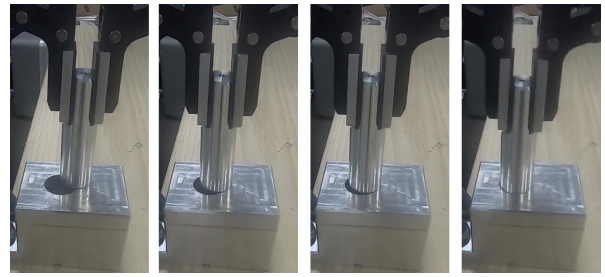


Fig. 9 Snapshots of the searching phase

As shown in Fig. 9, the peg position P_z was used to detect when the search was successful. If P_z became larger than 3 mm compared with the initial start point, the searching phase was complete and the peg was inside the hole. The four group experiments were all successful, with a small number of searching steps, as shown in Fig. 8, which proves the effectiveness and efficiency of the proposed searching hole method.

B. Inserting Phase

The inserting phase experiment assumed that the searching phase was successfully completed and there only existed the attitude error between the peg and hole. The aforementioned method based on fuzzy Q-learning was used to perform the inserting phase. An impedance controller was used in all six directions to realize the compliant behavior of the robot, so six damping parameters needed to be optimized. Consider the parameter B_d^{rx} as an example; the other directions' parameters are similar. The state input vector $\{V_{rx}, M_x\}$ is expressed with five sets using triangular membership functions, and their parameters were determined according to prior knowledge. The goal depth was set to 36 mm and the maximum number of steps was set to 50. The action set A was $\{50,75,100\}$ and the reward was computed according to Figs. 5 and 6. The learning rate and discount factor were set to 0.3 and 0.5, respectively. The virtual inertial parameter of the controller was constant. At the beginning of each episode, the robot moved to the fixed initial state with a small angle error around the x direction between the peg and hole. Then the proposed fuzzy Q-learning algorithm was continuously executed and output a damping parameter ranging from 50 to 100, which was provided to the low-level impedance controller until the peg moved to the goal depth. The results of the training process are shown in Fig. 10. Clearly, the algorithm rapidly converged after 160 episodes. The assembly steps decreased by approximately 18% and the accumulative reward of the entire episode increased by 21%, which shows the effectiveness of the fuzzy Q-learning algorithm.

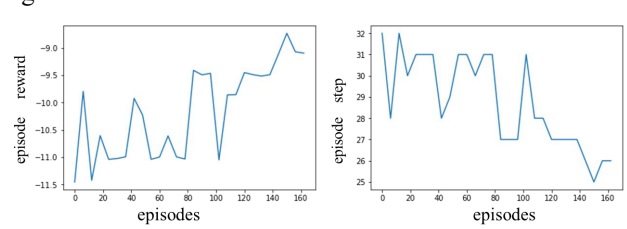


Fig. 10 Performance during the training process

A policy for the inserting phase was obtained after training to test the optimality of the trained policy. This policy was compared with a common impedance controller. The common impedance controller set the constant $M_d = [0.008\ 0.008\ 0.008\ 0.008\ 0.008\ 0.008]$, $B_d = [500\ 500\ 500\ 75\ 75\ 75]$. The robot moved to the same initial position as in the training process, and the learned policy and common impedance controller were used to perform the inserting phase separately. The experimental results are shown in Fig. 11. Although both two methods executed the inserting phase successfully, the learned policy with variable damping parameters performed the same peg-in-hole task using fewer assembly steps and smaller contact forces/torques than the common impedance controller, which had constant damping parameters.

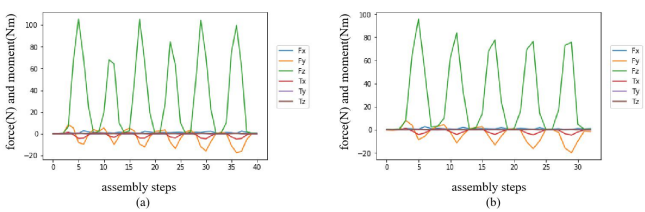


Fig. 11 Performance of tasks with the same initial position: (a) performance using constant parameters, and (b) performance using the learned policy of the fuzzy Q-learning algorithm.

Additionally, to test the robustness of the proposed algorithm against a new initial position, the pegs were rotated along the x-axis with a larger angle than the initial position in the training process. Then the aforementioned two methods were used to perform the task. As shown in Fig. 12, the new initial position led to larger contact forces and torques, and the common impedance controller failed. However, the trained policy of the reinforcement learning algorithm completed the task well, which shows the robustness and adaptability of the method to the new initial position.

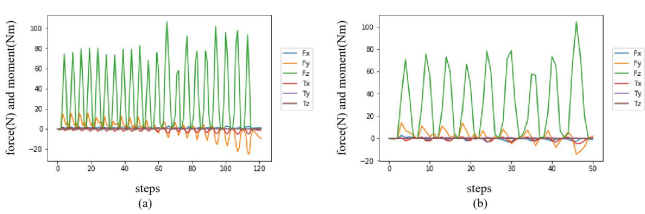


Fig. 12 Performance of tasks with the new initial position: (a) performance using constant parameters, and (b) performance using the learned policy of the fuzzy Q-learning algorithm.

C. Entire peg-in-hole process

To test the effectiveness and stability of the method for the peg-in-hole task, the task was executed 25 times with random initial positions. The results show that the proposed methods achieved a 100% success rate. Snapshots of one experiment are shown in Fig. 13 .The mean force/torques of the 25 tasks are shown in Fig. 14.

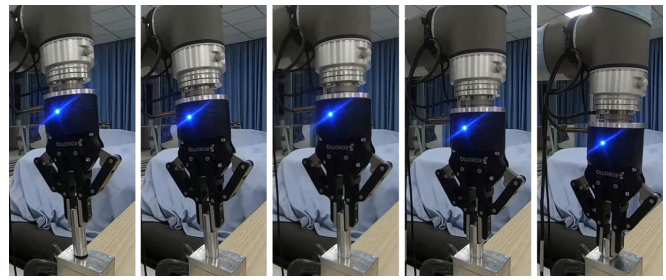


Fig. 13 Snapshots of the entire peg-in-hole task

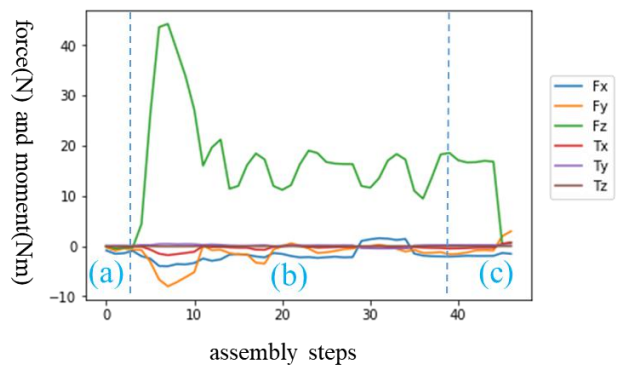


Fig. 14 Mean force and torques of the 25 tasks: (a) approaching phase, (b) searching phase, and (c) inserting phase.

V. Conclusion

In this paper, a method was proposed that combined a learning-based algorithm and force control strategy for peg-in-hole assembly. This method contains two main components. The first component is an MLP network for generating the hole searching trajectories, and a hybrid force/position controller was introduced to ensure a safe and stable interaction with the environment during the searching phase. The second component was designed for the inserting phase, and is a variable impedance controller based on fuzzy Q-learning. The effectiveness and robustness of the proposed method was verified by three experiments. First, a hole searching experiment was successfully completed using the MLP network and hybrid force/position controller, which showed the efficiency and effectiveness of our hole searching method. Then a hole inserting experiment was conducted and the results showed the optimality and robustness of the variable impedance controller based on fuzzy Q-learning for the inserting phase. Finally, the entire peg-in-hole task was performed, and the results showed the stability and adaptability of the proposed approach for peg-in-hole assembly. In conclusion, a practical and general approach was proposed to address complex peg-in-hole assembly tasks through learning based on algorithms combined with force control strategies, which can greatly improve the efficiency and safety of the industrial manufacturing process without identifying the unknown contact model and tuning the controller parameters.

References

[1] De Fazio T L. The instrumented remote center compliance[J]. The Industrial Robot, 1984, 11(4):238-242.

[2] De Luca A, Mattone R. Sensorless robot collision detection and hybrid force/motion control[C]. Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. Barcelona, Spain: IEEE, 2005: 999–1004.

[3] Lee H, Park J. An active sensing strategy for contact location without tactile sensors using robot geometry and kinematics[J]. Autonomous robots, 2014, 36(1-2): 109-121.

[4] Hogan N. Impedance control: An approach to manipulation: Part I—Theory[J]. 1985. Xxx

[5] Wu B, Qu D, Xu F, et al. Industrial robot high precision peg-in-hole assembly based on hybrid force/position control[J]. Journal of Zhejiang University(Engineering Science), 2018, v.52;No.334(02):178-185.

[6] Zhang K, Shi M H, Xu J, et al. Force control for a rigid dual peg-in-hole assembly[J]. Assembly Automation, 2017, 37(2):200-207.

[7] Petrović P B, Milačić V R. A concept of an intelligent fuzzy control for assembly robot[J]. CIRP Annals, 1998, 47(1): 9-12.

[8] Hou Z, Philipp M, Zhang K, et al. The learning-based optimization algorithm for robotic dual peg-in-hole assembly[J]. Assembly Automation, 2018, 38(4): 369-375.

[9] Zhu Z, Hu H. Robot learning from demonstration in robotic assembly: A survey[J]. Robotics, vol. 7, no. 2, p. 17, 2018.

[16] Gullapalli V, Grupen R A, Barto A G. Learning reactive admittance control[C]. Proceedings1992 IEEE International Conference on Robotics andAutomation. Nice, France: IEEE, 1992: 1475–1480 vol.2.

[17] Nuttin M, Van Brussel H. Learning the peg-into-hole assembly operation with a connectionist reinforcement technique[J]. Computers in Industry, 1997, 33(1): 101–109.

[18] Inoue T, Magistris G D, Munawar A, et al. Deep reinforcement learning for high precision assembly tasks[C]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS).Vancouver, BC, Canada, 2017: 819–825.

[19] Jing X , Zhimin H , Wei W , et al. Feedback Deep Deterministic Policy Gradient with Fuzzy Reward for Robotic Multiple Peg-in-hole Assembly Tasks[J]. IEEE Transactions on Industrial Informatics, 2018:1-1.

[10] Abu-Dakka F J , Nemec B , Kramberger A , et al. Solving peg-in-hole tasks by human demonstration and exception strategies[J]. Industrial Robot:An International Journal, 2014, 41(6):575-584.

[11] Aljaž Kramberger, Gams A , Nemec B , et al. Generalization of orientation trajectories and force-torque profiles for robotic assembly[J]. Robotics &Autonomous Systems, 2017, 98.

[12] Park D H, Hoffmann H, Pastor P , et al. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields[C]// Humanoids -ieee-ras International Conference on Humanoid Robots. IEEE, 2008.

[13] Tang T, Lin H C, Tomizuka M. A Learning-Based Framework for Robot Peg-Hole-Insertion[C]. ASME 2015 Dynamic Systems and Control Conference. 2015.

[14] Fei Y, Zhao X. An Assembly Process Modeling and Analysis for Robotic Multiple Peg-in-hole[J]. Journal of Intelligent and Robotic Systems: Theory and Applications, 2003, 36(2):175-189.

[15] Wan A, Xu J, Chen H, et al. Optimal Path Planning and Control of Assembly Robots for Hard-Measuring Easy-Deformation Assemblies[J]. IEEE/ASME Transactions on Mechatronics, 2017, 22(4):1600-1609.