



Deep Visual-guided and Deep Reinforcement Learning Algorithm Based for Multip-Peg-in-Hole Assembly Task of Power Distribution Live-line Operation Robot

Li Zheng¹ · Jiajun Ai¹ · Yahao Wang¹ · Xuming Tang² · Shaolei Wu³ · Sheng Cheng⁴ · Rui Guo⁵ · Erbao Dong¹

Received: 25 April 2023 / Accepted: 28 February 2024
© The Author(s) 2024

Abstract

The inspection and maintenance of power distribution network are crucial for efficiently delivering electricity to consumers. Due to the high voltage of power distribution network lines, manual live-line operations are difficult, risky, and inefficient. This paper researches a Power Distribution Network Live-line Operation Robot (PDLOR) with autonomous tool assembly capabilities to replace humans in various high-risk electrical maintenance tasks. To address the challenges of tool assembly in dynamic and unstructured work environments for PDLOR, we propose a framework consisting of deep visual-guided coarse localization and prior knowledge and fuzzy logic driven deep deterministic policy gradient (PKFD-DPG) high-precision assembly algorithm. First, we propose a multiscale identification and localization network based on YOLOv5, which enables the peg-hole close quickly and reduces ineffective exploration. Second, we design a main-auxiliary combined reward system, where the main-line reward uses the hindsight experience replay mechanism, and the auxiliary reward is based on fuzzy logic inference mechanism, addressing ineffective exploration and sparse reward in the learning process. In addition, we validate the effectiveness and advantages of the proposed algorithm through simulations and physical experiments, and also compare its performance with other assembly algorithms. The experimental results show that, for single-tool assembly tasks, the success rate of PKFD-DPG is 15.2% higher than the DDPG with functionized reward functions and 51.7% higher than the PD force control method; for multip-tools assembly tasks, the success rate of PKFD-DPG method is 17% and 53.4% higher than the other methods.

Keywords Power distribution live-line operation robot · Peg-in-hole assembly · Deep reinforcement learning · Admittance control · Fuzzy logic drive

1 Introduction

With the development of automation technology and artificial intelligence, robots have been gradually being applied to high-risk electrical maintenance tasks such as detection, installation and disassembly due to its exceptional efficiency, low accident rate, and high flexibility, with the expectation of replacing power repairman [1–3]. For the power system, the use of Power Distribution Live Operation Robot (PDLOR) has improved the level of automation and maintenance, as

shown in Fig. 1a and b. In live-line operations in distribution networks, many types of tasks such as live-line earthing operation, install arrester operation, and replacement operation tools can be abstracted as peg-in-hole assembly tasks, as shown in Fig. 1c, d, e, respectively. Peg-in-hole assembly is a common task, but the automation of the multip-peg-in-hole high-precision assembly process remains a challenge. The force control algorithm is the main approach for peg-in-hole assembly tasks, but it is difficult to apply in scenarios with complex contact models for multip-peg-in-hole assembly: adjusting controller parameters based on new assembly tasks requires a significant amount of time and effort [4]. Therefore, an advanced intelligent algorithm that does not rely on physical contact model analysis is required to effectively perform multip-peg-in-hole high-precision assembly tasks in unstructured power distribution live-line working scenarios.

Xuming Tang, Shaolei Wu, Sheng Cheng and Rui Guo contributed equally to this work

✉ Erbao Dong
ebdong@ustc.edu.cn

Extended author information available on the last page of the article

Fig. 1 Power distribution live-line operation robot and related peg-in-hole assembly works. (a) Robot operations in the distribution network and power distribution live-line operation robot; (b) Power distribution live-line operation robot. (c) Live-line earthing operation; (d) Install arrester operation; (e) Replacement operation tools



In recent years, with the rise of machine learning, artificial intelligence algorithms have played an important role in the industrial field [5, 6]. Inspired by the fact that humans can accomplish various assembly tasks through learning rather than analyzing the contact status and force of the model, the multip-peg-in-hole high-precision assembly problem of PDLOR can be combined with optimization algorithms based on reinforcement learning (RL) and deep learning (DL). RL algorithms involve developing an intelligent agent that interacts with the environment to generate sufficient training data through trial and error. The agent learns the desired strategy by maximizing the rewards obtained from the environment. When combined with the powerful non-linear fitting capability of DL, deep reinforcement learning (DRL) algorithms can effectively control the robot to output continuous and multidimensional actions in tasks with continuous action spaces. This enables PDLOR to autonomously handle the challenges of multip-peg-in-hole high-precision assembly of tools with dynamic and uncertain contact poses.

To fulfill the requirements of intricate and diverse high-risk live working maintenance tasks, PDLOR necessitates coordinated utilization of multiple tools during maintenance operations, thereby mandating the capacity for autonomous tool replacement. Therefore, this paper focuses

on researching multip-peg-in-hole assembly problem in automatic assembly operation tools based on an intelligent PDLOR system. The operational environment of PDLOR is typically dynamic and unstructured, with uncertain changes in the pose of the operation tools due to the vibration of the insulated bucket truck and deformation of the tool frame, which makes the assembly challenging. In this paper, we propose a multip-peg-in-hole assembly algorithm framework based on deep visual-guided coarse positioning and prior knowledge and fuzzy logic driven deep deterministic policy gradient (PKFD-DPG) high-precision assembly, targeting the challenge of adaptive assembly of operation tools for PDLOR. This algorithm enables robots to autonomously learn advanced assembly strategies in unstructured environments of electrified operations. During the initial stage, the distance and relative position between PDLOR's manipulators and the tool hole are uncertain. We first use a deep visual-guided algorithm based on MS-YOLOv5 to guide the manipulators to close quickly the position of the tool hole, reducing ineffective exploration in the early stage of training. When the distance between pegs and holes reaches a preset threshold, PDLOR executes multip-peg-in-hole assembly process based on the DRL-based variable admittance control strategy, reducing the number of training experiments

and avoiding risky exploration actions. In addition, a main-auxiliary combined reward function system is proposed, which evaluates the assembly process based on the main-line reward using hindsight experience replay and the auxiliary reward using fuzzy logic inference, effectively improving the convergence efficiency of the algorithm. The main contributions of this paper are as follows:

1. A long-distance multip-peg-in-hole assembly algorithm framework is proposed, which achieves complex PDLOR assembly task through deep visual-guided coarse localization and variable admittance control based on DRL high-precision assembly. This algorithm completes assembly tasks through robot-environment interaction learning without analyzing contact states.
2. A multiscale identification and localization network based on YOLOv5 (MS-YOLOv5) is proposed, which can quickly and accurately obtain the spatial position of pegs and holes centers. The deep visual-guided coarse localization algorithm based on MS-YOLOv5 guides the manipulator to close the tool quickly when the distance is far, effectively reducing the large amount of ineffective exploration in the assembly task.
3. A combined main-auxiliary reward system is designed, where the main-line reward uses the HER mechanism, and the auxiliary reward is based on fuzzy inference. A multilayer perceptron (MLP) is used to establish a non-linear mapping from actions and states to rewards, which solves the problems of sparse rewards and binary rewards in the robot learning process.
4. A digital twin model is built based on the *CoppeliaSim* simulation software for training and evaluation of the proposed algorithm. The effectiveness and practicality of the algorithm are demonstrated in the experiment and application of PDLOR tool autonomous assembly.

The remainder of this paper is structured as follows. Section 2 introduces relevant previous works. Section 3 provides a detailed description of the implementation process and network structure of the deep visual-guided coarse positioning algorithm based on MS-YOLOv5. In Section 4, we introduce a DRL method based on prior knowledge and fuzzy logic to accomplish peg-in-hole assembly tasks. In Section 5, we describe the construction and training process of the simulation model, analyse the results, and validate and evaluate the approach in real-world tasks. Section 6 concludes the paper and discusses some further works.

2 Related Work

With the continuous development of robot technology, robots have made significant advancements in accuracy, degrees

of freedom, and operational performance. Automated peg-in-hole assembly has been widely applied in industrial manufacturing based on robots. The solution strategies for automated robot assembly mainly include the hole-seeking strategy, vision-based servo control strategy, force/torque servo control strategy, and learning-based control strategy.

Robots can search around the target hole along a certain trajectory, which is known as the hole-seeking strategy. Chhatpar et al. [7] proposed a method that uses concentric circular search trajectories within a specific radius and search trajectories generated by a simulated annealing algorithm. They also described a tilted strategy for guided assembly to enhance blind search and facilitate assembly. In [8, 9], Kang et al., and Jasim et al. were inspired by the process of humans inserting plugs into sockets, and proposed an intuitive strategy of using uncertain-driven spiral trajectories around the hole to search for the hole. This strategy does not require precise knowledge of the hole's position or force/torque sensors. Chen et al. [10] proposed tilted and binary search strategies based on the actual assembly scenario, and achieved smooth insertion through moment control during the insertion phase, effectively reducing the search time and range. Although hole-seeking strategies are simple and easy to implement, they have low search efficiency and high randomness. Park et al. [11] proposed a compliant peg-hole assembly method based on spiral force trajectory (SFT) blind search. Instead of using the whole spiral trajectory, this method introduces a partial spiral trajectory within the sector to search.

Some researchers have also started to equip robots with environmental perception capabilities using vision sensors, which use RGB and depth information obtained from visual sensors to identify and locate target holes, and then guide the robot in assembly based on visual feedback. Abu et al. [12] and Park et al. [13] estimated the spatial pose between assembly parts using point cloud information obtained from depth cameras to complete assembly tasks. Jiang et al. [14] and Xu et al. [15] used a multilevel hybrid vision system to achieve fine alignment and assembly accuracy in robot peg-in-hole assembly. Lu et al. [16] proposed a coarse to fine visual servo (CFVS) pin hole method to achieve 3-DoF end-effector motion control based on 6D visual feedback. CFVS can handle arbitrary tilt angles and large initial alignment errors by performing fast attitude estimation prior to refinement. Yasutomi et al. [17] introduced a visual and proprioceptive data-driven robot peg-hole assembly control model. The proposed model can be adapted to different initial position and hole conditions. However, in the practical use of visual servoing, issues such as camera calibration errors, insufficient camera resolution, lighting interference, and occluded contact status often lead to inaccurate positioning of the visual system, making it difficult to achieve satisfactory servoing performance.

Currently, force/torque servo control is a major control strategy for automated robot assembly processes in industrial. The basic principle is to make decisions and control the robot's motion trajectory based on the magnitude of the contact force [18–20]. Some researchers have also proposed mixed assembly strategies based on visual recognition for rough positioning and force/torque perception for fine positioning to solve the problem of assembling complex-shaped objects [21–25]. Compared to algorithms based on visual sensors, force/torque servo control methods can improve the compliance of the robot manipulator and avoid excessive contact force on the environment. However, force/torque servo control methods have low efficiency and high hardware costs, making it difficult to perform mechanical modeling and optimize impedance parameters. They may also struggle to achieve assembly accuracy in complex or heavily interfered multip-peg-in-hole assembly tasks.

In recent years, the application of DRL algorithms in robot has gained attention from many researchers. Based on DRL, by developing an agent that interacts with the environment to learn assembly strategies, an effective and feasible approach has been provided for robot and assembly tasks. Fan et al. [26], Leyendecker et al. [27], Petrovic et al. [28], and Inoue et al. [29], among others, have used DRL frameworks to learn peg-in-hole assembly strategies in simulated environments and validated them on real robot manipulators using sim-to-real techniques. Xie et al. [30], Schoettler et al. [31], and Wang et al. [32], proposed visual-guided frameworks based on the imitation of human overt visual feedback behaviors, and designed RL-based controllers to achieve submillimeter-level unseen shape generalization for peg-in-hole assembly and complex industrial assembly tasks. In addition, Arik et al. [33] and Beltran et al. [34], combined RL techniques with traditional force control, and proposed a DRL-based force control framework for peg-in-hole assembly tasks.

However, most of the existing research on peg-in-hole assembly based on learning methods are limited to fixed scenarios with simple tasks. When implementing DRL algorithms for autonomous multip-peg-in-hole assembly processes with strong coupling and dynamic uncertainty in contact poses for PDLOR, there are still several challenges:

1. The working environment for PDLOR is unstructured, and PDLOR needs to adapt to various operating tools with uncertain poses of the end-flange of the manipulator and the end-flange of the tool.
2. Multip-peg-in-hole assembly tasks have strong coupling, making the assembly process more complex.
3. Mapping multidimensional and continuous observations of the actual multip-peg-in-hole assembly task to multidimensional and continuous actions requires a large amount of experimental data to train the agent, which can be extremely costly or not allowed.

4. In the exploration process, unstable training strategies in the early learning stages may result in a large number of ineffective explorations, leading to low sample efficiency of the algorithm.
5. It is difficult to find a perfect reward function to evaluate the long-term desirability and safety of exploration actions in the assembly process, which makes it challenging for the learning process to converge.

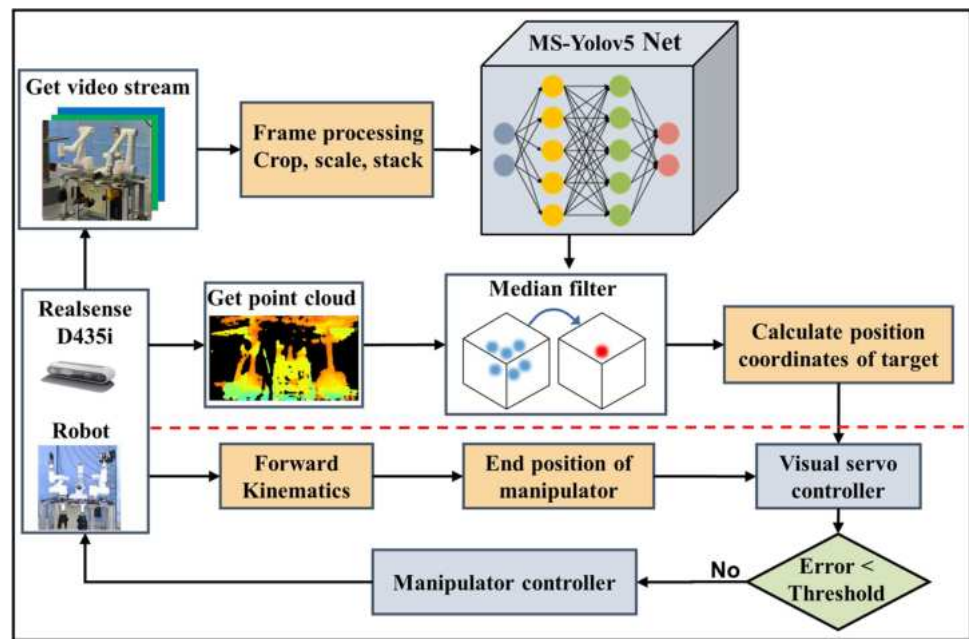
3 Deep Visual-Guided Rough Localization Algorithm

Compared to conventional assembly tasks, PDLOR faces the challenge of dealing with large distances between pegs and holes during tool assembly. To complete autonomous assembly tasks, PDLOR requires the ability to recognize and locate the end flanges of both the work tool and the manipulator in space. The visual system guides PDLOR's manipulators based on perception of the work environment and visual feedback, which reduces unnecessary exploration and accelerates the learning process. This section focuses on how to achieve deep visual-guided for PDLOR's coarse localization through the proposed MS-YOLOv5. First, we discuss the challenges of deep visual-guided and the currently used methods for solving them. Then, we elaborate on the proposed MS-YOLOv5 network structure and compare its performance with that of several classic object recognition algorithms. Finally, we introduce the deep visual-guided coarse localization algorithm based on the MS-YOLOv5 network, as shown in Fig. 2.

3.1 Basics on Object Detection

In the initial stages of PDLOR's manipulator and tool assembly, due to the distance between pegs and holes being relatively far apart, deep visual-guided technology is used to gradually bring the manipulator end effector closer to the tool. However, it is difficult to detect features because of the small size of the end flange pegs and tool flange holes, as well as the scarcity of surface feature points. Convolutional neural networks (CNNs), by extracting image features through deep convolutional neurons, can effectively address the challenge of detecting objects. In object detection, the application of CNNs can be categorized into two types: region proposal-based and regression-based. For region-based algorithms such as Faster R-CNN [35] and Mask R-CNN [36], the detector first identifies a set of candidate regions of interest (ROIs), which are then fed into a convolutional neural network for classification and bounding box regression. In contrast, regression-based algorithms, such as the YOLO series [37–39], are end-to-end processes that can directly regress the object's classification and position without the

Fig. 2 The framework deep visual-guided coarse localization algorithm based on the MS-YOLOv5 network



need to extract candidate regions of interest, resulting in faster runtime speeds.

3.2 Multiscale Identification and Localization YOLOv5

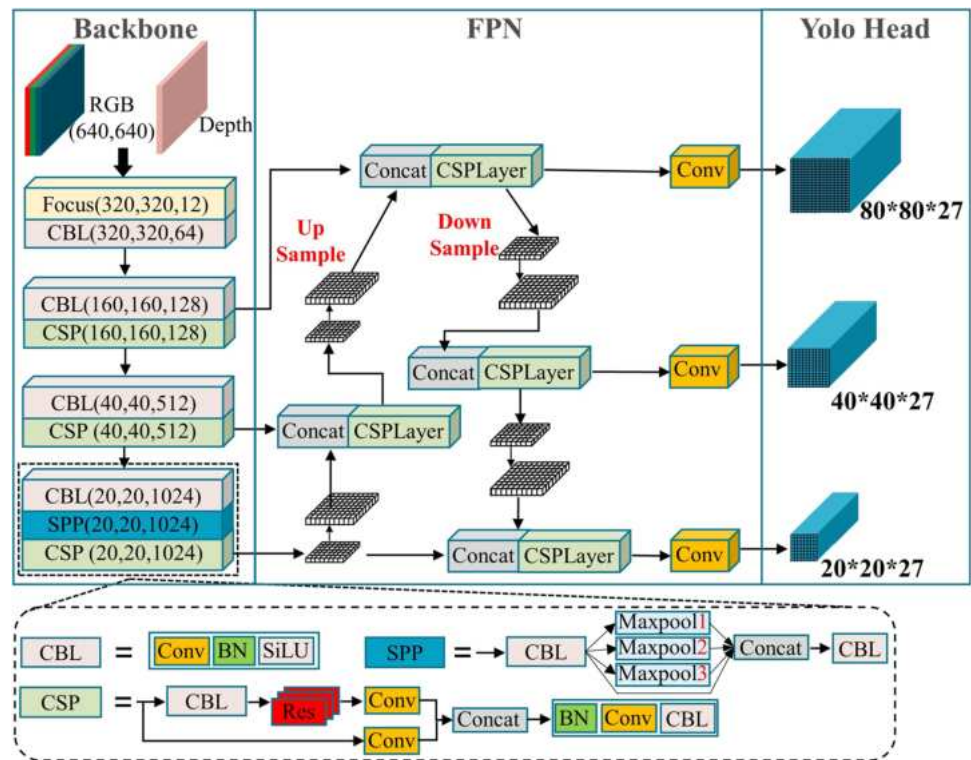
Compared to YOLOv3 and YOLOv4, YOLOv5 has adjusted its network structure and utilized mosaic data augmentation and adaptive image scaling techniques, which enable training with smaller batches of data and improve the accuracy of detecting small-scale objects. This paper proposes a multiscale identification and localization network based on YOLOv5 (MS-YOLOv5), which is designed to address the characteristics of small-scale, weak-texture of flange pegs at the end of manipulators and tools flange holes. MS-YOLOv5 performs depth estimation on the selected object detection bounding boxes, enabling quick and accurate determination of the spatial 3D position of small targets. MS-YOLOv5's network structure consists of Backbone, FPN, and YOLO Head. The Backbone serves as the main feature extraction network of MS-YOLOv5, which extracts three feature layers of different resolutions from the input image for subsequent feature extraction and processing. As an enhanced feature extraction network in MS-YOLOv5, FPN performs multi-scale feature fusion on the three effective feature layers to improve detection performance and accuracy. YOLO Head is the classifier and regressor of MS-YOLOv5, which generates object detection boxes containing information such as object size, object position, and probabilities for each class obtained through logistic regression by evaluating feature points.

The MS-YOLOv5 network structure is shown in Fig. 3. The CBL module consists of convolution, convolution, normalization, activation function, and SiLU activation function. The CSP module represents the structure of the cross-stage partial. The SPP module extracts features by performing max pooling with different kernel sizes, thereby increasing the receptive field of the network. Collecting target object images that include the surrounding scene and manually annotating them to create a training dataset is necessary. The dataset consists of 600 images and contains four categories: Flange, Peg, Hole, and Rack, as shown in Fig. 4.

We conducted object detection experiments on the constructed dataset for the four classes of objects using MS-YOLOv5, Faster R-CNN, SSD, YOLOv3, and YOLOv4. To evaluate the performance, we employed a 5-fold cross-validation method. The dataset was divided into 5 equally sized subsets. For each algorithmic experimental condition, we conducted 5 experiments to ensure that each subset was used as the testing set once and the remaining subset was used as the training set. The results are shown in Table 1.

To evaluate the performance of the object recognition algorithms, TP and TN represent the number of pixels correctly predicted as object and background, respectively, while FP and FN represent the number of pixels incorrectly classified as object and background, respectively. We calculated precision, recall, sensitivity, F1 score accuracy according to Eqs. 1, 2, 3. With Recall as the horizontal axis and precision as the vertical axis, the PR curve of each target object was drawn, and the AP value of each target object was calculated.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Fig. 3 MS-Yolov5 networks

$$Recall = Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 \times \frac{TP}{2 \times TP + FP + FN} \quad (3)$$

It can be demonstrated from Table 1 and Fig. 5a that the MS-YOLOv5 algorithm achieved state-of-the-art performance, especially in the recognition accuracy of small objects such as holes and pegs, which was significantly superior to the other four algorithms. The precision-recall (PR) curves for the four target objects in the 5-fold cross-validation experiments are shown in Fig. 5b, c, d, e. The PR curves of MS-YOLOv5

encompass the curves of the other four algorithms, indicating that MS-YOLOv5 performs better in peg-hole detection.

3.3 Deep Visual-guided Coarse Localization Algorithm

To enhance the performance of deep visual coarse localization, our PDLOR uses a Real Sense Depth Camera D435i to capture RGB images for each frame. After processing the RGB images with depth information through cropping, resizing, and stacking, we input them into the trained MS-YOLOv5 network to detect the two-dimensional position of the target objects by bounding boxes. In addition, some

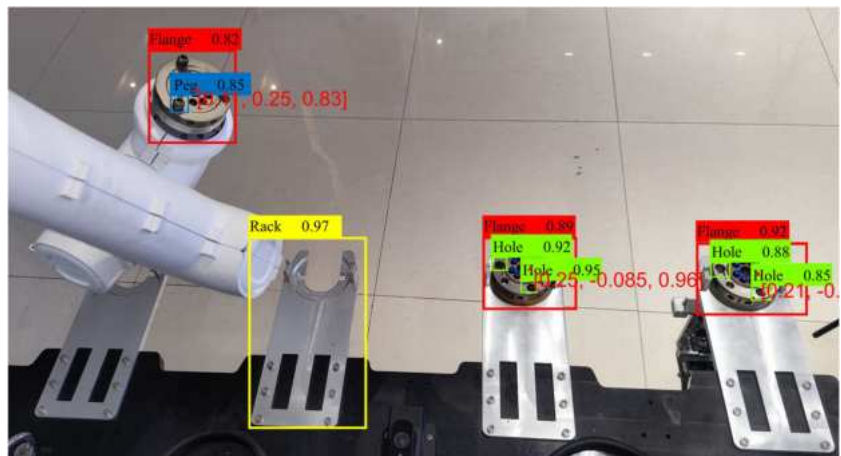
Fig. 4 Definition of dataset category labels

Table 1 Comparison of object detection performance of different algorithms under 5x cross validation experiment

Algorithm	AP				mAP	Precision	Sensitivity	F1-score
	Flange	Hole	Peg	Rack				
Faster R-CNN	73.15% ±2.58%	58.10% ±2.23%	66.32% ±1.79%	77.32% ±2.16%	68.72%	68.06%	75.11%	71.25%
SSD	82.49% ±1.89%	81.51% ±1.73%	71.28% ±1.56%	84.40% ±2.03%	79.92%	81.96%	81.19%	87.25%
YOLOv3	77.73% ±2.22%	71.28% ±2.36%	80.79% ±1.72%	81.38% ±2.24%	77.80%	75.81%	78.30%	83.28%
YOLOv4	86.34% ±2.13%	76.76% ±2.15%	77.85% ±1.64%	87.41% ±2.19%	82.09%	82.16%	83.53%	88.50%
MS-YOLOv5	92.27% ±1.63%	92.07% ±0.87%	86.97% ±1.21%	93.63% ±1.79%	91.24%	91.63%	96.25%	95.27%

random points are sampled around the center point of the target objects boxes, and the depth of these random points is median-filtered. This can avoid sudden transitions in depth values for certain points during the depth measurement process, where the depth value may be erroneously measured as 0. This greatly improves the stability and safety of the measured depth values around the center of the target during the measurement process. The obtained 3D position of the target

objects relative to the camera coordinate system is then transformed into the robot's base coordinate system. The overall framework of the PDLOR's deep visual-guided coarse positioning algorithm is shown in Fig. 2, where the manipulator is controlled by the deep visual-guided controller to close the tool flange until the deviation between the peg and hole position is within the threshold.

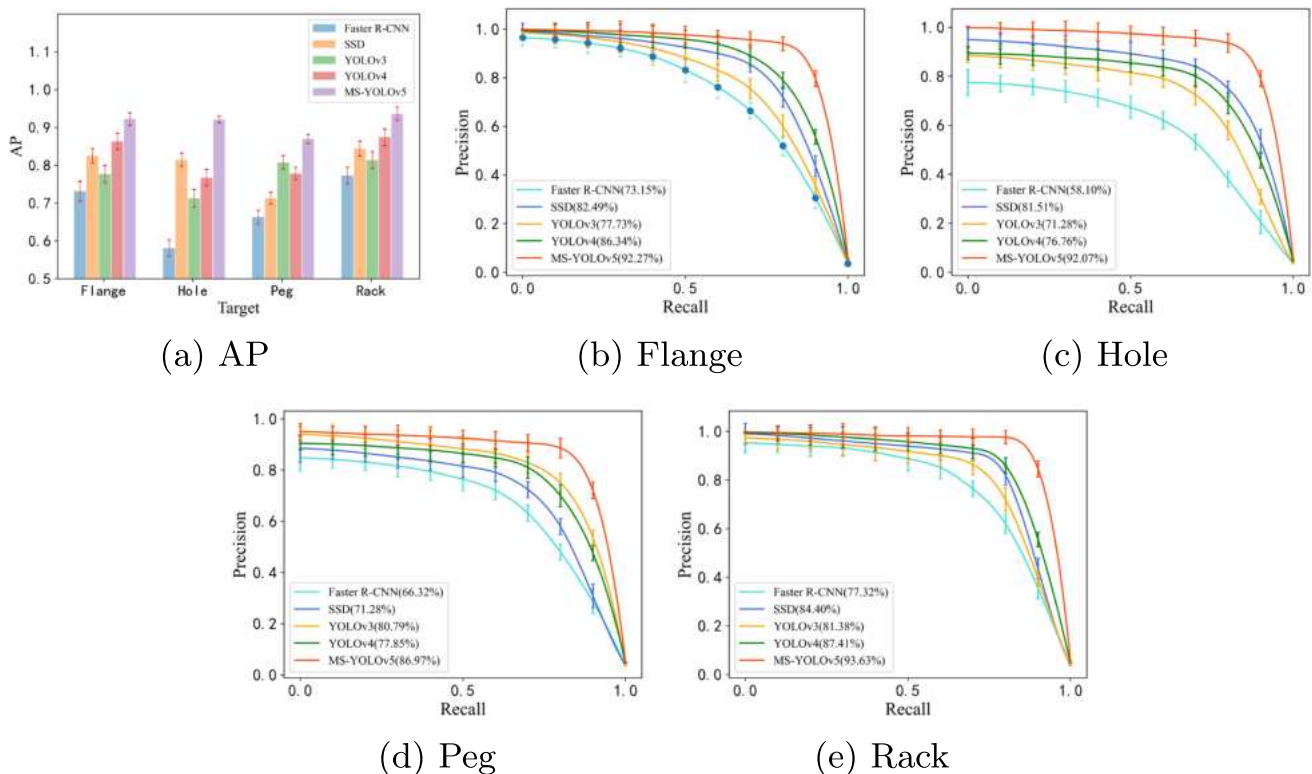


Fig. 5 Results of object recognition experiments under 5-fold cross-validation for different algorithms. (a) Comparison of average precision (AP) scores for the recognition of the four classes of objects by

the five algorithms. (b), (c), (d), and (e) show the precision-recall curves for the four classes of object recognition by the five algorithms

4 Prior Knowledge and Fuzzy Logic Driven Deep Deterministic Policy Gradient

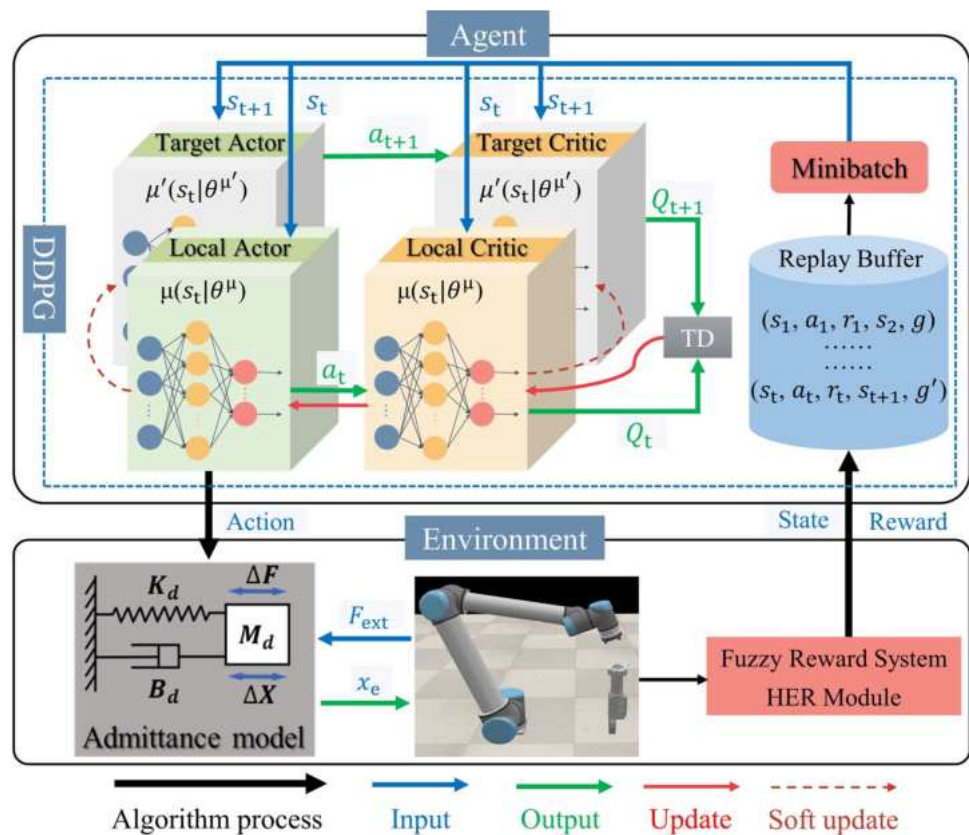
Deep visual-guided coarse positioning based on the Real Sense Depth Camera D435i has the ability to detect and estimate the positions of the pegs and holes. However, the deep visual-guided has certain limitations in terms of accuracy, as it cannot provide sufficiently precise spatial information. Moreover, when the robot manipulator end effector closes the tool holes, the visual signal from the camera may be lost, making it difficult to achieve high-precision multip-peg-in-hole assembly tasks solely relying on vision. Therefore, this section focuses on discussing how to accomplish precise multip-peg-in-hole assembly tasks through reinforcement learning methods driven by prior knowledge and fuzzy logic inference. First, we define the representation of the task for assembly of tool for PDLOR, and describe the action and state spaces for the assembly task. Second, we introduce a priori knowledge and fuzzy logic driven variable impedance control strategy, followed by a description of the main-auxiliary combined reward system design process for the algorithm. Finally, we present the pseudocode description and related details of the proposed PKFD-DPG algorithm. The overall framework is illustrated in Fig. 6.

4.1 Basics on Deep Reinforcement Learning

For the PDLOR tool assembly task, the goal is to develop an agent that can learn assembly strategies through interaction with the operating environment. The process of the agent performing the tool assembly task can be described as a Markov decision processes (MDPs) [40]: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, state space \mathcal{S} , action space \mathcal{A} , reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, state transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, represents the probability of transitioning between states. The assembly strategy of the agent can be represented as a mapping from states to actions $\pi : \mathcal{S} \rightarrow \mathbb{A}$. At each time step t , the agent selects and executes an assembly action $a_t \in \mathcal{A}$ based on the current state s_t . The environment transitions to a new state $s_{t+1} \in \mathcal{S}$ based on the state transition function $\mathcal{P}(s_{t+1} | s_t, a_t)$, while also receiving a scalar reward r_t , which represents a quantified evaluation of the action-state value. The agent improves the learned strategy by maximizing the sum G_t of the expected rewards r_t for future steps, where G_t is shown in the following equation:

$$G_t = r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} \quad (4)$$

Fig. 6 Autonomous tool assembly control strategy for PDLOR based on the PKFD-DPG algorithm



The $\gamma \in [0, 1)$ is the discount factor. The Q-function or action-value function is defined as $Q^\pi(s_t, a_t) = \mathbb{E}[G_t | s_t, a_t]$. Let π^* denote an optimal policy i.e. any policy π^* s.t. $Q^{\pi^*}(s, a) > Q^\pi(s, a)$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$ and any policy π . All optimal policies have the same Q-function which is called the optimal Q-function and denoted Q^* . It is easy to show that it satisfies the following Eq. 5 called the Bellman equation [40]:

$$Q^*(s, a) = \mathbb{E}_{s' \sim p(\cdot | s, a)} [r(s, a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')] \quad (5)$$

4.2 RL Algorithm of Variable Admittance Control

During the tool changing and assembly task of PDLOR, even slight uncertainty in the position of the pegs and holes can result in significant and unpredictable contact forces. To improve the compliance of PDLOR during tool assembly tasks and address the complex contact situations in live-line working scenarios, we designed an adaptive variable admittance controller to solve the tool assembly task. We first designed and implemented impedance control in the Cartesian space of the end-effector of the PDLOR's manipulator. Based on the impedance model, the manipulator can adjust its position and orientation according to the external forces applied at the end-effector, thereby avoiding blockages between pegs and holes or damage to the robot system due to excessive contact. The impedance controller can be established as follows:

$$M\ddot{x}_e + B\dot{x}_e + Kx_e = F_{ext} \quad (6)$$

where M , B , and K represent the required inertia, damping, and stiffness matrices, respectively. The vector F_{ext} denotes the measured actual contact force by the six-axis force sensor located at the end-effector of the manipulator, which is compensated for by gravity. The vector $x_e = [\delta p, \delta o]$ corresponds to the pose gain of the manipulator's end-effector.

We formulate the adaptive parameter tuning of the variable admittance control as a markov decision process (MDP), and the parameters of the inertia, damping, and stiffness matrix of the admittance model are learned by DRL. As shown in Fig. 6, the agent interacts with the environment, utilizing the DDPG algorithm to optimize the strategy and continuously identify the parameters of the admittance model. During the tool assembly task of PDLOR, the tool end flange holes are usually installed on the base platform of the robot workspace, and an ATI six-axis force sensor is installed between the manipulator end joint and the flange pegs to measure the assembly force and torque. And we optimize the pose gain x_e of the manipulator's end-effector to move the end flange pegs by adjusting the model parameters of the variable admittance controller, as shown in Fig. 7. To avoid excessive controllable

parameters that may cause instability of the robot system, we reduce the number of controllable parameters and lower the system's complexity based on prior knowledge. Specifically, we set each direction's inertia parameter M to a constant value and define a constant damping ratio $\zeta = B_d/2\sqrt{K_d M_d}$ to calculate the damping coefficient relative to the inertia and stiffness parameters. Therefore, the action space a_t can be defined by a 6-D vector of stiffness parameters, as shown in Eq. 7:

$$a_t = [K_d^x, K_d^y, K_d^z, K_d^\alpha, K_d^\beta, K_d^\gamma] \in \mathbb{R}^6 \quad (7)$$

During the tool assembly process, the state space s_t of the agent can be defined by a 12-D vector, as shown in Eq. 8:

$$s_t = [p_t^x, p_t^y, p_t^z, o_t^x, o_t^y, o_t^z, F_t^x, F_t^y, F_t^z, M_t^x, M_t^y, M_t^z] \in \mathbb{R}^{12} \quad (8)$$

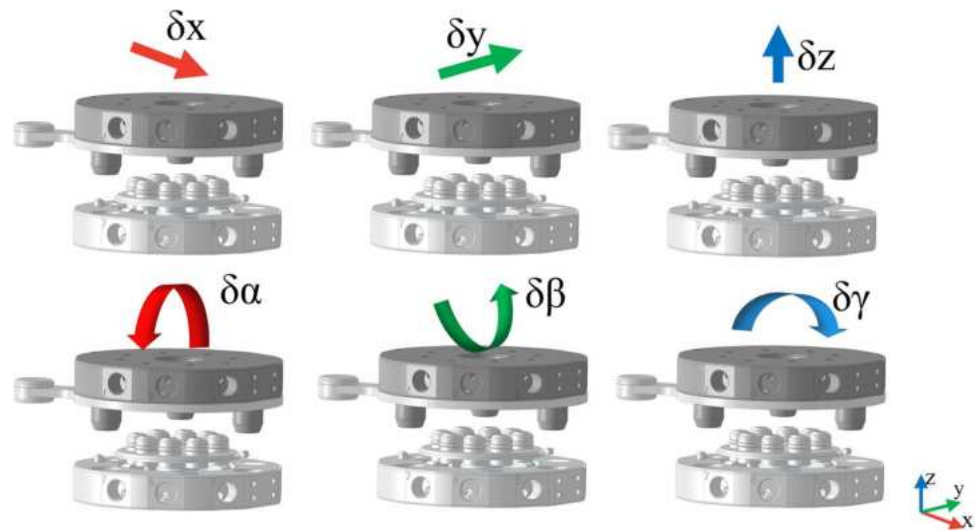
where p and o are the current position and orientation of the end effector flange in the PDLOR base coordinate system, which can be calculated by the robot forward kinematics; F and M are the force and torque measured by the six-axis force sensor.

4.3 Fuzzy Reward Function System and Hindsight Experience Replay

Reward sparsity is one of the major challenges that make it difficult to efficiently apply reinforcement learning in robotic tasks. Effective reward shaping can significantly enhance the convergence rate of the DRL algorithm [41, 42]. Indeed, it is difficult to implement a single explicit function expression as the reward function for the task of autonomous assembly of tools by PDLOR, where there are multiple criteria or factors that can affect or evaluate the quality of the assembly action. Therefore, this paper proposes a main-auxiliary combined reward system, which addresses the reward engineering problem from two perspectives. The main-line reward is based on the method of hindsight experience replay, which constructs a replay buffer to utilize the vast majority of failed exploration attempts of the manipulator. The auxiliary rewards are designed using a fuzzy feedback reward system, which establishes a nonlinear mapping from the agent's state and action to the reward using a three-layer MLP, and utilizes fuzzy logic inference reward as feedback for the MLP. This reward mechanism effectively adapts to the complexity and variability of the tasks, thereby improving the performance and adaptability of PDLOR. The main-auxiliary combined reward system is denoted by Eq. 9:

$$R = r_g + (1 - \mu)r_p + \mu r_f \quad (9)$$

Fig. 7 Relative motion between the flange pegs and holes



where the parameter $\mu = e^{-\lambda t}$ represents the decay coefficient, λ denotes the decay rate, and t refers to the current training iteration. During the training progresses, the decay coefficient gradually decreases to 0, and the predicted reward r_p constructed by the MLP gradually replaces the fuzzy feedback reward r_f . This approach significantly improves the perceptual capability of the reward system while reducing the system complexity in the later stages of training.

4.3.1 Hindsight Experience Replay

To improve the efficiency of utilizing the majority of failed exploration experiences, the hindsight experience replay (HER) technique [43] has been proposed in the RL community to improve the sample utilization efficiency of off-policy DRL algorithms. HER adds a goal space $g \in \mathcal{G}$ to the MDPs without altering the dynamics of the MDPs process. As for $\forall g \in \mathcal{G}$, $r_g : \mathcal{S} \rightarrow \{0, 1\}$, the goal of the agent is to reach a state such that $r_g : \mathcal{S} = 1$ s.t. $f_g : \mathcal{S} = [s = g]$, where the value equals 1 only when the state exactly equals the target, otherwise it is 0. For a given state s , there is a corresponding goal g mapped to it i.e. $m : \mathcal{S} \rightarrow \mathcal{G}$ s.t. $\forall s \in \mathcal{S}$, $r_{m(s)}(s) = 1$, and in HER, failed states are used as new goals for experience replay. In the task of learning autonomous assembly of tools by PDLOR, the state space of the agent consists of the position and orientation of the end effector flange pegs, as well as the force and torque measured by force sensors. The goal space is the pose of the tool flange holes in the robot base coordinate system. When the manipulator's end flange pegs reach the position $goal'$, the main-line reward function $r_g = r(s_t, a_t, g)$ is as shown in Eq. 10:

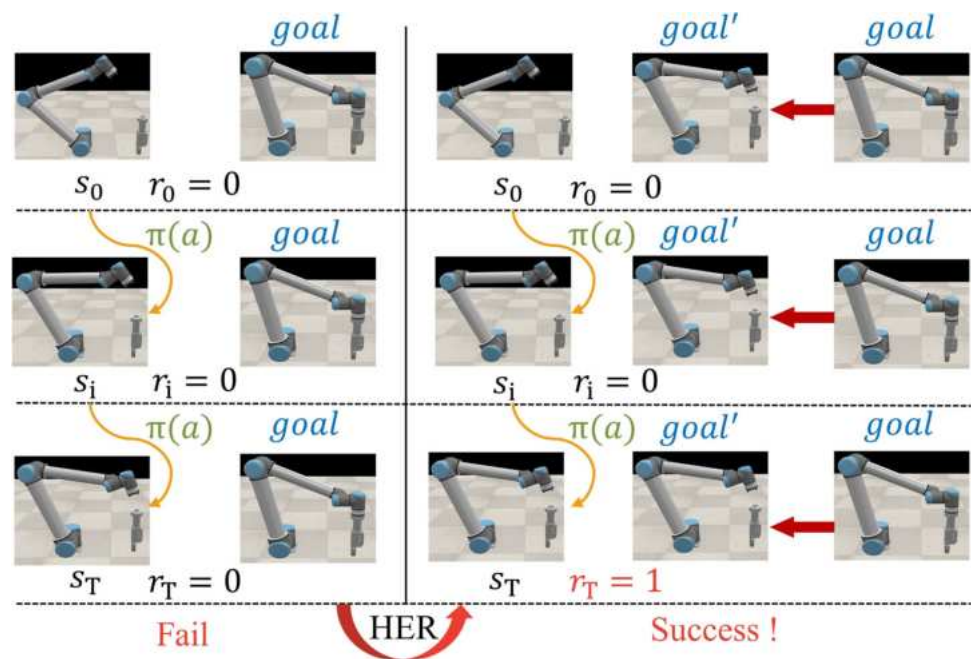
$$r_g = \begin{cases} 1, & |g' - s_T| < \epsilon \\ 0, & \text{Otherwise} \end{cases} \quad (10)$$

where the ϵ represents tolerance, which allows for a certain amount of deviation. The schematic process of the HER is shown in Fig. 8. First, we initialize an off-policy DRL algorithm A and clear the replay buffer \mathcal{D} . At each training episode, an initial state s_0 and a goal g_0 are uniformly sampled from the state space \mathcal{S} and the goal space \mathcal{G} , respectively. Then, during the environment steps $t = 1, 2, \dots, T$, the DRL algorithm A interacts with the environment to obtain transitions $\{(s_1, g_1, a_1, r_1, s_2), \dots, (s_T, g_T, a_T, r_T, s_{T+1})\}$. After each episode, the algorithm A stores the transitions from each training round in the replay buffer \mathcal{D} based on the knowledge in $\zeta = \{s_0, s_1, \dots, s_T\}$. And next a set of n additional goals $\varphi = \{g'_1, g'_2, \dots, g'_n\}$ uniformly sampled from the visited states $\zeta = \{s_0, s_1, \dots, s_T\}$ by HER. Afterward, these g' are iterated and $r'_T = r(s_t, a_t, g')$ is recalculated for each g' . And the g in the transition is replaced with g' as the new label to train the network, and then the new transition $(s_t, g'_t, a_t, r'_t, s_{t+1})$ is stored in the replay buffer \mathcal{D} . By converting failed experience samples into useful learning experience, HER technique can significantly reduce the trial-and-error required in the learning process, thus reducing the cost of learning and accelerating the learning process. HER process does not change the transition probability from the original state s_t to s_{t+1} when action a remains unchanged, which enhances the learning speed and success rate of the DRL algorithm.

4.3.2 Fuzzy Feedback Reward System

As there are many fuzzy criteria or factors that affect or evaluate the quality of assembly actions, using fuzzy logic inference to generate rewards provides an effective solution. However, it would become extremely complex due to the numerous fuzzy rules involved in the inference and defuzzification process by using fuzzy logic reasoning alone for the

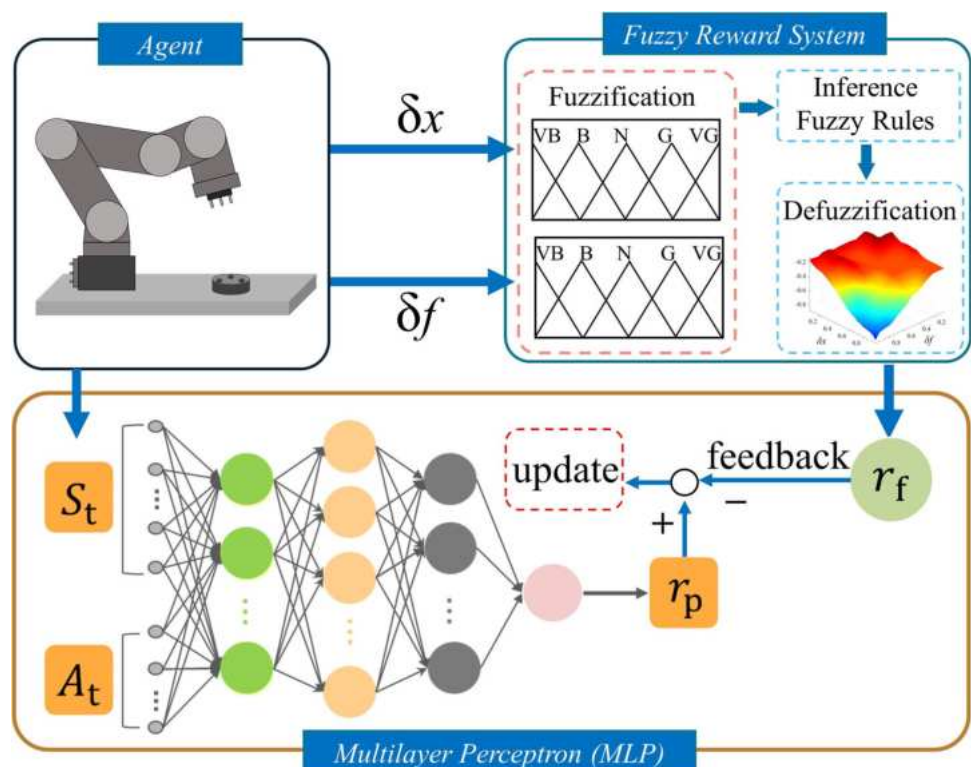
Fig. 8 The principle diagram of HER-based assembly task reward r_g



task of PDLOR autonomous assembly of tools. In addition, fuzzy logic inference is based on certain prior knowledge and uses linear membership functions, which may result in large gradients in the inference results and cause system instability. Therefore, a fuzzy feedback reward system is proposed in this paper as shown in Fig. 9, which uses an MLP with three hidden layers to construct a nonlinear mapping from the agent's

state and action to the reward. When the amount of system input increases, the computation amount of fuzzy system will have a significant gap [44], while the computation amount of MLP is very close. The reward r_f obtained from fuzzy logic inference is used as feedback for the MLP. It greatly increases the reward system's sensitivity to the agent's perception abilities without increasing the system's complexity. The reward

Fig. 9 Fuzzy feedback rewards system



r_f output from the fuzzy reward system is determined by four input variables of the system: the position deviation δp , orientation deviation δo , torque deviation $\delta \tau$, and force deviation δf , which are defined as follows equation:

$$\begin{aligned}\delta p &= [p_t^x - p_g^x, p_t^y - p_g^y, p_t^z - p_g^z]^\top, \\ \delta o &= [l * \theta, m * \theta, n * \theta]^\top, \\ \delta x &= \|\delta p, \delta o\|\end{aligned}\quad (11)$$

$$\begin{aligned}F &= [f_t^x, f_t^y, f_t^z, \tau_t^x, \tau_t^y, \tau_t^z]^\top, \\ \delta F &= \|F_t - F_{\max}\|\end{aligned}\quad (12)$$

where the symbol $\|\cdot\|$ denotes the Euclidean norm. The pose deviation δx is composed of the position deviation and the orientation deviation. The position deviation δp is obtained by subtracting the current position vector from the target position vector. The orientation deviation δo is obtained by multiplying the current orientation vector by the transpose of the target orientation vector and then converting it into an equivalent axis-angle representation. The force deviation is composed of the deviation between the current end-effector 3D force and 3D torque and their set maximum values.

Each input variable of the fuzzy reward system is divided into the same fuzzy set {VB, B, N, G, VG}, representing very bad, bad, neutral, good, and very good, respectively. To simplify the system complexity, the input variables of the system are reduced to two: the pose deviation δx , which consists of position deviation δp and orientation deviation δo , and the force deviation δF , which consists of torque deviation $\delta \tau$ and force deviation δf . The fuzzy reward value is obtained by inverse defuzzification through a defuzzifier (13) and is limited to $[-1, 0]$ to serve as a punishment for the agent.

$$f(X) = \frac{\sum_{i=1}^{25} U_i \prod_{j=1}^n t_j^i(x_j)}{\sum_{i=1}^{25} \prod_{j=1}^n t_j^i(x_j)} \quad (13)$$

where $\mathbf{X} = [x_1, x_2, \dots, x_n]^\top$ and $f(x)$ represent the input and output fuzzy sets, respectively. The $T(x)$ denotes the triangular membership function. The N represents the number of inputs, and the U_i represents the weight of the i th fuzzy rule, which is determined by prior knowledge.

4.4 PKFD-DPG: Prior Knowledge and Fuzzy Logic Driven Deep Deterministic Policy Gradient

The complete framework of the PKFD-DPG algorithm is shown in Algorithm 1, which is an improvement of the DDPG algorithm. The algorithm includes an Actor network and a Critic network, both of which have similar structures. Specifically, the Actor network consists of three linear layers, two

ReLU activation layers, and one tanh activation layer, while the Critic network consists of three linear layers and two ReLU activation layers. All of these layers are connected in the form of fully connected layers. The input of the Actor network includes the state space of the agent, which consists of 12 parameters. The input of the Critic network includes the state and action spaces of the manipulator, with a total of 18 parameters. The output of the Actor network is a stiffness matrix of the admittance model in the workspace of the manipulator, which consists of 6 parameters. And the output of the Critic network is a state-action value function Q_t . In PKFD-DPG, the parameters of the Actor network are updated by minimizing the negative of the state-action value Q_t , while the parameters of the Critic network are updated by minimizing the mean squared error between the predicted state-action value Q_p and the actual state-action value Q_t . In addition, the reward system of the PKFD-DPG algorithm includes a three-layer MLP, which is used to construct a nonlinear mapping from the agent's state and action to the reward. The input of the MLP is the agent's state and action, and the output value is the predicted reward value r_p . The network parameters are updated by minimizing the mean squared error between the predicted reward value r_p and the reward value r_f inferred by the fuzzy system.

To improve the stability and convergence of the PKFD-DPG algorithm, PKFD-DPG introduces target networks consisting of an Actor network and a Critic network, respectively, which provide the training target values. The update frequency of the local networks and target networks is different. When optimizing the Actor and Critic networks, the correlation between the network parameters can be cut off by fixing the sampling frequency to obtain long-period parameters from the target networks, thereby improving the stability of the PKFD-DPG algorithm. In addition, to prevent the output range of the Actor network from being too large and to ensure stable movement of the manipulator, a sigmoid function is chosen as the activation function for the output layer of the Actor network, and the output range can be normalized to $[0, 1]$.

5 Experiment and Analysis

In this section, we first constructed a digital twin model and a physical PDLOR for simulation experiment and physical experimental. Second, we simulated the tool assembly tasks in the *CoppeliaSim* simulation software to verify the feasibility and effectiveness of the proposed PKFD-DPG algorithm. Finally, we used the simulation results to guide real-world assembly tasks and compared them with some other classical assembly algorithms.

Algorithm 1 PKFD-DPG.

```

1: Initialize the network parameters of DDPG
2: Initialize replay buffer  $\mathcal{D}$ 
3: for episode = 1 to M do
4:   Sample a goal  $g \in \mathcal{G}$  and an initial state  $s_0 \in \mathcal{S}$ 
5:   for environment step = 1 to T do
6:     Sample an action  $a_t$  using the behavioral policy from DDPG:
7:      $a_t \sim \pi_\phi(a_t | s_t, g)$ 
8:     Execute the action  $a_t$  and observe a new state  $s_{t+1}$ :
9:      $s_{t+1} \sim p(s_{t+1} | a_t, s_t, g)$ 
10:    end for
11:    for environment step = 1 to T do
12:       $R = r_g + (1 - \mu)r_p + \mu r_f$ 
13:      Store the transition in  $\mathcal{D}$ :
14:       $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, g_t, a_t, r_t, s_{t+1})\}$ 
15:      Sample a set of additional goals  $\varphi$  from  $\zeta$  for replay
16:      for  $g' \in \varphi$  do
17:         $r_g = r(s_t, a_t, g')$ 
18:        Store the transition in  $\mathcal{D}$ :
19:         $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, g'_t, a_t, r_t, s_{t+1})\}$ 
20:      end for
21:    end for
22:    for gradient step = 1 to N do
23:      Sample a minibatch  $\mathcal{B}$  from the replay buffer  $\mathcal{D}$ 
24:      Perform one step of optimization using DDPG and minibatch  $\mathcal{B}$ 
25:    end for
26: end for

```

5.1 Simulation Setup and Environment Design

First, a physical and digital twin model of the PDLOR platform was established, including individual objects such as dual robot manipulators, two force sensors, a visual perception system, and various multip-peg-in-hole operating tools, as shown in Fig. 10. The manipulators are UR10, and the 6-axis force sensors ATI are installed between the end joint and the end flange, which can provide X , Y , and Z directional contact forces and torques based on the current contact state. The visual perception system consists of a 3D gimbal and a Real Sense Depth Camera D435i. The end-effector flange is used to simulate pegs, while the tool end flange is used to

simulate holes. The sizes of the pegs and holes are defined in Table 2. In our *CoppeliaSim* simulation environment, the friction coefficient of the holes and pegs is set to 0.01, and the maximum force and torque are set to 100 N and 100 Ncm , respectively. The multip-peg-in-hole tools have the following assumptions:

- The pegs are strictly rigid and circular, while the holes are elastic and strictly circular.
- The contact between the pegs and the holes is strictly point contact.
- The friction model is simplified to calculate the friction between the pegs and the holes.

5.2 Simulation Results Analysis

All simulation training experiment process for the entire model were run on a computer with an 8-core Intel(R) Core(TM) i9-10900 CPU @ 2.90 GHz, 64.0 GB RAM, and an NVIDIA GeForce RTX 3090 GPU. In each training episode, the initial position of the manipulator's end-effector flange pegs are obtained based on deep visual-guided, and the training initial position is ensured to be able to detect contact forces. In each step, the agent selects a stiffness coefficient matrix and calculates the end-effector position gain x_e in cartesian space based on the admittance model of the manipulator's end-effector, and then converts it into joint angle gain q_e through the inverse kinematics of the manipulator, which is a 6-D vector. The agent interacts with the environment, and the robot enters the next state based on the joint angle gain q_e . Gaussian noise is also added to the action to simulate the tool assembly process of PDLOR.

PDLOR used in power distribution networks usually need to be compatible with various types of operation tools, which can be installed on the tool rack in front of the robot. We simulate the assembly of a single operation tool and the assembly of multiple different operation tools, which correspond to fixed and unfixed tool flange hole positions, respectively.

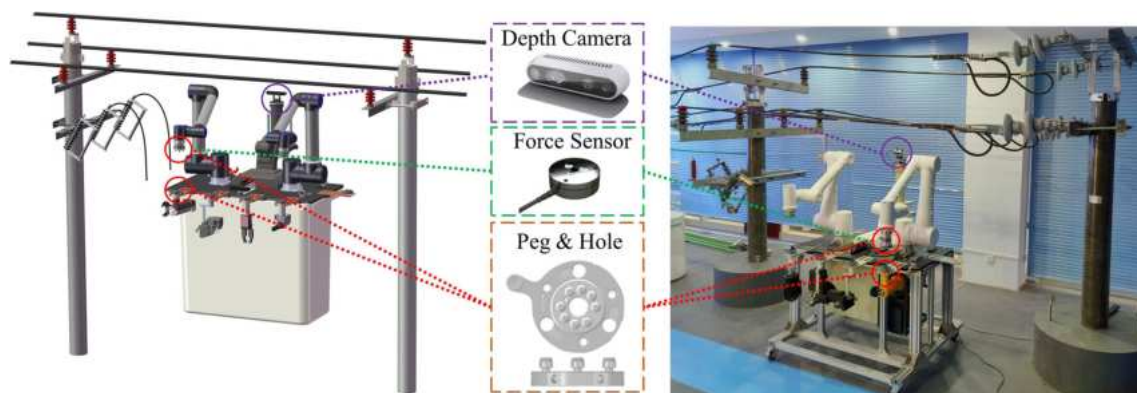


Fig. 10 The digital twin model and physical entity of the live-line robot platform

Table 2 Parameters of pegs and the holes

Name	Diameter	Length
End effector flange (Peg)	12.86 mm	14.40 mm
Tool end flange (Hole)	13.00 mm	18.60 mm

To evaluate the performance of the proposed main-auxiliary combined reward system of PKFD-DPG, we conducted comparative experiments on two assembly scenarios, comparing it with Func-DDPG, which utilizes functionized reward functions. Additionally, we compared the performance of the PKFD-DPG with that of the traditional model-based PD force controller in assembling single and multiple tools assembly tasks to validate the efficient and outstanding performance of the PKFD-DPG in autonomously assembling tools for PDLOR.

Multiple metrics are adopted to evaluate the learning performance of the agent's assembly strategy, namely, success rate, average reward value, average force, and average torque. The maximum reward in all experiments is set to 200. The tool assembly task is considered completed when the reward exceeds 0. In the *CoppeliaSim* simulation environment, PDLOR performs 1000 episodes of tool assembly tasks, with each assembly episode having 100 steps. In each episode, the assembly episode ends when one of the following conditions is met: 1) reaching the maximum time steps, 2) achieving the minimum distance error to the target pose with forces and torques within the allowed range, or 3) encountering a collision where forces and torques exceed the maximum threshold. Typically, a complete training experiment takes approximately 13 hours.

The results of various performance metrics for different assembly tasks after three repetitions of training in each experimental group are summarized in Table 3. The changing processes of the rewards in the training for each assembly

algorithm are displayed in Fig. 11. Table 3 clearly indicates that for the single-tool assembly task, both PKFD-DPG and Func-DDPG outperform the PD controller in terms of success rate and average reward value, with PKFD-DPG exhibiting lower standard deviation in success rate and average reward. However, when facing a multip-peg-in-hole assembly task, both Func-DDPG and the PD controller show inferior performance, making it difficult to determine the quality of actions performed by the agent solely based on the success rate and average reward. Nevertheless, PKFD-DPG still demonstrates commendable performance in this assembly task. Furthermore, from Fig. 11, it can be observed that the reward convergence of the PKFD-DPG algorithm is the fastest during the training process, and the reward curve of PKFD-DPG appears more stable in the later stages of training. This indicates that PKFD-DPG exhibits more efficient and superior performance in multip-peg-in-hole assembly tasks.

5.3 Experimental Evaluation and Validation

The effectiveness and robustness of PKFD-DPG were demonstrated through training in the *CoppeliaSim* simulation environment. Guided by the selection of hyperparameters for the proposed algorithm and the networks designed in the simulation environment, the performance of the PKFD-DPG algorithm was validated in a real-world environment of the PDLOR assembly task with multip-peg-in-hole. The assembly experiment platform for PDLOR is depicted in Fig. 10. To simplify the description, the assembly process of PDLOR can be divided into three stages: close, hole search, and insertion. In the close stage, the manipulator utilizes MS-YOLOv5 for object detection and localization to guide the descent of the end-effector flange pegs towards the tool flange holes. In the hole search stage, the flange pegs make contact with the tool frame and the tool flange, and there may be positioning errors between the pegs and the holes. In the insertion stage, the

Table 3 Comparison of peg hole assembly performance of different assembly algorithms for different assembly tasks

Assembly task	Algorithm	Success rate	Average reward	Average f_z (N)	Average τ_z (N/cm)
Single tool	PD	30.4%	-29.4	-42.8	-11.4
		± 43.8	± 42.0		
	Func-DDPG	66.9%	8.2	-31.3	-7.5
		± 41.7	± 32.4		
	PKFD-DPG	82.1%	60.7	-21.0	-5.1
		± 27.2	± 21.9		
Multiple tools	PD	19.2%	-68.8	-41.9	-15.4
		± 55.2	± 51.5		
	Func-DDPG	55.6%	-22.5	-32.1	-9.6
		± 43.1	± 12.7		
	PKFD-DPG	72.6%	29.9	-29.7	-7.3
		± 33.7	± 25.9		

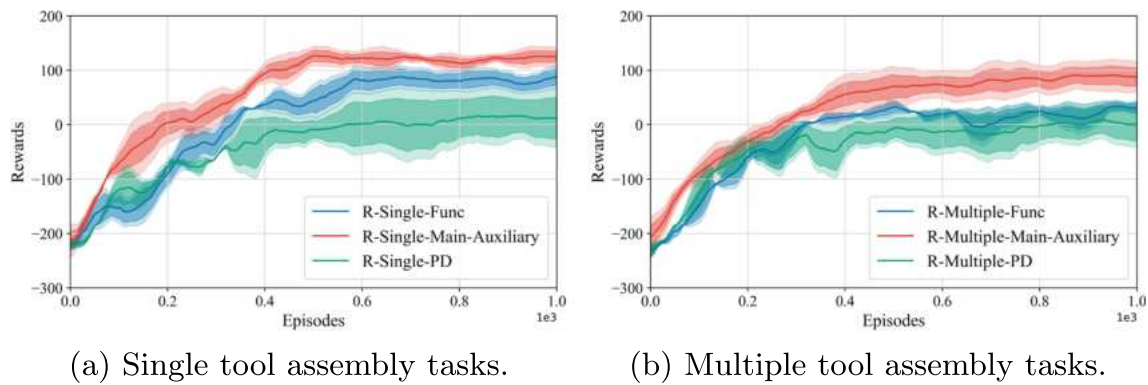


Fig. 11 Diagram of the reward convergence process for different assembly algorithms

positioning errors have been eliminated, but there still exist slight orientation errors between the pegs and the holes. With the feedback from force sensors, the PKFD-DPG algorithm learns optimal admittance parameters, allowing the manipulator to adjust its posture smoothly and achieve successful multip-peg-in-hole assembly. The entire assembly process is illustrated in Fig. 12.

To verify the efficient performance of the PKFD-DPG algorithm in the tool assembly task, we compared its performance with the PD controller and Func-DDPG algorithm on the same task. We collected force and torque data from the ATI six-axis force sensor at the end-effector of the manipulators during the assembly process, and synchronized the assembly steps with the assembly time steps, as shown in Fig. 13. From the figure, it can be observed that for the single-tool assembly task, PKFD-DPG, Func-DDPG, and the PD controller are all able to successfully complete the assembly task. The PKFD-DPG demonstrates the fastest assembly completion time, with the lowest force applied at the end-effector of the manipulator, indicating smoother performance of the manipulator during the assembly process. However, for the multip-peg-in-hole assembly task, the

assembly performance based on PKFD-DPG is still superior in terms of completion speed and force exerted compared to Func-DDPG. However, the PD controller fails to complete the assembly task effectively, possibly due to significant pose variations between the pegs and holes during the multip-peg-in-hole assembly process, making it challenging to establish a reliable contact model. As a result, the PD controller fails to complete the assembly within the limited assembly steps. Based on the PKFD-DPG algorithm in the tool assembly task, it can provide a suitable and generic solution for complex multip-peg-in-hole assembly tasks without analyzing unknown contact states and tuning program parameters, greatly improving the efficiency of the operation process of PDLOR by learning from experience.

6 Conclusion

In response to the demand for autonomous tool assembly by PDLOR, this paper proposes a deep visual-guided coarse localization and prior knowledge and fuzzy logic driven deep deterministic policy gradient high-precision

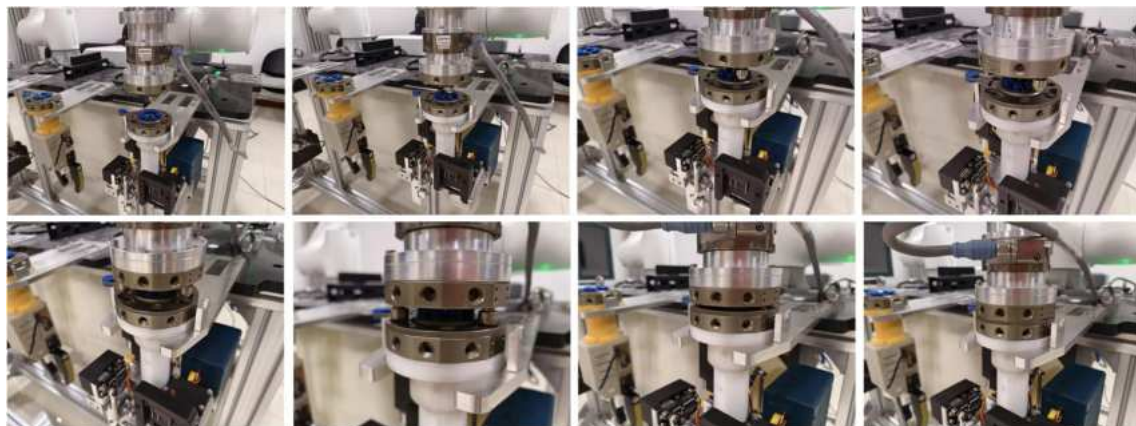
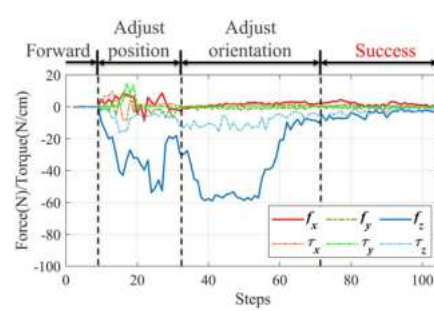
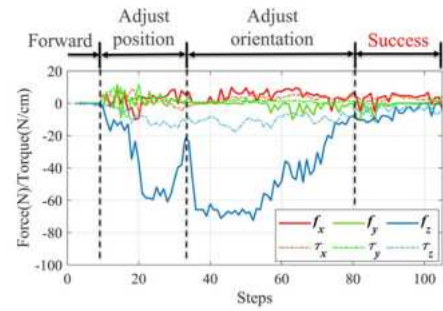


Fig. 12 Automatic assembly tool for Live-Line Robot

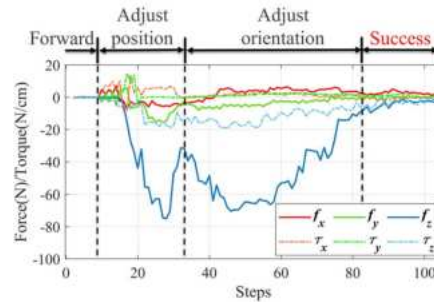
Fig. 13 Different algorithms for force/torque at the end of the robot manipulator for single and multiple tool assembly tasks



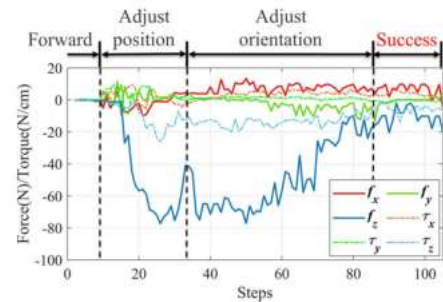
(a) PKFD-DPG for single-tool assembly tasks.



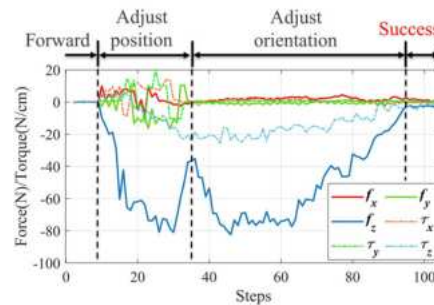
(b) PKFD-DPG for multiple-tool assembly tasks.



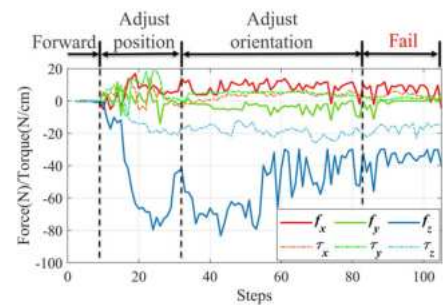
(c) Func-DDPG for single-tool assembly tasks.



(d) Func-DDPG for multiple-tool assembly tasks.



(e) PD for single-tool assembly tasks. (f) PD for multiple-tool assembly tasks.



assembly algorithm framework. This algorithm framework demonstrates excellent performance in multi-peg-in-hole tool assembly tasks. This approach avoids complex dynamic modelling and time-consuming parameter optimization processes, while achieving fast convergence through DRL. The main algorithmic contributions of this paper are twofold. First, a multiscale identification and localization network based on the YOLOv5 (MS-YOLOv5) is proposed, which enables rapid peg-in-hole approximation, allowing the agent to obtain more positive samples and avoid ineffective exploration during training. Second, a main-auxiliary combined reward function system is proposed, which effectively addresses the issues of sparse reward and binary reward through the main-line based on the HER mechanism and auxiliary reward based on the fuzzy inference mechanism, significantly accelerating the convergence speed and

improving the stability after convergence. In addition, experimental verification is conducted on both simulated and real PDLOR, and performance evaluation is compared with other assembly algorithms.

This method has a significant effect on improving the learning rate of autonomous tool replacement by PDLOR. However, due to the specificity of the model scene, its generalization ability is somewhat reduced. Therefore, with the advancement of transfer learning techniques, we expect the proposed PKFD-DPG algorithm to become a universal and practical approach for performing various peg-in-hole assembly tasks. In the future, we will further optimize the action selection strategy and explore reward construction mechanisms, with the aim of widespread application in industrial manufacturing scenarios, thus significantly improving the efficiency of industrial manufacturing processes.

Acknowledgements This work was supported by the National Key R&D Program of China (Grant number:2018YFB1307400) and State Grid Anhui Science and Technology Project.

Author Contributions Li Zheng: Conceptualization, Methodology, Software, Validation, Data curation, Writing- Original draft preparation, Investigation. Jiajun Ai: Conceptualization, Data curation. Xuming Tang: Resources, Data curation. Shaolei Wu: Resources, Data curation. Sheng Chen: Resources, Data curation. Rui Guo: Resources, Data curation. Erbao Dong: Supervision, Writing- Reviewing and Editing. All authors read and approved the final manuscript.

Funding This work was supported by the National Key R&D Program of China (Grant numbers[2018YFB1307400]).

Availability of data and materials Not applicable

Code availability The codes and datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest/Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethical standard Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alhassan, A.B., Zhang, X., Shen, H., Xu, H.: Power transmission line inspection robots: A review, trends and challenges for future research. *Int. J. Electr. Power Energy Syst.* **118**, 105862 (2020)
- Jenssen, R., Roverso, D., et al.: Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *Int. J. Electr. Power Energy Syst.* **99**, 107–120 (2018)
- Chen, Y., Wang, Y., Tang, X., Wu, K., Wu, S., Guo, R., Feng, Y., Dong, E.: Intelligent power distribution live-line operation robot systems based on stereo camera. *High Voltage* (2023)
- Jiang, Y., Huang, Z., Yang, B., Yang, W.: A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation. *Robot. Comput. Integr. Manuf.* **78**, 102366 (2022)
- Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., Tzovaras, D.: Machine learning and deep learning in smart manufacturing: The smart grid paradigm. *Comput. Sci. Rev.* **40**, 100341 (2021)
- Li, Z., YaHao, W., Run, Y., Shaolei, W., Rui, G., Dong, E.: An efficiently convergent deep reinforcement learning-based trajectory planning method for manipulators in dynamic environments. *J. Intell. Robot. Syst.* **107**(4) (2023)
- Chhatpar, S.R., Branicky, M.S.: Search strategies for peg-in-hole assemblies with position uncertainty. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium* (Cat. No. 01CH37180), vol. 3, pp. 1465–1470. IEEE (2001)
- Kang, H., Zang, Y., Wang, X., Chen, Y.: Uncertainty-driven spiral trajectory for robotic peg-in-hole assembly. *IEEE Robotics and Automation Letters* (2022)
- Jasim, I.F., Plapper, P.W., Voos, H.: Position identification in force-guided robotic peg-in-hole assembly tasks. *Procedia Cirp* **23**, 217–222 (2014)
- Chen, F., Cannella, F., Huang, J., Sasaki, H., Fukuda, T.: A study on error recovery search strategies of electronic connector mating for robotic fault-tolerant assembly. *J. Intell. Robot. Syst.* **81**(2), 257–271 (2016)
- Park, H., Park, J., Lee, D.-H., Park, J.-H., Bae, J.-H.: Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture. *IEEE Robot. Autom. Lett.* **5**(3), 4447–4454 (2020)
- Abu-Dakka, F.J., Nemec, B., Kramberger, A., Buch, A.G., Krüger, N., Ude, A.: Solving peg-in-hole tasks by human demonstration and exception strategies. *Industrial Robot: An International Journal* (2014)
- Park, H., Park, J., Lee, D.-H., Park, J.-H., Baeg, M.-H., Bae, J.-H.: Compliance-based robotic peg-in-hole assembly strategy without force feedback. *IEEE Trans. Ind. Electron.* **64**(8), 6299–6309 (2017)
- Jiang, T., Cui, H., Cheng, X., Tian, W.: A measurement method for robot peg-in-hole prealignment based on combined two-level visual sensors. *IEEE Trans. Instrum. Meas.* **70**, 1–12 (2020)
- Xu, J., Liu, K., Pei, Y., Yang, C., Cheng, Y., Liu, Z.: A noncontact control strategy for circular peg-in-hole assembly guided by the 6-dof robot based on hybrid vision. *IEEE Trans. Instrum. Meas.* **71**, 1–15 (2022)
- Lu, B.-S., Chen, T.-I., Lee, H.-Y., Hsu, W.H.: Cfv: Coarse-to-fine visual servoing for 6-dof object-agnostic peg-in-hole assembly. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12402–12408. IEEE (2023)
- Yasutomi, A.Y., Ichiwara, H., Ito, H., Mori, H., Ogata, T.: Visual spatial attention and proprioceptive data-driven reinforcement learning for robust peg-in-hole task under variable conditions. *IEEE Robot. Autom. Lett.* **8**(3), 1834–1841 (2023)
- Wang, J., Jiang, Y., Lin, S., Kong, F.: Geometric model-based joint angle selection criterion for force parameter identification & decoupling control method of position and posture in shaft-hole assembly. In: *2021 IEEE 11th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 7–12. IEEE (2021)
- Kim, M.-C., Choi, H., Piao, J., Kim, E.-S., Park, J.-O., Kim, C.-S.: Remotely manipulated peg-in-hole task conducted by cable-driven parallel robots. *IEEE/ASME Trans. Mechatron.* **27**(5), 3953–3963 (2022)
- Tang, X., Shang, W., Hu, J., Zhang, F., Zhang, X.: Error state probability-based compliance control for peg-in-hole assembly. *IEEE Trans. Autom. Sci. Eng.* (2023)
- Zhao, Y., Gao, F., Zhao, Y., Chen, Z.: Peg-in-hole assembly based on six-legged robots with visual detecting and force sensing. *Sensors* **20**(10), 2861 (2020)

22. Chen, Z., Xie, S., Zhang, X.: Position/force visual-sensing-based robotic sheet-like peg-in-hole assembly. *IEEE Trans. Instrum. Meas.* **71**, 1–11 (2021)
23. Lee, D.-H., Choi, M.-S., Park, H., Jang, G.-R., Park, J.-H., Bae, J.-H.: Peg-in-hole assembly with dual-arm robot and dexterous robot hands. *IEEE Robot. Autom. Lett.* **7**(4), 8566–8573 (2022)
24. Higuera, C., Ortiz, J., Qi, H., Pineda, L., Boots, B., Mukadam, M.: Perceiving extrinsic contacts from touch improves learning insertion policies. [arXiv:2309.16652](https://arxiv.org/abs/2309.16652) (2023)
25. Van der Merwe, M., Wi, Y., Berenson, D., Fazeli, N.: Integrated object deformation and contact patch estimation from visuo-tactile feedback. [arXiv:2305.14470](https://arxiv.org/abs/2305.14470) (2023)
26. Fan, Y., Luo, J., Tomizuka, M.: A learning framework for high precision industrial assembly. In: 2019 International conference on robotics and automation (ICRA), pp. 811–817. IEEE (2019)
27. Leyendecker, L., Schmitz, M., Zhou, H.A., Samsonov, V., Rittsteg, M., Lütticke, D.: Deep reinforcement learning for robotic control in high-dexterity assembly tasks—a reward curriculum approach. In: 2021 Fifth IEEE International Conference on Robotic Computing (IRC), pp. 35–42. IEEE (2021)
28. Petrovic, O., Schäper, L., Roggendorf, S., Storms, S., Brecher, C.: Sim2real deep reinforcement learning of compliance-based robotic assembly operations. In: 2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 300–305. IEEE (2022)
29. Inoue, T., De Magistris, G., Munawar, A., Yokoya, T., Tachibana, R.: Deep reinforcement learning for high precision assembly tasks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 819–825. IEEE (2017)
30. Xie, L., Yu, H., Zhao, Y., Zhang, H., Zhou, Z., Wang, M., Wang, Y., Xiong, R.: Learning to fill the seam by vision: Sub-millimeter peg-in-hole on unseen shapes in real world. In: 2022 International Conference on Robotics and Automation (ICRA), pp. 2982–2988. IEEE (2022)
31. Schoettler, G., Nair, A., Luo, J., Bahl, S., Ojeda, J.A., Solowjow, E., Levine, S.: Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5548–5555. IEEE (2020)
32. Wang, Y., Zhao, L., Zhang, Q., Zhou, R., Wu, L., Ma, J., Zhang, B., Zhang, Y.: Alignment method of combined perception for peg-in-hole assembly with deep reinforcement learning. *J. Sensors* **2021**, 1–12 (2021)
33. Lämmle, A., Tenbrock, P., Bálint, B., Nägele, F., Kraus, W., Váncza, J., Huber, M.F.: Simulation-based learning of the peg-in-hole process using robot-skills. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9340–9346. IEEE (2022)
34. 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.* **5**(4), 5709–5716 (2020)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28** (2015)
36. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969. (2017)
37. Redmon, J., Farhadi, A.: Yolo3: An incremental improvement. (2018) [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
38. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection (2020)
39. Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B.: A review of yolo algorithm developments. *Procedia Comput. Sci.* **199**, 1066–1073 (2022)
40. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, ??? (2018)
41. Eschmann, J.: Reward function design in reinforcement learning. *Reinforcement Learning Algorithms: Analysis and Applications*, 25–33 (2021)
42. Gupta, A., Pacchiano, A., Zhai, Y., Kakade, S., Levine, S.: Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Adv. Neural Inf. Process. Syst.* **35**, 15281–15295 (2022)
43. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., Zaremba, W.: Hindsight experience replay. *Adv. Neural Inf. Process. Syst.* **30** (2017)
44. Magdalena, L.: Fuzzy rule-based systems. *Springer handbook of computational intelligence*, 203–218 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Li Zheng is a PhD candidate at the University of Science and Technology of China. His research interests include robot motion planning and control and deep reinforcement learning.

Jiajun Ai is a master's student at the University of Science and Technology of China. His research interests include robot motion planning and control.

Yahao Wang is a PhD candidate at the University of Science and Technology of China. His research interests include robot motion planning and control.

Xuming Tang is senior engineer. His research interests include transmission line unmanned aerial vehicle inspection.



Shaolei Wu is a senior engineer at the Electric Power Research Institute of State Grid Anhui Electric Power Co., LTD.

Sheng Cheng is a senior engineer at the Artificial Intelligence Institute of China Electric Power Research Institute. His research direction is electric power artificial intelligence technology and application related professional work.

Rui Guo is the chief expert in the field of electric robots at the Shandong Electric Power Research Institute. The main research direction is a series of live-line robots such as overhead transmission line deicing robot, detection robot, anti-vibration hammer drag back robot, foreign object cleaning robot, insulator string detection robot, etc.

Erbao Dong is an associate professor at the University of Science and Technology of China. His research interests include bionic robots and live working robots.

Authors and Affiliations

Li Zheng¹  · Jiajun Ai¹ · Yahao Wang¹ · Xuming Tang² · Shaolei Wu³ · Sheng Cheng⁴ · Rui Guo⁵ · Erbao Dong¹ 

Li Zheng
zlsy@mail.ustc.edu.cn

Jiajun Ai
ajj@mail.ustc.edu.cn

Yahao Wang
wyh218@mail.ustc.edu.cn

Xuming Tang
ah_tangxuming@163.com

Shaolei Wu
wusl081x@ah.sgcc.com.cn

Sheng Cheng
chensheng@epri.sgcc.com.cn

Rui Guo
guoruihit@gmail.com

- ¹ CAS Key Laboratory of Mechanical Behavior and Design of Materials, Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, 96 Jinzhai Road, Hefei 230026, Anhui Province, China
- ² State Grid Anhui Electric Power Company Anhui Province, Hefei, China
- ³ State Grid Anhui Electric Power Company Electric Power Research Institute, Hefei, Anhui Province, China
- ⁴ China Electric Power Research Institute, Beijing, China
- ⁵ State Grid Intelligence Technology Co. Ltd, Jinan, Shandong Province, China