arXiv:2010.13056v1 [cs.RO] 25 Oct 2020

Proactive Action Visual Residual Reinforcement Learning for Contact-Rich Tasks Using a Torque-Controlled Robot

Yunlei Shi[1,2], Zhaopeng Chen[2,1] Hongxu Liu[3], Sebastian Riedel[2],
Chunhui Gao[2], Qian Feng[3], Jun Deng[2], Jianwei Zhang[1]

***Abstract*— Contact-rich manipulation tasks are commonly found in modern manufacturing settings. However, manually designing a robot controller is considered hard for traditional control methods as the controller requires an effective combination of modalities and vastly different characteristics. In this paper, we firstly consider incorporating operational space visual and haptic information into reinforcement learning(RL) methods to solve the target uncertainty problem in unstructured environments. Moreover, we propose a novel idea of introducing a proactive action to solve the partially observable Markov decision process problem. Together with these two ideas, our method can either adapt to reasonable variations in unstructured environments and improve the sample efficiency of policy learning. We evaluated our method on a task that involved inserting a random-access memory using a torque-controlled robot, and we tested the success rates of the different baselines used in the traditional methods. We proved that our method is robust and can tolerate environmental variations very well.**

I. INTRODUCTION

For high-precision assembly tasks, the robot needs to combine the high positioning accuracy with high flexibility. Designing a robot for these tasks is very challenging although such tasks can be easily performed by humans. Several torque-controlled robots have been designed for cooperative tasks to be performed in industrial environments [1], [2]. These torque-controlled robots have seven revolute joints with torque sensors, and similar control algorithms [3], [4], [5]. Currently, torque-controlled robots are already safe enough when collisions occur with environments or humans [1], [6]. However, their effectiveness in real-life and production scenarios is still not satisfactory.

Torque-controlled robots often serve computers, communication, and consumer electronics (3C) product lines, which usually involve small but complex assembly tasks, and need to be adjusted quickly and frequently. Nowadays there are a few 3C assemble factory lines [7] but they need a long time to build and setup in high precise, which are not suitable for small and medium-sized enterprises (SMEs) who have automation needs but cannot afford to upgrade the entire production line. Position uncertainty are quite normal in human-based traditional production lines. Some studies used simple fixed curves for exploring [8], [9] but they have low robustness against positional and angular errors for insertion tasks especially when targets are not fixed accurately. Schimmels and Peshkin [10], [11] designed an admittance matrix for force-guided assembly in the absence of friction and after two years they improved the admittance control law. However, there still existed a maximum limit requirement of friction value [12]. Stemmer et al. [13] proposed the region of attraction (ROA) method using vision and force perception to assemble the specified-shape objects, while geometry of the parts are required.

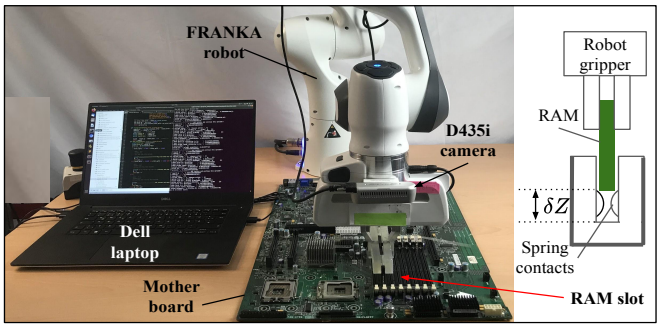

Fig. 1. A contact-Rich task scenario: RAM insertion. Such kind of tasks always have stuck problems due to the tight clearance and narrow space.

In this paper we equip a robot with a visual residual policy that combine multimodal feedback from vision and touch, two modalities with different frequencies, and characteristics. Our primary contributions are:

1) We propose a visual reinforcement learning (RL) method by combining a visual-based fixed policy with a contact-based parametric policy, this method greatly enhances the robustness and efficiency of the reinforcement learning (RL) algorithm.

2) We propose the proactive action in the visual residual RL policy to solve the partially observable Markov decision process (POMDP) problem, which could ensure the task success rate and the ability to tolerate environmental variations.

3) We implement ablative and comparative studies to give the effects of each modality on task success rate and prove the robustness of our method in experiment.

II. BACKGROUND AND RELATED WORK

A. Torque-controlled Robot Concepts

Torque-controlled robots have been developed for unstructured environments that are fundamentally different from the environments where classical industrial robotics have been used. The torque sensor in each joint plays a key role in robot controller. The basic controller consists of a torque

*This research has received funding from the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, DFG TRR-169/NSFC 61621136008, partially supported by European projects H2020 STEP2DYNA (691154) and ULTRACEPT (778602).

[1]TAMS (Technical Aspects of Multimodal Systems), Department of Informatics, Universitt Hamburg, [2]Agile Robots AG, [3]Technische Universitt Mnchen.

feedback loop, which can be interpreted as the scaling of the motor inertia B to the desired value B_θ [4]:

$$\boldsymbol{\tau}_m = \boldsymbol{B}\boldsymbol{B}_\theta^{-1}\boldsymbol{\tau_u} + (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}_\theta^{-1})\boldsymbol{\tau} \tag{1}$$

Here, τ_u is an intermediate control input that could shape the Cartesian or joint impedance behavior [3], and τ is the joint torque data measured by the torque sensor as well as the torque vector applied to manipulators joints. τ_m is the torque on demand of the motor controller. For the Cartesian impedance behavior, we have

$$\begin{aligned} \boldsymbol{\tau}_u &= -\boldsymbol{J}(\boldsymbol{\theta})^T(\boldsymbol{K}_x\tilde{\boldsymbol{x}}(\boldsymbol{\theta}) + \boldsymbol{D}_x\dot{\boldsymbol{x}}(\boldsymbol{\theta})) + \overline{\boldsymbol{g}}(\boldsymbol{\theta}) \\ \tilde{\boldsymbol{x}}(\boldsymbol{\theta}) &= f(\boldsymbol{\theta}) - \boldsymbol{x}_s \\ \dot{\boldsymbol{x}}(\boldsymbol{\theta}) &= \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \end{aligned} \tag{2}$$

$\boldsymbol{K_x}$ and $\boldsymbol{D_x}$ are the permutation and diagonal matrices of desired stiffness and damping; $\boldsymbol{x_s}$ is the desired end-effector (EE) pose, and $\boldsymbol{x}(\boldsymbol{\theta}) = f(\boldsymbol{\theta})$ is the EE pose computed based on the motor position. $\boldsymbol{J}(\boldsymbol{\theta}) = \partial f(\boldsymbol{\theta})/\boldsymbol{\theta}$ is the manipulator Jacobian; $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_s$ are the measured and desired motor positions, respectively. $\overline{\boldsymbol{g}}(\boldsymbol{\theta})$ is the gravity function that always comes from the CAD model or the parameter identification; this function inevitably has errors.

B. Visual Servo Control in Manufacturing Application

Vision sensor allows robot to measure the environment with noncontact method. Shirai and Inoue [14] described an idea on how to use visual feedback to correct the position of a robot in order to increase assembly task accuracy. Position-based visual servo (PBVS) systems and image-based visual servo (IBVS) systems are two major classes of visual servo control systems. The typical control structure of PBVS can be found in [15].

An end-effector mounted camera could acquire the target depth and orientation information which can be used directly for PBVS [16], [17]. While the lens and the imaging sensors, the calibration of intrinsic/extrinsic parameters, the reflection, shadow and occlusion will exert a strong influence on the precision of the visual guidance [18].

C. Reinforcement Learning for Assembly Tasks

RL offers a set of tools for the design of sophisticated robotic behaviors that are difficult to engineer. RL has been applied previously and has gained great success in solving a variety of problems in robotic manipulations [19], [20], [21], [22], [23]. Newman et al. [24] inverted the mapping from the relative positions to the observed moments and trained the neural net to guide the robotic assembly. Inoue et al. [22] used long short-term memory to learn algorithms with two threads (an action thread and a learning thread) for searing and inserting a peg into a tight hole; however, their methods required several pre-defined heuristics, flat searching surfaces, and also a long training time.

Residual RL could take advantage of the efficiency of conventional controllers and the flexibility of RL [25]. The idea is to try to inject prior information into an RL algorithm to speed up the training process instead of randomly exploring from scratch. Specifying goals via images makes it possible to specify goals with minimal manual effort

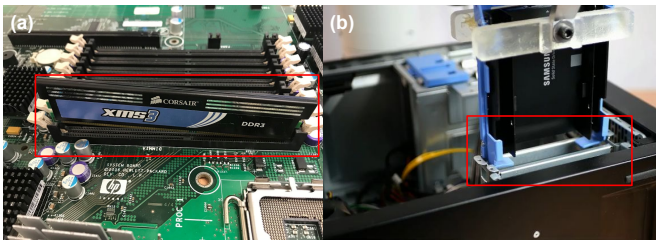

Fig. 2. Inside the computer host, there is no sliding surface for insertion tasks. (a) RAM slot. (b) Solid State Disk (SSD) slot.

such as taking a photo [21]. Combining the sense of vision and touch could endow robots with a similar ability as humans to complete the assembly tasks [19], which could provide robustness to sensor and actuator noises [21] as well as position uncertainty. However, only a few studies have focused on real industrial production contact-rich tasks, and they also require a sliding surface for the algorithms to search [19], [22], [26].

III. PROBLEM STATEMENT AND METHOD OVERVIEW

A. Problem Statement

1) Position Uncertainty in Unstructured Environments: As we talked in Section II-A, position uncertainty are quite normal in human based production lines. Workers could perform high-precision robotic assembly tasks with their strong intelligence, excellent visual ability, and dexterous hands. while these tasks are very challenging to robots especially in these unstructured production environments.

Also, torque-controlled robots have low position robustness to friction and obstruction in contact-rich tasks due to the low stiffness design concepts as we described in Section II-A. The limited control stiffness together with the friction and obstruction in contact-rich tasks give the position control error at the millimeter level. Torque-controlled robots are expected to achieve a desired dynamical relationship between environmental forces and movements of the robot in order to avoid breaking the environments or targets, thus the desired position and contact force can not be satisfied in the same dimension simultaneously. Moreover, the location of the targets is uncertain sometimes due to insufficient accuracy of industrial assembly line.

Using visual method to correct the positions of the targets is an intuitive solution, while we still have position control problems when robot contact with targets due to the reason as we explained in section II-B, even we have implemented some explore actions (e.g., the spiral explore method [8]).

In 3C production lines, the insertion scenarios are different with the typical simplification settings of peg-in-hole [19], [22]. For example, the random-access memory (RAM) insert-type task has problems as follow:

1) The RAM slot or other slots does not have a proper surface for sliding behavior of the robot in alignment stage [19], [20] as shown in Figure 2 which makes sliding type algorithm not work any more.

2) The objects (like the RAM or hard disk) would be easily stuck by the structure near the slot or the slot itself in the explore/alignment stage as shown in Figure 2.
3) Compared with previous studies, the slot has a long and narrow shape with tight clearance which is hard to insert by random and traditional search algorithm [8], [27].

2) Uncertainty POMDP States: The main challenge of the traditional policy is to design adaptable, yet robust algorithms in the face of inherent difficulties for modeling all possible interaction behaviors. RL enabled us to find new control policies automatically for contact-rich problems where traditional heuristics had been used, but the results were not satisfactory.

Contact states are hard to estimate due to the sensor noise and robot modeling error, changing the Markov decision process (MDP) to POMDP, which makes it significantly harder to find an optimal policy [28], and it also requires more training time. Belief state tracking is one way to deal with the POMDP problem [29], [30], [31], but this method takes too much time to find an optimal policy.

B. Method Overview

An eye-in-hand camera is helpful for solving the problem of position uncertainty in unstructured Environments in contact-rich tasks. The camera could try to align the characters of the target and compensate for the position error of the robot. Visual feedback control could provide geometric object properties for the pre-reaching target phase, whereas the camera aligning accuracy would always be disturbed by the target material or light. Force feedback control is quite helpful for providing contact information between the object and environment for accurate localization and control under occlusions or bad vision conditions, and force information could be obtained easily from the proprioceptive data in the torque-controlled robot controller. Visual feedback and force feedback are complementary and sometimes concurrent during contact-rich manipulation. In this paper, we implemented the visual-based fixed policy combined with contact-based parametric policy (see Figure 3) as follow:
1) For roughly locating the slot, we use one global image take from the teach mode with the RGB-D camera and rely only on the PBVS method [15] (i.e. the visual-based fixed policy) control in this phase, because in free space, the contact-based parametric policy can not receive proper contact information.
2) After rough location phase finished, the robot will move to target slot according to the pre-recorded transformation ${}^{g}\boldsymbol{x}_d$ from global image pose to detailed image pose, where ${}^{g}\boldsymbol{x}_d$ is recorded in teach phase. When the RAM in EE contact with the target slot, the detailed image which more accuracy for locating a slot, will be used to insert the RAM into the slot according to our method descirbed in Section IV.

IV. POLICY AND CONTROLLER DESIGN

A. Policy Design

*1) **Visual Residual Reinforcement Learning**:* To take advantage of the high flexibility of RL and the high efficiency of conventional controllers, we introduce the idea of residual RL from [25] with vision information; our method is expected to perform better compared with original residual RL in a variable environment due to the position uncertainty problem in Section III-A.1.

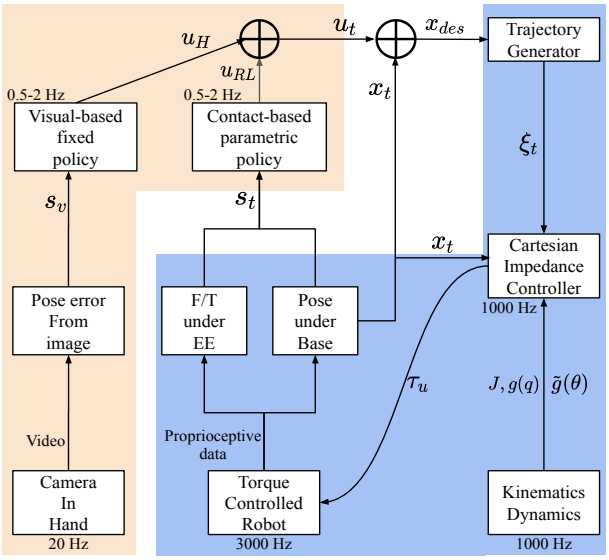

Fig. 3. Representation of policies and controller scheme. Blue region part is the real-time controller, and the wheat region part is non-real-time trained policy.

In residual RL, the policy are chosen by additively combining a fixed policy $\pi_H(s_v)$ with a parametric RL policy $\pi(\theta)(u_t|s_t)$: $u_t = \pi_H(s_v) + \pi_\theta(s_t)$. The fixed policy can help the agent move to the target, but prevent the agent from exploring more states. To balance the exploration and exploitation between the fixed policy and parametric RL policy, we design the weighted residual RL as

$$u_t = (1-\alpha)\pi_H(s_v) + \alpha * \pi_\theta(s_t). \tag{3}$$

Here, α is the action weight between the fixed policy and the parametric RL policy; The parametric policy is learned in the RL process to maximize expected returns on the task. We use a P-controller as the hand-designed controller $\pi_H(s_v)$ in our experiments for the visual-based fixed policy.

Firstly, we explain the detailed design of $\pi_H(s_v)$. s_v represents a geometric relationship of robot states which is a Euclidean distance calculated by visual and estimated depth information. We introduce the method from [32] which used depth information in PBVS. Combined with feature extraction and features' depth information Z_N, we could get estimated target feature set ${}^{c}P^* = (X_1^*, Y_1^*, Z_1^*, ..., X_N^*, Y_N^*, Z_N^*)$ and current feature set ${}^{c}P = (X_1, Y_1, Z_1, ..., X_N, Y_N, Z_N)$ whose coordinates are expressed with respect to the camera coordinate frame c following the perspective projection method [15]:

$$\begin{bmatrix} X_N \\ Y_N \end{bmatrix} = \frac{Z_N}{f} \begin{bmatrix} u_N \\ v_N \end{bmatrix}. \tag{4}$$

Where f is the focal length of the camera lens. $[u_N, v_N]^T$ gives the coordinates of the image feature set expressed in pixel units. Iterative Closest Point (ICP) [33] could be used to get the coordinate transformation ${}^{c*}\boldsymbol{x}_c$ by the feature set

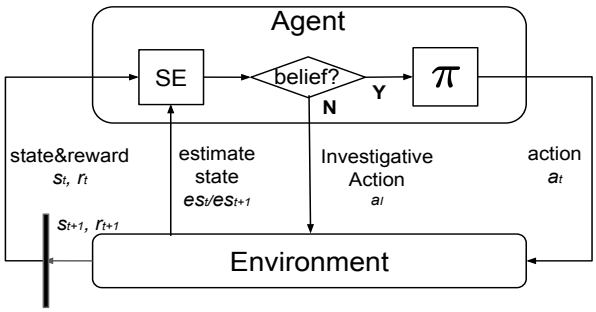

Fig. 4. Investigative action idea for solving POMDP problem. SE: state estimator. The states will be estimated by SE function, then the policy receive the clear states and output an action.

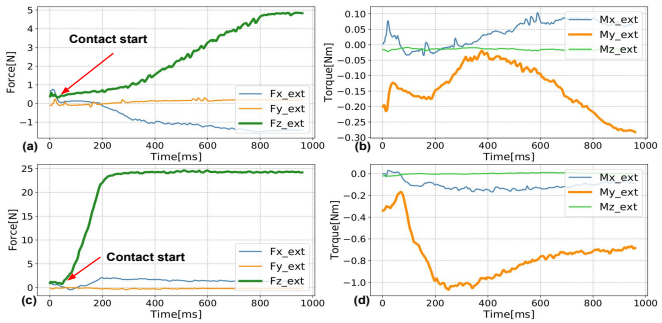

Fig. 5. (a) RAM contacts with one slot side in **movement action** with 5 N force feedback in the Z direction. (b) External moment data M_y which is hard to detect the torque contact status (goes up first and goes down later during contact force increase). (c) RAM contacts with the same side of (a) using an **investigative action** with 25 N press force. (d): External moment M_y reaches to -1 Nm which could clearly detect contact status

${}^{c}\boldsymbol{P}$ and ${}^{c}\boldsymbol{P}^{*}$.

$$ {}^{c*}\boldsymbol{x}_c = \begin{pmatrix} {}^{c*}\boldsymbol{R}_c & {}^{c*}\boldsymbol{t}_c \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{5} $$

Here we set $s_v = ({}^{c*}\boldsymbol{t}_c, \theta\boldsymbol{u})$ depends on Equation (5), where ${}^{c*}\boldsymbol{t}_c$ is the translation error vector, and $\theta\boldsymbol{u}$ gives the angle/axis representation for the rotation error [34]. Then a velocity control scheme is designed by using an exponential and decoupled decrease of the error(i. e., $\dot{\boldsymbol{e}} = -\lambda\boldsymbol{e}$) as:

$$ \begin{aligned} \boldsymbol{v_c} &= -\lambda({}^{c*}\boldsymbol{R}_c)^T {}^{c*}\boldsymbol{t}_c \\ \boldsymbol{w_c} &= -\lambda\theta\boldsymbol{u} \end{aligned} \tag{6} $$

Equation (6) is used in rough location phase in section III-B. $[\boldsymbol{v}_c, \boldsymbol{w}_c]^T$ is the camera frame velocity command under current camera frame $\mathcal{F}_c$, which could be easily transfer to robot EE frame $\mathcal{F}_e$. In this paper, we calculate robot movement commands under robot EE frame $\mathcal{F}_e$ first, and then transfer them to base frame before sending to Equation (2). Secondly, we directly use $s_v = ({}^{c*}\boldsymbol{t}_c, \theta\boldsymbol{u})$ as the states of fixed policy in accurate location phase,

$$ \pi_H(s_v) = -\boldsymbol{k}_p \cdot s_v, \tag{7} $$

which is quite convenient to implement.

In this paper, we use a value-based RL called Q-learning algorithm as the contact-based parametric RL policy $\pi(\theta)(u_t|s_t)$, the Q-function is implemented as a table with states as rows and actions as columns, then we can update the table by using the Bellman equation:

$$ Q^{\pi}(s_t, u_t) = \mathbb{E}_{r_t, s_{t+1}\sim E}[r_t + \gamma\mathbb{E}_{u_{t+1}\sim\pi}[Q^{\pi}(s_{t+1}, u_{t+1})]]. \tag{8} $$

*2) **Proactive Action**:* Most studies [25], [23], and [19] have modeled the robot manipulation task as a finite-horizon discounted Markov Decision Process (MDP) $\mathcal{M}$ in an environment E, with a state space $\mathcal{S}$, an action space $\mathcal{A}$, state transition dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, a discount factor $\gamma \in (0, 1]$, and a reward function $r : \mathcal{S}\times\mathcal{A} \to \mathcal{R}$ to determine an optimal stochastic policy π. In practice, many contact states s_t cannot be observed directly in the manipulation tasks that are close to a POMDP problem. However, the POMDP problem is confined to the modeling error of the torque-controlled robot, which makes it difficult to detect the contact states. Inspired by wild gorillas, who tried to cross a pool of water using a walking stick to test the water depth [35], we improved our RL process by adding a proactively investigative action (a_I) that could detect the clear states (es_t) involved in the RL process as Figure 4 which is different with [22] who continues push the target get a detectable moment; here, investigative action is one kind of the proactive actions.

We use the investigative action a_I combined with u_t to construct a new policy $u_t^I(s_t)$ instead of the original $u_t(s_t)$, which can be written as $a_I, u_t \to E \to s_{t+1}^I$. Where s_{t+1}^I is determined by adding an investigative action a_I of the torque-controlled robot to the environment. Consequently, the heuristic design of the investigative action prevents the learning process from falling into multiple unclear states.

In particular, the torque-controlled robot outputs either the movements or the forces. In our experiments, the movements are taken as the actions in the action space $\mathcal{A}$, and the forces are taken as the investigative actions. Instead of using 20 N force continuously to detect the values of the moments in the search phase [22], we only command the controller to exert a force (10-25 N) in some directions in a short time (0.5-1 s) as the investigative action while the feedback movements or force/moments are used to verify the contact states when the states are not clear. Our investigative action method can greatly reduce the friction and the probability of getting stuck when the robot performs the movement actions.

B. Controller Design:

We use the increment equation $x_{des} = x_t + u_t$ to avoid the potential "far away" problem for safety concerns; x_{des} is the desired EE pose, and x_t is the current EE pose; u_t is the increment action command from the agent. The Cartesian impedance controller takes the Cartesian EE movement u_t from agent at 0.5 to 2 Hz, and the output joint torque gives the command τ_u to the robot at 1000 Hz. We calculate the desired EE pose x_{des} by combining u_t with the current EE pose x_t. The trajectory generator bridges the low frequency output x_{des} of the agent and the high frequency impedance control of the robot and outputs $\xi_t = x_s$ to the Cartesian impedance controller in Equation (2). x_k is the position and q_k is the quaternion representation of the orientation given by a simple linear interpolator:

$$ \xi_t = \{\boldsymbol{x}_k, \boldsymbol{q}_k\}_{k=t}^{t+T}. \tag{9} $$

Algorithm 1 Visual Residual Reinforcement Learning with Investigative Action

Require: RL policy π_θ, fixed policy π_H.

for iteration=1 to M episodes **do**
  Copy latest policy π_θ from learning thread
  Sample initial state s_0
  for step=1 to N **do**
    Get action u_{RL} by greedily picking from $\pi_\theta(s_t)$
    Get action u_H from $\pi_H(s_v)$
    Output policy action: $u_t = (1-\alpha)u_H + \alpha * u_{RL}$
    if belief ==true **then**
      Get next state $u_t \rightarrow s_{t+1}$
    else
      Get next state $a_I, u_t \rightarrow s_{t+1}$
    end if
    Optimize π_θ with Equation (8)
    if EpisodeEnd == true **then**
      break
    end if
  end for
end for

V. EXPERIMENTS: DESIGN AND SETUP

We consider the experiment for the insertion task here. The task can be described as moving the already-grasped parts to their goal pose as shown in Figure 1. This is the most common setting in manufacturing. The success of such tasks can be measured by minimizing the distance between the objects and their goal pose especially in the Z direction (see Figure 1).

A. Experiment Algorithm Design

In our weighted residual RL, actions u_t are designed by adding the fixed policy $u_H = \pi_H(s_v)$ with the parametric policy $u_{RL} \sim \pi(\theta)(u_t|s_t)$:

$$u_t = (1-\alpha)u_H + \alpha * u_{RL}. \tag{10}$$

The fixed policy output u_H is calculated by a hand-designed controller as given in Equation (7); α helps to adjust the balance between exploration and exploitation. We set $\boldsymbol{k}_p$ to (1,1,0.3,0,0,0) when the fixed policy is calculated. To identify a reasonable weight between the two components, we initially experimented with the weighted residual RL by introducing a group of action weight parameters, such as 0.3, 0.5, and 0.7. The training experiments suggested an optimum policy output with a weight of 0.5, whereas the weight could increase or decrease around 0.5 according to the visual condition in the implementation phase. We utilized the algorithm to detect states and implemented its slightly-modified version where the trained policies were constructed by two aforementioned components. Here the flag belief is set to 0 or 1, according to the moment threshold settings, a detectable moment(over threshold) always gives true belief state. Combined with the investigative action mentioned in Section IV-A.2, the modified Q-learning algorithm was trained at a high speed, and it easily resulted in optimization.

1) Action Design: We design Cartesian movement actions for this experiment. Each Cartesian movement dimension was set to +1 for a positive movement and -1 for a negative movement; therefore, we had $6 * 2 = 12$ actions. We set λ as the scale parameter to adjust the amplitude of actions as

$$\boldsymbol{a} = \lambda[P^d_{\sigma x}, P^d_{\sigma y}, P^d_{\sigma z}, R^d_{\sigma x}, R^d_{\sigma y}, R^d_{\sigma z}]. \tag{11}$$

Here, P and R are positional and orientational movements under EE frame, respectively. λ is easy to choose because it is closely related to assembly clearance and visual accuracy, normally we set $\lambda = 0.002$, then we have movements resolution at 0.002 mm and 0.002 rad level. We found that orientational movements accuracy were enough by using the fixed policy u_H, so we only output positional movements actions in our RL idea, this is normal setting because the visual feedback and force feedback are complementary during contact-rich manipulation.

The *Investigative action* was designed as the force action ${}^eF_z = -25N$ under robot EE frame $\mathcal{F}_e$ for 1 s. The robot will try to add force but will stop moving if the force is greater than 25N or the movement is more than 3 mm. Then, the agent will obtain clear state feedback because of the large contact force and torque amplitude, as shown in Figure 5.

2) Reward Design: Depending on the pose error between the current picture and the target picture and the depth information, the reward function was set as follows:

$$r = \begin{cases} 1, & \text{(success)} \\ -2, & \text{(failed)}. \\ 1 - 150\|s_{xy}\|_2 - s/s_{max}, & \text{(otherwise)}. \end{cases}$$

Here, s_{xy} is the norm of the x and y errors of the images, s is the number of steps in one episode, and s_{max} is the maximum steps in one episode.

3) State Design: We get the estimated 6-DoF external force and moments along the X, Y, and Z axis under the EE frame from Franka controller. Here we consider the contact force and the moments between the robot's EE (i.e., the grasping RAM) and the slot as the MDP states as

$$\boldsymbol{s} = [F_x, F_y, F_z, M_x, M_y, M_z] \tag{12}$$

We assume that the EE contacts the slot when the external force $|F| > 4$ N or the external moments $|M| > 0.4$ Nm, a value of ± 1 means that a contact is made, and 0 means that there is no contact with the encoding states.

B. Experiment Environment and Task Setup

1) Environment Setup: We used the Franka robot [2] for real robot experiments and set the Cartesian stiffness as 3000 N/m and 300 Nm/rad (Recommended upper limit). Two sensor modalities were available in the real hardware, including proprioception and redgreenblue (RGB) depth camera. The RGB and depth information were recorded using the eye-in-hand Intel RealSense Depth Camera D435i. The policy ran on a Dell Precision 5510 laptop and sent the updated position to the real-time controller, which calculated the joint torque command and sent it to the robot controller at 1000 Hz. We used a CORSAIR DDR3 RAM and a motherboard as training and testing environment.

TABLE I

ABLATION STUDY OF POLICY EVALUATION STATISTICS

Baselines	Result(success/total)	Time Cost
No vision	92/200	1.09 h
No RL policy	112/200	0.65 h
Random RL policy	77/200	2.59 h
No investigative action	66/200	0.85 h
Our method	**179/200**	1.18 h

TABLE II

COMPARISON OF SUCCESS RATES FOR DIFFERENT BASELINES

Baselines	Fix motherboard	Move motherboard
Baseline 1	97/100	0/20
Baseline 2	100/100	0/20
Baseline 3	98/100	81/100
Baseline 4	100/100	88/100
Our method	**100/100**	**100/100**

2) Tasks Setup: In ablation study experiment, We evaluated our trained policy by masking different modalities as 4 baselines given below:

1) *No vision*: masks out the visual part action; we set $\alpha = 1$.
2) *No RL policy*: masks out the RL part action; we set $\alpha = 0$.
3) *Random policy*: generates a random Q table.
4) *No investigative action*: masks out the investigative action and chooses random action when the state is not clear.

We set maximum steps as 10 and add initial random errors($|error| \in [2, 3]mm$) in x and y directions for each baseline only in ablation study experiment.

In comparison study experiment, we compared the task success rates of our method with the other four baselines in the real scenarios (no maximum steps limit and no initial random errors for each baseline) by moving the motherboard, which are as follows:

1) Baseline 1: For normal teaching and direct insertion
2) Baseline 2: For normal teaching with spiral exploration
3) Baseline 3: For teaching with vision and direct insertion
4) Baseline 4: For teaching with vision and spiral exploration

VI. EXPERIMENTS: RESULTS AND DISCUSSION

We trained our policy with 500 episodes, and each episode lasted a maximum of 50 steps. The training time for the exploration was approximately 150 minutes which is much less than [19]. We specified discrete actions in this experiment, and the action execution had errors. Our policy can increase the probability of success and decrease the cost steps but cannot guarantee success every time. We set random errors for the initial pose of the robot; sometimes, the robot will successfully insert by chance and obtain a high reward in the early stage of training.

Table I shows the ablation study result of the policy evaluation statistics. *Random RL policy* and *No investigative action* had poor performances with success rates of 38.5% and 33%, respectively. *No vision* had a 46% success rate because of discrete overshooting actions whereas *No RL policy* had a 56% success rate because the RAM was always stuck by the short side of the slot. Our method had a success rate of 89.5%. It should be noted that the success rate of our method is limited buy the maximum steps in the experiment.

We observed that the absence of either visual or correct forces/moments information negatively affected the task success rate, and wrong policy performance was even worse than without RL policy. Therefore, the *Random RL policy* and *No investigative action* had similar performances because the RL policy is always in conflict with the visual output action. None of the four baselines has reached the same level of performance as the final method. With visual input alone, the robot sometimes cannot overcome the last small distance because of either the limited movement accuracy of the robot or contact friction, while RL policy are capable of recovering from such issues whcih could be proved in our method. Without the visual input, the robot will require more steps to find the proper pose for insertion and will always overshoot for some actions (i.e., drop out of the slot).

Table II shows a comparison of the success rates of different traditional method baselines. In order to simulate industrial scenario, the additional random error and maximum steps limit in ablation study are removed. Obviously, The baselines 1&2 work well only when the motherboard are fixed in the same position as in the teaching phase, so we only test 20 times in "move motherboard" case for baselines 1&2 for saving time. The success rates for the baselines 3&4 are increased with vision correction, but they still have failure cases due to the visual error. Our method shows a strong ability to tolerate environmental variations and resilience from stuck with full success which really meets the requirements of industrial scenarios. Please note that in the comparison study, the increase of success rates is also related to the removal of initial errors and the removal of limit of the maximum steps.

VII. CONCLUSION AND FUTURE WORK

In this paper, we combined RL with an operational space visual controller to solve position uncertainty problem in high-precision assembly tasks, and we proposed a proactive action idea to solve the POMDP problem by using an investigative action.

Our method could solve the shortage of traditional visual servoing method by using our visual residual RL algorithm, we inherits some traditional controller parameters which makes setting up not fast enough, we will extend our method to be trained towards end-to-end approach in next step.

Unfortunately, space does not permit more generalize test in this paper, while we test the SSD insertion scenario as Figure 2 with our policy and achieve full success with 100 times test. We will continue to generalize the model and policy so that they could handle different parts and robot manipulators in the next step. Then, the skill could be packaged as a service that will be delivered to robots in new factory lines with a short setup time. Our method uses a discrete number of actions to perform the insertion task, as an obvious next step, we will analyze the difference between this method and continuous space learning techniques.

REFERENCES

[1] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: an international journal*, vol. 34, no. 5, pp. 376–385, 2007.

[2] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, 2019.

[3] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A passivity based cartesian impedance controller for flexible joint robots-part ii: Full state feedback, impedance design and experiments," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2666–2672.

[4] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stamigioli, and G. Hirzinger, "A passivity based cartesian impedance controller for flexible joint robots-part i: Torque feedback and gravity compensation," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2659–2665.

[5] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *The international journal of robotics research*, vol. 26, no. 1, pp. 23–39, 2007.

[6] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: Detection, isolation, and identification," *Submitted to IEEE Transactions on Robotics*, 2015.

[7] L. Robot. Desktop computer host automatic assembly line. Youtube. [Online]. Available: https://www.youtube.com/watch?v=GNqNVgLk1Mg

[8] H. Park, J.-H. Bae, J.-H. Park, M.-H. Baeg, and J. Park, "Intuitive peg-in-hole assembly strategy with a compliant manipulator," in *IEEE ISR 2013*. IEEE, 2013, pp. 1–5.

[9] F. EMIKA. Ram. Youtube. [Online]. Available: https://www.youtube.com/watch?time_continue=1&v=HQt7XZB-rts&feature=emb_logo

[10] M. A. Peshkin, "Programmed compliance for error corrective assembly," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 473–482, 1990.

[11] J. M. Schimmels and M. A. Peshkin, "Admittance matrix design for force-guided assembly," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 213–227, 1992.

[12] ——, "Force-assembly with friction," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 465–479, 1994.

[13] A. Stemmer, G. Schreiber, K. Arbter, and A. Albu-Schaffer, "Robust assembly of complex shaped planar parts using vision and force," in *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2006, pp. 493–500.

[14] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *Pattern recognition*, vol. 5, no. 2, pp. 99–108, 1973.

[15] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[16] C. Teulière and E. Marchand, "Direct 3d servoing using dense depth maps," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1741–1746.

[17] H. Fujimoto, "Visual servoing of 6 dof manipulator by multirate control with depth identification," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 5. IEEE, 2003, pp. 5408–5413.

[18] R. Li and H. Qiao, "A survey of methods and strategies for high-precision robotic grasping and assembly taskssome new trends," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2718–2732, 2019.

[19] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," *arXiv preprint arXiv:1810.10191*, 2018.

[20] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2062–2069.

[21] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," *arXiv preprint arXiv:1906.05841*, 2019.

[22] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 819–825.

[23] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel, "Reinforcement learning on variable impedance controller for high-precision robotic assembly," *arXiv preprint arXiv:1903.01066*, 2019.

[24] W. S. Newman, Y. Zhao, and Y.-H. Pao, "Interpretation of force and moment signals for compliant peg-in-hole assembly," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1. IEEE, 2001, pp. 571–576.

[25] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," *arXiv preprint arXiv:1812.03201*, 2018.

[26] M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, and D. Fox, "Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning," *arXiv preprint arXiv:2005.10872*, 2020.

[27] H. Park, J. Park, D.-H. Lee, J.-H. Park, M.-H. Baeg, and J.-H. Bae, "Compliance-based robotic peg-in-hole assembly strategy without force feedback," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6299–6309, 2017.

[28] A. Y. Ng, "Shaping and policy search in reinforcement learning," Ph.D. dissertation, University of California, Berkeley Berkeley, 2003.

[29] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Efficient dynamic-programming updates in partially observable markov decision processes," 1995.

[30] C. C. White, "A survey of solution techniques for the partially observed markov decision process," *Annals of Operations Research*, vol. 32, no. 1, pp. 215–230, 1991.

[31] W. S. Lovejoy, "A survey of algorithmic methods for partially observed markov decision processes," *Annals of Operations Research*, vol. 28, no. 1, pp. 47–65, 1991.

[32] P. Martinet, J. Gallice, and K. Djamel, "Vision based control law using 3d visual features," in *World Automation Congress, WAC'96, Robotics and Manufacturing Systems*, vol. 3, 1996, pp. 497–502.

[33] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[34] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[35] T. Breuer, M. Ndoundou-Hockemba, and V. Fishlock, "First observation of tool use in wild gorillas," *PLoS Biology*, vol. 3, no. 11, 2005.