micromachines

Article

Adaptive Sliding Mode Disturbance Observer and Deep Reinforcement Learning Based Motion Control for Micropositioners

Shiyun Liang [1,†], Ruidong Xi [1,†], Xiao Xiao [2] and Zhixin Yang [1,*]

[1] State Key Laboratory of Internet of Things for Smart City and Department of Electromechanical Engineering, University of Macau, Macau 999078, China; mb95407@um.edu.mo (S.L.); yb57466@umac.mo (R.X.)
[2] Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China; xiaox@sustech.edu.cn
* Correspondence: zxyang@um.edu.mo; Tel.: +853-8822-4456
† These authors contributed equally to this work.

Abstract: The motion control of high-precision electromechanitcal systems, such as micropositioners, is challenging in terms of the inherent high nonlinearity, the sensitivity to external interference, and the complexity of accurate identification of the model parameters. To cope with these problems, this work investigates a disturbance observer-based deep reinforcement learning control strategy to realize high robustness and precise tracking performance. Reinforcement learning has shown great potential as optimal control scheme, however, its application in micropositioning systems is still rare. Therefore, embedded with the integral differential compensator (ID), deep deterministic policy gradient (DDPG) is utilized in this work with the ability to not only decrease the state error but also improve the transient response speed. In addition, an adaptive sliding mode disturbance observer (ASMDO) is proposed to further eliminate the collective effect caused by the lumped disturbances. The micropositioner controlled by the proposed algorithm can track the target path precisely with less than 1 μm error in simulations and actual experiments, which shows the sterling performance and the accuracy improvement of the controller.

Keywords: micropositioners; reinforcement learning; disturbance observer; deep deterministic policy gradient

Citation: Liang, S.; Xi, R.; Xiao, X.; Yang, Z. Adaptive Sliding Mode Disturbance Observer and Deep Reinforcement Learning Based Motion Control for Micropositioners. *Micromachines* **2022**, *13*, 458. https://doi.org/10.3390/mi13030458

Academic Editor: Duc Truong Pham

Received: 28 February 2022
Accepted: 15 March 2022
Published: 17 March 2022

1. Introduction

Micropositioning technologies based on smart materials in precision industries have gained much attention for numerous potential applications in optical steering, micro-assembly, nano-inscribing, cell manipulation, etc. [1–7]. One of the greatest challenge in this research field is the uncertainties produced by various factors such as the dynamic model, environmental temperature, sensors performance, and the actuators' nonlinear characteristics [8,9], which make the control of micropositioning system a demanding problem.

To address the uncertain problem, different kinds of control approach have been developed, such as the PID control method [10], sliding mode control [11,12], and adaptive control [13]. In addition, many researchers have integrated these control strategies to further improve the control performance. Victor et al. have proposed a scalable field-programmable gate array-based motion control system with a parabolic velocity profile [14]. A new seven-segment profile algorithm was developed by Jose et al. to improve the performance of the motion controller [15]. Combined with the backstepping strategy, Fei et al. proposed an adaptive fuzzy sliding mode controller in [16]. Based on the radial basis function neural network (RBFNN) and sliding mode control (SMC), Ruan et al. developed a RBFNN-SMC for nonlinear electromechanical actuator systems [17]. Gharib et al. designed

a PID controller with a feedback linearization technique for path tracking control of a micropositioner [18]. Nevertheless, the performance and robustness of such model-based control strategies are still limited by the precision of the dynamics model. On the other hand, a sophisticated system model frequently leads to a complex control strategy. Although many researchers have considered the factors of uncertainties and disturbances, it is still difficult for the system to provide a precise and comprehensive process.

As the rapid development in artificial intelligence in recent years has roundly impacted the traditional control field, learning-based and data-driven approaches, especially reinforcement learning (RL) and neural networks, have become a promising research tropic. Different from traditional control strategies that need to make assumptions based on the dynamics model [19,20], reinforcement learning can directly learn the policy by interacting with the system. Back in 2005, Adda et al. presented a reinforcement learning algorithm for learning control of stochastic micromanipulation systems [21]. Li et al. designed a state–action–reward–state–action (SARSA) method using linear function approximation to generate an optimal path by controlling the choice of the micropositioner [22]; however, the reinforcement learning algorithms such as Q-learning [23] and SARSA [24] utilized in the aforementioned works are unable to deal with complex dynamics problems, especially the continuous state action space problem. With the spectacular improvement enjoyed by deep reinforcement learning (DRL), primarily driven by deep neural networks (DNN) [25], the DRL algorithms, such as the deep Q network (DQN) [26], policy gradient (PG) [27], deterministic policy gradient (DPG) [28], and deep deterministic policy gradient (DDPG) [29] with the ability to approximate the value function, have played an important role in continuous control tasks.

Latifi et al. introduced a model-free neural fitted Q iteration control method for micromanipulation devices; in this work, the DNN is adopted to represent Q-value function [30]. Leinen introduced the concept of experience playback in DQN and the approximate value function of the neural network into the SARSA algorithm for the control of a scanning probe microscope [31]. Both simulation and real experimental results have shown that their proposed RL algorithm based on the neural network could achieve better performance compared to traditional control methods to some extent; however, due to the collective effects of disturbances generated from nonlinear systems and deviations in value functions [29,32,33], the RL control method could induce significant inaccuracies in the tracking control tasks [34]. To improve the anti-disturbance capability and control accuracy, disturbance rejection control [35], time-delay estimation based control [36], disturbance observer-based controllers [37,38] have been proposed successively. To deal with this issue, a deep reinforcement learning controller integrated with an adaptive sliding mode disturbance observer (ASMDO) is developed in this work. Previous research on trajectory tracking control of DRL has shown that apparent state errors have always existed [39–42]. One of the main reasons is the inaccurate estimation of the action value function in DRL structure. As indicated in [43], even in elementary control tasks, accurate action values cannot be attained from the same action value function; therefore, in this work, the DDPG algorithm is developed with an integral differential compensator (DDPG-ID) added to cope with this situation. In addition, the comparison of the reinforcement learning control method with various common state-of-the-art control methods are listed in Table 1, which shows the pros and cons of these different methods.

In this study, deep reinforcement learning is leveraged into a novel optimal control scheme for complex systems. An anti-disturbance, stable, and precise control strategy is proposed for the trajectory tracking task of the micropositioner system. The contribution of this works are presented as follows:

(1) A DDPG-ID algorithm based on deep reinforcement learning is introduced as a basic micropositioner system motion controller, which avoided the limitation of traditional control strategies to the accuracy and comprehensiveness of the dynamic model;

(2) To eliminate the collective effect caused by the lumped disturbances from the micropositioner system and inaccurate estimation of the value function in deep reinforcement learning, an adaptive sliding mode disturbance observer (ASMDO) is proposed;
(3) An integral differential compensator is introduced in DDPG-ID to compensate for the feedback state of the system, which improves the accuracy and response time of the controller, and further improves the robustness of the controller subject to external disturbances.

The manuscript is structured as follows. Section 2 presents the system description of the micropositioner. In Section 3, we develop a deep reinforcement learning control method combined with ASMDO and compensator, and parameters of the DNNs are illustrated. Then, simulation parameters and tracking results are given in Section 4. To further evaluate the performance of the proposed control strategy in the micropositioner, tracking experiments are presented in Section 4. Lastly, conclusions are given in Section 5.

Table 1. Comparison of different control algorithms.

Method	Advantages	Disadvantages
PID control	Simple design structure Easy to implementation	Mainly used in linear systems Requirement of full-state feedback Lack of adaptivity
SMC control	Simple design structure Easy to implementation High robustness	Excessive chattering effect Lack of adaptivity
Adaptive control	Lower initial cost Lower cost of redundancy High reliability and performance	Stability is not treated rigorously High gain observes needed Slow convergence
Backstepping control	Global stability Simple design structure Easy to be integrated	Low anti-interference ability Sensitive to system models Lack of adaptivity
RL control	No need of accurate model Improved control performance High adaptivity	Poor anti-interference ability Easy to generate state error

2. System Description

The basic structure of micropositioner is shown in Figure 1, which consists of a base, a platform, and a kinematic device. The kinematic device is composed with an armature, an electromagnetic actuator, and a chain mechanism driven by electromagnetic actuator. As shown in Figure 1, there are mutual-perpendicular compliant chains actuated by the electron-magnetic actuator (EMA) in the structure. The movement of the chain mechanism is in accordance with the working air gap y. The EMA generates the magnetic force T_m, which can be approximated as:

$$T_m = k\left(\frac{I_c}{y+p}\right)^2 \tag{1}$$

where k and p are constant parameters related to the electronmagnetic actuator, I_c is the excitation current, and y is the working air gap between the armature and the EMA. Then, the electrical model of the system can be given as:

$$V_i = RI_c + \frac{d}{dt}(HI_c) \tag{2}$$

where V_i is the input voltage from the EMA, R is the resistance of the coil and H denotes the coil inductance, which can be given as:

$$H = H_1 + \frac{pH_0}{y+p} \tag{3}$$

where H_1 is the coil inductance while the air gap is infinite, and H_0 is the incremental inductance when the gap is zero. The motion equation for the micropositioner can be expressed as:

$$m\frac{d^2y}{dt^2} = \iota(\alpha_0 - y) - T_m \tag{4}$$

where ι is the stiffness along the motion direction in the system, and α_0 is the initial air gap.

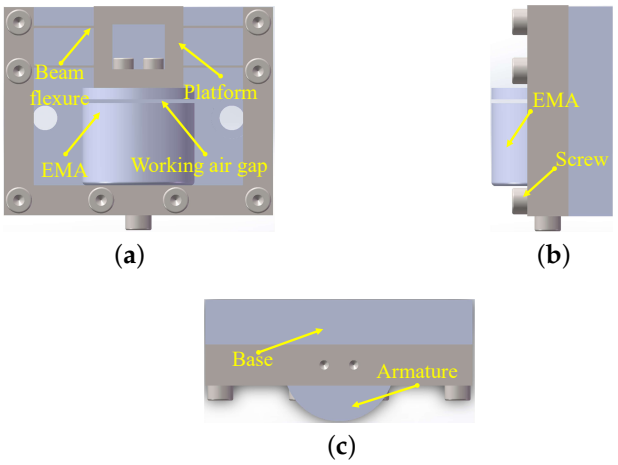


Figure 1. The diagrammatic model of EMA actuated micropositioner. (**a**) The front view of micropositioner. (**b**) The end view of micropositioner. (**c**) The vertical view of micropositioner.

According to Equations (1)–(4), they define $x_1 = y$, $x_2 = \dot{y}$, $x_3 = I_c$ as the state variables and the control input $u = V_i$. Then, the dynamics model of the electromagnetic actuator can be written as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{\iota}{m}(\alpha_0 - x_1) - \frac{k}{m}\left(\frac{x_3}{x_1+p}\right)^2 \\ \dot{x}_3 = \frac{1}{H}\left(-Rx_3 + \frac{H_0 p x_2 x_3}{(x_1+p)^2} + u\right) \end{cases} \tag{5}$$

Define the variables $z_1 = x_1$, $z_2 = x_2$, $z_3 = \frac{\iota}{m}(\alpha_0 - x_1) - \frac{k}{m}\left(\frac{x_3}{x_1+p}\right)^2$, then we have

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = f(x) + g(x)u \end{cases} \tag{6}$$

where $f(x) = -\frac{\iota x_2}{m} + \frac{2kx_3^2}{m(x_1+p)^2}\left(\frac{H(x_1+p)-pH_0}{H(x_1+p)^2}x_2 + \frac{R}{H}\right)$, $g(x) = -\frac{2kx_3}{Hm(x_1+p)^2}$, and z_1 is the system output.

In realistic engineering application, there always exist some uncertainties of the system, then system Equation (6) can be rewritten as:

$$\begin{cases} \dot{z}_i = z_{i+1}, i = 1,2 \\ \dot{z}_3 = f_0(x) + g_0(x)u + (\Delta f(x) + \Delta g(x)u) + d \end{cases} \tag{7}$$

where $f_0(x)$ and $g_0(x)$ denote the nominal part of the micropositioner system and $\Delta f(x)$, $\Delta g(x)$ denote the uncertainties of the modeling system; d denotes the external disturbances. Then, defining $D = (\Delta f(x) + \Delta g(x)u) + d$, we have

$$\begin{cases} \dot{z}_i = z_{i+1}, i = 1,2 \\ \dot{z}_3 = f_0(x) + g_0(x)u + D \end{cases} \tag{8}$$

where D is the lumped system disturbances. The following assumption is exploited [44]:

Assumption 1. *The lumped interference D is bounded and its upper bound is less than a fixed parameter β_1 and the derivative of D is unknown but bounded.*

Remark 1. *Assumption 1 is reasonable since all micropositioner platforms are accurately designed and parameter identified, and all disturbances are remained in a controllable domain.*

3. Design of ASMDO and DDPG-ID Algorithm

In this section, the adaptive sliding mode disturbance observer (ASMDO) is introduced based on the dynamics of the micropositioner. Then, the DDPG-ID control method and pseudocode are given.

3.1. Design of Adaptive Sliding Mode Disturbance Observer

To develop the ASMDO, a virtual dynamic is firstly designed as

$$\begin{cases} \dot{\eta}_i = \eta_{i+1}, i = 1, 2 \\ \dot{\eta}_3 = f(z) + g(z)u + \hat{D} + \rho \end{cases} \tag{9}$$

where $\eta_i, i = 1, 2, 3$ are auxiliary variables, $\hat{D}$ is the estimation of lumped disturbances, ρ denotes the sliding mode term, which is introduced afterwards.

Define a sliding variable $S = \sigma_3 + k_2\sigma_2 + k_1\sigma_1$, where $\sigma_i = x_i - \eta_i, i = 1, 2, 3$, k_1 and k_2 are positive design parameters. Then the sliding mode term ρ is designed as

$$\rho = \lambda_1 S + k_2\sigma_3 + k_1\sigma_2 + \lambda_2 \mathrm{sgn}(S) \tag{10}$$

where λ_1, λ_2 are positive design parameters with $\lambda_2 \geq \beta_1$.

Choosing an unknown constant β_2 to present the upper bound of $\dot{D}$, the ASMDO is proposed as:

$$\dot{\hat{D}} = k(\dot{x}_3 - f_0(z) - g_0(z)u - \hat{D}) + (\hat{\beta}_2 + \lambda_3)\mathrm{sgn}(\rho) \tag{11}$$

where k and λ_3 are positive design parameters and $\hat{\beta}_2$ is defined as the estimation of β_2 given by $\dot{\hat{\beta}}_2 = -\delta_0\hat{\beta}_2 + \|\rho\|$, with δ_0 is a small positive number.

Then, the output $\hat{D}$ of the ASMDO is used as a compensation of the control input to eliminate the uncertainties generated by the system and external disturbances.

Remark 2. *Choosing $V_1 = \frac{1}{2}S^2$ and $V_2 = \frac{1}{2}(\tilde{D}^2 + \tilde{\beta}_2^2)$, where $\tilde{D} = D - \hat{D}$, $\tilde{\beta}_2 = \beta_2 - \hat{\beta}_2$ as two Lyapunov function, derivative V_1 and V_2 with respect to time, it is easy to prove that both S and $\tilde{D}$ will exponentially converge to the equilibrium point, so the proof process is not repeated.*

3.2. Design of DDPG-ID Algorithm for Micropositioner

The goal of reinforcement learning is to obtain a policy for the agent that could maximizes the cumulative reward through interactions with the environment. The environment is usually formalized as a Makov decision process (MDP) described by a four-tuple (S, A, P, R), where S, A, P, and R represent the state space of environment, set of actions, state transition probability function, and reward function separately. At each time step t, the agent in current state $s_t \in S$ takes action $a_t \in A$ from policy $\pi(a_t|s_t)$, then the agent acquires a reward $r_t \leftarrow R(s_t, a_t)$ and enters the next state s_{t+1} according to the state transition probability function $P(s_{t+1}|s_t, a_t)$. Based on the Markov property, the Bellman equation of action–value function $Q_\pi(s_t, a_t)$, which is used for calculating the future expected reward, can be given as:

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi(r_t + \gamma Q_\pi(s_{t+1}, a_{t+1})) \tag{12}$$

where $\gamma \in [0, 1]$ denotes the discount factor.

In trajectory tracking control task of micropositioner, state s_t is state array about the air gap y of micropositioner at time t. Action a_t is the voltage u applied by the controller

to micropositioner. As shown in Figure 2, DDPG is one of actor–critic algorithms, which has an actor and a critic. The actor is responsible for generating actions and interacting with the environment, and the critic evaluates the performance of the actor and guides the action in the next state.

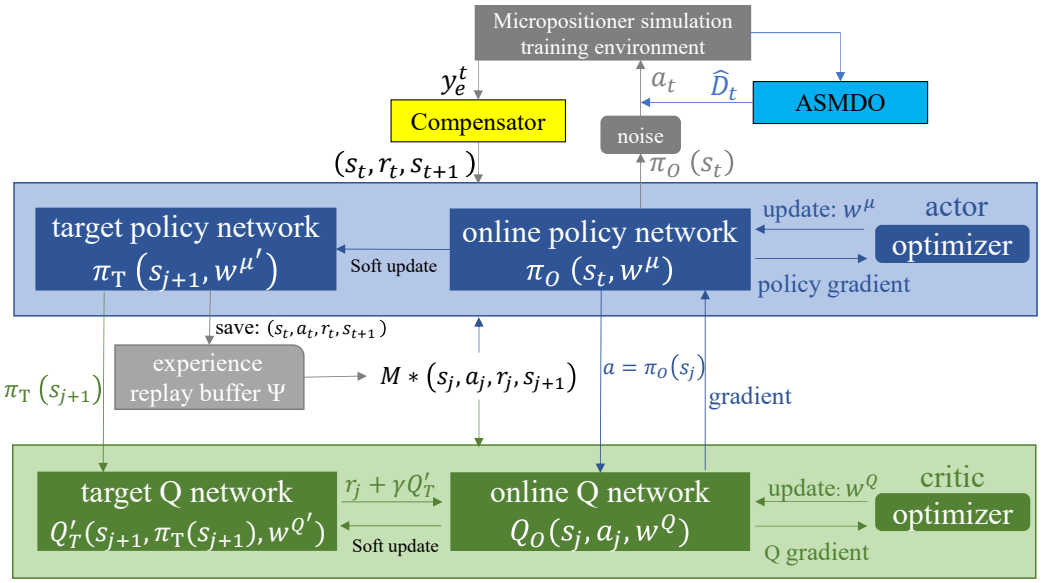


Figure 2. The structure diagram of DDPG-ID algorithm.

The action–value function and policy approximation are parameterized by DNN to solve the continuous states and actions problem in micropositioner with $Q(s_t, a_t, w^Q) \doteq Q_\pi(s_t, a_t)$, $\pi_{w^\mu}(a_t|s_t) \doteq \pi(a_t|s_t)$, where w^Q and w^μ are the parameters of neural networks in action–value function and policy function. Under the prerequisite of using the neural network approximation representation policy function, the neural network gradient update method is used to seek the optimal policy π.

DDPG-ID uses deterministic policy $\pi(s_t, w^\mu)$ rather than traditional stochastic policy $\pi_{w^\mu}(a_t|s_t)$, where the output of policy is the action a_t with highest probability to current state s_t, $\pi(s_t, w^\mu) = a_t$. The policy gradient is given as

$$\nabla_{w^\mu} J(\pi) = \mathbb{E}_{s\sim\rho^\pi}[\nabla_{w^\mu} \pi(s, w^\mu) \nabla_a Q(s, a, w^Q)\Big] \tag{13}$$

where $J(\pi) = \mathbb{E}_\pi[\sum_{t=1}^{T} \gamma^{(t-1)} r_t]$ is the expectation of discount accumulative rewards, T denotes the final time of a whole process, ρ^π is the distribution of state following the deterministic policy. Value function $Q(s_t, a_t, w^Q)$ is updated by calculating time temporal-difference error (TD-error), which can be defined as

$$e_{TD} = r_t + \gamma Q(s_{t+1}, \pi(s_{t+1})) - Q(s_t, a_t) \tag{14}$$

where e_{TD} is the TD-error, $r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}))$ represents the TD target value. By minimizing the TD-error, the parameters are updated backwards through the neural network gradient.

To avoid the convergence problem of single network caused by correlation between TD target value and current value [45,46], A target Q network $Q'_T(s_{t+1}, a'_{t+1}, w^{Q'})$ is introduced to calculate network portion of TD target value and an online Q network $Q_O(s_t, a_t, w^Q)$ is used to calculate current value in critic. Both these two DNN have the same structure. The actor also has an online policy network $\pi_O(s_t, w^\mu)$ to generate current action and a target policy network $\pi_T(s_t, w^{\mu'})$ to provide the target action a'_{t+1}. $w^{\mu'}$ and $w^{Q'}$ separately represent the parameters of target policy and target Q networks.

In order to improve the stability and efficiency during RL training, experience replay technology is utilized in this work, which saves transition experience (s_t, a_t, r_t, s_{t+1}) into the experience replay buffer Ψ at each interaction with the environment for subsequent updates. In each training time t, a minibatch of M transitions (s_j, a_j, r_j, s_{j+1}) from the experience replay buffer are extracted to calculate the gradients and update neural networks.

An integral differential compensator is developed in deep reinforcement learning structure to improve the accuracy and responsiveness of tracking tasks in this work, which is shown in Figure 2. The integral portion of the state is utilized to increase the control input continuously, which would eventually reduce tracking error. The differential part is integrated to reduce the system oscillation and accelerates stability. The proposed compensator is designed as follows:

$$s_{ID}^t = y_e^t + \alpha \sum_{n=1}^{t} y_e^t + \beta\left(y_e^t - y_e^{t-1}\right) \tag{15}$$

where s_{ID}^t represents the compensator error at time t, $y_e^t = \sqrt{\left(y_d^t - \hat{y}^t\right)^2}$, y_d^t represents the desired trajectory at time t, $\hat{y}^t$ is the measured air gap at time t and y_e^t is the error between them. α is the integral gain and β is the differential gain.

Then the state s_t at time t can be described as:

$$s_t = \begin{bmatrix} s_{ID}^t & \hat{y}^t & \dot{\hat{y}}^t & y_d^t & \dot{y}_d^t \end{bmatrix}^T \tag{16}$$

where $\dot{\hat{y}}^t$ and $\dot{y}_d^t$ represent the derivatives of $\hat{y}^t$ and y_d^t.

The reward r_t function designed is to measure the tracking error:

$$r_t = \begin{cases} -4, y_e^t > 0.005 \\ +5, 0.003 < y_e^t \leqslant 0.005 \\ +10, 0.001 < y_e^t \leqslant 0.003 \\ +18, y_e^t \leqslant 0.001 \end{cases} \tag{17}$$

As shown in Figure 3, the adaptive sliding mode disturbance observer (ASMDO) is embedded into the DDPG-ID between the actor and micropositioner system environment. Action a_t with the environment is expressed as

$$a_t = \pi_O(s_t, w^\mu) + \hat{D}_t + \mathcal{N}_t \tag{18}$$

where w^μ is the parameters of online policy network π_O, $\hat{D}_t$ is the estimation of the micropositioner system at time t, and $\mathcal{N}_t$ is Gaussian noise for action exploration.

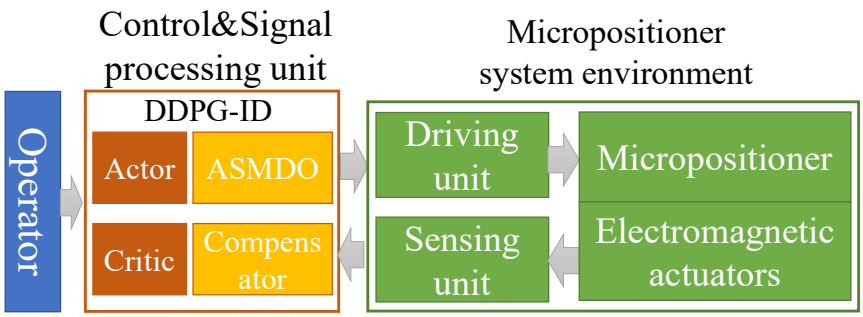


Figure 3. System signal flow chart.

3.2.1. Critic Update

After selecting M transitions (s_j, a_j, r_j, s_{j+1}) samples from experience replay buffer Ψ, the Q value is calculated. The online Q network is responsible for calculating the current Q value, which is as follows:

$$Q_O(s_j, a_j, w^Q) = w^Q \phi(s_j, a_j) \tag{19}$$

where $\phi(s_j, a_j)$ represents the input of online Q network, which is an eigenvector consisting of state s_j and action a_j.

The target Q network Q'_T is defined as:

$$Q'_T(s_{j+1}, \pi_T(s_{j+1}, w^{\mu'}), w^{Q'}) = w^{Q'} \phi(s_{j+1}, \pi_T(s_{j+1}, w^{\mu'})) \tag{20}$$

where $\phi(s_{j+1}, \pi_T(s_{j+1}, w^{\mu'}))$ is the input of the target Q network, which is a eigenvector consisting state s_{j+1} and target policy network output $\pi_T(s_{j+1}, w^{\mu'})$.

For target policy network π_T, the equation is:

$$\pi_T(s_{j+1}, w^{\mu'}) = w^{\mu'} s_{j+1} \tag{21}$$

Then, we rewrite the target Q value Q_T as:

$$Q_T = r_j + \gamma Q'_T(s_{j+1}, \pi_T(s_{j+1}, w^{\mu'}), w^{Q'}) \tag{22}$$

where r_j is the reward from the selected samples.

Since M transitions (s_j, a_j, r_j, s_{j+1}) are sampled from experience buffer Ψ, the loss function of the update critic is shown in Equation (23).

$$\mathcal{L}\left(w^Q\right) = \frac{1}{M} \sum_{j=1}^{M} \left(Q_T - Q_O\left(s_j, a_j, w^Q\right)\right)^2 \tag{23}$$

where $\mathcal{L}\left(w^Q\right)$ is the loss value of critic.

In order to smooth the target network update process, the soft update is applied without copying parameters periodically as:

$$w^{Q'} \leftarrow \tau w^Q + (1 - \tau) w^{Q'} \tag{24}$$

where τ is the update factor, usually a small constant.

The diagram of Q network is shown in Figure 4 , which is a parallel neural network. The Q network includes both state and action portions, and the output value of Q network is based on state and action. The state portion of the neural network consists of a state input layer, three full connection layers, and two ReLU layers clamped between the three full connection layers. The neural network of the action portion contains an action input layer and a full connection layer. The output layers of the above two portions are combined entering the neural network of the common part, which contains a ReLU layer and one output layer.

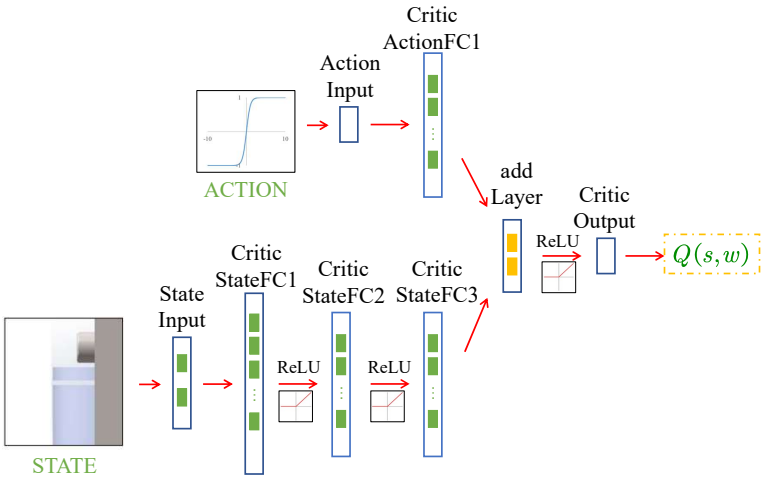


Figure 4. The diagram of Q network.

The parameters of each layer in the Q network are shown in Table 2.

Table 2. Q network parameters.

Network Layer Name	Number of Nodes
StateLayer	5
CriticStateFC1	120
CriticStateFC2	60
CriticStateFC3	60
ActionInput	1
CriticActionFC1	60
addLayer	2
CriticOutput	1

3.2.2. Actor Update

The output of online policy network is

$$\pi_O = w^{\mu} s_j \tag{25}$$

On account of using deterministic policy, the calculation of the policy gradient has no integrals of action a, but instead has the derivatives of the value function Q_O with respect to action a in comparison with stochastic policy. The gradient formula can be rewritten as follows:

$$\nabla_{w^{\mu}} J \approx \frac{1}{M} \sum_{j}^{M} (\nabla_{a_j} Q_O(s_j, aj, w^{Q}) \nabla_{w^{\mu}} \pi_O(s_j, w^{\mu})) \tag{26}$$

where the weights w^{μ} are updated with the gradient back-propagation method. The target policy network is also updated with soft update pattern as follows:

$$w^{\mu'} \leftarrow \tau w^{\mu} + (1 - \tau) w^{\mu'} \tag{27}$$

where τ is the update factor, usually a small constant.

Figure 5 shows the diagram of the policy network in this paper, which contains a state input layer, a full connection layer, a tanh layer, and an output layer. The parameters of each layer in the policy network are shown in Table 3.

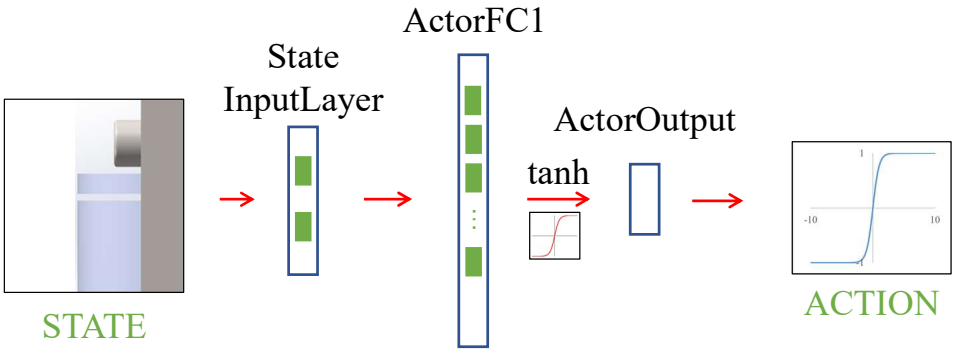


Figure 5. The diagram of policy network.

Table 3. Policy network parameters.

Network Layer Name	Number of Nodes
StateLayer	5
ActorFC1	30
ActorOutput	1

The Algorithm 1 pseudocode can be shown as:

Algorithm 1 DDPG-ID Algorithm.

Randomly initialize online Q network with weights w^Q
Randomly initialize online policy network with weights w^μ
Initialize the target Q network by $w^{Q'} \leftarrow w^Q$
Initialize the target policy network by $w^{\mu'} \leftarrow w^\mu$
Initialize the experience replay buffer Ψ
Load the simplified micropositioner dynamic model
for episode = 1, MaxEpisode **do**
  Initialize a noise process $\mathcal{N}$ for exploration
  Initialize ASMDO and ID compensator
  Randomly initialize micropositioner states
  Receive initial observation state s_1
  for step = 1, T **do**
    Select action $a_t = \pi_O(s_t) + \hat{D}_t + \mathcal{N}_t$
    Use a_t to run micropositioner system model
    Process errors with integral differential compensator
    Receive reward r_t and new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer Ψ
    Randomly sample a minibatch of M transitions (s_j, a_j, r_j, s_{j+1}) from Ψ
    Set $Q_T = r_j + \gamma Q'_T(s_{j+1}, \pi_T(s_{j+1}, w^{\mu'}), w^{Q'})$
    Minimize loss: $\mathcal{L}(w^Q) = \frac{1}{M}\sum_{j=1}^{M}(Q_T - Q_O(s_j, a_j, w^Q))^2$ to update online Q network
    Use the sampled policy gradient to update online policy network:
    $\nabla_{w^\mu} J = \frac{1}{M}\sum_j^M(\nabla_{a_j} Q_O(s_j, a_j, w^Q)\nabla_{w^\mu}\pi_O(s_j, w^\mu))$
    Update the target networks: $w^{Q'} \leftarrow \tau w^Q + (1-\tau)w^{Q'}, w^{\mu'} \leftarrow \tau w^\mu + (1-\tau)w^{\mu'}$
  end for
end for

4. Simulation and Experimental Results

In this section, two kinds of periodic external disturbances were added to verify the practicability of the proposed ASMDO and three distinct desired trajectories were utilized to evaluate the performance of proposed deep reinforcement learning control strategy. An traditional DDPG algorithm and a well-tuned PID strategy were adopted for comparison.

To further verify the spatial performances of the proposed algorithm, two kinds of different trajectories were introduced in the experiments.

4.1. Simulation Results

The parametric equations of two kinds of periodic external disturbances are defined as $d_1 = 0.1\sin(2\pi t) + 0.1\sin(0.5\pi t + \frac{\pi}{3})$, and $d_2 = 0.1 + 0.1\sin(0.5\pi t + \frac{\pi}{3})$. Based on the micropositoner model proposed in [44], the effectiveness of the observer is presented in Figures 6 and 7

The disturbance estimation results from the proposed ASMDO are presented in Figures 6a and 7a, it is can be seen that the observer could track the given disturbance rapidly. The estimation errors are less than 0.01 mm in Figures 6b and 7b, which shows the effectiveness of the ASMDO as interference compensation.

The dynamics model of micropositioner is given in Section 2, and its basic system model parameters are from our previous research [44,47], which is shown in Table 4. The DDPG algorithm is defined in same neural network structure and training parameters as DDPG-ID in this paper. The training parameters of the DDPG-ID and DDPG are shown in Table 5.

The first desired trajectory designed for tracking control simulation is a waved signal. According to the initial conditions, the parametric equation of the waved trajectory is defined as:

$$y_d(t) = 0.985 - 0.015\sin(\frac{\pi t}{4} - \frac{\pi}{2}) \tag{28}$$

The training process of both DDPG-ID and DDPG are run on the same model with stochastic initialized micropositioner states. During the training evaluation, a larger episode reward indicates a more accurate and lower error control policy. It is shown in Figure 8 that DDPG-ID reaches the maximum reward score with fewer episodes compared to DDPG, which reveals that DDPG-ID algorithm converge faster than DDPG algorithm. Comparing Figure 8a with Figure 8b, the average reward of DDPG-ID training process is larger than DDPG's average reward in stable state, which further indicates that policy learned by DDPG-ID algorithm has better performance. The trained algorithms are employed for tracking control of micropositioner system simulation experiments.

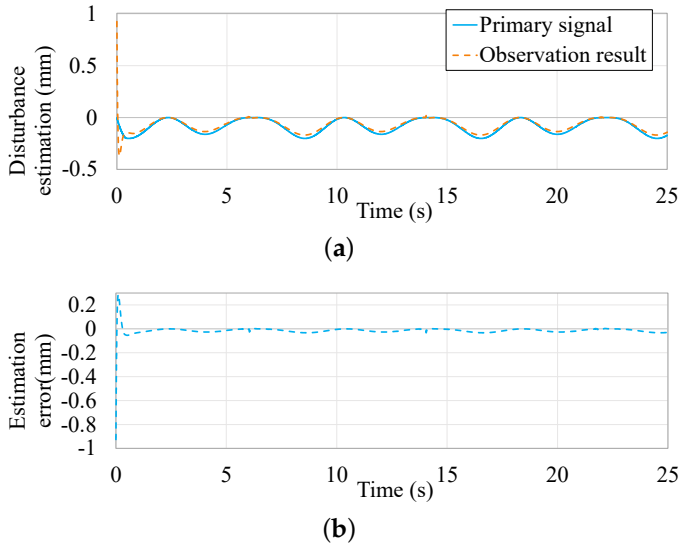


Figure 6. Observation result of ASMDO with d_2. (**a**) Observing result based on the ASMDO. (**b**) Observing error based on the ASMDO.

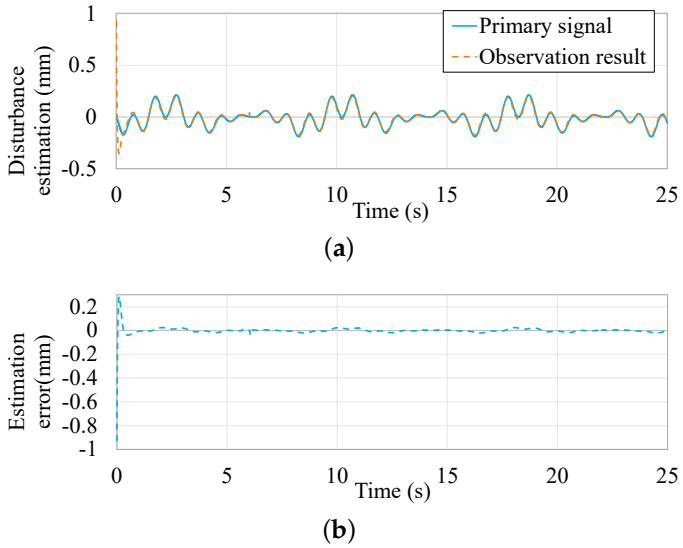


Figure 7. Observation result of ASMDO with d_1. (**a**) Observing result based on the ASMDO. (**b**) Observing error based on the ASMDO.

Table 4. Parameters of the micropositioner model.

Notation	Value	Unit
L_1	13.21	H
L_0	0.67	H
a	1.11×10^{-5}	m
R	43.66	Ω
c	8.83×10^{-5}	$Nm^2\ A^{-2}$
k	1.803×10^{N5}	Nm^{-1}
m	0.0272	Kg

Table 5. Training parameters of DDPG-ID and DDPG.

Hyperparameters	Value
Learning rate for actor φ_1	0.001
Learning rate for critic φ_2	0.001
Discount factor γ	0.99
Initial exploration ε	1
Experience replay buffer size ψ	100,000
Minibatch size M	64
Max episode ϖ	1500
Soft update factor τ	0.05
Max exploration steps T	250 (25 s)
Time step T_s	0.01 s
Intergal gain α	0.01
Differential gain β	0.001

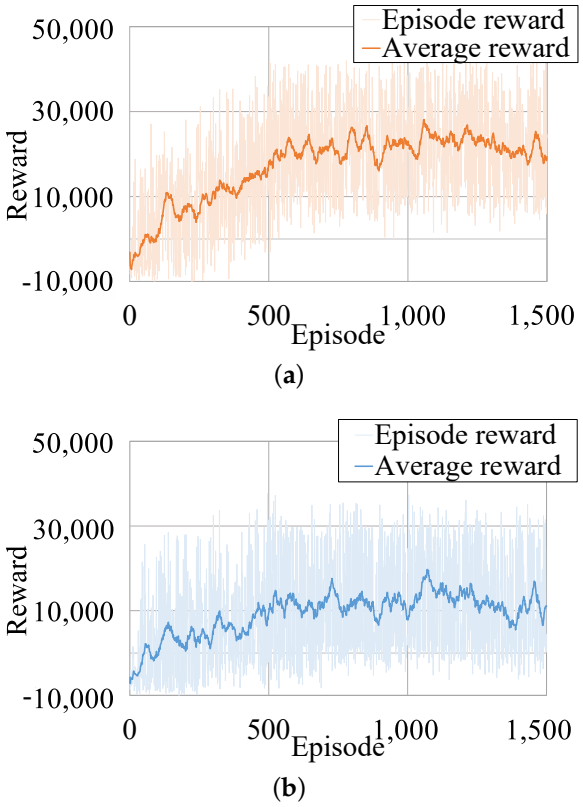


Figure 8. The training rewards of two RL schemes. (**a**) The training rewards generated by DDPG-ID. (**b**) The training rewards generated by DDPG.

The tracking results of the waved trajectory is shown in Figure 9. The RMSE value, MAX value, and mean value of the tracking errors for these three control methods are provided in Table 6. In terms of tracking accuracy, the trained DDPG-ID controller has a better performance compared to DDPG and PID, which has smaller state error and smoother tracking trajectory. The tracking error of the DDPG-ID algorithm ranges from -8×10^{-4} to 9×10^{-4} mm, which is almost about a half of the DDPG policy. In the interim, the DDPG controller has a lesser tracking error than PID. A huge oscillation has been induced by the PID controller, which will affect the hardware to a certain extent in the actual operation process. This huge oscillation input signal is much larger than a normal control input signal, which typically ranges from 0 to 11 V. Based on the characteristics of reinforcement learning, it is hard for a well-trained policy to generate such a shock signal.

Table 6. Tracking errors comparison of different controllers in the waved trajectory.

	RMSE	MAX	MEAN
DDPG-ID	3.658×10^{-4}	4.758×10^{-4}	1.003×10^{-4}
DDPG	1.093×10^{-3}	2.615×10^{-3}	4.414×10^{-4}
PID	1.654×10^{-3}	3.144×10^{-4}	3.104×10^{-4}

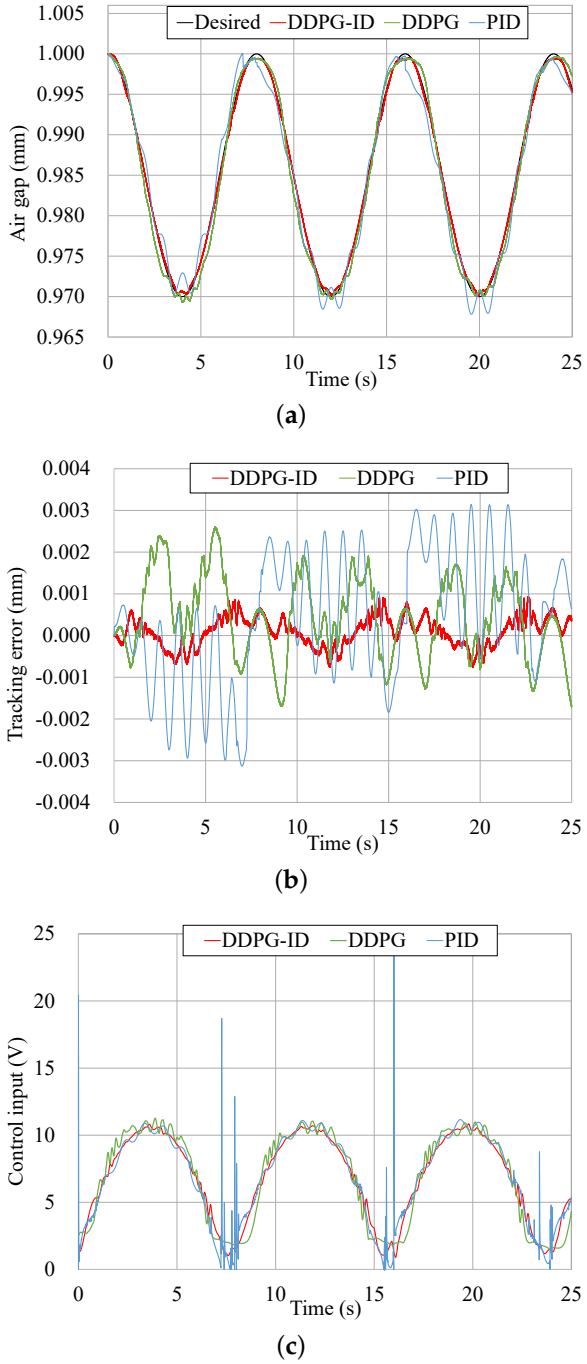


Figure 9. Tracking results comparison of the waved trajectory. (**a**) Tracking results comparison based on three control schemes. (**b**) Tracking error comparison based on three control schemes. (**c**) Control input comparison based on three control schemes.

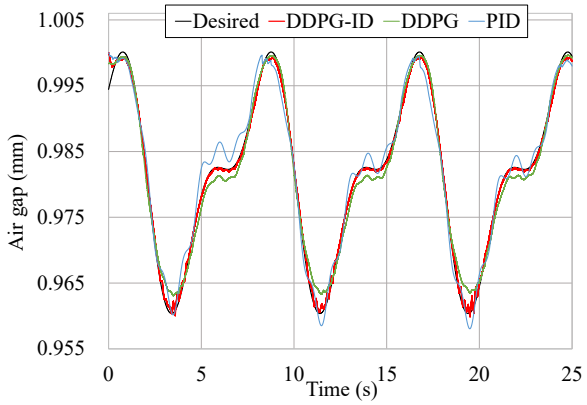
As can be seen in these figures, the tracking error of DDPG-ID in periodic trajectory is still less than the others, which ranges from -1.6×10^{-4} to 9×10^{-4} mm. Similar to the previous waved trajectory, the control input based on DDPG has shown better performance in terms of oscillations.

Another tracking results of a periodic trajectory is illustrated in Figure 10, and the tracking errors comparison of these three control methods are given in Table 7. The parametric equation of the periodic trajectory is defined as

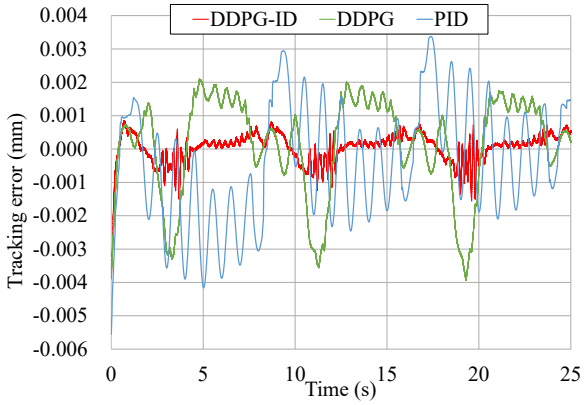
$$y_d(t) = 0.981 - 0.015\sin(\frac{\pi t}{4} - \frac{\pi}{2}) + 0.008\sin(\frac{\pi t}{2} - \frac{\pi}{16}). \quad (29)$$

Table 7. Tracking errors comparison of different controllers in the periodic trajectory.

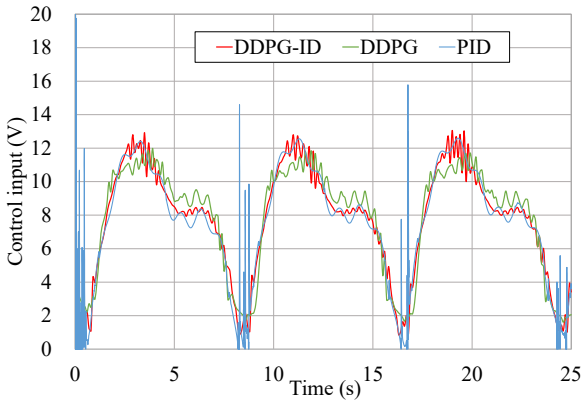
	RMSE	MAX	MEAN
DDPG-ID	4.272×10^{-4}	8.471×10^{-4}	5.404×10^{-5}
DDPG	1.545×10^{-3}	3.102×10^{-3}	1.610×10^{-4}
PID	1.923×10^{-3}	3.376×10^{-3}	3.311×10^{-4}



(**a**)



(**b**)



(**c**)

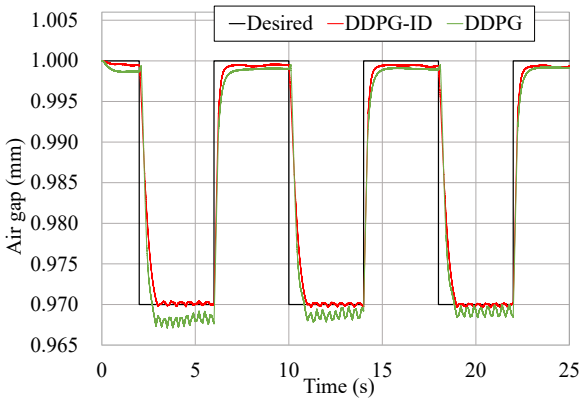
Figure 10. Tracking results comparison of the periodic trajectory. (**a**) Tracking results comparison based on three control schemes. (**b**) Tracking error comparison based on three control schemes. (**c**) Control input comparison based on three control schemes.

To further demonstrate the universality of the DDPG-ID policy, a periodic step trajectory is also utilized for comparison. The step signal with a period of 8 s is designed as the desired trajectory, which is shown in Figure 11a. The well-tuned PID controller is also tested in this step trajectory simulation. Since intense oscillations emerge, the results of PID show extremely worse performance are not shown in this paper.

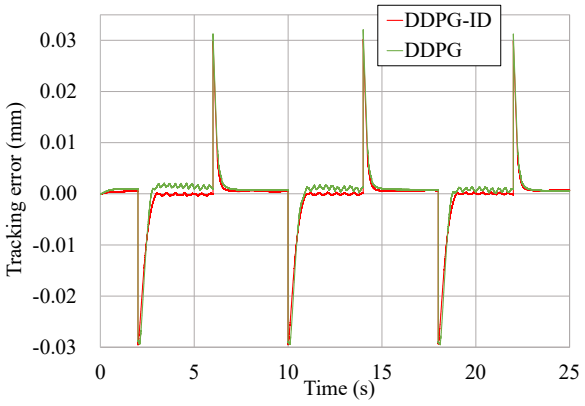
According to Figure 11, the tracking result of DDPG-ID algorithm remains stable with the tracking error bounded in -2×10^{-4} to 9×10^{-4} mm, which is still as a half of DDPG's performance. Due to the characteristic of the step signal, the state error will become tremendous during the step transition. Errors of DDPG-ID and DDPG are observed dropping quickly after step transition. It can be seen from Table 8 that the errors of DDPG-ID algorithm are substantially less than that of DDPG algorithm. As to the control inputs, the value of DDPG still fluctuates considerably when the state converges stable.

Table 8. Tracking errors comparison of different controllers in the step trajectory.

	RMSE	MAX	MEAN
DDPG-ID	4.612×10^{-3}	0.02953	6.938×10^{-4}
DDPG	5.279×10^{-3}	0.02986	1.437×10^{-3}

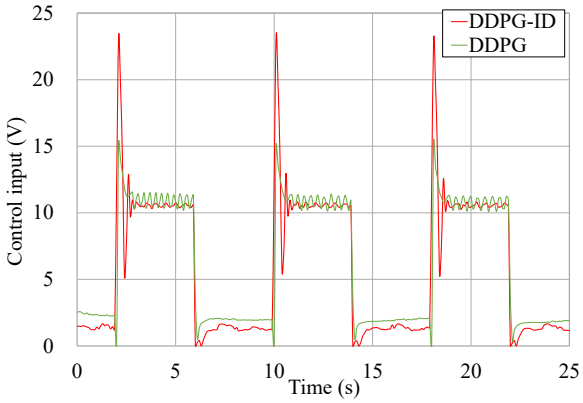


(**a**)



(**b**)

Figure 11. *Cont.*



(**c**)

Figure 11. Tracking results comparison of the step trajectory. (**a**) Tracking results comparison based on two control schemes. (**b**) Tracking error comparison based on two control schemes. (**c**) Control input comparison based on two control schemes.

According to above simulation results, it can be concluded that the control policy of DDPG-ID has triumphantly dealt with collective effect caused by disturbance and inaccurate estimation of deep reinforcement learning comparing to DDPG. The comparison results also have demonstrated the excellent control performance of the policy learned by DDPG-ID algorithm.

4.2. Experimental Results

The speed, acceleration, and direction of these designed trajectories vary with time, which makes the experiments results more trustworthy. In each test, the EMA in micropositioner is regulated for tracking the desired path of working air gap.

As shown in Figure 12, a laser displacement sensor is utilized to detect the motion states. Then DDPG-ID algorithm was administered through a SimLab board transplanted with Matlab-Simulink. The EMA controls the movement of the chain mechanism by executing the control signal, which is from the analog output port of SimLab board. The analog input port of SIMLAB board is connected with the signal output from the laser displacement sensor.

Figure 13 shows the tracking experiment results of the waved trajectory. It reaches the starting point on a straight track with a speed of 5.6 μm/s. At time 5 s, it begins to track the desired waved trajectory in three periods, and the waved trajectory can be described as $y_d(t) = 28 + 25\sin(\frac{\pi t}{10} + \frac{\pi}{2})$. The tracking error fluctuates within ±1.5 μm, which is demonstrated in Figure 13b. Except for several particular points of time, the tracking errors could range from ±1 μm.

Another periodic trajectory tracking experiment was also executed. As shown in Figure 14, the desired periodic trajectory starts at time 5 s, and it is defined as $y_d(t) = 35 - 25\sin(\frac{\pi t}{7.5} - \frac{2\pi}{3}) - 5\sin(\frac{\pi t}{15} + \frac{\pi}{6})$. The tracking error of the periodic trajectory still range from ±1.5 μm.

The experimental results show that the proposed DDPG-ID algorithm is able to closely track above two trajectories. Compared with the simulation results, the tracking error does not increase significantly, and it can be maintained between −1 μm and +1 μm.

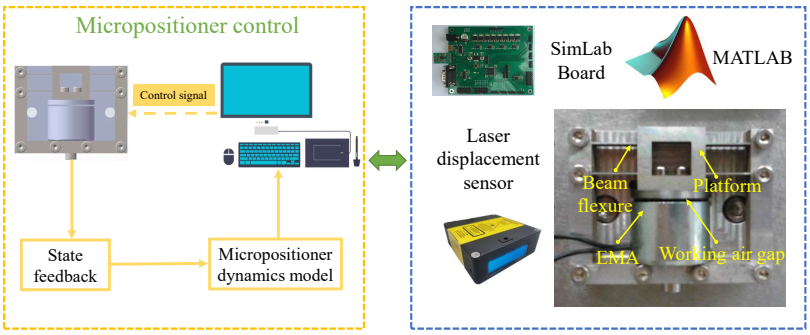


Figure 12. The schematic diagram of experiment system.

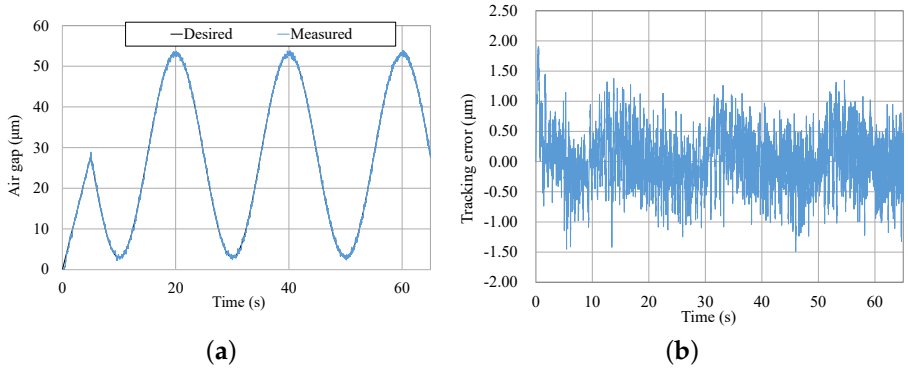


Figure 13. Tracking results of the waved trajectory. (**a**) Tracking result of desired trajectory. (**b**) Tracking error of desired trajectory.

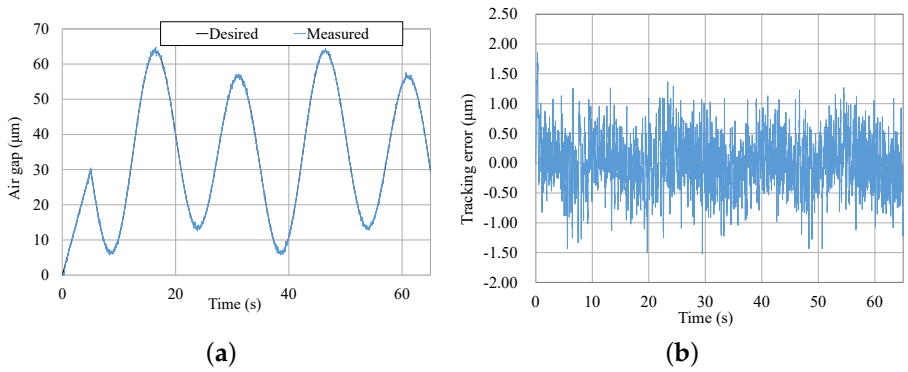


Figure 14. Tracking results comparison of the step trajectory. (**a**) Tracking result of desired trajectory. (**b**) Tracking error of desired trajectory.

5. Conclusions and Future Works

In this paper, a composite controller is developed based on an adaptive sliding mode disturbance observer and a deep reinforcement learning control scheme. A deep deterministic policy gradient is utilized to obtain the optimal control performance. To improve the tracking accuracy and transient response time, an integral differential compensator is applied during the learning process in the actor–critic framework. An adaptive sliding mode disturbance observer is developed to further retrench the influence of modeling uncertainty, external disturbances, and the effect of inaccurate value function. In comparison with the existing DDPG and the most commonly used PID controller, the trajectory tracking results has successfully indicated the satisfactory performances and the precision of the control policy based on the DDPG-ID algorithm in the simulation. The tracking errors are less than 1 μm, which shows the significant tracking efficiency of the proposed methods. The experimental results also indicate the high accuracy and strong anti-interference capabil-

ity of the proposed deep reinforcement learning control scheme. To further improve the tracking effect and realize micro-manipulation tasks in the future work, specific operation experiments will be performed such as cell manipulation, micro-assembly, etc.

Author Contributions: Writing—original draft preparation, S.L., R.X., X.X. and Z.Y.; writing—review and editing, S.L. and R.X.; data collection, S.L. and R.X.; visualization, S.L., R.X., X.X. and Z.Y.; supervision, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0018/2019/AKP and SKL-IOTSC(UM)-2021-2023), in part by the Ministry of Science and Technology of China (Grant No. 2019YFB1600700), in part by the Guangdong Science and Technology Department (Grant No. 2018B030324002 and 2020B1515130001), in part by the Zhuhai Science and Technology Innovation Bureau (Grant no. ZH22017002200001PWC), Jiangsu Science and Technology Department (Grant No. BZ2021061), and in part by the University of Macau (Grant No. MYRG2020-00253-FST).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

PID	Proportional–integral–derivative control
$RBFNN$	Radial basis neural network
RL	Reinforcement learning
$SARSA$	State-Action-Reward-State-Action
Q	The Value of Action in reinforcement learning
DRL	Deep reinforcement learning
DNN	Deep neural networks
DQN	Deep Q network
PG	Policy gradient
$DDPG$	Deep deterministic policy gradient
ID	Integral differential compensator
T_m	The magnetic force
y	The working air gap in micropositioner
I_c	The excitation current in micropositioner
EMA	The electron-magnetic actuator
V_i	The input voltage from the electron-magnetic actuator
R	The resistance of the coil in micropositioner
H	The coil inductance in micropositioner
u	The control input
D	The lumped system disturbance
$ASMDO$	Adaptive Sliding Mode Disturbance Observer
s_t	The state at time t in reinforcement learning
a_t	The action at time t in reinforcement learning
r_t	The reward at time t in reinforcement learning
$ReLU$	Rectified linear unit activation function
$tanh$	Hyperbolic tangent activation function

References

1. Català-Castro, F.; Martín-Badosa, E. Positioning Accuracy in Holographic Optical Traps. *Micromachines* **2021**, *12*, 559. [CrossRef] [PubMed]
2. Bettahar, H.; Clévy, C.; Courjal, N.; Lutz, P. Force-Position Photo-Robotic Approach for the High-Accurate Micro-Assembly of Photonic Devices. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6396–6402. [CrossRef]
3. Cox, L.M.; Martinez, A.M.; Blevins, A.K.; Sowan, N.; Ding, Y.; Bowman, C.N. Nanoimprint lithography: Emergent materials and methods of actuation. *Nano Today* **2020**, *31*, 100838. [CrossRef]
4. Dai, C.; Zhang, Z.; Lu, Y.; Shan, G.; Wang, X.; Zhao, Q.; Ru, C.; Sun, Y. Robotic manipulation of deformable cells for orientation control. *IEEE Trans. Robot.* **2019**, *36*, 271–283. [CrossRef]

5. Zhang, P.; Yang, Z. A robust adaboost. rt based ensemble extreme learning machine. *Math. Probl. Eng.* **2015**, *2015*, 260970. [CrossRef]
6. Yang, Z.; Wong, P.; Vong, C.; Zong, J.; Liang, J. Simultaneous-fault diagnosis of gas turbine generator systems using a pairwise-coupled probabilistic classifier. *Math. Probl. Eng.* **2013**, *2013*, 827128. [CrossRef]
7. Wang, D.; Zhou, L.; Yang, Z.; Cui, Y.; Wang, L.; Jiang, J.; Guo, L. A new testing method for the dielectric response of oil-immersed transformer. *IEEE Trans. Ind. Electron.* **2019**, *67*, 10833–10843. [CrossRef]
8. Roshandel, N.; Soleymanzadeh, D.; Ghafarirad, H.; Koupaei, A.S. A modified sensorless position estimation approach for piezoelectric bending actuators. *Mech. Syst. Signal Process.* **2021**, *149*, 107231. [CrossRef]
9. Ding, B.; Yang, Z.X.; Xiao, X.; Zhang, G. Design of reconfigurable planar micro-positioning stages based on function modules. *IEEE Access* **2019**, *7*, 15102–15112. [CrossRef]
10. García-Martínez, J.R.; Cruz-Miguel, E.E.; Carrillo-Serrano, R.V.; Mendoza-Mondragón, F.; Toledano-Ayala, M.; Rodríguez-Reséndiz, J. A PID-type fuzzy logic controller-based approach for motion control applications. *Sensors* **2020**, *20*, 5323. [CrossRef]
11. Salehi Kolahi, M.R.; Gharib, M.R.; Heydari, A. Design of a non-singular fast terminal sliding mode control for second-order nonlinear systems with compound disturbance. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 7343–7352. [CrossRef]
12. Nguyen, M.H.; Dao, H.V.; Ahn, K.K. Adaptive Robust Position Control of Electro-Hydraulic Servo Systems with Large Uncertainties and Disturbances. *Appl. Sci.* **2022**, *12*, 794. [CrossRef]
13. Cruz-Miguel, E.E.; García-Martínez, J.R.; Rodríguez-Reséndiz, J.; Carrillo-Serrano, R.V. A new methodology for a retrofitted self-tuned controller with open-source fpga. *Sensors* **2020**, *20*, 6155. [CrossRef]
14. Montalvo, V.; Estévez-Bén, A.A.; Rodríguez-Reséndiz, J.; Macias-Bobadilla, G.; Mendiola-Santíbañez, J.D.; Camarillo-Gómez, K.A. FPGA-Based Architecture for Sensing Power Consumption on Parabolic and Trapezoidal Motion Profiles. *Electronics* **2020**, *9*, 1301. [CrossRef]
15. García-Martínez, J.R.; Rodríguez-Reséndiz, J.; Cruz-Miguel, E.E. A new seven-segment profile algorithm for an open source architecture in a hybrid electronic platform. *Electronics* **2019**, *8*, 652. [CrossRef]
16. Fei, J.; Fang, Y.; Yuan, Z. Adaptive Fuzzy Sliding Mode Control for a Micro Gyroscope with Backstepping Controller. *Micromachines* **2020**, *11*, 968. [CrossRef] [PubMed]
17. Ruan, W.; Dong, Q.; Zhang, X.; Li, Z. Friction Compensation Control of Electromechanical Actuator Based on Neural Network Adaptive Sliding Mode. *Sensors* **2021**, *21*, 1508. [CrossRef] [PubMed]
18. Gharib, M.R.; Koochi, A.; Ghorbani, M. Path tracking control of electromechanical micro-positioner by considering control effort of the system. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 984–991. [CrossRef]
19. Han, M.; Tian, Y.; Zhang, L.; Wang, J.; Pan, W. Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee. *Automatica* **2021**, *129*, 109689. [CrossRef]
20. de Orio, R.L.; Ender, J.; Fiorentini, S.; Goes, W.; Selberherr, S.; Sverdlov, V. Optimization of a spin-orbit torque switching scheme based on micromagnetic simulations and reinforcement learning. *Micromachines* **2021**, *12*, 443. [CrossRef]
21. Adda, C.; Laurent, G.J.; Le Fort-Piat, N. Learning to control a real micropositioning system in the STM-Q framework. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 4569–4574.
22. Li, J.; Li, Z.; Chen, J. Reinforcement learning based precise positioning method for a millimeters-sized omnidirectional mobile microrobot. In Proceedings of the International Conference on Intelligent Robotics and Applications, Wuhan, China, 15–17 October 2008; pp. 943–952.
23. Shi, H.; Shi, L.; Sun, G.; Hwang, K.S. Adaptive Image-Based Visual Servoing for Hovering Control of Quad-Rotor. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *12*, 417–426. [CrossRef]
24. Zheng, N.; Ma, Q.; Jin, M.; Zhang, S.; Guan, N.; Yang, Q.; Dai, J. Abdominal-waving control of tethered bumblebees based on sarsa with transformed reward. *IEEE Trans. Cybern.* **2018**, *49*, 3064–3073. [CrossRef]
25. Tang, L.; Yang, Z.X.; Jia, K. Canonical correlation analysis regularization: An effective deep multiview learning baseline for RGB-D object recognition. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *11*, 107–118. [CrossRef]
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
27. Sutton, R.S.; McAllester, D.A.; Singh, S.P.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 27–30 November 2000; pp. 1057–1063.
28. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning (PMLR), Bejing, China, 22–24 June 2014; pp. 387–395.
29. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
30. Latifi, K.; Kopitca, A.; Zhou, Q. Model-free control for dynamic-field acoustic manipulation using reinforcement learning. *IEEE Access* **2020**, *8*, 20597–20606. [CrossRef]
31. Leinen, P.; Esders, M.; Schütt, K.T.; Wagner, C.; Müller, K.R.; Tautz, F.S. Autonomous robotic nanofabrication with reinforcement learning. *Sci. Adv.* **2020**, *6*, eabb6987. [CrossRef] [PubMed]

32. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
33. Zeng, Y.; Wang, G.; Xu, B. A basal ganglia network centric reinforcement learning model and its application in unmanned aerial vehicle. *IEEE Trans. Cogn. Dev. Syst.* **2017**, *10*, 290–303. [CrossRef]
34. Guo, X.; Yan, W.; Cui, R. Event-triggered reinforcement learning-based adaptive tracking control for completely unknown continuous-time nonlinear systems. *IEEE Trans. Cybern.* **2019**, *50*, 3231–3242. [CrossRef]
35. Zhang, J.; Shi, P.; Xia, Y.; Yang, H.; Wang, S. Composite disturbance rejection control for Markovian Jump systems with external disturbances. *Automatica* **2020**, *118*, 109019. [CrossRef]
36. Ahmed, S.; Wang, H.; Tian, Y. Adaptive high-order terminal sliding mode control based on time delay estimation for the robotic manipulators with backlash hysteresis. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1128–1137. [CrossRef]
37. Chen, M.; Xiong, S.; Wu, Q. Tracking flight control of quadrotor based on disturbance observer. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1414–1423. [CrossRef]
38. Zhao, Z.; He, X.; Ahn, C.K. Boundary disturbance observer-based control of a vibrating single-link flexible manipulator. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 2382–2390. [CrossRef]
39. Alibekov, E.; Kubalík, J.; Babuška, R. Policy derivation methods for critic-only reinforcement learning in continuous spaces. *Eng. Appl. Artif. Intell.* **2018**, *69*, 178–187. [CrossRef]
40. Hasselt, H. Double Q-learning. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, 2613–2621.
41. Zhang, S.; Sun, C.; Feng, Z.; Hu, G. Trajectory-Tracking Control of Robotic Systems via Deep Reinforcement Learning. In Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Bangkok, Thailand, 18–20 November 2019; pp. 386–391.
42. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2042–2062. [CrossRef] [PubMed]
43. Yang, X.; Zhang, H.; Wang, Z. Policy Gradient Reinforcement Learning for Parameterized Continuous-Time Optimal Control. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 59–64.
44. Xiao, X.; Xi, R.; Li, Y.; Tang, Y.; Ding, B.; Ren, H.; Meng, M.Q.H. Design and control of a novel electromagnetic actuated 3-DoFs micropositioner. *Microsyst. Technol.* **2021**, *27*, 1–10. [CrossRef]
45. Tommasino, P.; Caligiore, D.; Mirolli, M.; Baldassarre, G. A reinforcement learning architecture that transfers knowledge between skills when solving multiple tasks. *IEEE Trans. Cogn. Dev. Syst.* **2016**, *11*, 292–317.
46. Srikant, R.; Ying, L. Finite-time error bounds for linear stochastic approximation andtd learning. In Proceedings of the Conference on Learning Theory (PMLR), Phoenix AZ, USA, 25–28 June 2019; pp. 2803–2830.
47. Feng, Z.; Ming, M.; Ling, J.; Xiao, X.; Yang, Z.X.; Wan, F. Fractional delay filter based repetitive control for precision tracking: Design and application to a piezoelectric nanopositioning stage. *Mech. Syst. Signal Process.* **2022**, *164*, 108249. [CrossRef]