Article

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

Shi-Yun Liang [1,‡], Rui-Dong Xi [1,‡], Xiao Xiao [2], and Zhi-Xin Yang [1,*]

[1] State Key Laboratory of Internet of Things for Smart City and Department of Electromechanical Engineering, University of Macau, Macau SAR, China; mb95407@um.edu.mo (S.L.); yb57466@um.edu.mo (R.X.)
[2] Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China.;xiaox@sustech.edu.cn (X.X.) ; xiaox@sustech.edu.cn
[*] Correspondence: zxyang@um.edu.mo; Tel.: +00853-8822-4456
‡ These authors contributed equally to this work.

Abstract: The robust control of high precision electromechanical 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 p arameters. 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 improves 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 sterling performance is revealed with intensive tracking simulation experiments and demonstrates the improvement in the accuracy and response time of the controller.

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

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 [1–4], etc. One of the greatest challenge in this research field is the uncertainties produced by various factors like dynamic model, environmental temperature, sensors performance and the actuators' nonlinear characteristics [5] [6], 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 PID control method, backstepping controller [7], sliding mode control (SMC) approach [8] and neural network based controller [9]. In addition, many researchers have integrated these control strategies to further improve the control performance. Combined with backstepping strategy, Fei proposed an adaptive fuzzy sliding mode controller in [10]. Based on backstepping technique and neural networks, Chen developed an event-triggered adaptive control scheme with prescribed performance [11]. Liu combined a membrane structure genetic algorithm (MSGA) method with adaptive inverse neuro-control to identify the parameters of micropositioning system [12]. 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 most of control strategies have considered the factors of uncertainties and disturbances, the system is still problematic to achieve precise and comprehensive process.

As the rapid development in artificial intelligence in recent years have 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 assumption on dynamics model [13] [14], reinforcement learning can directly learn the policy by interacting with the system. Back in 2005, Adda presented a reinforcement learning algorithm for learning control of stochastic micromanipulation systems [15]. 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 [16]. However, the reinforcement learning algorithms such as Q-learning [17] and SARSA [18] 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) [19], the DRL algorithms, such as deep Q network (DQN) [20], policy gradient (PG) [21], deterministic policy gradient (DPG) [22] and deep deterministic policy gradient (DDPG) [23] with the ability to approximate the value function, have played an important role in continuous control tasks.

Latifi 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 [24]. Leinen introduced the concept of experience playback in DQN and approximate value function of neural network into SARSA algorithm for control of a scanning probe microscope [25]. Both simulation and real experimental results have shown that their proposed RL algorithm based on the neural network could achieve better performance than traditional control methods to some extent. However, due to the collective effects of disturbances generated from non-linear systems and deviations in value functions [23,26,27], the RL control method could induce significant inaccuracies in the tracking control tasks [28]. To improve the anti-disturbance capability and control accuracy, disturbance rejection control [29], time-delay estimation based control [30], disturbance observer based controllers [31][32] 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. To cope with an apparent state error occurred in trajectory tracking tasks of DRL [33–35], which is induced by the imprecise estimation of the action value function. The DDPG with integral differential compensator (DDPG-ID) is developed for decreasing the state error.

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 trajectory tracking task of micropositioner system. The contribution of this work are presented as follows:

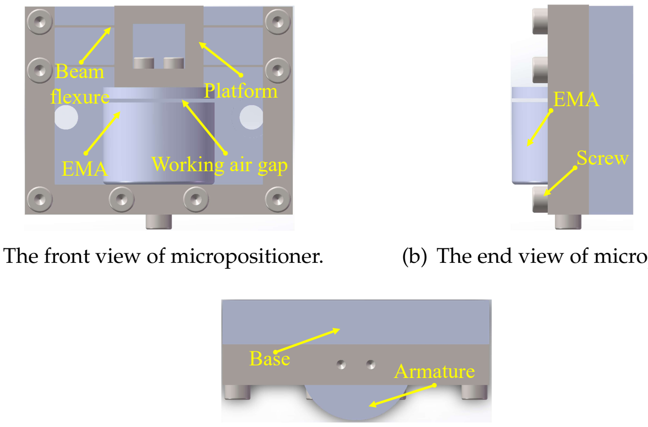
(1) A DDPG-ID algorithm based on deep reinforcement learning is introduced as a basic micropositioner system motion controller, which avoided the limitation of traditional control strategies to the accuracy and comprehensiveness of the dynamic model;

(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 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 letter is structured as follows. Sect. 2 presents the system description of the micropositioner. In Sect. 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 Sect. 4. To further evaluate the performance of the proposed control strategy in the micropositioner, tracking experiments are conducted in Sect. 5. Lastly, conclusion remarks are driven in Sect. 6.

2. System Description



(a) The front view of micropositioner.

(b) The end view of micropositioner.

(c) The vertical view of micropositioner.

Figure 1. The diagrammatic model of EMA actuated micropositioner.

The basic structure of micropositioner is shown in Fig. 1, which consists 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 Fig. 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:

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

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

doi:10.20944/preprints202203.0199.v1

According to the equations (1)-(4) ,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 the system equations (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 define $D = (\Delta f(x) + \Delta g(x)u) + d$, 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 [36]:

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. Approach

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

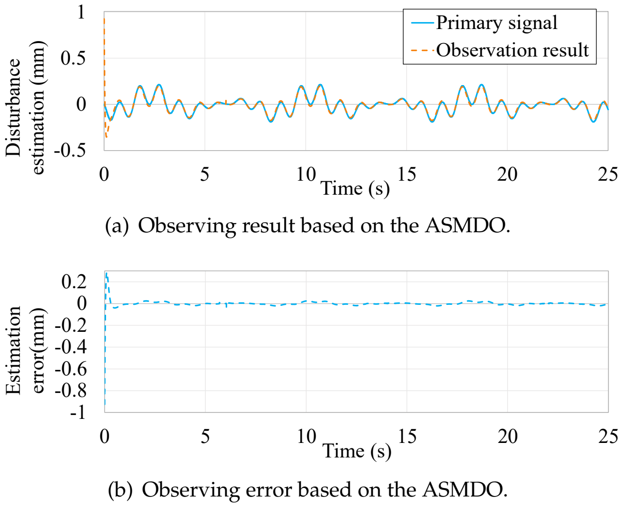
$$\begin{cases} \dot{\eta}_i = \eta_{i+1}, i = 1, 2 \\ \dot{\eta}_3 = f(z) + g(z)u + \hat{D} + \rho \end{cases} \tag{9}$$

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

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

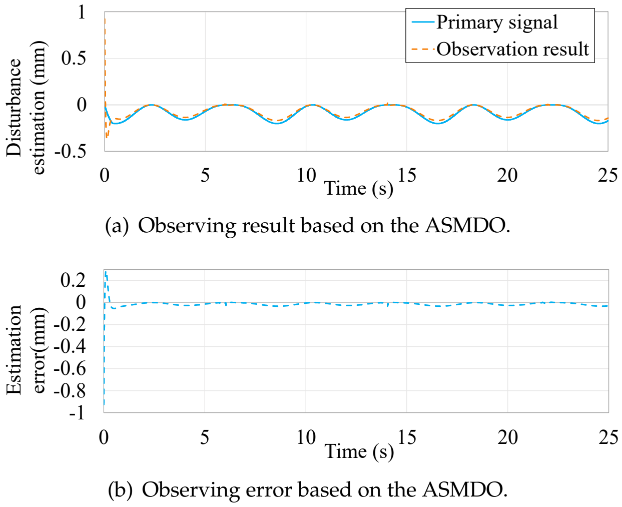
where λ_1, λ_2 are positive design parameters with $\lambda_2 \geq \beta_1$.



(a) Observing result based on the ASMDO.

(b) Observing error based on the ASMDO.

Figure 2. Observation result of ASMDO with d_1.



(a) Observing result based on the ASMDO.

(b) Observing error based on the ASMDO.

Figure 3. Observation result of ASMDO with d_2.

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 will not be repeated.

Two kinds of periodic external disturbances are added to verify the practicability of the proposed ASMDO with $d_1 = 0.1\sin(2\pi t) + 0.1\sin(0.5\pi t + \frac{\pi}{3})$, $d_2 = 0.1 + 0.1\sin(0.5\pi t + \frac{\pi}{3})$, based on the micropositoner model proposed in [36]. The effectiveness of the observer is presented as observation results in Fig. 2 and Fig. 3, obviously, the ASMDO can be used as interference compensation.

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 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 $s_t \in S$ take action $a_t \in A$ from policy $\pi(a_t|s_t)$, then 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:

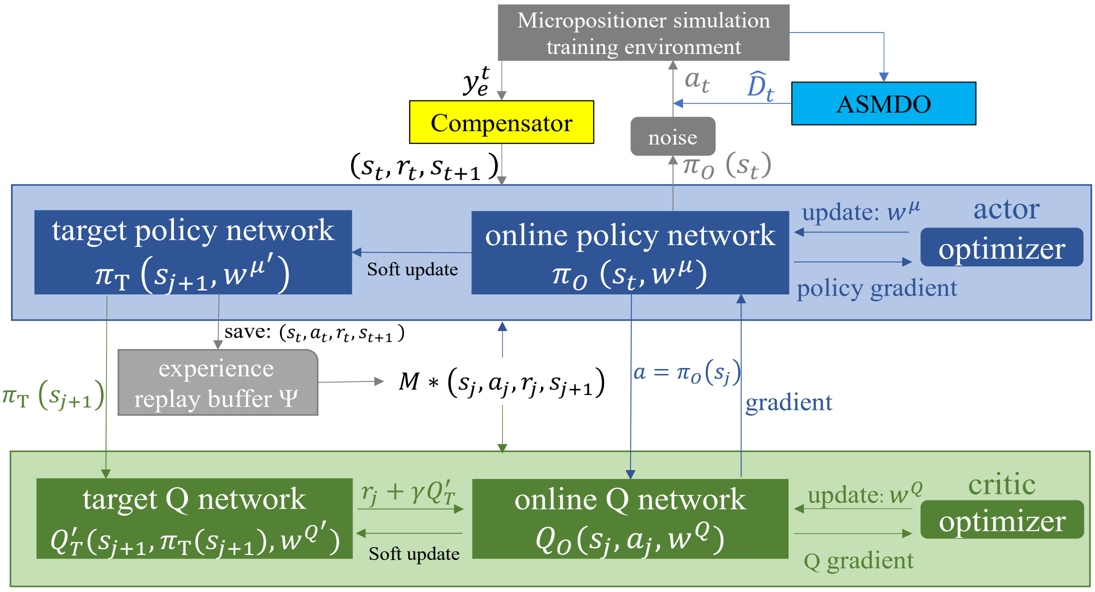


Figure 4. The Structure diagram of DDPG-ID algorithm.

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

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

In trajectory tracking control task of micropositioner, state s_t is sate 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 Fig. 4, DDPG is one of Actor-Critic algorithms which has actor and critic. The actor is responsible for generating actions and interacting with the environment, and critic evaluates the performance of the actor and guide the action in the next state.

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 $J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^{(t-1)} r_t\right]$ 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

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

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 [37], 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 Fig. 4. 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 sate 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 Fig. 5, the adaptive sliding mode disturbance observer (ASMDO) is embedded in DDPG-ID between 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.

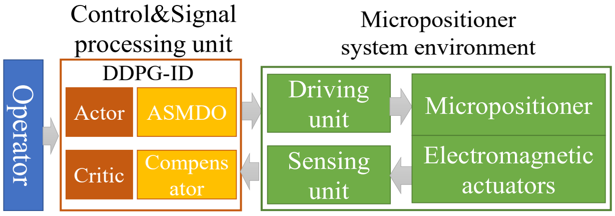

Control&Signal processing unit
Micropositioner system environment
DDPG-ID
Operator
Actor
ASMDO
Critic
Compensator
Driving unit
Sensing unit
Micropositioner
Electromagnetic actuators

Figure 5. 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 shown 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 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 rewritten 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 $\mathcal{L}(w^Q)$ 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 Fig. 6, 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 is consisted with 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. The parameters of each layer in Q network are shown in Table 1.

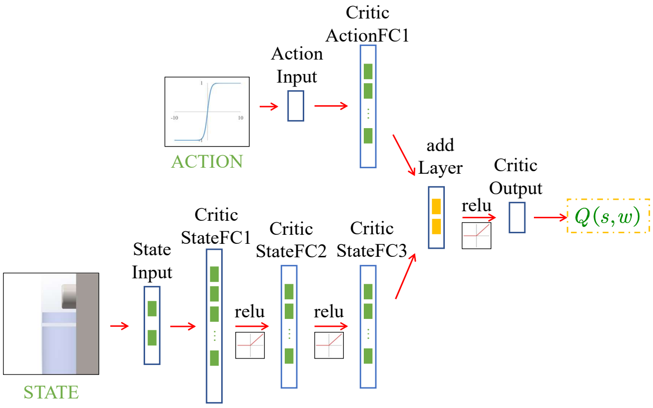


Figure 6. The diagram of Q network.

Table 1. 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

Table 2. Policy network parameters.

Network Layer Name	Number of Nodes
StateLayer	5
ActorFC1	30
ActorOutput	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 the derivatives of 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 following:

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

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

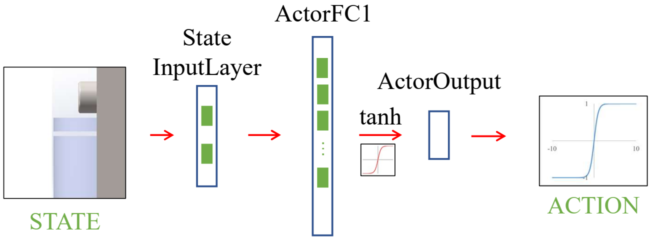


Figure 7. The diagram of policy network.

Fig. 7 shows the diagram of 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 policy network are shown in Table 2.

The DDPG-ID algorithm 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 $w^\mu$
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 $\Psi$
Load the simplified micropositioner dynamic model
for episode=1, MaxEpisode do
    Initialize a noise process $\mathcal{N}$ for exploration
    Initialize adaptive sliding mode disturbance observer
    Initialize integral differential 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 $\Psi$
        Randomly sample a minibatch of $M$ transitions $(s_j, a_j, r_j, s_{j+1})$ from $\Psi$
        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 Results

In this section, three distinct desired trajectories are designed for thoroughly evaluating the performances of proposed deep reinforcement learning control strategy in positioning and tracking simulation experiments. An traditional DDPG algorithm and a well-tuned PID strategy are taken in experiments for comparison. The dynamics model of micropositioner is given in Section II, and its basic system model parameters are from our previous research [36], which is shown in Table 3.

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 4.

Table 3. Parameters of the micropositioner model.

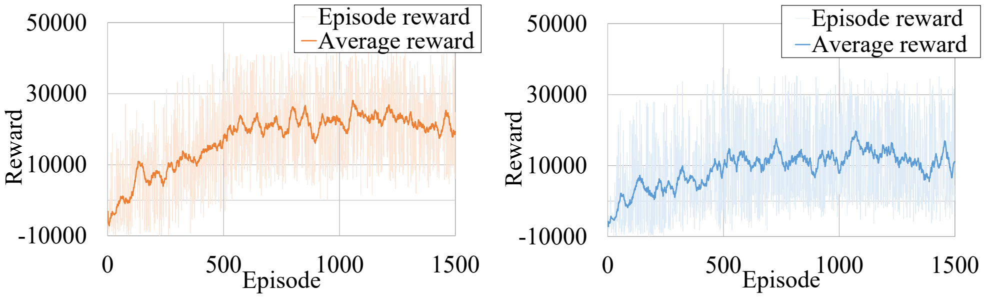
Notation	Value	Unit
L_1	13.21	H
L_0	0.67	H
a	1.11×10^{-5}	m
R	43.66	Ω
c	8.83×10^{-5}	$\mathrm{Nm}^2\ \mathrm{A}^{-2}$
k	$1.803 \times 10^{\mathrm{N}5}$	Nm^{-1}
m	0.0272	Kg

Table 4. 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	100000
Minibatch size M	64
Max episode ϖ	1500
Soft update factor τ	0.05
Max exploration steps T	250 (25s)
Time step T_s	0.01s
Intergal gain α	0.01
Differential gain β	0.001

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:

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



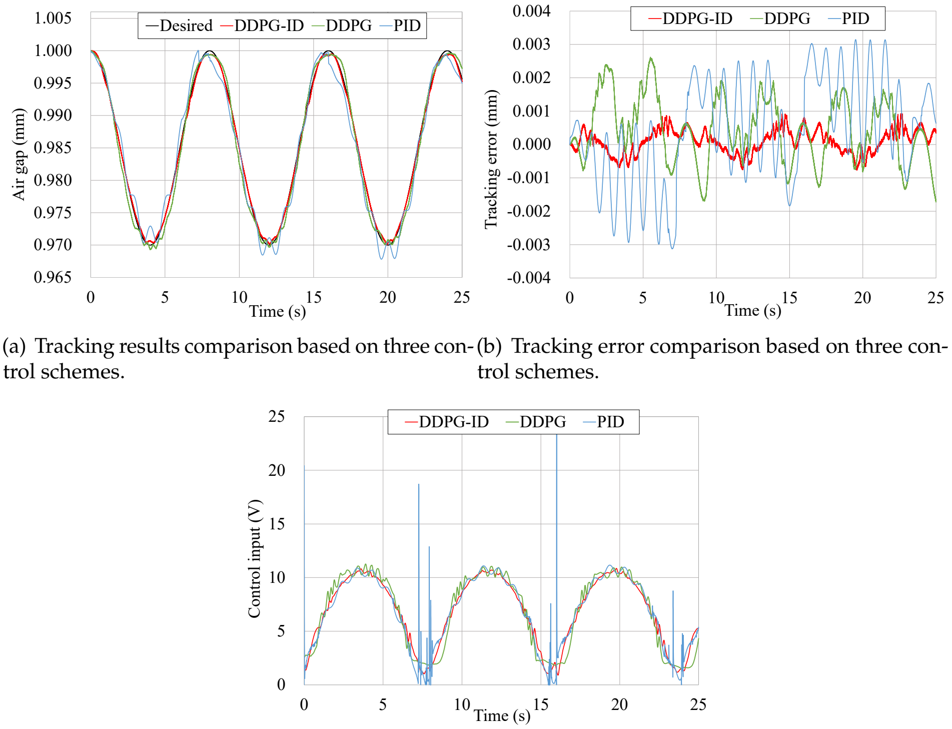
(a) The training rewards generated by DDPG-ID.

(b) The training rewards generated by DDPG.

Figure 8. The training rewards of two RL schemes.

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 Fig. 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 Fig. 8 (a) with Fig. 8 (b), 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.

The tracking results of the waved trajectory is shown in Fig. 9. In terms of tracking accuracy, the trained DDPG-ID controller has a better performance comparing with 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 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. The unnormal huge oscillation input signal always much larger than normal control input signal which is range from 0 to $11V$. Based on the characteristics of reinforcement learning, it is hard for a well-trained policy to generate such a shock signal.

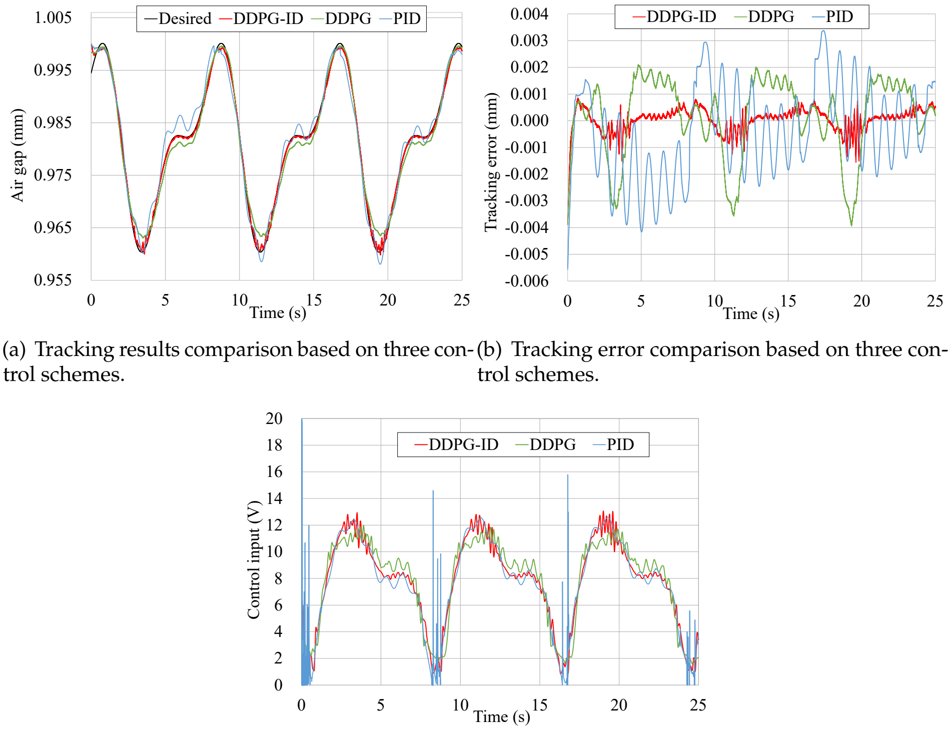


(a) Tracking results comparison based on three control schemes.

(b) Tracking error comparison based on three control schemes.

(c) Control input comparison based on three control schemes.

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.

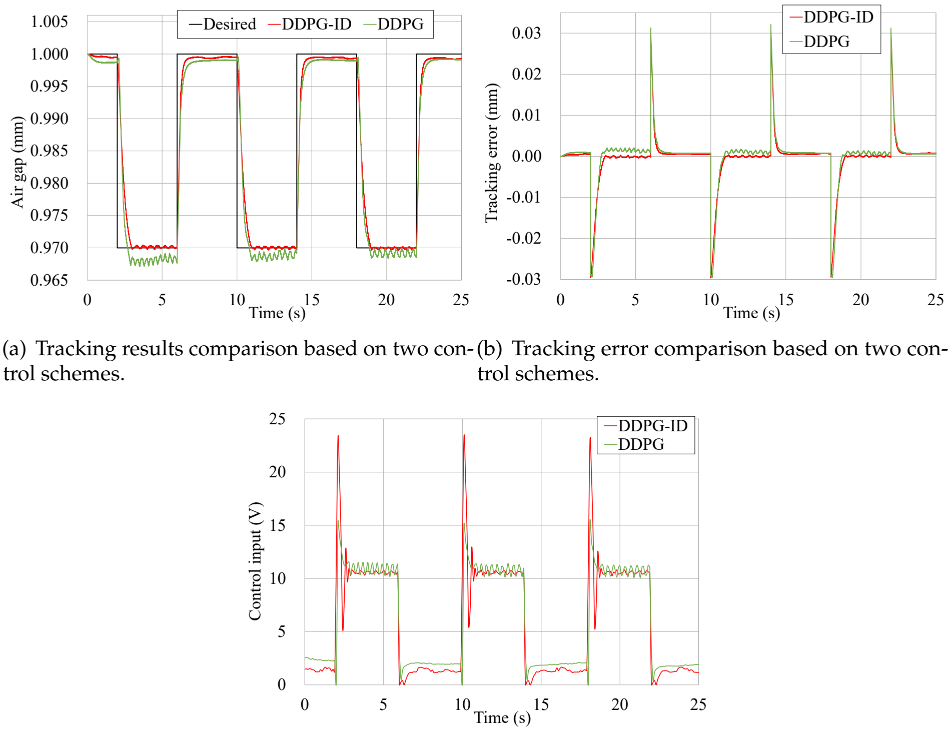
(c) Control input comparison based on three control schemes.

Figure 10. Tracking results comparison of the periodic trajectory.

Another tracking results of a periodic trajectory is illustrated in Fig. 10. The parametric equation of the periodic trajectory is defined as

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

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.



(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.

Figure 11. Tracking results comparison of the step trajectory.

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 8s is designed as the desired trajectory which is shown in Fig. 11 (a). The well-tuned PID contorller 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 Fig. 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. As to the control inputs, the value of DDPG still fluctuates considerably when the state converges stable.

Based on the simulation results, 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 compare results also have demonstrate excellent control performance of the policy learned by DDPG-ID algorithm.

5. Experimental Results

To further verify the performances of the proposed DDPG-ID algorithm, trajectory tracking experiments are carried out. Two desired trajectories are designed and employed. 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 Fig. 12, it used a laser displacement sensor to detect the motion states. Then DDPG-ID algorithm was administered through a SimLab board transplanted with

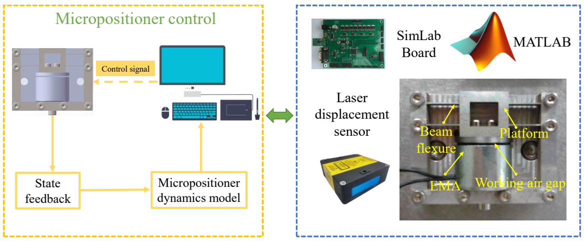
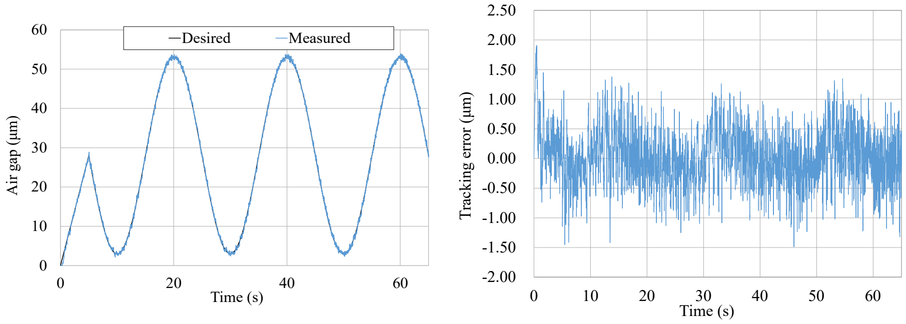


Figure 12. The schematic diagram of experiment system.

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.

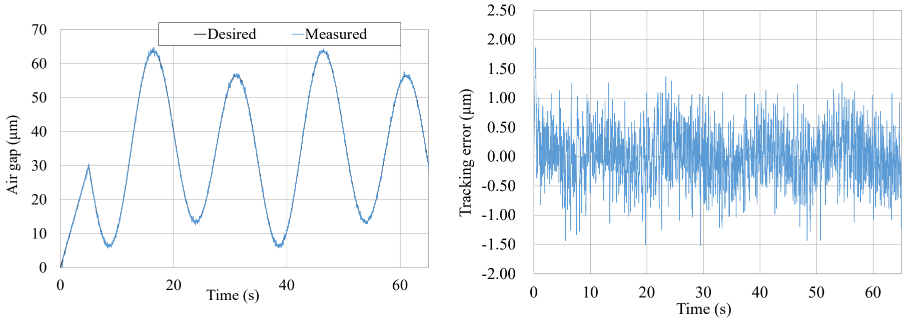


(a) Tracking result of desired trajectory.

(b) Tracking error of desired trajectory.

Figure 13. Tracking results of the waved trajectory.

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



(a) Tracking result of desired trajectory.

(b) Tracking error of desired trajectory.

Figure 14. Tracking results comparison of the step trajectory.

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

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 $-1\mu m$ and $+1\mu m$.

6. Conclusion

In this article, 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 retrenching 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 experimental results also indicate high-accuracy and strong anti-interference capability of the proposed deep reinforcement learning control scheme.

Acknowledgments: 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).

References

1. Català-Castro, F.; Martín-Badosa, E. Positioning Accuracy in Holographic Optical Traps. *Micromachines* **2021**, *12*, 559.
2. Bettahar, H.; Clévy, C.; Courjal, N.; Lutz, P. Force-Position Photo-Robotic Approach for the High-Accurate Micro-Assembly of Photonic Devices. *IEEE Robotics and Automation Letters* **2020**, *5*, 6396–6402.
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.
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 Transactions on Robotics* **2019**, *36*, 271–283.
5. Roshandel, N.; Soleymanzadeh, D.; Ghafarirad, H.; Koupaei, A.S. A modified sensorless position estimation approach for piezoelectric bending actuators. *Mechanical Systems and Signal Processing* **2021**, *149*, 107231.
6. 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.
7. Fang, Y.; Fu, W.; An, C.; Yuan, Z.; Fei, J. Modelling, simulation and dynamic sliding mode control of a mems gyroscope. *Micromachines* **2021**, *12*, 190.
8. Xie, M.; Yu, S.; Lin, H.; Ma, J.; Wu, H. Improved sliding mode control with time delay estimation for motion tracking of cell puncture mechanism. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2020**, *67*, 3199–3210.
9. Sheng, G.; Gao, G.; Zhang, B. Application of improved wavelet thresholding method and an RBF network in the error compensating of an MEMS gyroscope. *Micromachines* **2019**, *10*, 608.
10. Fei, J.; Fang, Y.; Yuan, Z. Adaptive Fuzzy Sliding Mode Control for a Micro Gyroscope with Backstepping Controller. *Micromachines* **2020**, *11*, 968.
11. Chen, X.; Liu, Y.; Zhang, L.; Gao, J.; Yang, B.; Chen, X. Event-Triggered Adaptive Control Design With Prescribed Performance for Macro-Micro Composite Positioning Stage. *IEEE Transactions on Industrial Electronics* **2020**.
12. Liu, D.; Fang, Y.; Wang, H.; Dong, X. Adaptive novel MSGA-RBF neurocontrol for piezo-ceramic actuator suffering rate-dependent hysteresis. *Sensors and Actuators A: Physical* **2019**, *297*, 111553.
13. 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.
14. Huang, X.; Wu, W.; Qiao, H. Computational modeling of emotion-motivated decisions for continuous control of mobile robots. *IEEE Transactions on Cognitive and Developmental Systems* **2020**, *13*, 31–44.
15. Adda, C.; Laurent, G.J.; Le Fort-Piat, N. Learning to control a real micropositioning system in the STM-Q framework. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005, pp. 4569–4574.
16. Li, J.; Li, Z.; Chen, J. Reinforcement learning based precise positioning method for a millimeters-sized omnidirectional mobile microrobot. International Conference on Intelligent Robotics and Applications. Springer, 2008, pp. 943–952.
17. Shi, H.; Shi, L.; Sun, G.; Hwang, K.S. Adaptive Image-Based Visual Servoing for Hovering Control of Quad-Rotor. *IEEE Transactions on Cognitive and Developmental Systems* **2019**, *12*, 417–426.
18. 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 transactions on cybernetics* **2018**, *49*, 3064–3073.

19. Tang, L.; Yang, Z.X.; Jia, K. Canonical correlation analysis regularization: an effective deep multiview learning baseline for RGB-D object recognition. *IEEE Transactions on Cognitive and Developmental Systems* **2018**, *11*, 107–118.
20. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* **2013**.
21. Sutton, R.S.; McAllester, D.A.; Singh, S.P.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 2000, pp. 1057–1063.
22. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. International conference on machine learning. PMLR, 2014, pp. 387–395.
23. 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 preprint arXiv:1509.02971* **2015**.
24. Latifi, K.; Kopitca, A.; Zhou, Q. Model-free control for dynamic-field acoustic manipulation using reinforcement learning. *IEEE Access* **2020**, *8*, 20597–20606.
25. Leinen, P.; Esders, M.; Schütt, K.T.; Wagner, C.; Müller, K.R.; Tautz, F.S. Autonomous robotic nanofabrication with reinforcement learning. *Science advances* **2020**, *6*, eabb6987.
26. 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.
27. Zeng, Y.; Wang, G.; Xu, B. A basal ganglia network centric reinforcement learning model and its application in unmanned aerial vehicle. *IEEE Transactions on cognitive and developmental systems* **2017**, *10*, 290–303.
28. Guo, X.; Yan, W.; Cui, R. Event-triggered reinforcement learning-based adaptive tracking control for completely unknown continuous-time nonlinear systems. *IEEE transactions on cybernetics* **2019**, *50*, 3231–3242.
29. 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.
30. 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 Transactions on Systems, Man, and Cybernetics: Systems* **2019**.
31. Chen, M.; Xiong, S.; Wu, Q. Tracking flight control of quadrotor based on disturbance observer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2019**.
32. Zhao, Z.; He, X.; Ahn, C.K. Boundary disturbance observer-based control of a vibrating single-link flexible manipulator. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2019**.
33. Alibekov, E.; Kubalík, J.; Babuška, R. Policy derivation methods for critic-only reinforcement learning in continuous spaces. *Engineering Applications of Artificial Intelligence* **2018**, *69*, 178–187.
34. Hasselt, H. Double Q-learning. *Advances in neural information processing systems* **2010**, *23*, 2613–2621.
35. Zhang, S.; Sun, C.; Feng, Z.; Hu, G. Trajectory-Tracking Control of Robotic Systems via Deep Reinforcement Learning. 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE, 2019, pp. 386–391.
36. 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. *Microsystem Technologies* **2021**, pp. 1–10.
37. Tommasino, P.; Caligiore, D.; Mirolli, M.; Baldassarre, G. A reinforcement learning architecture that transfers knowledge between skills when solving multiple tasks. *IEEE Transactions on Cognitive and Developmental Systems* **2016**, *11*, 292–317.