IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 9, 2020, accepted December 15, 2020, date of publication January 4, 2021, date of current version January 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3049072

Adaptive Controller of PEMFC Output Voltage Based on Ambient Intelligence Large-Scale Deep Reinforcement Learning

JIAWEN LI[1], TAO YU[1], AND BO YANG[2]
[1]College of Electric Power, South China University of Technology, Guangzhou 510640, China
[2]Faculty of Electric Power Engineering, Kunming University of Science and Technology, Kunming 650093, China

Corresponding author: Tao Yu (taoyul@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51777078.

ABSTRACT In this article, an adaptive Proportion integration (PI) controller for varying the output voltage of a proton exchange membrane fuel cell (PEMFC) is proposed. The PI controller operates on the basis of ambient intelligence large-scale deep reinforcement learning. It functions as a coefficient tuner based on an ambient intelligence exploration multi-delay deep deterministic policy gradient (AIEM-DDPG) algorithm. This algorithm is an improvement on the original deep deterministic police gradient (DDPG) algorithm, which incorporates ambient intelligence exploration. The DDPG algorithm serves as the core, and the AIEM-DDPG algorithm runs on a variety of deep reinforcement learning algorithms, including soft actor-critic (SAC), deep deterministic policy gradient (DDPG), proximal policy optimization (PPO) and double deep Q-network (DDQN) algorithms, to attain distributed exploration in the environment. In addition, a classified priority experience replay mechanism is introduced to improve the exploration efficiency. Clipping multi-Q learning, policy delayed updating, target policy smooth regularization and other methods are utilized to solve the problem of Q-value overestimation. A model-free algorithm with good global searching ability and optimization speed is demonstrated. Simulation results show that the AIEM-DDPG adaptive PI controller attains better robustness and adaptability, as well as a good control effect.

INDEX TERMS Distributed deep reinforcement learning, ambient intelligence exploration multi-delay deep deterministic policy gradient, proton exchange membrane fuel cell, air mass flow control, intelligent controller.

I. INTRODUCTION

In recognition of the serious environmental pollution caused by conventional fuels, many countries have invested into R&D on new energy fuels [1], [2]. The proton exchange membrane fuel cell (PEMFC) is a device that converts the chemical energy generated by hydrogen and oxygen into electric energy, with no harmful substances discharged. The PEMFC is characterized by higher generating efficiency and specific energy than those of ordinary lithium batteries, making it an ideal power generation candidate. Due to its characteristics of cleanliness, efficiency and low working temperature, PEMFCs could be used in a wide range of applications in the foreseeable future [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Huai-Zhi Wang.

According to the electrochemical reaction principle of the PEMFC, the output voltage will change in accordance with the change of load. As a PEMFC mainly uses hydrogen and oxygen as the reactive gases, and generates electric energy via an electrochemical reaction, its output voltage is affected by hydrogen flow, and thus a reasonable hydrogen flow control is required for stabilizing the output voltage.

The PEMFC is a nonlinear system which has multiple inputs and outputs. In recent years, scholars have conducted a number of studies into PEMFC flow control.

In [5], [6], the authors proposed and modeled an air supply system for a PEMFC and proposed an optimal feedback control method for air flow control. In [7], the authors elaborated on the method in [5], [6] by developing a robust servo control method. In [8], the authors proposed a coordinated control method for attenuating cathode flow and pressure in

a small PEMFC. In [9], the authors designed a filter based on the control method in [5], [6], and demonstrated the effectiveness of its control system [9]. However, their feedback control method fails to take into account certain characteristics of hybrid fuel cell vehicles. Some control methods based on model predictive control are presented in [10]–[16], including model predictive control (MPC) [10], [11], internal model control (IMC) [12], model-based adaptive control [13], nonlinear model predictive control (NMPC) [14], nonlinear multivariable control [15], time delay control [16], and generalized model control [17].

In addition, many linear controllers gas control within the PEMFC have been proposed. A linear quadratic Gaussian (LQG) control method proposed in [18]. In [19], the authors have developed a multivariable linear quadratic regulator control (LQR) coupled with feed-forward control. In [20], the authors have proposed a linear parameter varying (LPV) control method. The above linear controls are able to control a PEMFC to a certain extent but cannot address the nonlinearity of that PEMFC.

A large number of sliding-mode controls for PEMFC flow control have been developed. In [20], the authors have demonstrated a one-order sliding mode control, while a high-order ultra-twisted sliding mode control is proposed in [22]. In [23], the authors present a high order sliding mode control combing observer. However, due to the high-frequency discontinuous switching of a sliding-mode controller, the actual controlling effect of such a system is inconsistent over time. A sliding mode controller produces adverse ''buffeting'' to the controlled object, which makes this method unsuitable for precise control.

An adaptive control algorithm can automatically adjust the control policy in accordance with the real-time state of the system; thus, such algorithms are widely used in the field of PEMFC control. Several researchers have proposed a variety of adaptive control algorithms: In [24], the authors proposed a data-driven adaptive control method; In [25], the authors have developed an adaptive control method based on parameter identification and pole assignment; an adaptive extremum search control method have proposed in [26]; In [27], the authors have produced a PID-neural network control method; In [28], the authors have proposed an interval type-2 fuzzy-PID control method; and, In [29], the authors have proposed a fuzzy adaptive PID control method.

PID and its associated algorithms have been widely used in fuel cell control and other practical engineering applications due to their simple control policy and good robustness. A number of PID control methods have been proposed, including neural PID control [30], a fuzzy PID control [31], an algorithm combining PID and fuzzy control [32], a feedback linearization control policy (for transforming a nonlinear control model into a linear model) [33], and a fraction-order PID (FOPID) control based on nonlinear observer with unknown input [34].

These proposed PIDs and associated algorithms can achieve better PEMFC control, but they cannot be adapted easily to address nonlinearity in the PEMFC [35]. An adaptive algorithm has superior robustness and adaptability, but often has an excessively complex control policy. In order to overcome such defects and obtain better control performance, it is necessary to choose an adaptive algorithm with simple control policy, better identification and decision-making ability.

Deep deterministic policy gradient (DDPG) in deep reinforcement learning is a model-free method [36]–[38], which combines the perception of deep learning with the decision-making ability of reinforcement learning, and which has excellent self-adaptive ability, thus achieving timely response and accurate control. In contrast with conventional control methods, DDPG develops control policies through full interaction with the environment [39], [40] without identifying the model; this function makes DDPG compatible with a nonlinear control environment. Nevertheless, although DDPG is applied in various control fields [41]–[43], the algorithm is affected by the common problems associated with many deep reinforcement learning algorithms: it requires long time off-line training before practical application, and it cannot be generalized to every environment when training is insufficient; these lead to poor robustness whenever this algorithm is employed for decision-making. Therefore, this algorithm is seldom directly employed as a control algorithm in the control of the PEMFC, which in turn requires precise control. As mentioned above, an improved DDPG algorithm, and an output voltage adaptive PI controller (tuner) based on the AIEM-DDPG algorithm, are proposed in this article. This controller capitalizes on the excellent sensing ability and decision-making ability of AIEM-DDPG algorithm. It can actively regulate the coefficients of PID control according to the system state, so that it can regulate the hydrogen flow of anode in real time in order to control the output voltage.

This article makes two unique contributions to the field:

1. An adaptive PI controller (of output voltage) based on deep reinforcement learning is proposed. The controller employs the deep reinforcement learning algorithm as the tuner, which regulates the output voltage of the PI controller by adjusting the coefficient of the PI controller in real time. It avoids the poor robustness caused by direct use of the deep reinforcement learning algorithm as the control algorithm of the controller, thus ensuring that the PEMFC can satisfy the real-time control requirements under different working conditions and improving the output characteristics of the fuel cell.

2. For the above framework, an AIEM-DDPG algorithm is proposed. This is a large-scale deep reinforcement learning algorithm based on DDPG, which has better global searching ability and optimization speed. The AIEM-DDPG algorithm adopts an ambient intelligence exploration policy, in that the algorithm enables multiple explorations running on the DDPG algorithm as well as other deep reinforcement learning algorithms with different principles such as SAC, DDPG, PPO and DDQN, in order to perform distributed exploration in the environment. In addition, the classified priority experience replay mechanism is introduced in order to improve the

exploration efficiency. Clipping multi-Q learning, deferred policy updating, target policy smooth regularization and other methods are utilized to address overestimation of the Q-value. Simulation results show that the AIEM-DDPG controller achieve better control performance and robustness than controllers based on other control principles

II. MODEL OF THE PEMFC

A. PEMFC DYNAMIC MODEL

The voltage of a PEMFC is the sum of thermodynamic electromotive force, polarization overvoltage and Ohmic overpotential. When liquid water is produced, the ideal standard potential of a PEMFC is 1.229 V. During the electricity generation of an actual fuel cell, there is some irreversible voltage loss, called polarization overvoltage, which includes activation polarization overvoltage η_{ac}, Ohmic polarization overvoltage η_{ohm} and concentration polarization overvoltage η_{con}. Such voltage loss will result in a smaller cell voltage than the ideal standard potential. The effect of activation polarization is severe at low current density, but the effect of concentration polarization is dominant at high current density. The output voltage V_{cell} of a single cell can be basically expressed as follows [44]:

$$V_{cell} = E - \eta_{act} - \eta_{ohm} - \eta_{con} \tag{1}$$

For a fuel cell stack composed of N single cells in series connection, the output voltage V can be calculated as:

$$V = NV_{cell} \tag{2}$$

1) THERMODYNAMIC ELECTROMOTIVE FORCE

According to the Nernst hydrogen/oxygen fuel cell equation, the thermodynamic electromotive force can be determined as follows:

$$E = \frac{\Delta G}{2F} + \frac{\Delta S}{2F}(T - T_{\text{ref}}) + \frac{RT}{2F}\left(\ln p_{H_2} + \frac{1}{2}\ln p_{O_2}\right) \tag{3}$$

where ΔG is Gibbs free energy, ΔS is standard molar entropy, R is gas constant, F is Faraday constant, T is the operation temperature of the stack, T_{ref} is the reference temperature, P_{H_2} and P_{O_2} are the differential pressures of hydrogen and oxygen respectively. After specific data are substituted, the equation can be converted as follows:

$$E = 1.229 - 0.85 \times 10^{-3}(T - 298.15) + 4.3085 \times 10^{-5} T \left(\ln p_{\text{H}_2} + \ln p_{O_2}/2\right) \tag{4}$$

2) ACTIVATION OVERVOLTAGE

When electrochemical reaction occurs on the electrode surface of PEMFC, the electrons pass through the external circuit load, and the protons pass through the proton exchange membrane. During the transfer process, the electrons and protons have to overcome the chemical energy of the reaction. Specifically, when a certain current pass through a cell, the electrode potential deviates from the reversible potential which results in the activation polarization overvoltage. The activation overvoltage of PEMFC consists of anodic overvoltage and cathodic overvoltage. The η_{act} is shown as follows

$$\eta_{act} = x_1 + x_2 T + x_3 T \ln c\,(O_2) + x_4 T \ln I \tag{5}$$

where I is the load current of PEMFC, ξ_i is the model coefficient fitted by experimental data on basis of hydrodynamic force, thermal power and electrochemistry; $c(O_2)$ is the dissolved oxygen concentration on the cathode catalyst interface. It is a function of temperature and oxygen differential pressure. According to Henry's law it is shown as follows:

$$c\,(O_2) = P_{O_2}/5.08 \times 10^6 \exp(-498/T) \tag{6}$$

3) OHMIC POLARIZATION OVERVOLTAGE

η_{ohm} mainly consists of the voltage due to the equivalent membrane impedance R_{m} of proton membranes and the voltage drop due to the resistance R_{c} against the passage of protons through proton membranes [45]:

$$\eta_{\text{ohm}} = IR_{\text{int}} = I\,(R_{\text{m}} + R_{\text{c}}) \tag{7}$$

The internal resistance of a cell can be empirically expressed as follows:

$$R_{int} = 0.01605 - 3.5 \times 10^{-5} T + 8 \times 10^{-5} i \tag{8}$$

4) CONCENTRATION POLARIZATION OVERVOLTAGE

Concentration overvoltage is caused by mass transfer, which affects hydrogen and oxygens concentrations. The η_{con} can express as:

$$\eta_{\text{con}} = -\beta \ln\,(J/J_{\text{max}}) \tag{9}$$

where β is decided by PEMFC and its working status; J is current density, and J_{max} is the maximum current density.

5) CONCENTRATION POLARIZATION OVERVOLTAGE

Charge double-layer exists in PEMFC: hydrogen ions accumulate on the electrolyte surface and electrons gather on the electrode surface. The voltage resulting from the above phenomenon is equivalent to a parallel connection of capacitance C on two ends of the polarization over-resistance R_d, so that charge and energy can be stored on the electrode and electrolyte surfaces as well as in the nearby charge layer. This capacitance is called equivalent capacitance which can effectively "smooth" the voltage drop on the equivalent resistance. It is the double-layer charge capacitance that endows PEMFC with excellent dynamic properties. Hence, the fuel cell voltage dynamic property model can be established by adding capacitance C and inductance L under the stable status (Figure 1).

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

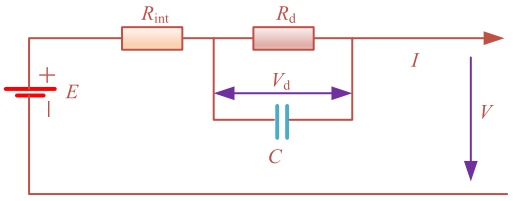

FIGURE 1. Equivalent circuit of PEMFC.

As shown in Figure 1, let the polarizing voltage of R_d be V_d, so that the voltage change of a single cell can be expressed as a differential equation:

$$\mathrm{d}V_\mathrm{d}/\mathrm{d}t = I/C - V_\mathrm{d}/R_\mathrm{d}C \tag{10}$$

The role of the DC/DC converter is not considered in the model, so the output voltage is equal to the stack voltage.

B. HYDROGEN FLOW RATE OF PEMFC

Research implies that the output voltage of a system is largely related to the flow of reaction gases. Since the wind-cooling PEMFC cathode gases are mainly ventilated by blast apparatuses, the space-time gas flow is controlled by linear feedback during modeling, which can promptly trace hydrogen flow variation. Therefore, hydrogen flow is the main factor that determines the output voltage of PEMFC. In this study, together with the electrochemical model of PEMFC, a fuel cell dynamic model is built. The pressure at the hydrogen inlet is altered by controlling the flow of the hydrogen inlet, so as to indirectly control the output voltage of the cell and stabilize the voltage. According to the law of mass conservation and the ideal-gas equation, it can be shown as follows [36]:

$$\frac{V_o}{8.314T} \times \frac{dP_{H_2}}{dt} = m_{H_2} - K\left(P_{H_2} - P_{EH_2}\right) - \frac{0.5NI}{F} \tag{11}$$

where V_0 is the total volume of the anode flow field; m_{H_2} is the hydrogen flow rate; P_{EH_2} is the hydrogen elimination pressure; K is the anode flow coefficient.

C. PEMFC OUTPUT VOLTAGE CONTROL PRINCIPLE

Studies have shown that the output voltage of the system is mainly related to the flow rate of the reaction gas. Because the cathode gas of the air-cooled PEMFC is mainly delivered by the blower, linear feedback control is adopted for the air flow in modeling, which changes according to the needs of the hydrogen supply system in time. Therefore, the hydrogen flow is the main factor determining the output voltage of PEMFC in this article. A dynamic model of the PEMFC is established according to the electrochemical model of PEMFC. The coefficients of the adaptive PI controller are regulated in real time by the coefficient tuner based on AIEM-DDPG algorithm. Then the flow rate of the hydrogen is controlled by the adaptive PI controller thus indirectly controlling the output voltage of the PEMFC to achieve the purpose to stable output voltage. According to the conservation of mass and the conservation of ideal gas, formula (11) is obtained [36].

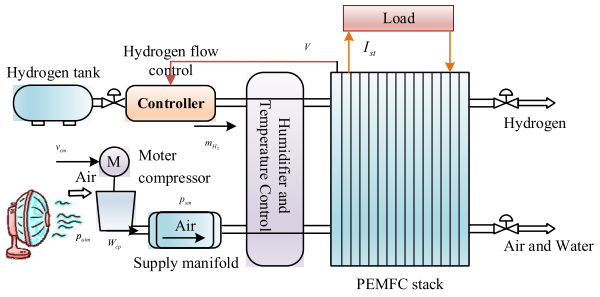

FIGURE 2. PEMFC output voltage control system.

III. INTELLIGENCE CONTROL OF AIEM-DDPG

A. DEEP REINFORCEMENT LEARNING

The purpose of deep reinforcement learning is to realize end-to-end learning from the input to the output of decision-making results. Conventional deep reinforcement learning methods can be divided into two categories: ones based on value function, and those based on policy gradient. The former type often suffers from an unstable training process and cannot deal with the task of continuous action space; the latter type is designed to parameterize the policy, employ a deep neural network for approaching the policy, and seek the optimal policy by following the direction of the policy gradient. The latter type of algorithm is more stable in the training process, but its implementation is more complicated, and the variance resulting from learning by sampling is large [46]–[49].

B. COMMON POLICY GRADIENT ALGORITHMS

1) DDPG

In [50], a DDPG algorithm is proposed, which is a deterministic policy algorithm. DDPG employs two deep neural networks, namely, policy network and value function network. They correspond to policy function $\pi_\phi(\mathrm{s})$ and value function $Q_\theta(s, a)$ respectively, with their parameters of ϕ and θ. DDPG is designed to find an optimal policy π_ϕ which maximizes the expected return value $J(\phi) = \mathrm{E}_{s_i \sim p_\pi a_i \sim \pi}[R_0]$.

The parameter updating of policy network by DDPG through the gradient $\nabla_\phi J(\phi)$ is expressed as the following formula:

$$\nabla_\phi J(\phi) = \mathrm{E}_{s-p_*}\left[\nabla_a Q^\tau(s, a)\big|_{a-\pi(s)} \nabla_\phi \pi_\phi(s)\right] \tag{12}$$

where $Q^\pi(s, a) = \mathrm{E}_{s_j - p_r, a_i - \pi}[r_t \mid s, a]$ means the expected return value after the action a is taken in the state s under the condition of following the policy π.

The parameter updating of value network is realized by the loss minimization function $L(\theta)$.

$$\begin{cases} L(\theta) = \mathrm{E}_{s_s, a_l, r(s_t, a_t), s_{t+1}}\left[(y_t - Q_\theta(s_t, a_t))^2\right] \\ y_t = r(s_t, a_t) + \gamma Q_{\theta^*}(s_{t+1}, a_{t+1}) \\ a_{t+1} \sim \pi_{\phi'}(s_{t+1}) \end{cases} \tag{13}$$

where the ϕ' and θ' represent the parameter of target policy network and target value network respectively. DDPG sends the gradient information of the Q value function to the policy

network through the value function network and formulate the policy along the direction of increasing the Q value according to formula (12).

2) SAC

SAC is developed based on DDPG. For a policy of SAC, the greater the entropy, the higher the randomness of action selection. The expression of entropy is as follows:

$$H\left(\pi\left(\cdot \mid s_t\right)\right) = -\sum_{t=1}^{\infty} \pi\left(\cdot \mid s_t\right) \log \pi\left(\cdot \mid s_t\right) \tag{14}$$

Then, the expression of optimal policy of SAC algorithm is as follows:

$$\pi^* = \arg\max_{\pi} E_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R\left(s_t, a_t, s_{t+1}\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right)\right] \tag{15}$$

3) PPO

Proximal policy optimization (PPO) is a stochastic policy algorithm to deal with the difficulty in determining the learning rate in conventional policy gradient algorithm. PPO algorithm uses the ratio of new policies to old policies to restrict the update range of new policy, making the algorithm insensitive to the learning rate and improving the training efficiency. PPO algorithm is improved based on TRPO, as follows: the reward is as follows:

$$r_t(\theta) = \pi_\theta\left(a_t \mid s_t\right) / \pi_{\theta_{tu}}\left(a_t \mid s_t\right) \tag{16}$$

the target function of conventional TRPO is as follows:

$$L^{CPI}(\theta) = \hat{E}_t\left[\frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{ou}}\left(a_t \mid s_t\right)} \hat{A}_i\right] = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] \tag{17}$$

where the A_2 is the advantage function, that is, the gradient oscillation of the selection probability TRPO controlling the action in (s_t, a_t) is extremely large. Therefore, in order to further control the update rate of policy gradient, PPO adds a CLIP function in the target function, so that the update of $r_t(\theta)$ is restrained at the interval $[r_t(\theta), 1+\varepsilon]$ or $[1+\varepsilon, r_t(\theta)]$, so PPO proposes that the KL divergence should not be too large), and puts forward the following target function:

$$L^{CLIP}(\theta) = E_t\left[\min\left(r_t(\theta)\hat{A}_t, clip\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right)\right] \tag{18}$$

PPO is an on-policy stochastic policy algorithm, and it has the advantages of stable convergence and online update.

4) DDQN AND DQN

DQN learning is to ensure that the Q-estimate of the current value network is as close as possible to the target Q-value of the target value network. This process can be expressed as follows:

$$Loss(\theta) = E\left[\left(Q_{\text{target}} - Q\left(s_t, a_t; \theta\right)\right)^2\right] \tag{19}$$

where $Q(s_t, a_t; \theta)$ is the Q-estimate of the current state, and Q_{target} is the target value, denoted as

$$Q_{\text{target}} = r + \gamma \max_{a_{t+1}} Q\left(s_{t+1}, a_{t+1}; \theta\right) \tag{20}$$

The function of target value is expressed as follows:

$$Y_t^{\text{DQN}} \equiv R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \theta_t^-\right) \tag{21}$$

In order to solve the value estimation of DQN, double-Q estimation is adopted in DDQN, and the objective function value is expressed as follows:

$$Y_t^{DDQN} \equiv r + \gamma Q\left(S_{t+1}, \max_a Q\left(S_{t+1}, a; \theta_t\right); \theta_t^-\right) \tag{22}$$

where r is the reward function value acquired by the agent from the environment; γ is the discount factor; θ_t is the network parameter at the t^{th} iteration; θ_t is the target network parameter at the t^{th} iteration.

C. AIEM-DDPG

Ambient intelligence exploration multi-delay deep deterministic policy gradient (AIEM-DDPG) is a deep reinforcement learning algorithm based on DDPG [50]. In order to solve the Q-value overestimation in DDPG algorithm, this algorithm adopts three tricks: clipping multi-Q learning, policy delayed updating and target policy smooth regularization, which realize better stability and training efficiency of the algorithm.

Conventional DDPG algorithm only employs one actor network to explore the environment, which makes it difficult to guarantee the diversity of samples, and the algorithm is easy to subject to local optimum. To solve this problem, three tricks including ambient intelligent exploration policy, classified priority experience replay, distributed training framework are introduced into the AIEM-DDPG algorithm, which realize better exploration effect of the algorithm.

1) CLIPPING MULTIPLE Q-LEARNING

Inspired by the double-deep Q-learning (DDQN) method, we integrate the current actor network with AIEM-DDPG to select optimal action. Afterward, the policy is evaluated by the target critic network.

$$y_t = r\left(s_t, a_t\right) + \gamma Q_{\theta'}\left(s_{t+1}, \pi_\phi\left(s_{t+1}\right)\right) \tag{23}$$

In the DDPG, the "soft update" [25] method is applied to the target critic and actor networks, which share great similarity to the real network. Moreover, the separation of action policy evaluation becomes hard. For this reason, the target value is calculated by the clipped multiple Q-learning method in the AIEM-DDPG.

$$y_t^1 = r\left(s_t, a_t\right) + \gamma \min_{i=1,2,3} Q_{\theta_i'}\left(s_{t+1}, \pi_{\phi_1}\left(s_{t+1}\right)\right) \tag{24}$$

2) DELAYED POLICY UPDATING

After the critic network is updated for d times, the actor network will be updated once so that it is able to achieve update under the low error of Q value, thus increasing the actor network's update efficiency.

3) SMOOTH REGULARIZATION OF TARGET POLICY

Moreover, random noise is added to the target policy and the values of a mini-batch are averaged for the implementation of smooth regularization.

$$y_t = r\left(s_t, a_t\right) + \mathrm{E}_{\varepsilon}\left[Q_{\theta'}\left(s_{t+1}, \pi_{\phi'}\left(s_{t+1}\right) + \varepsilon\right)\right] \quad (25)$$

Also, a stochastic noise is added to the target strategy, and the values of a mini-batch are averaged for the implementation of smooth regularization.

$$y_t = r\left(s_t, a_t\right) + \gamma \min_{i=1,2} Q_{\theta_t}\left(s_{t+1}, \pi_{\phi'}\left(s_{t+1}\right) + \varepsilon\right) \quad (26)$$

$$\varepsilon \sim clip(N(0, \sigma), -c, c) \quad (27)$$

4) AMBIENT INTELLIGENCE EXPLORATION POLICY AND DISTRIBUTED TRAINING FRAMEWORK

Distributed reinforcement learning, also known as large-scale deep reinforcement learning, is to obtain more generalized deep reinforcement learning policy by using neural network to approximately fit the policy function and the large-scale computing resources to achieve efficient training of neural network models. Ambient intelligence exploration policy has two meanings: one is that agents can continuously learn by themselves in various ways to enrich their own policies and experience; the other is that agents can obtain different information or experience samples by crossing different unrelated environments.

Ambient intelligence exploration policy and large-scale deep reinforcement learning framework are combined together to promote agent learning. The training framework of AIEM-DDPG algorithm is as follows. This framework takes DDPG algorithm as main algorithm and employs explorers composed of many other deep reinforcement learning algorithms with different principles such as SAC, DDPG, PPO and DDQN to explore in different environments. These explorers are named SAC-explorer, DDPG-explorer, PPO-explorer and DDQN-explorer respectively. There are one leader and two experience pools, in which the leader includes three critic networks and one actor network. The framework uses these explorers to collect samples to enrich the learning samples of the leader, so as to achieve better exploration efficiency and training efficiency.

a: DDPG-EXPLORER

Every DDPG-explorer contains one actor network with its own network model and environment. Different environment is explored by several different explorers in parallel. The actor network in different DDPG-explorers adopts different exploration policies, including greedy strategy, gaussian noise and OU noise.

In Q-learning, the ε-greedy strategy indicates choosing any action within the action space with certain probability. Therefore, to imitate the exploration policy of Q learning, the exploration policy of actor network in 6 explorers is set as the greedy strategy, and it is named as the ε-explorer.

The action of ε-DDPG-explorer is shown as follows:

$$a_{\varepsilon}^{l} = \begin{cases} \pi_{\theta}^{l}(s) & \text{With}\varepsilon \text{ probability} \\ a_{\text{rand}}^{l} & \text{With 1-}\varepsilon\text{probability} \end{cases} \quad (28)$$

where the $\pi_{\theta}^{l}(s)$ is the actor network policy of *lth* ε-explorer, the a_{rand} is the action in the total action space.

In addition, the exploration policy of actor network in 6 explorers is set as the OU noise, and it is named as the OU-explorer. By using random OU noise with different variance, the noises among different explorers are different, which can reduce repetition among samples.

The action of OU-DDPG-explorer is shown as follows:

$$a_{OU}^{j} = \pi_{\theta}^{j}(s) + \mathcal{N}_{OU}^{j} \quad (29)$$

where the $\pi_{\theta}^{j}(s)$ is the actor network policy of *jth* OU -explorer, the N_{OU} is the OU noise.

Moreover, the optimization policy of actor network in 6 explorers is set as the Gaussian noise, and it is named as the Gaussian-DDPG-explorer. Random Gaussian noise with different variance is used in different explorers.

$$a_{\text{Gaussian}}^{m} = \pi_{\theta}^{\text{m}}(s) + \mathcal{N}_{Gaussian}^{m} \quad (30)$$

where the $\pi_{\theta}^{m}(s)$ is the actor network policy of *mth* Gaussian -explorer, the $N_{Gaussian}$ is the Gaussian noise.

As a result, by employing the above exploration policy based on different principles, the randomness and diversity of samples explored by explorers can be enhanced.

In offline training, all DDPG-explorers generates the samples $e_i^{DDPG} = (s_t^{i-DDPG}, a_t^{i-DDPG}, r_t^{i-DDPG}, s_{t+1}^{i-DDPG})$ based on their own environment, and add the samples to the public experience pools according to the classified experience priority replay mechanism. Then the learner draws training samples from the pool according to the standard and keep learning. Finally, the actor network in DDPG-explorer regularly updates its network parameters from the latest actor network of the leader.

b: SAC-EXPLORER

There is six SAC-explorers, each of which includes overall SAC structure and different network models and environments. Each SAC-explorer utilizes its own policy to explore in the environment, and puts the explored sample $e_i^{SAC} = (s_t^{i-SAC}, a_t^{i-SAC}, r_t^{i-SAC}, s_{t+1}^{i-SAC})$ into its own experience pool and the public experience pool of the leader at the same time. In addition, SAC-explorer regularly collects samples from its own experience pool for training and updating its parameters. In this article, the excellent exploration ability of SAC algorithm is utilized to explore in different environments and generate different samples, so as to diversify the samples.

c: PPO-EXPLORER

There are six PPO-explorers, each of which includes complete PPO algorithm structure, different network models and environments, each PPO-explorer adopts its own policy to

explore in the environment, and puts the explored sample $e_i^{PPO} = (s_t^{i-PPO}, a_t^{i-PPO}, r_t^{i-PPO}, s_{t+1}^{i-PPO})$ into the its own experience pools and the public experience pools of the leader at the same time. In addition, PPO-explorer regularly updates its parameters and empties each episode of its own experience pool. In this article, PPO algorithm which has policy constraint ability is utilized to obtain the samples explored with more smooth policy, so as to make the policy of the actor in leader converge quickly.

d: DDQN-EXPLORER

There are six DDQN-explores, each of which has complete algorithm structure and different network models, parameters and environments. Each DDQN-explorer uses its own policy and different greedy coefficient ε to explore in the environment, and puts the explored sample $e_i^{DDQN} = (s_t^{i-DDQN}, a_t^{i-DDQN}, r_t^{i-DDQN}, s_{t+1}^{i-DDQN})$ into its own experience pools and the public experience pools of the leader at the same time. Since DDQN is a discrete deep reinforcement learning algorithm, it has the advantage of quick convergence, but is not feasible for solving the problem of continuous space. In this article, DDQN which has the advantage of quick convergence is utilized to quickly obtain the samples with higher value, thus guiding the training of the leader. Nevertheless, this algorithm is not applicable to continuous space, so the samples collected by it only play a guiding role.

e: LEADER

The leader adopts the three tricks to deal with Q-value overestimation in DDPG. In addition, in offline training, the actor network in the leader converges to an optimal solution by continuously sampling and learning from the public experience pool according to the classified experience priority replay mechanism. The leader regularly transmits the parameters of the latest actor network to DDPG-explorer, without communicating with other explorers.

5) CLASSIFICATION EXPERIENCE REPLAY

Classification experience replay standard for the average reward: Under the guidance of the ε-greed methods in Q-learning, the experiential samples are reserved by two independent buffer pools in AIEM-DDPG. In the process of network initialization, the average reward value of the samples in these two pools is marked to be 0. Every time a new sample is brought in, the value will be calculated again. Subsequently, the sample's reward and its average value are compared. When the sample's reward is less than or equal to the average value r_a, then the sample should be put into the pool 1. If not, the sample should be put into pool 2.

During offline training, as to pool 1, n_ξ samples can be gotten with the probability of ξ. In pool 2, $n_{(1-\xi)}$ samples can be gotten with the probability of 1-ξ. The detailed framework is displayed in Figure 3. The explicit process is displayed in Figure 4.

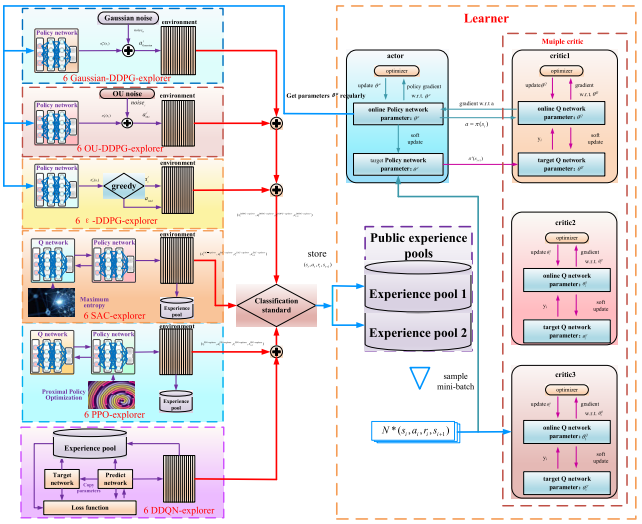

FIGURE 3. Distributed training framework of PEMFC intelligent controller based on AIEM-DDPG.

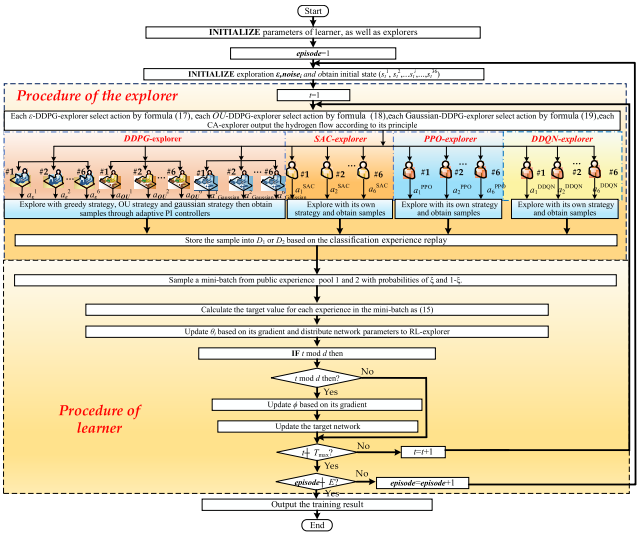

FIGURE 4. AIEM-DDPG method flow.

IV. DESIGN OF AIEM-DDPG CONTROLLER

The adaptive PI controller in this article takes the anode hydrogen flow rate of PEMFC as the control amount and the output voltage of PEMFC as input. AIEM-DDPG algorithm is used as the tuner of the adaptive PI controller, and the coefficients of the PI controller are regulated in real time by the tuner. The input of the tuner includes output voltage and reference voltage, and the output is proportion coefficient and integral coefficient. The control interval of the tuner is 0.01s. The adaptive PI controller controls the flow rate of hydrogen, so that the output voltage of PEMFC can reach a predetermined value, thus ensuring the stability of the system. The objective of the controller is to make the output voltage of PEMFC strictly and accurately follow the reference voltage. Figure 5 shows the detailed control diagram.

A. ACTION SPACE

The action space is expressed as formula (31):

$$\begin{cases} a = [k_p \; k_i] \\ 0 \le k_p \le k_p^{\max} \\ 0 \le k_i \le k_i^{\max} \end{cases} \tag{31}$$

where k_p is the proportionality coefficient of the PI controller, k_p^{max} is the integral coefficient of the PI controller, and k_p^{max} and k_i^{max} are the maximum of these two parameters.

B. STATE SPACE

State is the error $e(t)$ (output voltage error) between the output voltage of the input controller and the reference voltage, its integral to t, and the output voltage v_{st}, as shown in formula (32).

$$\begin{cases} [e(t) \quad \int_0^t edt \quad v_{st}(t)] \\ e(t) = v_{st}^*(t) - v_{st}(t) \end{cases} \tag{32}$$

C. SELECTION OF REWARD FUNCTION

According to formula (33), a composite reward function is formed by the quadratic term of the output voltage error of each control interval, the linear weighting of the square of the action a made in the last control cycle and the control reward term.

The reward function is expressed as follows:

$$\mathrm{r}(t) = -\left[\mu_1 e^2(t) + \mu_2 \sum_{i=1}^{2} a_i^2(t-1)\right] + \beta \tag{33}$$

$$\beta = \begin{cases} 0.8 & e^2(t) \le 0.02 \\ 0 & e^2(t) > 0.02 \end{cases} \tag{34}$$

where t is the discrete time; e(t) is the error of output voltage at time t; $a(t\text{-}1)$ is the action of the agent at time t-1, β is the control reward term. When the control error e(t) is not greater than 0.02, the agent will give a positive reward.

V. SIMULATION

The parameters used in the model are given in Table AI. The fuel cell stack employs 75kW stacks used in the FORD P2000 fuel cell prototype vehicle [51]. The active area of the fuel cell is calculated from the peak power of the stack. The compressor model is based on the Allied Signal compressor detailed in [52]. The membrane properties of Nafion 117 membrane are obtained from [53]. The values of volumes are approximated from the dimensions of the P2000 fuel cell system. The training graph is shown in Figure 5 and the relevant parameters are listed in Table 1. Both the simulation model and programs described in this article have been developed using a server consisting of 48 CPUs. The single CPU is a 2.10GHz Intel Xeon Platinum processor, and the RAM of the server is 192GB. The simulation software package used is MATALB/Simulink version 9.8.0 (R2020a).

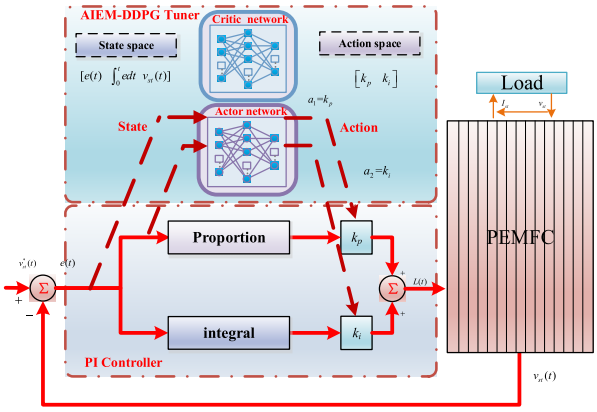

FIGURE 5. Diagram of output voltage control of PEMFC based on AIEM-DDPG tuner.

TABLE 1. Parameter settings.

Parameter	Value
Learning rate of critic	0.001
Learning rate of actor	0.001
Discount factor	0.8
Quantity of explorers	36
Noise variance of the i^{th} Gaussian-explorer	0.07+0.007*i
Noise variance of the i^{th} OU-explorer	0.05+0.005*i
Probability of the i^{th} ε-explorer	0.9
Selection probability of experience pool 1	0.85
Update interval of policy network	2
Volume of experience pools 1 and 2	200000
Noise variance of target action	0.01
Learning rate of critic	0.001

A. PARAMETER SETTING

The parameters of the AIEM-DDPG algorithm are shown in Table 1.

B. ONLINE APPLICATION WITH STEP LOAD

The effectiveness of the algorithm is verified via simulation under two working conditions: 1. step load; and, 2. random load disturbance. This novel method is compared with the following methods: APEX-DDPG [54] adaptive PI (APEX-DDPG-PI) controller, TD3 [55] adaptive PI (TD3-PI) controller, DDPG [50] adaptive PI (DDPG-PI) controller, DQN [36] adaptive PI (DQN-PI) controller, and REF [30] adaptive PI (REF-PI) controller and DDPG controller. These controllers are all of the adaptive controller type.

The PSO-optimized fuzzy PI (PSO-FPI) controller, fuzzy PI (FPI) controller [29] and PI controller are all conventional controllers. The simulation time of the step load is 5s. At 1s, the load current increases from 100A to 120A. Figures 6(a)~(d) and Table 2 show the results of the simulation. The Rising time represents the first time at which 99% of the reference value is reached. The stabilization time is the time of stabilizing in the range of 0.1% of the reference value.

TABLE 2. The response results of the output voltage control of PEMFC.

Parameter	AIEM-DDPG -PI	APEX-DDPG -PI	TD3 -PI	DDPG -PI	DDPG	DQN-PI	REF-PI	PSO-fuzzy-PID	Fuzzy-PID	PID
Rise time T_r/s	**0.01**	0.01	0.02	0.02	0.01	0.02	0.02	0.11	0.13	0.03
Stable time T_s/s	**0.06**	0.08	0.08	0.09	0.21	0.36	0.37	0.97	1.09	0.41
Overshoot σ/%	**0.0674**	0.0833	0.0920	0.108	0.258	0.151	0.149	0.635	0.585	0.756

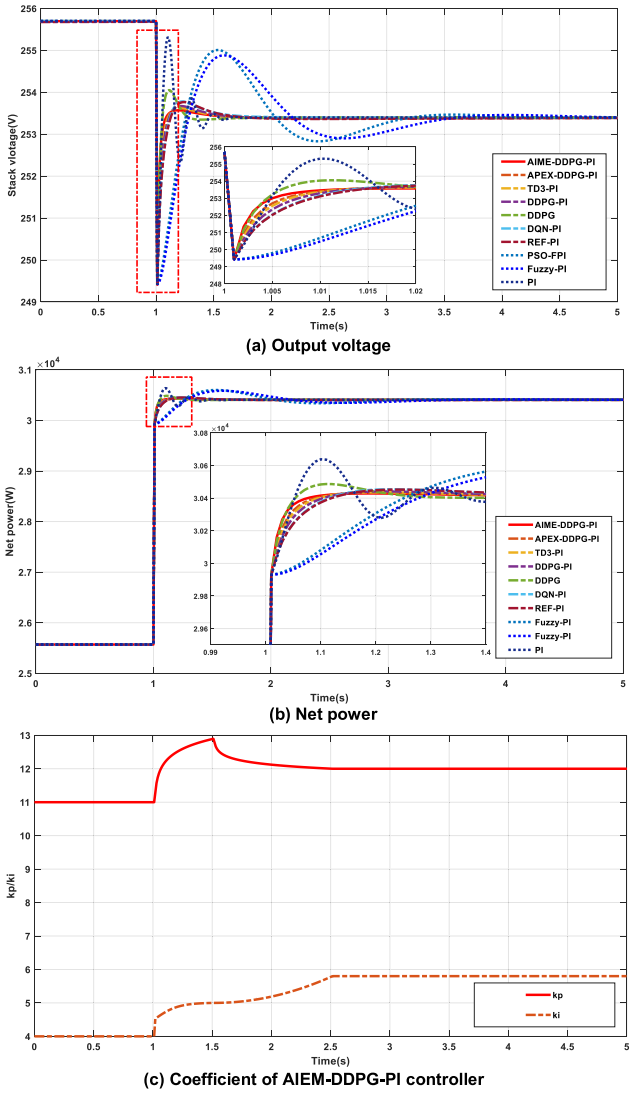

FIGURE 6. Diagram of simulation results of PEMFC under step disturbance.

As shown in Figure 6(a) and Table 2, compared with the other adaptive controllers, the AIEM-DDPG-PI controller has better control performance, smaller overshoot and faster response speed under load disturbance. In addition, it can be seen that the AIEM-DDPG-PI controller has the best control performance. This is because the AIEM-DDPG-PI controller adopts various tricks for improving exploration efficiency and attaining better control performance in the training. By contrast, the exploration policies of the APEX-DDPG-PI controller, TD3-PI controller and DDPG-PI controller are too narrow; as a result, these controllers are more likely to converge on a local optimal solution, making it impossible to obtain a control policy with better control performance.

The DQN-PI controller adopts the discrete reinforcement learning algorithm DQN as the tuner, which leads to failure in the continuous regulation of k_p and k_i, resulting in poor control performance. Its stabilization time and overshoot are 6 times and 2.24 times that of AIEM-DDPG-PI controller, respectively. As shown in Figure 6(c), k_p and k_i of the AIEM-DDPG-PI controller constantly change with the system state, whereby k_p is increased at the early stage of the load sudden change for rapid response, and is gradually decreased at the end of the disturbance in order to attain smooth regulation.

In the REF-PI controller, a conventional REF neural network serves as the tuner of PI controller, which makes the controller vulnerable to the sample accuracy. Its stabilization time and overshoot are 6.17 times and 2.21 times that of the AIEM-DDPG-PI controller, respectively.

The overshoot of AIEM-DDPG-PI is obviously smaller (3.83 times) than that of the DDPG controller, even though the latter has a similar response speed compared with the former (rising time equaling 0.01s), and the DDPG controller exhibits significant oscillation. This is because the DDPG algorithm is not generalized, which leads to poor robustness of the controller; meanwhile, the AIEM-DDPG-PI controller can smoothly regulate the output voltage.

Compared with conventional controllers, the AIEM-DDPG-PI controller exhibits better control performance in terms of response speed, stability and overshoot. The maximum rising time of the controllers of conventional algorithms is about 11 times that of the AIEM-DDPG-PI controller, the maximum stabilization time is 18.17 times and the maximum overshoot is 11.22 times that of other controllers.

The performance of controllers operating on conventional algorithms is affected by large overshoot and oscillation, due to their innate inability to handle nonlinear systems (for example, the PEMFC). As can be seen in Figure 6(b), the controller based on the AIEM-DDPG algorithm can ensure stability in the net power. Therefore, the AIEM-DDPG controller proposed in this article achieve better control performance and stability under step load.

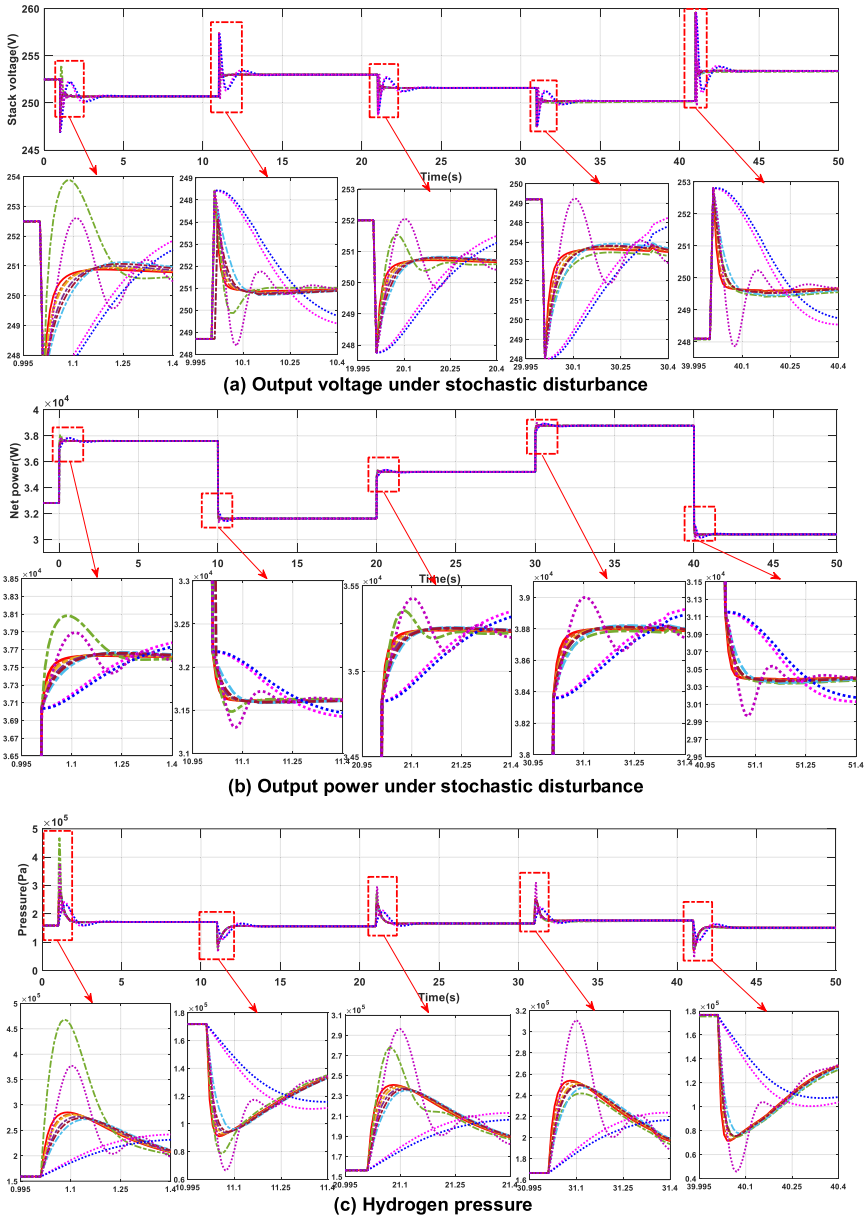

FIGURE 7. Diagram of simulation results of PEMFC under stochastic load.

C. ONLINE APPLICATION WITH STOCHASTIC LOAD

In order to verify the robustness and control performance of the AIEM-DDPG controller, a simulation has been carried out under a stochastic load working condition. Appendix AI shows the stochastic load, where each condition lasts for 10s and the total simulation time is 51s. Figures.8(a)∼(d) show the results.

As shown in Figures 7(a)∼(b), under different loads the AIEM-DDPG controller achieves the best control performance in terms of output voltage and power, with the most stable regulation curve and the smallest overshoot. This is because a large number of samples under different loads have been learned during offline training, which makes the controller highly adaptive and robust. In addition, it is found that the DDPG controller has poor robustness, therefore it attains completely different control performances under different load working conditions. At 31∼41s and 41∼51s, its response speed to output voltage is fast and overshoot is small; but under other conditions, especially at 1∼11s, its response speed is extremely slow and there is a great overshoot in output voltage. By comparison, the AIEM-DDPG-PI controller, in which the PI controller is the carrier, is an improvement on the DDPG framework. The AIEM-DDPG algorithm is only used as the tuner of the PI controller. The algorithm is not directly used as the control algorithm for developing the control strategy; as a consequence, the controller has greater adaptability and robustness. Because the AIEM-DDPG-PI controller maintains the output voltage, its net power is also relatively stable (Figure 7(b)). As can be seen in Figure 7(c), the dynamic response is in direct

TABLE 3. Parameter of PEMFC.

symbol	parameter	value
$\rho_{m,dry}$	Number of cells in fuel-cell stack	0.002kg/cm^3
$M_{m,dry}$	membrane dry equivalent weight	1.1kg/mol
t_m	membrane thickness	0.01275cm
n	number of cell in fuel cell stack	381
A_{fc}	fuel cell active area	280cm^2
d_c	compressor diameter	0.2286m
J_{cp}	compressor and motor inertia	5*10^{-5}kg*m^2
V_{an}	anode volume	0.005m^3
V_{ca}	cathode volume	0.01m^3
V_{sm}	supply manifold volume	0.02m^3
V_{rm}	return manifold volume	0.005m^3
$C_{D,\,rm}$	return manifold throttle discharge coefficient	0.0124
$A_{T,\,rm}$	return manifold throttle discharge coefficient	0.002m^2
$k_{rm,out}$	supply manifold outlet orifice constant	0.3629*10^{-5} kg/(s*Pa)
$k_{ca,out}$	cathode outlet orifice constant	0.2177*10^{-5} kg/(s*Pa)
M_a	Air molar mass	29*10^{-3} kg*mol^{-1}
M_{O_2}	Oxygen molar mass	32*10^{-3} kg*mol^{-1}
M_{N_2}	Nitrogen molar mass	28*10^{-3} kg*mol^{-1}
M_v	Oxygen molar mass	18.02 kg*mol^{-1}
p_{atm}	Atmospheric pressure	101.325 kPa
T_{atm}	Atmospheric temperature	298.15 K
γ	Ratio of specific heat of air	1.4
C_p	Constant pressure specific heat of air	1004 J/(mol*K)
ρ_a	Air density	1.23 kg/m^3
R	Universal gas constant	8.3145 J/(mol*K)
R_a	Air gas constant	286.9 J/(mol*K)
R_{O_2}	Oxygen gas constant	259.8 J/(mol*K)
R_{N_2}	Nitrogen gas constant	296.8 J/(mol*K)
R_v	Vapor gas constant	461.5 J/(mol*K)
R_{H_2}	Hydrogen gas constant	4124.3 J/(mol*K)

proportion to the overshoot of hydrogen pressure of the controller. The AIEM-DDPG-PI controller will have a reasonable hydrogen overshoot at the initial stage of load change due to its ability to maintain fast response speed, but it will soon return to the reference value. The DDPG controller will also exhibit a large overshoot of hydrogen pressure when its performance is poor; however, this emanates from the problem of the algorithm focusing excessively on the response speed and neglecting the static error.

VI. CONCLUSION

In this article, an adaptive PI controller for PEMFC output voltage is proposed, and an AIEM-DDPG algorithm is proposed as the tuner of this PI controller for regulating the coefficient of the controller. This algorithm is an improvement on the DDPG algorithm. Its innovations include the ambient intelligence exploration policy (AIEM), which uses explorers with various exploration policies to conduct distributed exploration in the environment. In addition, the classified priority experience replay mechanism is introduced as it improves the efficiency of exploration and training. In addition, clipping multi-Q learning, delay policy updating, and target policy smooth regularization are used to solve the problem of Q value overestimation. Ultimately, a coefficient regulation algorithm with outstanding adaptability is obtained, which can actively regulate the coefficient of

the PI controller in alignment with the varying state of the PEMFC.

2) By simulating the PEFMC under different working conditions, it is found that the AIEM-DDPG controller can meet the real-time control requirements under different operating load disturbances in the output voltage control system of the PEMFC.

The AIEM-DDPG-PI controller in this article overcomes the problems of slow convergence and poor generalization of the DDPG algorithm, combines the robustness of PI controller with the perception of deep reinforcement learning, improves the performance of conventional PI controller, and realizes accurate and stable millisecond control of output voltage. The problem of low robustness of the DDPG controller (which directly applies the DDPG algorithm to formulate control strategies) is thus avoided. For that reason alone, the AIEM-DDPG algorithm is of great practical significance. The authors will apply this algorithm to the actual PEMFC system in future work.

APPENDIX

See Table 3.

REFERENCES

[1] D. Zhao, Y. Huangfu, M. Dou, and F. Gao, "Cathode partial pressure estimation of a proton exchange membrane fuel cell for transportation applications," in *Proc. IEEE Conf. Expo Transp. Electrific. Asia–Pacific (ITEC Asia–Pacific)*, Beijing, China, Aug./Sep. 2014, pp. 1e–5e.

[2] L. Sun, Y. Jin, L. Pan, J. Shen, and K. Y. Lee, "Efficiency analysis and control of a grid-connected PEM fuel cell in distributed generation," *Energy Convers. Manage.*, vol. 195, pp. 587–596, Sep. 2019.

[3] Y. Qiu, P. Wu, T. Miao, J. Liang, K. Jiao, T. Li, J. Lin, and J. Zhang, "An intelligent approach for contact pressure optimization of PEM fuel cell gas diffusion layers," *Appl. Sci.*, vol. 10, no. 12, p. 4194, Jun. 2020.

[4] R. M. Aslam, D. B. Ingham, M. S. Ismail, K. J. Hughes, L. Ma, and M. Pourkashanian, "Simultaneous direct visualisation of liquid water in the cathode and anode serpentine flow channels of proton exchange membrane (PEM) fuel cells," *J. Energy Inst.*, vol. 91, no. 6, pp. 1057–1070, Dec. 2018.

[5] J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng, "Control of fuel cell breathing," *IEEE Control Syst.*, vol. 24, no. 2, pp. 30–46, Apr. 2004.

[6] J. T. Pukrushpan, H. Peng, and A. G. Stefanopoulou, "Control-oriented modeling and analysis for automotive fuel cell systems," *J. Dyn. Syst., Meas., Control*, vol. 126, no. 1, pp. 14–25, Mar. 2004.

[7] J. Sun and I. V. Kolmanovsky, "Load governor for fuel cell oxygen starvation protection: A robust nonlinear reference governor approach," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 6, pp. 911–920, Nov. 2005.

[8] Y. Xiong and X. Deng, "Research on the control of the cathode gas flow and pressure of a small PEM fuel cell," in *Proc. 6th World Congr. Intell. Control Automat.*, Jun. 2006, pp. 7711–7715.

[9] K.-W. Suh and A. G. Stefanopoulou, "Performance limitations of air flow control in power-autonomous fuel cell systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 465–473, May 2007.

[10] A. Vahidi, A. Stefanopoulou, and H. Peng, "Model predictive control for starvation prevention in a hybrid fuel cell system," in *Proc. Amer. Control Conf.*, vol. 1, Jun./Jul. 2004, pp. 834–839.

[11] N. Chatrattanawet, T. Hakhen, S. Kheawhom, and A. Arpornwichanop, "Control structure design and robust model predictive control for controlling a proton exchange membrane fuel cell," *J. Cleaner Prod.*, vol. 148, pp. 934–947, Apr. 2017.

[12] H. Beirami, A. Z. Shabestari, and M. M. Zerafat, "Optimal PID plus fuzzy controller design for a PEM fuel cell air feed system using the self-adaptive differential evolution algorithm," *Int. J. Hydrogen Energy*, vol. 40, no. 30, pp. 9422–9434, Aug. 2015.

[13] J. Han, S. Yu, and S. Yi, "Adaptive control for robust air flow management in an automotive fuel cell system," *Appl. Energy*, vol. 190, pp. 73–83, Mar. 2017.

[14] J. K. Gruber, C. Bordons, and A. Oliva, "Nonlinear MPC for the airflow in a PEM fuel cell using a volterra series model," *Control Eng. Pract.*, vol. 20, no. 2, pp. 205–217, Feb. 2012.

[15] Y.-X. Wang and Y.-B. Kim, "Real-time control for air excess ratio of a PEM fuel cell system," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 852–861, Jun. 2014.

[16] Y.-B. Kim, "Improving dynamic performance of proton-exchange membrane fuel cell system using time delay control," *J. Power Sources*, vol. 195, no. 19, pp. 6329–6341, Oct. 2010.

[17] M. A. Danzer, J. Wilhelm, H. Aschemann, and E. P. Hofer, "Model-based control of cathode pressure and oxygen excess ratio of a PEM fuel cell system," *J. Power Sources*, vol. 176, no. 2, pp. 515–522, Feb. 2008.

[18] S. Rodatz, G. Paganelli, and L. Guzzella, "Optimizing air supply control of a PEM fuel cell system," in *Proc. Amer. Control Conf.*, Denver, CO, USA, Jun. 2003, pp. 2043–2048.

[19] A. Arce, D. R. Ramirez, A. J. Del Real, and C. Bordons, "Constrained explicit predictive control strategies for PEM fuel cell systems," in *Proc. 46th IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 6088–6093.

[20] M. Abdullah and M. Idres, "Fuel cell starvation control using model predictive technique with Laguerre and exponential weight functions," *J. Mech. Sci. Technol.*, vol. 28, no. 5, pp. 1995–2002, May 2014.

[21] G. Park and Z. Gajic, "A simple sliding mode controller of a fifth-order nonlinear PEM fuel cell model," *IEEE Trans. Energy Convers.*, vol. 29, no. 1, pp. 65–71, Mar. 2014.

[22] R. J. Talj, D. Hissel, R. Ortega, M. Becherif, and M. Hilairet, "Experimental validation of a PEM fuel-cell reduced-order model and a motor-compressor higher order sliding-mode control," *IEEE Trans. Ind. Electron.*, vol. 57, no. 6, pp. 1906–1913, Jun. 2010.

[23] J. Liu, W. Luo, X. Yang, and L. Wu, "Robust model-based fault diagnosis for PEM fuel cell air-feed system," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3261–3270, May 2016.

[24] L. Sun, J. Shen, Q. Hua, and K. Y. Lee, "Data-driven oxygen excess ratio control for proton exchange membrane fuel cell," *Appl. Energy*, vol. 231, pp. 866–875, Dec. 2018.

[25] K. Ou, Y.-X. Wang, and Y.-B. Kim, "Performance optimization for open-cathode fuel cell systems with overheating protection and air starvation prevention," *Fuel Cells*, vol. 17, no. 3, pp. 299–307, Jun. 2017.

[26] J. G. Williams, G.-P. Liu, K. Thanapalan, and D. Rees, "Design and implementation of on-line self-tuning control for PEM fuel cells," *World Electr. Vehicle J.*, vol. 2, no. 4, pp. 242–252, Dec. 2008.

[27] M. Hatti and M. Tioursi, "Dynamic neural network controller model of PEM fuel cell system," *Int. J. Hydrogen Energy*, vol. 34, no. 11, pp. 5015–5021, Jun. 2009.

[28] P. E. M. Almeida and M. G. Simoes, "Neural optimal control of PEM fuel cells with parametric CMAC networks," *IEEE Trans. Ind. Appl.*, vol. 41, no. 1, pp. 237–245, Jan. 2005.

[29] J. G. Williams, G. Liu, S. Chai, and D. Rees, "Intelligent control for improvements in PEM fuel cell flow performance," *Int. J. Autom. Comput.*, vol. 5, no. 2, pp. 145–151, May 2008.

[30] C. Damour, M. Benne, C. Lebreton, J. Deseure, and B. Grondin-Perez, "Real-time implementation of a neural model-based self-tuning PID strategy for oxygen stoichiometry control in PEM fuel cell," *Int. J. Hydrogen Energy*, vol. 39, no. 24, pp. 12819–12825, Aug. 2014.

[31] K. Ou, Y.-X. Wang, Z.-Z. Li, Y.-D. Shen, and D.-J. Xuan, "Feedforward fuzzy-PID control for air flow regulation of PEM fuel cell system," *Int. J. Hydrogen Energy*, vol. 40, no. 35, pp. 11686–11695, Sep. 2015.

[32] Z. Baroud, M. Benmiloud, and A. Benalia, "Fuzzy self-tuning PID controller for air supply on a PEM fuel cell system," in *Proc. 4th Int. Conf. Elect. Eng. (ICEE)*, Boumerdes, Algeria, Dec. 2015, pp. 1–4.

[33] J. Chen, Z. Liu, F. Wang, Q. Ouyang, and H. Su, "Optimal oxygen excess ratio control for PEM fuel cells," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1711–1721, Sep. 2018.

[34] D. Zhao, F. Li, R. Ma, G. Zhao, and Y. Huangfu, "An unknown input nonlinear observer based fractional order PID control of fuel cell air supply system," *IEEE Trans. Ind. Appl.*, vol. 56, no. 5, pp. 5523–5532, Sep./Oct. 2020.

[35] E. D. Sontag, *Input to State Stability: Basic Concepts and Results*. Berlin, Germany: Springer, 2008, pp. 163–220.

[36] X. Zhang, Z. Xu, T. Yu, B. Yang, and H. Wang, "Optimal mileage based AGC dispatch of a GenCo," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2516–2526, Jul. 2020.
[37] X. Zhang, T. Tan, B. Zhou, T. Yu, B. Yang, and X. Huang, "Adaptive distributed auction-based algorithm for optimal mileage based AGC dispatch with high participation of renewable energy," *Int. J. Electr. Power Energy Syst.*, vol. 124, Jan. 2021, Art. no. 106371.
[38] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, and D. Wierstra, "Continuous control with deep reinforcement learning," *Comput. Sci.*, vol. 8, no. 6, p. A187, Sep. 2015.
[39] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 348–368, Dec. 2018.
[40] P. Chen, Z. He, C. Chen, and J. Xu, "Control strategy of speed servo systems based on deep reinforcement learning," *Algorithms*, vol. 11, no. 5, p. 65, May 2018.
[41] L. Xi, J. Wu, Y. Xu, and H. Sun, "Automatic generation control based on multiple neural networks with actor-critic strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 14, 2020, doi: 10.1109/TNNLS.2020.3006080.
[42] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 814–817, Jan. 2020.
[43] H. Shi, Y. Sun, G. Li, F. Wang, D. Wang, and J. Li, "Hierarchical intermittent motor control with deterministic policy gradient," *IEEE Access*, vol. 7, pp. 41799–41810, Mar. 2019.
[44] J. T. Pukrushpan, *Modeling and Control of Fuel Cell Systems and Fuel Processors*. Ann Arbor, MI, USA: Univ. Michigan, 2003.
[45] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, May 1979.
[46] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "A review of deep learning for renewable energy forecasting," *Energy Convers. Manage.*, vol. 198, Oct. 2019, Art. no. 111799.
[47] H. Wang, Y. Liu, B. Zhou, C. Li, G. Cao, N. Voropai, and E. Barakhtenko, "Taxonomy research of artificial intelligence for deterministic solar power forecasting," *Energy Convers. Manage.*, vol. 214, Jun. 2020, Art. no. 112909.
[48] H. Z. Wang, G. B. Wang, G. Q. Li, J. C. Peng, and Y. T. Liu, "Deep belief network based deterministic and probabilistic wind speed forecasting approach," *Appl. Energy*, vol. 182, pp. 80–93, Nov. 2016.
[49] D. Xu, Q. Wu, B. Zhou, C. Li, L. Bai, and S. Huang, "Distributed multi-energy operation of coupled electricity, heating, and natural gas networks," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2457–2469, Oct. 2020, doi: 10.1109/TSTE.2019.2961432.
[50] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, Feb. 2015.
[51] J. A. Adams, W. Yang, K. A. Oglesby, and K. D. Osborne, "The development of Ford's P2000 fuel cell vehicle," *SAE Trans.*, vol. 6, no. 109, pp. 1634–1645, Mar. 2000.
[52] J. M. Cunningham, M. A. Man, R. M. Moore, and D. J. Friedman, "Requirements for a flexible and realistic air supply model for incorporation into a fuel cell vehicle (FCV) system simulation," *SAE Int.*, vol. 108, pp. 3191–3196, Aug. 1999.
[53] T. Nguyen and R. E. White, "A water and heat management model for proton-exchange-membrane fuel cells," *J. Electrochem. Soc.*, vol. 140, no. 8, pp. 2178–2186, Aug. 1993.
[54] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, "Distributed prioritized experience replay," in *Proc. Int. Conf. Learn. Represent.*, Mar. 2018, pp. 1–19.
[55] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," Feb. 2018, *arXiv:1802.09477*. [Online]. Available: http://arxiv.org/abs/1802.09477

JIAWEN LI received the M.S. degree in electrical engineering from Northeast Electric Power University, Jilin, China, in 2016. He is currently pursuing the D.Eng. degree in electrical engineering with the School of Electric Power, South China University of Technology. His research interest includes automatic generation control.

TAO YU is currently a Professor of power system with the School of Electric Power, South China University of Technology (SCUT), Guangzhou, China. His research interest includes nonlinear and coordinated control theory.

BO YANG received the B.Eng. degree in electrical engineering from the South China University of Technology, Guangzhou, China, in 2010, and the Ph.D. degree in electrical engineering from the University of Liverpool, Liverpool, U.K., in 2015. He joined the Faculty of Electric Power, Kunming University of Science and Technology, as a Lecturer. His research interests include nonlinear adaptive control, VSC-HVDC systems, and wind generations. He won the Full Scholarship from the China Scholarship Council in 2011.