sensors

MDPI

Article

Sensors Integrated Control of PEMFC Gas Supply System Based on Large-Scale Deep Reinforcement Learning

Jiawen Li and Tao Yu *

College of Electric Power, South China University of Technology, Guangzhou 510640, China; eplijiawen@mail.scut.edu.cn
* Correspondence: taoyu1@scut.edu.cn; Tel.: +86-13002088518

Abstract: In the proton exchange membrane fuel cell (PEMFC) system, the flow of air and hydrogen is the main factor influencing the output characteristics of PEMFC, and there is a coordination problem between their flow controls. Thus, the integrated controller of the PEMFC gas supply system based on distributed deep reinforcement learning (DDRL) is proposed to solve this problem, it combines the original airflow controller and hydrogen flow controller into one. Besides, edge-cloud collaborative multiple tricks distributed deep deterministic policy gradient (ECMTD-DDPG) algorithm is presented. In this algorithm, an edge exploration policy is adopted, suggesting that the edge explores including DDPG, soft actor-critic (SAC), and conventional control algorithm are employed to realize distributed exploration in the environment, and a classified experience replay mechanism is introduced to improve exploration efficiency. Moreover, various tricks are combined with the cloud centralized training policy to address the overestimation of Q-value in DDPG. Ultimately, a model-free integrated controller of the PEMFC gas supply system with better global searching ability and training efficiency is obtained. The simulation verifies that the controller enables the flows of air and hydrogen to respond more rapidly to the changing load.

Keywords: distributed deep reinforcement learning; edge-cloud collaborative multiple tricks distributed deep deterministic policy gradient; PEMFC; integrated control of gas supply system

Citation: Li, J.; Yu, T. Sensors Integrated Control of PEMFC Gas Supply System Based on Large-Scale Deep Reinforcement Learning. *Sensors* **2021**, *21*, 349. https://doi.org/10.3390/s21020349

Received: 8 December 2020
Accepted: 31 December 2020
Published: 6 January 2021

1. Introduction

The accelerating growth of the global economy in recent years has been accompanied by increasing consumption of fossil fuels and attendant carbon emissions. The environmental pollution caused by traditional fossil fuel energy consumption (represented by oil, gas, and coal) poses an existential threat to humanity and ecosystems. In response, governments and scientists worldwide are directing more resources into renewable energy ('green') technology. Fuel cells that can directly convert chemical energy into electrical energy via chemical reactions are widely recognized as a valuable energy source. Among them, the proton exchange membrane fuel cell (PEMFC) has attracted more focus due to its fast start-up ability, high energy conversion rate, low working temperature, lightweight properties, and resistance to external natural conditions [1].

The PEMFC system generally includes four parts: gas supply system, thermal management system, water management system, and the energy control system. The gas supply system includes airflow control and hydrogen flow control. The airflow affects the oxygen excess ratio (OER) of the PEMFC. If the airflow is insufficient, oxygen starvation will occur, causing the output voltage of the PEMFC to decline sharply; if the airflow is excessive, oxygen saturation will occur, leading to an increase in the parasitic power, which affects the output characteristics of the PEMFC. Therefore, reasonable control of airflow is required so that the net power of the PEMFC remains stable [2–4]. Similarly, hydrogen flow has a significant impact on the output voltage of the PEMFC, in that reasonable hydrogen flow control enables the output voltage to respond timeously to load changes, thus stabilizing the output voltage and power. A large number of previous studies on PEMFC control

have focused mainly on control of air flow and hydrogen flow. There are two general consensuses concerning these variables. Firstly, it is widely assumed that, provided that the hydrogen storage is sufficient and the flow control response is rapid, the PEMFC can accurately and rapidly satisfy the airflow demand; secondly, provided that there is a sufficient supply of air and its flow can be controlled rapidly, the PEMFC can meet the demand of the hydrogen flow control system timeously. However, with air or hydrogen supply systems it is often difficult to ensure sufficient gas storage and guarantee instantaneous response at any time because the relationship between gas flow rate and output voltage is non-linear and the response of the output voltage is delayed. Furthermore, the flow of the two gases jointly affects the output voltage and net power of the PEMFC. If the coordination between the two supply systems is insufficient, the output stability of the PEMFC will be seriously affected.

To solve the above problems, Woon et al. proposed a multi-input multi-output (MIMO) high order sliding mode algorithm to simultaneously control the air flow and hydrogen flow of the PEMFC. They considered the coordination between the control of the two gas streams, enabling the output voltage to remain in a stable state under different load conditions [5]. However, the high-frequency discontinuous switching items in the sliding mode controller make the actual level of control by the system discontinuous over time. As a result, the sliding mode controller will inevitably cause unfavorable chattering problems to affect the output voltage of the PEMFC. Therefore, this method cannot be applied to an actual PEMFC system. Sankar et al. put forward a sliding mode observer (SMO)-based nonlinear multivariable sliding mode controller (SMC) and globally linearizing controller (GLC) for a PEMFC; it employs a non-linear observer to improve the control performance of the algorithm and reduce chattering, thus realizing comprehensive optimal control of oxygen and hydrogen [6]. However, this method is complicated and to date has not proved to be a practical real-world solution. Wang et al. adopted a robust controller which regulates the flow of oxygen and hydrogen. Their PEMFC was modeled as a MIMO system, with air and hydrogen flow rates as the inputs and stack voltage and current as the outputs, which enables stable output voltage as well as lower hydrogen consumption [7,8]. Nevertheless, the steady-state accuracy of their PEMFC system proved to be weak since the robust controller generally does not work in the optimal state, and the underlying calculations are complicated. Thus, it is difficult for this method to be implemented in practice. On the basis of the designed state-space mathematical model, He et al. used a proportion integral (PI) controller and state feedback controller to control the flow rate of hydrogen and air respectively, so as to realize the coordinated control of the two gases [9]. However, their PI controllers and feedback controllers are incompatible with nonlinear systems (i.e., PEMFC), and their experimentation did not consider the control performance in achieving optimal control. Drawing on expert system and fuzzy control theory, Almeida et al. proposed a comprehensive intelligent controller that comprehensively considers the coordination of four systems: gas control, thermal management, water management, and energy control. However, the method cannot easily be applied to real-world systems due to the inefficient search ability and absence of self-learning in traditional expert systems [10]. Additionally, Almeida et al. proposed a method for realizing optimal control of PEMFC gas streams based on the parameterized cerebella model articulation controller (P-CMAC), whereby the voltage can be controlled by manipulating the flow rates of hydrogen and air [11]. This control strategy amounts to an approximate optimal control strategy; nevertheless, the neural network used in this method was a simple artificial neural network, resulting in extremely low robustness of the controller. In addition, the specific performance of this method was determined by the quality of the samples during offline training. Therefore, the method is lacking in self-correction ability, and it cannot ensure optimal control due to insufficient training.

Although the aforementioned algorithms enable coordinated control of the gas system, most of them do not employ a simple, high-performance control strategy, which would realize optimal control of the output voltage as well as appropriate oxygen excess rate. Thus,

a simple model-free control algorithm is required so that a simpler, more practical method for controlling the gas supply system of the PEMFC can be realized, one which ensures coordination between hydrogen and oxygen supplies in the control process. The deep deterministic policy gradient (DDPG) algorithm in deep reinforcement learning is a model-free algorithm [12–14]. It combines the perception of deep learning with the decision-making ability of reinforcement learning and is characterized by strong adaptive ability, timely response, and accurate control. In contrast with conventional control methods, DDPG permits full interaction with the environment [15,16] and can be used to address the uncertainty in nonlinear control. Besides, controllers based on DDPG can realize MIMO control and are now being applied to various control fields [17–19]; to date, however, a working DDPG-based controller for a PEMFC has yet to be successfully demonstrated.

Furthermore, the DDPG algorithm still has many shortcomings, which make it an unsuitable candidate for the field of precision control. First of all, there is the problem of over-estimation of the Q value, causing the algorithm to fall into the local optimal solution, causing the final control strategy to deviate and yield poor control performance or even oscillation. Second, the DDPG algorithm is not sufficiently robust during offline training, and so the controller that directly uses this algorithm as a control algorithm has very low robustness. Third, the DDPG algorithm requires a long offline training period, which greatly limits its potential application in the industry.

In order to address these problems, several scholars have carried out a number of research studies. In response to the first problem, Fujimoto et al. proposed the twin delayed deep deterministic policy gradient (TD3) algorithm, whereby they introduced a policy delay update technique [20]. Their experimental results were moderately successful; however, they opined that solving Q value overestimation remains an intractable problem. Regarding the second problem, Horgan et al. proposed a new algorithm operating on a distributed reinforcement learning framework, which enables greater exploration ability [21]. Their results show that this algorithm did improve the algorithm's exploration ability; however, it also learned more low-value samples, boosting offline training costs. Addressing the third problem, Schau et al. proposed a prior experience replay technique [22], which can sort samples according to their value and preferentially select samples with high value for training. Nevertheless, to date, the algorithm which only employs this technique remains ineffective. Another technique is the edge-cloud collaborative framework, a form of collaborative computing that has recently caught the imagination of the academic community. This framework can be used as a hardware architecture for deep reinforcement learning to obtain better computing power and to increase the exploration ability and optimization speed of the algorithm. A distributed reinforcement learning framework can be embedded in such an architecture and used to obtain better training results [23].

In view of the fact that DDPG suffers from the problem of Q value overestimation as well as the problem of low exploration ability (ascribed to its single exploration ability, in that it employs only one actor network to explore the environment), a new algorithm which can solve these problems and achieve better coordination control of air and hydrogen flow is proposed. A number of improvements to the original DDPG algorithm are presented. In this paper, an integrated gas supply system controller for a PEMFC based on the ECMTD-DDPG algorithm is presented.

This paper makes the following novel contributions to the field:

1. An integrated controller based on deep reinforcement learning is proposed. This is a combined controller that integrates the flow controllers for air and hydrogen. It can formulate the control strategy according to different PEMFC states and output the motor voltage of the air compressor and hydrogen flow simultaneously. Hence, the real-time control requirements of the PEMFC under different operating conditions can be met.

2. An ECMTD-DDPG algorithm is proposed for the aforementioned framework. This is a distributed deep reinforcement learning algorithm with the edge-cloud collaborative framework. It is designed with multiple edge explorers and a cloud learner. Second, the edge exploration policy is adopted in edge explorers in order to increase exploration effi-

ciency. Third, the classified experience replay mechanism is introduced in order to improve training efficiency. Fourth, the cloud learner employs clipping multi-Q learning, delay policy updating, and smooth regularization of target policy to solve overestimation of Q value in DDPG. Finally, a control algorithm with better robustness and adaptability is proposed. The simulation results below demonstrate that the proposed method can effectively control the output voltage of the PEMFC and ensure stable operation of the system.

2. The Principle of the Gas Supply System

The chemical reactions are shown in Figure 1. First, hydrogen enters the PEMFC polar plate flow channel from the anode inlet. (The polar plate flow channel is attached to the anode gas diffusion layer.) Then, the hydrogen passes through the gas diffusion layer to the catalytic layer and undergoes a reduction reaction under the action of the catalyst, separating into hydrogen ions and electrons. The proton exchange membrane has selective permeability, which allows hydrogen ions to pass through whilst blocking the electrons. Therefore, hydrogen ions can pass through the proton exchange membrane to reach the cathode, while electrons can only reach the cathode through the external circuit. The flow of electrons in the external circuit constitutes direct current and supplies power to the load. At the cathode, the hydrogen ions passing through the proton exchange membrane and the electrons arriving through the external circuit merge, before arriving at the cathode entrance. The oxygen passing through the cathode gas diffusion layer to the catalytic layer is combined with hydrogen, and the oxidation reaction occurs under the action of the catalyst to generate water and heat, which are then removed [1,2].

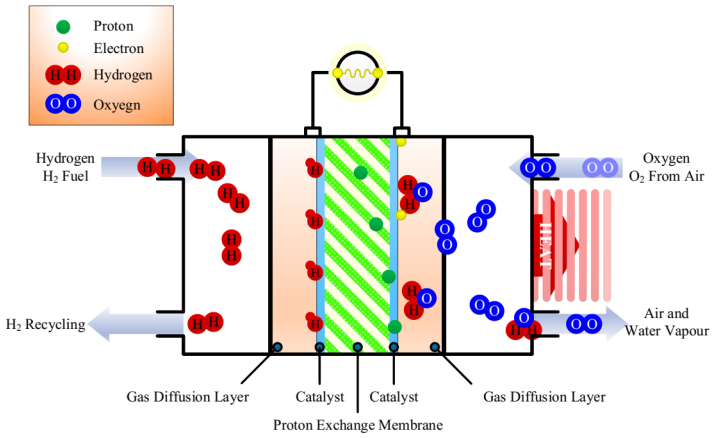

Figure 1. Chemical reaction in proton exchange membrane fuel cell (PEMFC).

2.1. The Dynamic Control Model of the Voltage

In the PEMFC, the pressure of hydrogen is subject to the influence of the inlet flux of hydrogen, the outlet flux of hydrogen, and the flux of hydrogen being consumed by the chemical reaction. According to the principle of conservation of matter, the ideal gas state can be expressed as follows:

$$\frac{V_a}{RT}\frac{dP_{H_2}}{dt} = m_{H_2,i} - K_a\left(P_{H_2} - P_{H_2,B}\right) - \frac{0.5Ni}{F} \quad (1)$$

where V_a is total volume of flow field in the anode, R is the gas constant, T is the working temperature, P_{H2} is the partial pressure of hydrogen, $m_{H2,i}$ is the inflow flux of hydrogen, K_a is the flux coefficient in the anode, $P_{H2,B}$ is the hydrogen removal pressure, N is the number of single fuel cells, F is the Faraday constant, and i is the load current.

Likewise, the oxygen pressure is influenced by the inlet flux of air, the outlet flux of air and the flux of air consumed by the chemical reaction. According to the continuity principle, the masses of the two kinds of components in the cathode, oxygen and nitrogen,

are balanced. Based on the conservation of matter, the continuity equation for oxygen and nitrogen can be expressed as follows:

$$\frac{dm_{O_2}}{dt} = W_{O_2,in} - W_{O_2,\,out} - W_{O_2,\,ret} \tag{2}$$

$$\frac{\mathrm{d}m_{\mathrm{N}_2}}{\mathrm{d}t} = W_{\mathrm{N}_2,in} - W_{\mathrm{N}_2,\,out} \tag{3}$$

The OER is shown as follows:

$$\lambda_{O_2} = \frac{W_{O_2,in}}{W_{O_2,ret}} \tag{4}$$

where m_{O_2} and m_{N_2} are the masses of oxygen and nitrogen respectively; $W_{O_{2,in}}$ and $W_{N_{2,in}}$ represent the flows of oxygen and nitrogen entering the stack; $W_{O_{2,out}}$ and $W_{N_{2,out}}$ are the flows of oxygen and nitrogen flowing out of stack; and, $W_{O_{2,ret}}$ is the flow of oxygen consumed due to reaction in the stack.

The oxygen pressure characteristic equation can be obtained as follows:

$$\frac{V_{\mathrm{c}}}{RT}\frac{\mathrm{d}P_{\mathrm{O}_2}}{\mathrm{d}t} = m_{\mathrm{O}_2,\mathrm{i}} - K_{\mathrm{c}}\left(P_{\mathrm{O}_2} - P_{\mathrm{O}_2,\mathrm{B}}\right) - \frac{0.2Ni}{F} \tag{5}$$

where V_{c} is the total volume of flow field in the anode, P_{O_2} is the partial pressure of hydrogen, K_{c} is the flux coefficient in the anode, and $P_{\mathrm{O}_2,\mathrm{B}}$ is the hydrogen removal pressure.

The thermodynamic electromotive force is expressed as follows:

$$E_{\mathrm{Nernst}} = 1.229 - 0.00085(T - 298.15) + 0.000043T\left(\ln P_{\mathrm{H}_2} + \frac{1}{2}\ln P_{\mathrm{O}_2}\right) \tag{6}$$

where E_{Nernst} is the thermodynamic electromotive force. The ohmic overpotential is as follows:

$$V_{\mathrm{ohmic}} = -i(R_{\mathrm{M}} + R_{\mathrm{c}}) \tag{7}$$

where V_{ohmic} is the ohmic overpotential, R_{M} is the equivalent membrane impedance of the proton membrane, and R is the impedance that prevents the proton from passing through the proton membrane. There exists an electrical double layer in the PEMFC [1,2]. If equivalent capacitors C are connected to both ends of the polarization resistor in parallel, the PEMFC will exhibit excellent dynamic properties. The differential equation of a single fuel cell is as follows:

$$\frac{\mathrm{d}v_{\mathrm{d}}}{\mathrm{d}t} = \frac{i}{C} - \frac{v_{\mathrm{d}}}{q} \tag{8}$$

where v_{d} is the total polarization overvoltage.

Having taken into account the thermodynamic properties, mass transfer and dynamic properties, the output voltage of the PEMFC can be expressed as follows:

$$V_{\mathrm{cell}} = E_{\mathrm{Nernst}} - V_{\mathrm{ohmic}} - v_d \tag{9}$$

2.2. Supply Pipe

Taking into account the principles of conservation of mass and conservation of energy, the supply pipe in the cathode can be represented by the following equations:

$$\frac{\mathrm{d}m_{\mathrm{sm}}}{\mathrm{d}t} = W_{\mathrm{cp}} - W_{\mathrm{sm}} \tag{10}$$

$$\frac{\mathrm{d}p_{\mathrm{sm}}}{\mathrm{d}t} = \frac{\gamma R_a}{V_{\mathrm{sm}}}\left(W_{\mathrm{cp}}T_{\mathrm{cp}} - W_{\mathrm{sm}}T_{\mathrm{sm}}\right) \tag{11}$$

where m_{sm} is the mass of air in the supply pipe, p_{sm} is the pressure in the supply pipe, γ is the heat ratio coefficient of air, R_a is the gas constant of air, V_{sm} is the supply pipe volume, T_{cp} is the temperature at which the compressor presses in air, T_{sm} is the temperature of the air in the supply pipe, W_{cp} is the flux through the compressor, and W_{sm} is the mass flux through the supply pipe.

For the supply pipe, the flow rate of the gas flowing into the pipe is equal to the flow rate of the air compressor W_{cp}, and the flow rate of the gas flowing out is $W_{sm,\,out}$. Since the pressure difference between the supply pipe and the cathode is relatively small, the flow rate of the gas flowing out is expressed as follows:

$$W_{sm,\mathrm{out}} = k_{sm,\mathrm{out}}(p_{\mathrm{sm}} - p_{\mathrm{ca}}) \tag{12}$$

where $k_{sm,out}$ is the outlet flow constant of the supply pipe, and P_{ca} is the cathode gas pressure.

2.3. Return Pipe

The change of temperature needs to be considered in the design of the return pipe. The gas temperature T_m in the return pipe is equal to the temperature of the gas leaving the cathode. Pressure p_{rm} of return pipe can be obtained based on the l conservation of mass and the ideal gas law, i.e.,

$$\frac{\mathrm{d}p_{\mathrm{rm}}}{\mathrm{d}t} = \frac{R_a T_{\mathrm{rm}}}{V_{\mathrm{m}}}(W_{\mathrm{ca}} - W_{\mathrm{rm}}) \tag{13}$$

where W_{ca} is the air flux in the stack cathode, W_{rm} is the air flux at the outlet of the return pipe, and V_{rm} is the volume of the return pipe.

Since the pressure drop between the return manifold and the atmospheric is relatively large, the equations of return manifold exit flow are as follows:

$$W_{rm,out} = \frac{C_{D,rm}A_{T,rm}p_{rm}}{\sqrt{RT_{rm}}}\left(\frac{p_{atm}}{p_{rm}}\right)^{\frac{1}{\gamma}}\left\{\frac{2\gamma}{\gamma-1}\left[1-\left(\frac{p_{atm}}{p_{rm}}\right)^{\frac{\gamma-1}{\gamma}}\right]\right\}^{\frac{1}{2}} \quad \text{for} \quad \frac{p_{atm}}{p_{rm}} > \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{14}$$

and

$$W_{rm,out} = \frac{C_{D,rm}A_{T,rm}p_{rm}}{\sqrt{RT_{rm}}}\gamma^{\frac{1}{2}}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \quad \text{for} \quad \frac{p_{atm}}{p_{rm}} \le \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{15}$$

where $A_{T,rm}$ is the return manifold throttle area, $C_{D,rm}$ is the return manifold throttle discharge coefficient, p_{rm} is the return manifold pressure, and T_{sm} is the return manifold temperature, p_{atm} is atmospheric pressure.

2.4. Air Compressor

The dynamic properties of the compressor can be described using a rotation model [1], i.e.,

$$\begin{cases} J_{\mathrm{cp}}\frac{\mathrm{d}\omega_{\mathrm{cp}}}{\mathrm{d}t} = \tau_{\mathrm{cm}} - \tau_{\mathrm{cp}} \\ \tau_{\mathrm{cm}} = \eta_{\mathrm{cm}}\frac{k_{\mathrm{t}}}{R_{\mathrm{cm}}}(v_{\mathrm{cm}} - k_{\mathrm{v}}\omega) \\ \tau_{\mathrm{cp}} = W_{\mathrm{cp}}\frac{C_{\mathrm{p}}T_{\mathrm{atm}}}{\omega\eta_{\mathrm{cp}}}\left[\left(\frac{p_{\mathrm{sm}}}{p_{\mathrm{atm}}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right] \end{cases} \tag{16}$$

where J_{cp} is the rotational inertia of the compressor; ω_{cp} is the speed of the compressor; τ_{cm} is the motor torque of the compressor; τ_{cp} is the load torque of the compressor; k_{t}, R_{cm}, and k_{v} are the motor constants; η_{cm} is the mechanical efficiency of the motor; v_{cm} is the control voltage of the motor; C_{p} is the specific heat capacity of air; η_{cp} represents the efficiency of the compressor; and p_{atm} and T_{atm} represent the atmospheric pressure and temperature, respectively.

2.5. The Control Principle of Gas Supply System

Research has indicated that there is an important correlation between the output voltage of the system and the flux of reactant gas. Since the output voltage of PEMFC is mainly affected by hydrogen flux and air flux, during modeling, both air flux and hydrogen flux are controlled by an intelligent controller based on EILMMA-DDPG. In this paper, a dynamic model of PEMFC is set up. The control objective is to keep output voltage and OER of PEMFC to stabilize at the optimal value v_{st}* and W_{O2}* by controlling the voltage of the motor for air compressor and the hydrogen flux. The control framework is shown in Figure 2.

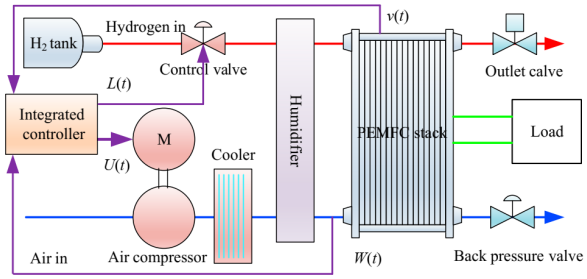

Figure 2. The control framework of integrated controller.

3. ECMTD-DDPG Framework

3.1. Deep Reinforcement Learning

Deep reinforcement learning (DRL) is a kind of machine learning that is not model-based. This method aims at maximizing the accumulation of long-term rewards and allows agents to continuously learn the optimal actions in different states. In deep reinforcement learning, there are two types of commonly used algorithms. One type is value-based DRL, which includes deep Q learning (DQN) and deep double Q learning (DDQN). This type of algorithm is generally used to deal with problems with discrete action spaces. The advantage of value-based DRL is that convergence is fast and it does not fall easily into a local optimal solution. The other is policy-based DRL, which includes DDPG and TD3, etc. Policy-based DRL is often used to deal with problems with continuous action space, and is therefore a commonly used algorithm; however, this algorithm can fall easily into a local optimal solution.

In recent years, deep reinforcement learning algorithms have continued to develop, and distributed deep reinforcement learning using computer parallel computing capabilities have been proposed. The chief characteristic of distributed reinforcement learning is to use each agent in the system as the main body of learning. These agents learn the response policy to the environment and the mutual cooperation policy. In addition, they use the CPU or GPU of multiple computers to realize better computing processing power, so as to improve the performance of the algorithm. Moreover, they usually adopt the model of centralized training and decentralized execution. All agents in the algorithm correspond to a leader for centralized training such as the APEX-DDPG algorithm [22].

Edge cloud collaboration is a form of collaborative computing and is a relatively mature technology model.

An edge-oriented edge cloud collaboration approach is used in this paper, in that the Cloud is only responsible for initial training work, and the model is downloaded to the edge after the training is completed. Computing tasks are performed at the edge. The edge cloud collaboration framework is introduced in distributed deep reinforcement learning. In the proposed ECMTD-DDPG framework, the leader in distributed reinforcement learning is the cloud. It contains a variety of neural networks and is responsible for the initial centralized training model. The edge is the edge explorer, and each edge explorer contains a neural network. These networks are responsible for collecting data to assist the cloud for training while performing their own computing tasks in the online application process.

During training, the role of DRL is to use a large amount of computer computing power at the edge to continuously search for optimization and exploration, and to deliver the explored information to the cloud for centralized training. Following the initial training, the DRL agent can directly make decisions through the trained model. Through the combination of such technologies, the integration of distributed reinforcement learning and edge-cloud collaboration technology is realized. Related common algorithms and edge-cloud collaboration frameworks are described below.

3.2. Common Policy Gradient Algorithms

3.2.1. DDPG

In [14], 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 $\pi_{\phi}{}^{j}(s)$ 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 $J(\phi) = E_{s_i \sim p_{\pi}, a_i \sim \pi}[R_0]$. It employs the loss function to update the critics as Formula (17) and employs the policy gradient to update the actor policy as Formula (18)

$$L = \frac{1}{N}\sum_{i}\left(y_i - Q\left(s_i, a_i \middle| \theta^{Q}\right)\right)^2 \tag{17}$$

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i} \nabla_a Q\left(s, a \middle| \theta^{Q}\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i} \tag{18}$$

where the $\mu(s|\theta^{\mu})$ is the policy which has the parameter of θ^{μ}, L is the loss function, y_i is the target values.

3.2.2. SAC

SAC, which is developed based on DDPG, is a stochastic policy gradient algorithm. Its typical feature is that entropy regularization is introduced to improve the randomness of action selection in the training process. For a policy of SAC, the greater the entropy, the higher the randomness of action selection. The expression of entropy is as follows:

$$H(\pi(\cdot|s_t)) = -\sum_{t=1}^{\infty} \pi(\cdot|s_t) \log \pi(\cdot|s_t) \tag{19}$$

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

$$\pi^* = \text{argmax}_{\pi} E_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot|s_t)))\right] \tag{20}$$

SAC algorithm adopts five neural networks, namely, policy network, two Q networks, and two V networks. Among them, the policy network outputs actions, the state value network $V(s)$ outputs the value of current state, the target state value network outputs the value of next state, and two action value networks $Q_1(s,a)$, $Q_2(s,a)$ output the value of action selection.

3.3. ECMTD-DDPG

3.3.1. Clipped Multi-Q Learning

Inspired by double deep Q-learning (DDQN) [12], ECMTD-DDPG employs the current actor network to select the optimal action and uses the target critic network to evaluate the policy. This process is encapsulated by Formula (21):

$$y_t = r(s_t, a_t) + \gamma Q_{\theta'}\left(s_{t+1}, \pi_{\phi}(s_{t+1})\right) \tag{21}$$

In DDPG, the target actor network and the target critic network adopt the soft update mechanism [14], which establishes similarity between the current network and the target network. This, however, makes it hard to effectively separate the action selection from the policy evaluation. Therefore, the clipped double Q-learning method is adopted to calculate the target value, as expressed in Formula (22):

$$y_t^1 = r(s_t, a_t) + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s_{t+1}, \pi_{\phi_1}(s_{t+1})\right) \tag{22}$$

The Cloud leader in ECMTD-DDPG employs an independent actor network and three critic networks. The actor network π_{ϕ_1} is updated according to the first critic network, and the target values ${y_t}^2$ and ${y_t}^3$ of other critic networks are equal to ${y_t}^1$.

3.3.2. Policy Delayed Updating

The Cloud leader of ECMTD-DDPG updates the actor network after the critic network is updated for d times, so as to ensure that the actor network can be updated with low Q value error and thus improve the updating efficiency of the actor network.

3.3.3. Smooth Regularization of Target Policy

ECMTD-DDPG introduces a regularization method to reduce the variance of the target, as expressed by Formula (23):

$$y_t = r(s_t, a_t) + \mathrm{E}_\varepsilon\left[Q_{\theta'}\left(s_{t+1}, \pi_{\phi'}(s_{t+1}) + \varepsilon\right)\right] \tag{23}$$

At the same time, by adding a random noise to the target policy and averaging on mini-batch, smooth regularization is realized:

$$y_t = r(s_t, a_t) + \gamma \min_{i=1,2} Q_{\theta_t}\left(s_{t+1}, \pi_{\phi'}(s_{t+1}) + \varepsilon\right) \tag{24}$$

$$\varepsilon \sim \mathrm{clip}(N(0, \sigma), -c, c) \tag{25}$$

3.3.4. Distributed Training Method Based on Edge-Cloud Collaborative Framework

(1) Edge computing and cloud computing

Cloud computing is a centralized service, which has a large number of computing resources. All data is transmitted to the cloud computing center for processing, but it is difficult to ensure the timeliness and computing speed because of mass data.

Edge computing is a distributed computing architecture, which decomposes large-scale services that are originally handled by central nodes and distribute them to edge nodes. The purpose is to fully mobilize the computing power of network terminals, deal with computing problems more quickly and accurately in real time and reduce the cloud data communication.

Edge computing and cloud computing supplement each other. Specifically, cloud computing is a unified leader responsible for big data analysis of long-term data, while edge computing focuses on real-time processing and execution.

(2) Computing architecture

By referring to the above computing architecture, ECMTD-DDPG algorithm adopts a distributed reinforcement learning and training framework based on edge-cloud collaboration. There are two roles in the algorithm. The first role is the edge explorer based on the multi-exploration principle, which corresponds to the edge computing terminal in the edge-cloud collaborative framework. It acts to realize distributed exploration of the environment, so as to obtain more abundant training samples. The edge explorer includes 24 DDPG-edge explorers, 8 SAC-edge explorers, and 8 control algorithm edge explorers (CA-edge explorers) corresponding to different CPUs. They have fast computing speed and can realize real-time sampling.

Another role in the algorithm is a cloud leader, which includes two public experience pools, among which the cloud leader includes three critics and one actor, corresponding to one GPU. The cloud leader deals with a large amount of data in the public experience pools by collecting mini-batch and obtains the optimal control strategy by training. The parameter updating is slow without updating the neural network in real time.

In this paper, DDPG-edge explorer architecture includes only one actor network, and each has its own network model as well as environment. Different DDPG-edge explorer adopts different exploration principles, including greedy strategy, Gaussian noise, and Ornstein–Uhlenbeck (OU) noise. In different DDPG-edge explorers, the actor network adopts different exploration policies. The exploration policy of actor network in 8 edge-explorers is set as greedy strategy, which is named ε-DDPG-edge-explorers, that is, any action is selected in the action space with a certain probability.

$$a_{\varepsilon}^{l} = \begin{cases} \pi_{\theta}^{l}(s) & \text{With } \varepsilon \text{ probability} \\ a_{\text{rand}}^{l} & \text{With } 1-\varepsilon \text{ probability} \end{cases} \tag{26}$$

The optimization policy of the actor network in 8 DDPG-edge explorers is set as OU noise, which is named OU-DDPG -edge-explorer. Different OU-DDPG-edge-explorers adopt random OU noise with different variance.

$$a_{OU}^{j} = \pi_{\theta}^{j}(s) + \mathcal{N}_{OU}^{j} \tag{27}$$

The optimization policy of the actor network in 8 DDPG-edge explorers is set as Gaussian noise, which is named Gaussian-DDPG-edge explorer. Different Gaussian-DDPG-edge explorers adopt random Gaussian noise with different variance.

$$a_{\text{Gaussian}}^{m} = \pi_{\theta}^{\text{m}}(s) + \mathcal{N}_{Gaussian}^{m} \tag{28}$$

By adopting the exploration schemes based on the above different principles, the randomness and diversity of the explored samples can be increased.

SAC-edge explorer contains a complete SAC agent architecture, different network models, and environments, as described in Section 3.2.2. SAC-edge explorer adopts its own policy for exploration in different environments and puts the samples obtained from exploration into its own experience pool and the public experience pools in cloud leader. In addition, SAC-edge explorer regularly draws samples from its own experience pool for training and updates its own parameters according to the Formula (18).

CA-explorer has a control algorithm with different principles. By interacting with different environments, it can generate corresponding demonstration samples and guide cloud leader to learn.

A variety of edge explorers explore the environment in parallel. First, DDPG-edge explorer and SAC-edge explorer generates training samples $e_i^{DDPG} = ({s_t}^{i\text{-}DDPG}, {a_t}^{i\text{-}DDPG}, {r_t}^{i\text{-}DDPG}, {s_{t+1}}^{i\text{-}DDPG})$ and $e_i^{SAC} = ({s_t}^{i\text{-}SAC}, {a_t}^{i\text{-}SAC}, {r_t}^{i\text{-}SAC}, {s_{t+1}}^{i\text{-}SAC})$ based on their own environments, and CA-explorer generates the demonstration sample $e_i^{CA} = ({s_t}^{i\text{-}CA}, {a_t}^{i\text{-}CA}, {r_t}^{i\text{-}CA}, {s_{t+1}}^{i\text{-}CA})$ based on the environment. The conversion experience is added to two public experience pools according to the standard. Then, cloud leader draws mini-batch from the public experience pool according to the classification experience replay mechanism and keeps learning. Finally, the actor network in DDPG-edge explorer regularly updates its network parameters according to the latest network of actor from cloud leader.

3.3.5. Guided Exploration Policy-Based Imitation Learning

In imitation learning, agents utilize human expert demonstration samples to improve their training efficiency. In order to increase the effect of human experts in the demonstration, the imitation learning proposed in this paper is to imitate the control strategy of which the parameters have been artificially regulated and considered to have optimized collaboration. The control output and data received by the controller in CA-edge explorer

will be converted to the demonstration sample $e_i^{CA} = (s_t^{i\text{-}CA}, a_t^{i\text{-}CA}, r_t^{i\text{-}CA}, s_{t+1}^{i\text{-}CA})$ which is stored in the public experience pool.

As shown in Figure 3, different CA-edge explorers adopt different control algorithms, including PID, fuzzy PID, PSO-optimized fuzzy PID (artificially setting air flow controller parameters and PSO-optimized hydrogen flow control parameters), fractional-order PID (FOPID), adaptive PID, fuzzy adaptive PID and fuzzy control, which are respectively named PID-CA-explorer, FPID-CA-explorer, PSO-FPID-CA-explorer, FOPID-CA-explorer, APID-CA-explorer, FAPID-CA-explorer, and FC-CA-explorer. These CA-explorers only collect samples and do not need to receive any information from the cloud leader. They provide diversified learning samples for the cloud leader. The specific algorithm flow is shown in Figure 4.

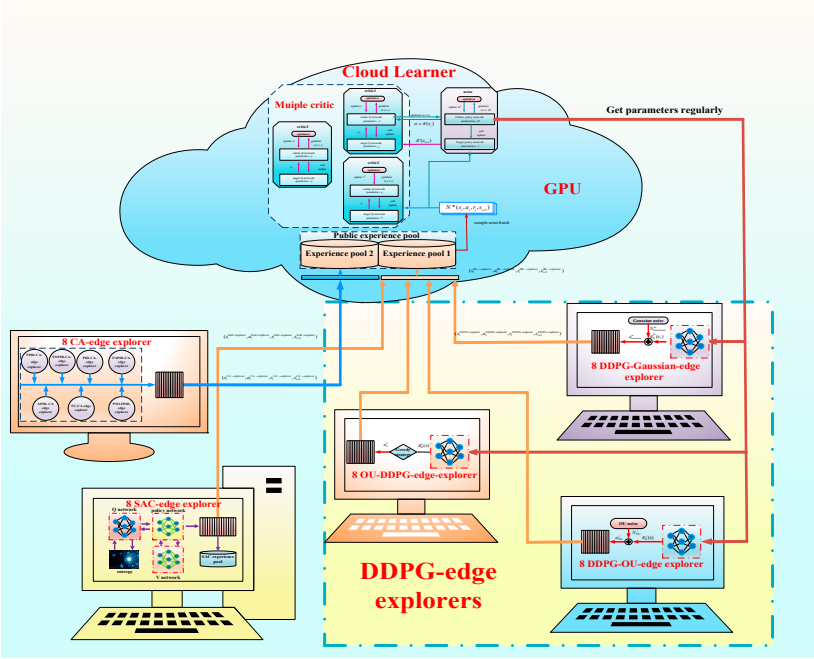

Figure 3. Distributed training framework of ECMTD-DDPG.

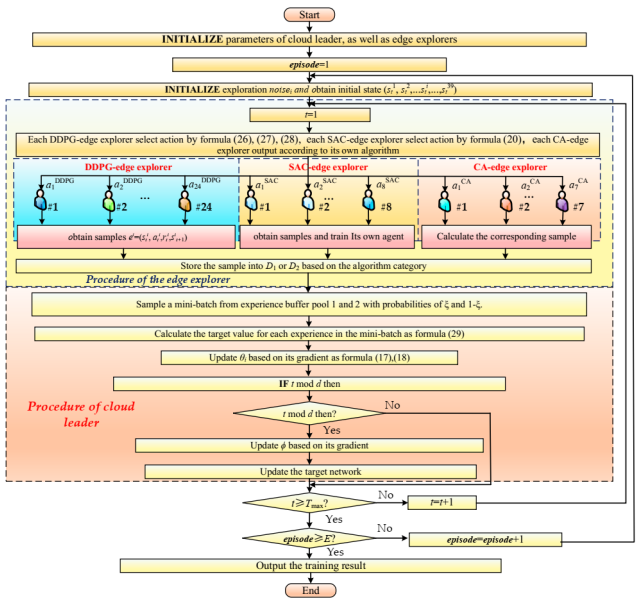

Figure 4. ECMTD-DDPG algorithm flow chart.

3.3.6. Classified Experience Replay

In cloud leader, two independent public experience pools are employed for storing samples. When the network model is initialized, all samples in these two public experience pools are cleared. During training, samples collected by DDPG-edge explorer and SAC-edge explorer are put into experience pool 1, and samples collected by CA-edge-explorer are stored in experience pool 2. In the pre-learning, with respect to pool 1, the probability of getting n_{ξ} samples is ξ. In pool 2, the probability of obtaining $n_{(1-\xi)}$ samples is (1-ξ).

In order to enable agents to get more demonstration samples generated by CA-explorer in the early stage of training and learn more samples collected by DDPG-edge explorer as well as SAC-edge explorer in the later stage, the probability increases gradually with the increase of episodes, as shown in Formula (29).

$$\xi = \begin{cases} 0.7 & \text{episodes} < 1000 \\ 0.8 & 1000 < \text{episodes} < 2000 \\ 0.9 & 2000 < \text{episodes} < 3000 \\ 1 & \text{episodes} > 3000 \end{cases} \tag{29}$$

4. Design of ECMTD-DDPG Integrated Controller

The integrated controller of the gas supply system described in this paper mainly has two control objectives: (1) to ensure proper OER or air flux; (2) to ensure stable output voltage under the condition of proper OER. The integrated controller indirectly controls the OER and output voltage of PEMFC by simultaneously controlling the motor voltage of air compressor and the flow of hydrogen according to the control error of air flow and output voltage in the system as well as the related states. The control interval is 0.01 s. The specific control framework, state space, and action space are shown in Figure 5.

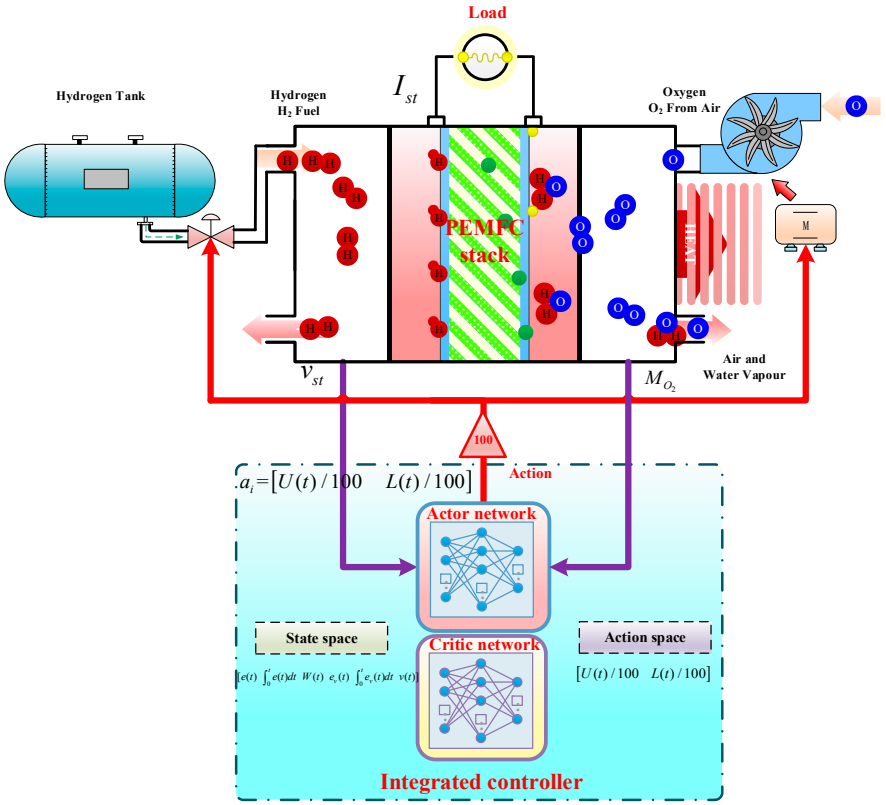

Figure 5. Integrated control of PEMFC gas supply system based on ECMTD-DDPG algorithm.

4.1. Action Space

The action space is set as shown in Formula (30).

$$\begin{cases} a = [U(t)/100L(t)/100] \\ 0 \le U(t) \le U_{\max} \\ 0 \le L(t) \le L_{\max} \end{cases} \tag{30}$$

where the t is the discrete time, $U(t)$ is the motor voltage of air compressor; $U_{\max}$ is the upper limit of the motor voltage of the air compressor; $L(t)$ is the hydrogen flow; and $L_{\max}$ is the upper limit of hydrogen flow.

4.2. State Space

State refers to the control error between air flow and ideal air flow of the input controller ($e_{o2}(t)$), ($W(t)$), ($e_v(t)$), its integral to t and output voltage $v(t)$, state space is shown in Formula (31).

$$[e_{O_2}(t) \int_0^t e_{O_2} dt W(t) e_v(t) \int_0^t e_v dt v(t)] \tag{31}$$

where the $e_{O_2}(t)$ refers to the control error between air flow and ideal air flow of the input controller, $\int_0^t e_{O_2}(t)$ refer to the $e_{O_2}(t)$ integral to t, $W(t)$ refer to the air flow. $e_v(t)$ refer to the control error between output voltage and output voltage reference value of input controller. $\int_0^t e_v(t)$ refer to the $e_v(t)$ integral to t, $v(t)$ refers to the output voltage of PEMFC.

4.3. Selection of Reward Function

The reward function is expressed as follows:

$$r(t) = -\left[\mu_1 e_{O_2}^2(t) + \mu_2 e_v^2(t)\right] + \beta \tag{32}$$

$$\beta = \begin{cases} 0 & e_{O_2}^2(t) > 0.01 \text{and} e_v^2(t) > 0.09 \\ 2.2 & e_{O_2}^2(t) \le 0.01 \text{and} e_v^2(t) \le 0.09 \\ 1.1 & otherwise \end{cases} \tag{33}$$

where the β means the control reward item. According to the state of $e_{o2}(t)$ and $e_v(t)$, agents are given a proper positive reward.

5. Simulation

The parameters of the PEMFC in the simulation are from the reference [24] which is based on the real experimental data of FORD P2000 vehicle with 75 kW PEMFC [25–27]. The working temperature of PEMFC is considered to be approximately constant which is 353 K. In addition, the gases in the PEMFC are fully humidified. The simulation software package used is MATALB/Simulink version 9.9.0 (R2020b). The pre-learning parameters are shown in Table 1.

Table 1. Parameter setting.

Parameter	Value
Learning rate of critic	0.001
Learning rate of actor	0.001
Discount factor	0.99
Number of CA-edge-explorers	7
Number of DDPG-edge-explorers	24
variance of mth Gaussian-RL-explorer	$0.06 + 0.007 \times m$
variance of jth OU-RL-explorer	$0.03 + 0.005 \times j$
Probability ε of lth ε-RL-explorer	0.9
Interval of policy network update	2
Sizes of experience pools 1 and 2	1,000,000
Target action noise variance	0.005

5.1. Pre-Learning

To ensure the randomness and diversity of samples obtained, the step load current with different amplitudes (0–200 A) is added for training, and the training interval of each episode is 10 s. The training diagram is exhibited in Figure 6.

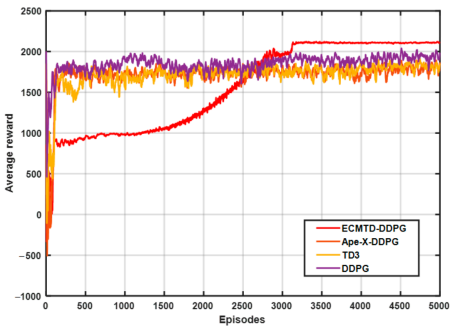

Figure 6. Training diagram.

In Figure 6, the curve represents the average value of rewards in the episode corresponding to algorithms. Among them, the learning speed of Ape-X-DDPG, TD3, and DDPG algorithms is slow, and the learning process has significant fluctuations. In contrast, ECMTD-DDPG has a more stable learning process and a higher final average reward value, indicating that ECMTD-DDPG with edge exploration policy can obtain the optimal solution with a higher value. Meanwhile, due to the distributed exploration method, a large variety of explorers simultaneously explore the optimal solution, making ECMTD-DDPG converge to the optimal solution earlier. This demonstrates that the distributed training method can improve the quality of the trained solution.

5.2. Online Application

Two conditions (step load and stochastic load) are used in simulation to verify the effectiveness of the method. Besides, the reinforcement learning (RL) integrated controller (such as Ape-X-DDPG [21], TD3 [20], and DDPG [14] controller), the conventional controller (such as PSO optimized fuzzy-PID (PSO-fuzzy-PID)), fuzzy PID controller (fuzzy-PID), PSO-optimized PID(PSO-PID) controller, and PID controller are introduced as comparisons.

5.2.1. Step Load Condition

(1) As presented in Figure 7a–d, the ECMTD-DDPG controller can adapt to a rapid change of OER and output voltage with small overshoot and short dynamic response time compared with all comparative algorithms. Besides, it has fast response, small overshoot, short stabilization time, and does not cause oxygen saturation or starvation in the air flow control. In contrast, the control performance of conventional control algorithms is lower, and the stabilization time of OER is longer.

(2) According to the result of the conventional controller in Table 2, their OER steady time is up to 1.09 times of ECMTD-DDPG, and their output voltage steady time can reach 0.224 s, which is more than 9.6667 times of ECMTD-DDPG. The output voltage overshoot of conventional controllers is all higher than 0.06%, which is more than 20 times that of the ECMTD-DDPG and up to 148.8 times that of ECMTD-DDPG. This is because the serious discordance between controllers leads to the slow response and oscillation in the process of disturbance and easily causes severe instability of the stack voltage.

(3) Furthermore, according to the result of the RL controller in Table 2, the OER steady time of some RL controllers exceeds 4 s, and their output voltage steady time is up to 0.5 s, which is more than 16.7 times of ECMTD-DDPG. Moreover, the output voltage overshoot of other RL controllers are all greater than 0.096%, which is more than 32 times that of the ECMTD-DDPG and up to101 times that of ECMTD-DDPG. Consequently, the ECMTD-DDPG controller has a smaller overshoot and faster response rate compared with

other RL controllers. Particularly, the control performance of some RL controllers, such as the DDPG controller, is worse than conventional controllers since the DDPG controller does not have robustness to adapt to every load condition. However, the ECMTD-DDPG controller can have better control performance than all the other controllers.

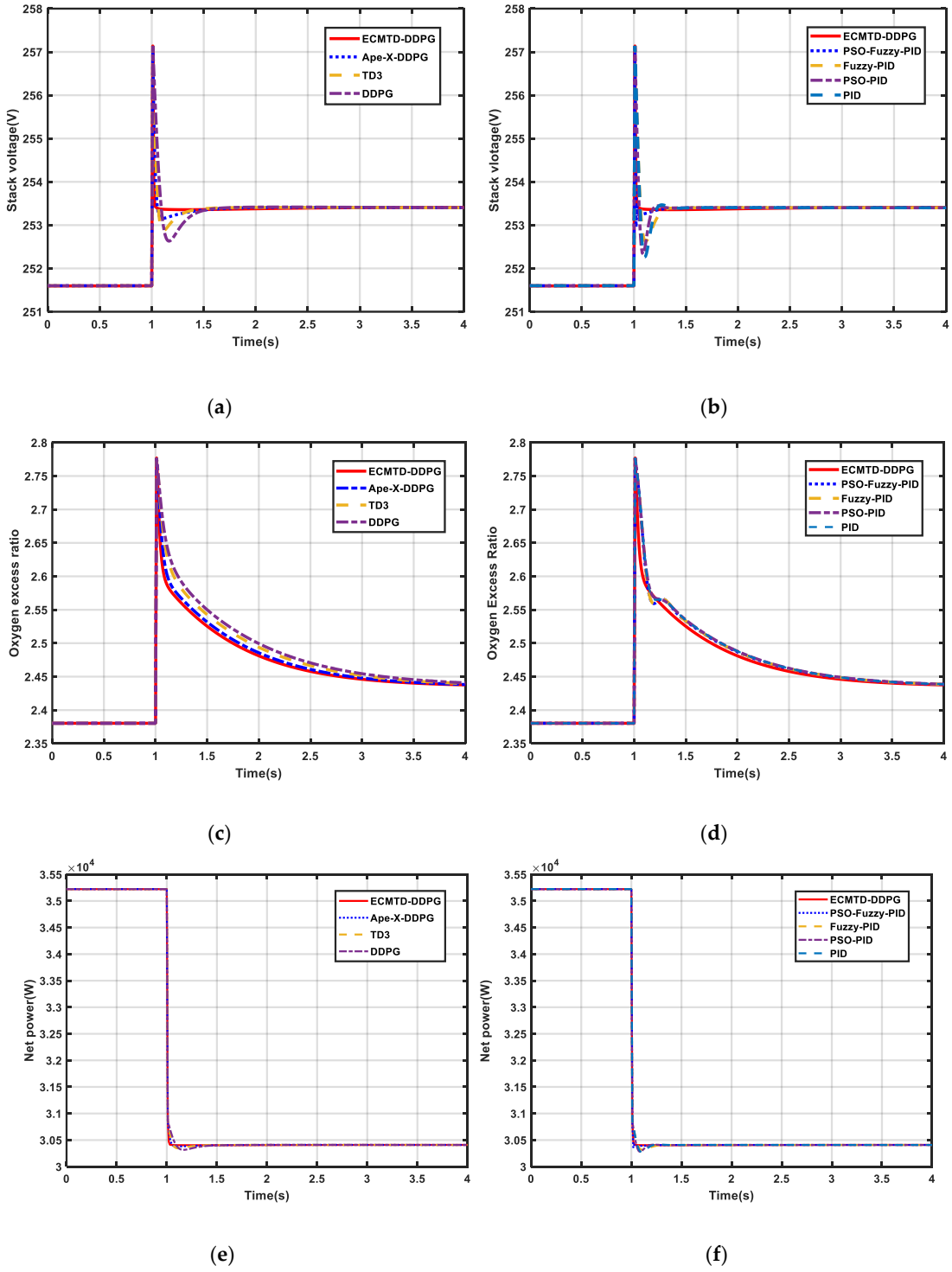

Figure 7. Result of PEMFC under step load condition (**a**) Stack voltage of PEMFC by RL algorithm (**b**) Stack voltage of PEMFC by conventional algorithm (**c**) OER of PEMFC by RL algorithm (**d**) OER of PEMFC by conventional algorithm (**e**) Net power of PEMFC by RL algorithm (**f**) Net power of PEMFC by conventional algorithm.

Table 2. Response parameters of PEMFC gas supply system.

Type	Parameters	EILMMA-DDPG	APE-X-DDPG	TD3	DDPG	PSO-Fuzzy-PID	Fuzzy-PID	PSO-PID	PID
OER	Rise time $T_r/10^{-3}$ s	1.02	1.09	1.12	1.18	1.29	1.22	1.5	1.60
	Stable time T_s/s	3.52	3.73	>4	>4	3.79	3.80	3.84	3.83
	Overshoot σ/%	14.0068	14.0068	14.0068	14.0069	14.0068	14.0068	14.0069	14.0068
Output voltage	Rise time $T_r/10^{-3}$ s	31	90	110	170	40	110	80	100
	Stable time T_s/s	0.03	0.46	0.39	0.50	0.21	0.29	0.18	0.224
	Overshoot σ/%	0.003	0.096	0.2035	0.3030	0.0669	0.3062	0.4210	0.4464

This is because the exploration policies of Ape-X-DDPG, TD3, and DDPG are too narrow, resulting in convergence to the local optimal solution. Meanwhile, ECMTD-DDPG has been improved to obtain a control policy with better performance. It can be observed by carefully analyzing that the ECMTD-DDPG controller significantly improves the efficiency of algorithm exploration by adopting the distributed edge-cloud collaborative training framework in the training to make full use of all the computing power of GPU and CPU. Additionally, there is no coordination problem between controllers in the control process.

Similarly, it can be observed from Figure 7e–f that the ECMTD-DDPG controller can also ensure the stability of net power in the course of change load. Therefore, the ECMTD-DDPG controller has a better control performance and stability under the condition of step load.

5.2.2. Stochastic Load Condition

Simulation is performed with stochastic load adding to the system in the simulation to verify the robustness and control performance of the ECMTD-DDPG controller. The simulation time is 50 s. The load current is shown in Figure A1 of Appendix A. The results are illustrated in Figure 8a–f.

As indicated in Figure 8a–d, the ECMTD-DDPG controller has extremely high adaptive ability and robustness. By replacing the conventional dual-controller structure with agent that can realize MIMO control, it can achieve the coordination between different controllers in the gas supply system and automatically make the optimal decision in the current state according to the state of PEMFC, realizing millisecond decision-making of control. Therefore, the proposed algorithm can still output proper OER under stochastic load condition and ensure stable control over the output voltage to achieve better control performance. Particularly, the DDPG controller exhibits completely different control performance under different load conditions because the DDPG controller does not have robustness; this may lead to great overshoot or longer steady time of OER and output voltage.

Under stochastic load condition, the ECMTD-DDPG controller can still realize stable regulation and accurate track of output voltage (Figure 8a–b). Thus, the change in output power is further stabilized (Figure 8e–f).

To sum up, ECMTD-DDPG controller has the characteristics of short response time and rapid response in gas supply control under stochastic load disturbance.

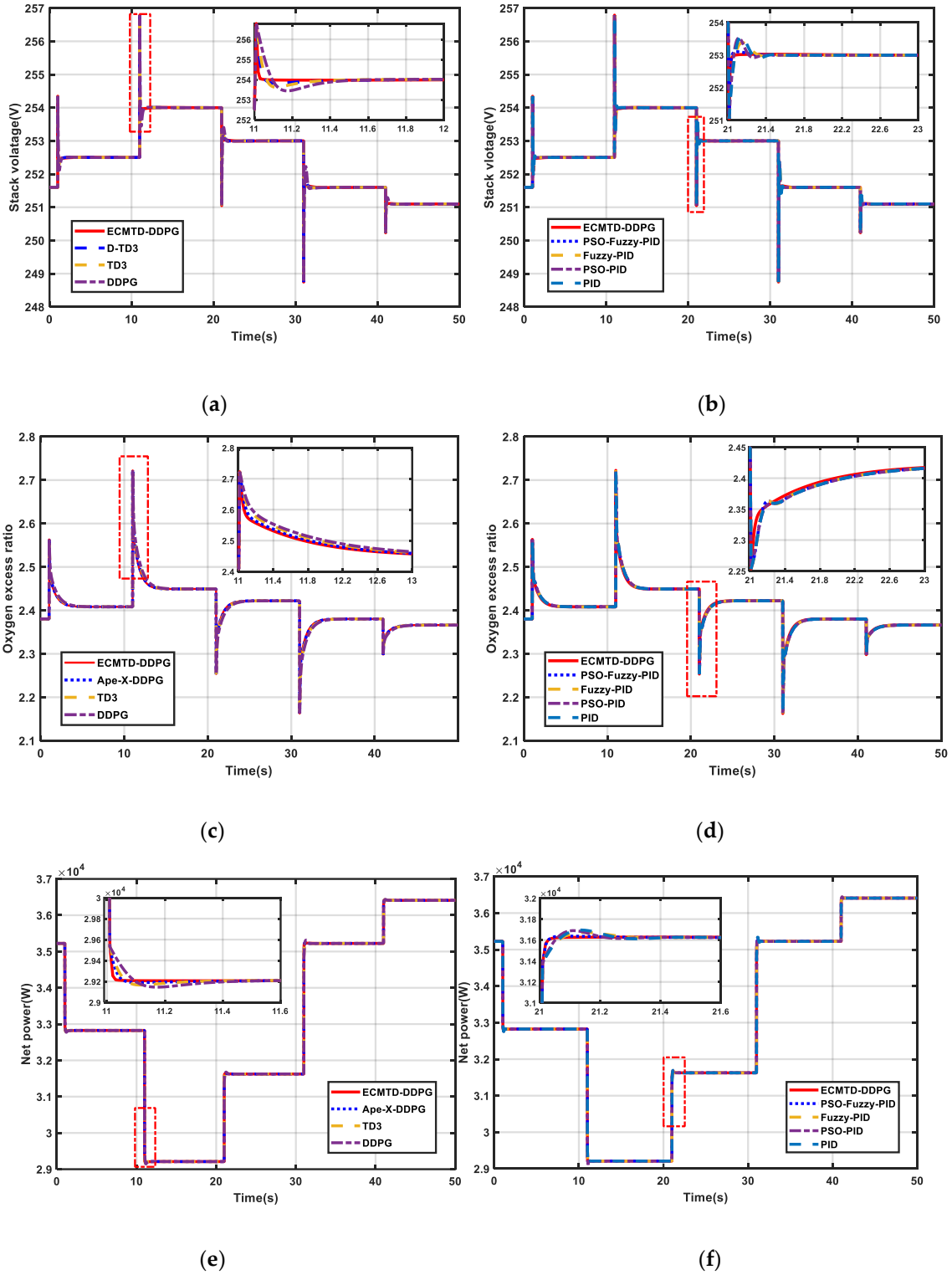

Figure 8. Simulation result of PEMFC under stochastic load (**a**) Stack voltage by RL algorithm (**b**) Stack voltage by conventional algorithm (**c**) OER by RL algorithm (**d**) OER by conventional algorithm (**e**) Net power by RL algorithm (**f**) Net power by conventional algorithm.

6. Conclusions

(1) In this paper, an integrated controller for PEMFC gas supply system control is presented. The original airflow controller and hydrogen flow controller are combined using

a deep reinforcement learning algorithm, realizing the coordination between controllers in the gas supply system.

(2) For this framework, the ECMTD-DDPG algorithm is proposed. In this algorithm, the edge exploration policy is adopted, suggesting that the edge explorers including DDPG algorithm, SAC algorithm, and conventional control algorithms are utilized to conduct distributed exploration in the environment, so as to improve the exploration efficiency. Moreover, the classified experience replay mechanism is introduced to improve training efficiency. Besides, clipped multi-Q learning, delay policy updating, and smooth regularization of target policy are combined with the cloud centralized training policy to address the overestimation of Q-value in DDPG. Finally, an adaptive reinforcement learning control algorithm with better global searching ability and training efficiency is obtained, ensuring stable output voltage and OER by realizing MIMO control in PEMFC.

(3) According to the simulation results, it is concluded that ECMTD-DDPG integrated controller can meet the real-time control requirements in PEMFC gas supply control system under different load conditions. It has small overshoot and can effectively avoid sudden change of output voltage, as well as problems of oxygen saturation and starvation. Thus, it can more efficiently control the output voltage of PEMFC.

Author Contributions: Methodology, J.L.; supervision, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant. 51777078).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors gratefully acknowledge the support of the National Natural Science Foundation of China.

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

Appendix A

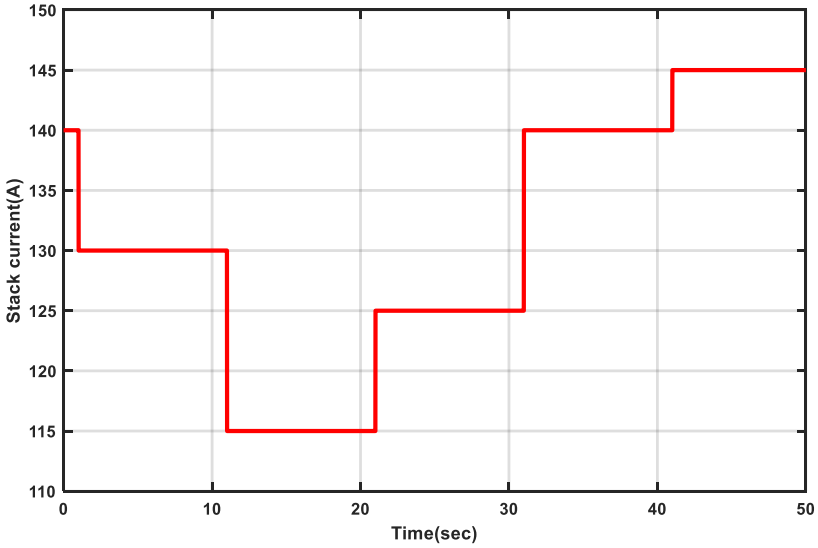

Figure A1. Stack current of PEMFC with Stochastic load condition.

References

1. Lee, P.; Han, S.; Hwang, S. Three-Dimensional Transport Modeling for Proton Exchange Membrane (PEM) Fuel Cell with Micro Parallel Flow Field. *Sensors* **2008**, *8*, 1475–1487. [CrossRef] [PubMed]
2. Feroldi, D.; Serra, M.; Riera, J. Performance improvement of a PEMFC system controlling the cathode outlet air flow. *J. Power Sources.* **2007**, *169*, 205–212. [CrossRef]
3. Pukrushpan, J.T.; Stefanopoulou, A.G.; Peng, H. *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis, and Feedback Design*; Springer: Berlin, Germany, 2004.
4. Talj, R.J.; Hissel, D.; Ortega, R.; Becherif, M.; Hilairet, M. Experimental Validation of a PEM Fuel-Cell Reduced-Order Model and a Moto-Compressor Higher Order Sliding-Mode Control. *IEEE Trans. Ind. Electron.* **2010**, *57*, 1906–1913. [CrossRef]
5. Na, W.K.; Gou, B. Feedback-linearization-based nonlinear control for PEM fuel cells. *IEEE Trans. Energy Convers.* **2008**, *23*, 179–190.
6. Sankar, K.; Thakre, N.; Singh, S.M.; Jana, A.K. Sliding mode observer based nonlinear control of a PEMFC integrated with a methanol reformer. *Energy* **2017**, *139*, 1126–1143. [CrossRef]
7. Wang, F.-C.; Yang, Y.-P.; Huang, C.-W.; Chang, H.-P.; Chen, H.-T. System identification and robust control of a portable proton exchange membrane full-cell system. *J. Power Sources* **2007**, *164*, 704–712. [CrossRef]
8. Wang, F.-C.; Chen, H.-T.; Yang, Y.-P.; Yen, J.-Y. Multivariable robust control of a proton exchange membrane fuel cell system. *J. Power Sources* **2008**, *177*, 393–403. [CrossRef]
9. He, J.; Choe, S.-Y.; Hong, C.-O. Analysis and control of a hybrid fuel delivery system for a polymer electrolyte membrane fuel cell. *J. Power Sources* **2008**, *185*, 973–984. [CrossRef]
10. Almeida, P.E.; Simoes, M.G. Neural Optimal Control of PEM-Fuel Cells with Parametric CMAC Networks. In Proceedings of the 38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, Salt Lake City, UT, USA, 12–16 October 2003; IEEE: New York, NY, USA, 2003; pp. 723–730.
11. Zhan, Y.; Zhu, J.; Guo, Y.; Wang, H. Performance Analysis and Improvement of a Proton Exchange Membrane Fuel Cell Using Comprehensive Intelligent Control. In Proceedings of the 2008 International Conference on Electrical Machines and Systems, Wuhan, China, 17–20 October 2008; IEEE: New York, NY, USA, 2008; pp. 2378–2383.
12. Qi, X. Rotor resistance and excitation inductance estimation of an induction motor using deep-Q-learning algorithm. *Eng. Appl. Artif. Intell.* **2018**, *72*, 67–79. [CrossRef]
13. Hu, Y.; Li, W.; Xu, K.; Zahid, T.; Qin, F.; Li, C. Energy management strategy for a hybrid electric vehicle based on deep reinforcement learning. *Appl. Sci.* **2018**, *8*, 187. [CrossRef]
14. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *Comput. Sci.* **2015**, *8*, A187.
15. Zhu, M.; Wang, X.; Wang, Y. Human-like autonomous car-following model with deep reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 348–368. [CrossRef]
16. Chen, P.; He, Z.; Chen, C.; Xu, J. Control strategy of speed servo systems based on deep reinforcement learning. *Algorithms* **2018**, *11*, 65. [CrossRef]
17. Xi, L.; Wu, J.; Xu, Y.; Sun, H. Automatic Generation Control Based on Multiple Neural Networks With Actor-Critic Strategy. *IEEE Trans. Neural. Netw. Learn. Syst.* **2020**. [CrossRef] [PubMed]
18. Duan, J.; Shi, D.; Diao, R.; Li, H.; Wang, Z.; Zhang, B.; Bian, D.; Yi, Z. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Trans. Power Syst.* **2019**, *35*, 814–817. [CrossRef]
19. Shi, H.; Sun, Y.; Li, G.; Wang, F.; Wang, D.; Li, J. Hierarchical Intermittent Motor Control With Deterministic Policy Gradient. *IEEE Access* **2019**, *7*, 41799–41810. [CrossRef]
20. Fujimoto, S.; Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In *PMLR: Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018*; Jennifer, D., Andreas, K., Eds.; International Machine Learning Society: Bellevue, WA, USA, 2018; Volume 80, pp. 1587–1596.
21. Horgan, D.; Quan, J.; Budden, D.; Barth-Maron, G.; Hessel, M.; Van Hasselt, H.; Silver, D. Distributed prioritized experience replay. *arXiv* **2018**, arXiv:1803.00933.
22. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
23. Lin, B.; Zhu, F.; Zhang, J.; Chen, J.; Chen, X.; Xiong, N.N.; Mauri, J.L. A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. *IEEE Trans. Industr. Inform.* **2019**, *15*, 4254–4265. [CrossRef]
24. Pukrushpan, J.T. *Modeling and Control of Fuel Cell Systems and Fuel Processors*; University of Michigan: Ann Arbor, MI, USA, 2003.
25. Adams, J.A.; Yang, W.-C.; Oglesby, K.A.; Osborne, K.D. The development of Ford's P2000 fuel cell vehicle. *SAE Trans.* **2000**, 1634–1645. [CrossRef]
26. Cunningham, J.M.; Hoffman, M.A.; Moore, R.M.; Friedman, D.J. Requirements for a flexible and realistic air supply model for incorporation into a fuel cell vehicle (FCV) system simulation. *SAE Trans.* **1999**, *108*, 3191–3196.
27. Nguyen, T.V.; White, R.E. A water and heat management model for proton-exchange-membrane fuel cells. *J. Electrochem. Soc.* **1993**, *140*, 2178. [CrossRef]