Hindawi
International Transactions on Electrical Energy Systems
Volume 2023, Article ID 4295384, 14 pages
https://doi.org/10.1155/2023/4295384

Research Article

Multilayer Deep Deterministic Policy Gradient for Static Safety and Stability Analysis of Novel Power Systems

Yun Long,[1] **Youfei Lu**,[1] **Hongwei Zhao**,[1] **Renbo Wu**,[1] **Tao Bao**,[2] **and Jun Liu**[3]

[1]*Guangzhou Power Supply Bureau, Guangdong Power Grid Co. Ltd., Guangzhou 510630, China*
[2]*Digital Grid Research Institute, China Southern Power Grid, Guangzhou 510630, China*
[3]*School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Correspondence should be addressed to Yun Long; longyun_grid@163.com

Received 19 January 2023; Revised 16 February 2023; Accepted 11 April 2023; Published 21 April 2023

Academic Editor: Xueqian Fu

More and more renewable energy sources are integrated into novel power systems. The randomness and fluctuation of such renewable energy sources bring challenges to the static stability and safety analysis of novel power systems. In this work, a multilayer deep deterministic policy gradient is proposed to address the fluctuation of renewable energy sources. The proposed method is stacked with multilayer deep reinforcement learning methods that can be continuously updated online. The proposed multilayer deep deterministic policy gradient is compared with other deep learning algorithms. The feasibility, effectiveness, and superiority of the proposed method are verified by numerical simulations.

1. Introduction

More and more countries are joining carbon-peaking and carbon-neutral programs [1]. Building a new type of power system with mainly renewable energy, or even a 100% renewable energy power system, has become imperative [2, 3]. Although the carbon emissions of renewable energy are very small, the dispatch center of the power system has to suffer from the dispatch difficulties caused by the fluctuation of renewable energy [4]. There are already developed countries that have to restart their thermal power generators to meet the demand for electricity for living and production activities [5].

An increasing number of sensors have been installed in novel power systems [6]. These sensors bring a huge amount of data to the dispatch center [7]. The current methods of data processing are still far from adequate to fully utilize the data that the dispatch center can receive. A variety of methods that do not rely on traditional models, i.e., data-driven methods, are constantly being implemented into novel power systems [8].

Currently, a data-driven approach cannot avoid involving deep learning methods [9, 10]. Deep learning can be classified as convolutional neural networks [8], deep neural networks [11], deep reinforcement learning [12], and deep forest algorithms [13]. Deep learning can in turn be classified as classification algorithms, prediction algorithms, and control algorithms [14]. The deep reinforcement learning method is a control method. In this work, the deep reinforcement learning method is applied to solve the safety and stability control problems of novel power systems.

Deep reinforcement learning is developed through reinforcement learning. The series of reinforcement learning methods evolved from being trained to get a look-up table method to an actor-critic network method consisting of a deep neural network or a convolutional neural network [15]. Although it contains deep neural networks or convolutional neural networks internally, deep reinforcement learning is still a control- or policy-based method in general [16].

The deep deterministic policy gradient (DDPG) method is a deep reinforcement learning method based on actor-critic that has been applied very effectively [17]. For example, DDPG can obtain small energy costs in peer-to-peer energy trading [18]. In general, a well-trained deep neural network can represent the control system signal at a specific time scale

[19]. Therefore, to obtain better control performance, a deep reinforcement learning method based on the actor-critic structure capable of observing control signals at multiple time scales is proposed in this study. In this study, DDPG is chosen as the control method mainly based on (1) the fact that the output of DDPG is a deterministic strategy while proximal policy optimization (PPO) is a probability distribution; (2) the critic output in PPO is a value function, and the input is only state, while the critic output in DDPG is a behavior-state value similar to deep Q-networks (DQN); therefore, the input of DDPG contains action. The characteristics of different types of deep reinforcement learning methods are given in Table 1.

Recently, numerous deep reinforcement technologies have been combined to achieve better control performances in more complex scenarios. For example, traditional controllers + deep reinforcement learning [4], modal decomposition + generative adversarial networks [12], and twin-delayed DDPG + DDPG [20] are combined to address the frequency control problems of novel power systems; Markov chain and isoprobabilistic transformation are combined for capacitor planning [21]. Overall, the primary contributions of this work are summarized as follows:

(1) This work proposes a deep reinforcement learning method based on multilayer DDPG. The proposed MDDPG can represent and predict control signals at multiple time scales using multiple deep neural networks. The proposed MDDPG observes more state variables of the control system, and thus has a stronger ability in responding to stochastic perturbations.

(2) This work is the first application of a deep reinforcement learning method based on actor criticism to the stability control of novel power systems. The proposed MDDPG can obtain better power system stability control performances through multiple time scales of representation and observation.

(3) The MDDPG proposed in this study is also a framework for combining multiple deep reinforcement learning. The proposed framework can integrate deep reinforcement learning with different characteristics or different parameters.

The stability control model of novel power systems is presented in the next section. Then, the next section shows the proposed MDDPG method. Then, the simulation calculations and their results are shown. The final section concludes this study.

2. Model of Rotor Angle Stability Control of Novel Power Systems

The modeling process of angle stability control of novel power systems is described in this section.

2.1. Model of Single-Machine Infinite Bus System. A single infinity system is one of the simplest and most basic systems in power systems with infinite power, constant voltage, and constant frequency [22]. A classical generator model is shown in Figure 1. E_B is infinity bus voltage; E_t is generator terminal voltage; $X_d^{'}$ is transient reactance; X_E is the reactance of external network; δ is angle over infinity bus voltage E_B; and $E^{'}$ is reference vector. If the system is affected by a disturbance, δ will be changed.

The stator current I_t is obtained as

$$\widetilde{I}_t = \frac{E^{'}\angle 0^{\circ} - E_B\left(\cos\delta - j\sin\delta\right)}{jX_T}. \tag{1}$$

After the stator resistance is ignored, the air gap power P is equal to the terminal power P_e. The air gap torque is equal to the air gap power when expressed per unit. Then,

$$T_{\mathrm{e}} = P = \frac{E^{'}E_B}{X_{\mathrm{T}}}\sin\delta. \tag{2}$$

Substituting the initial conditions $\delta = \delta_0$, linearize equation (2) as

$$\Delta T_e = \frac{\partial T_e}{\partial\delta}\Delta\delta = \frac{E^{'}E_B}{X_{\mathrm{T}}}\cos\delta_0\left(\Delta\delta\right) = K_s. \tag{3}$$

The motion equations of rotor rotation angle and angle deviation of synchronous generator in per unit, respectively, are

$$\frac{d}{dt}\begin{bmatrix}\Delta\omega_r \\ \Delta\delta\end{bmatrix} = \begin{bmatrix}-\dfrac{K_d}{2H} & -\dfrac{K_s}{2H} \\ \omega_0 & 0\end{bmatrix}\begin{bmatrix}\Delta\omega_r \\ \Delta\delta\end{bmatrix} + \begin{bmatrix}\dfrac{1}{2H} \\ 0\end{bmatrix}\Delta T_{m.} \tag{4}$$

The control block diagram of a single infinity bus system represented by the classical generator model is represented in Figure 2.

In Figure 2, K_s is synchronous torque coefficient; K_D is damped torque coefficient; H is inertia constant; $\Delta\omega_r$ is the standardized value of angular velocity offset; $\Delta\delta$ is rotor angular offset; s is Laplace operator; and ω_0 is rotor reference speed. The expression for the rotor angular offset is obtained from Figure 2 as follows:

$$\Delta\delta = \frac{\omega_0}{s}\left[\frac{1}{2Hs}\left(-K_s\Delta\delta - K_D s\frac{\Delta\delta}{\omega_0} + \Delta T_m\right)\right]. \tag{5}$$

Considering the effect of the variation of system excitation flux on system performance, and neglecting the effect of damping winding on the circuit, the excitation voltage is assumed to be constant. The rotor angle δ is the angle at which the q-axis exceeds the reference quantity E_B, δ_i is the sum of the internal angle and the angle at which E_t exceeds E_B. The equivalent circuit related to the magnetic chain and current of the synchronous motor is shown in Figure 3.

The magnetic chain of the stator and rotor can be expressed as

$$\begin{aligned}
\psi_d &= -L_l i_d + L_{ads}\left(-i_d + i_{\mathrm{fd}}\right) = -L_l i_d + \psi_{ad},\\
\psi_q &= -L_l i_q + L_{aqs}\left(-i_q\right) = -L_l i_q + \psi_{aq},\\
\psi_{fd} &= L_{fd} i_{fd} + L_{ads}\left(-i_d + i_{fd}\right) = L_{fd} i_{fd} + \psi_{ad},
\end{aligned} \tag{6}$$

Table 1: The classification of deep reinforcement learning methods.

Type	Action space	Methods
Value-based	Discrete	*Q* learning, DQN, state-action-reward-state-action
Policy-based	Discrete or continuous	Policy gradient
Actor-critic	Discrete or continuous	Actor-critic, PPO, trust region policy optimization
Actor-critic	Continuous	DDPG, twin-delayed DDPG, soft actor-critic

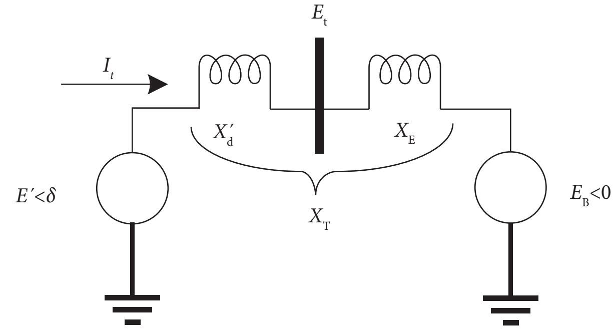

Figure 1: Classic generator model.

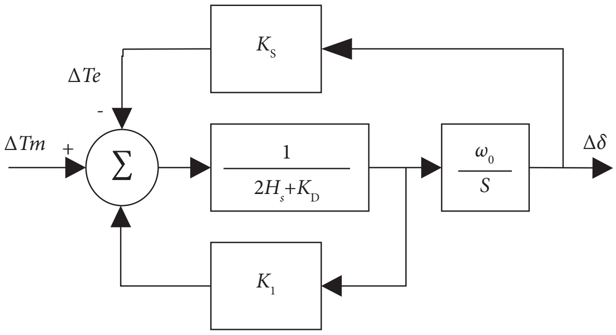

Figure 2: Control framework of single-machine infinite bus system.

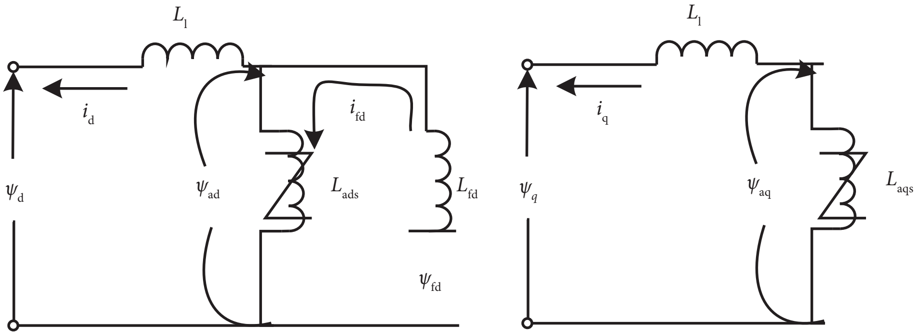

Figure 3: Equivalent circuit diagram of motor magnetic chain and current.

where ψ_{ad} and ψ_{aq} are the air gap magnetic chains; L_{ads} and L_{aqs} are the saturation values of mutual inductances; L_l is the stator leakage inductance; and L_{fd} is the rotor circuit inductance. The excitation current is expressed as

$$i_{fd} = \frac{\psi_{fd} - \psi_{ad}}{L_{fd}}. \tag{7}$$

The air gap magnetic chain of d-axis is expressed by ψ_{fd} and i_d, as

$$\psi_{ad} = -L_{ads} i_d + \frac{L_{ads}}{L_{fd}}\left(\psi_{fd} - \psi_{ad}\right) = L'_{ads}\left(-i_d + \frac{\psi_{fd}}{L_{fd}}\right), \tag{8}$$

where

$$L'_{ads} = \frac{1}{1/L_{ads} + 1/L_{fd}}. \tag{9}$$

Since the rotor circuit is not considered in q-axis, the air-gap magnetic chain is expressed as

$$\psi_{aq} = -L_{aqs} i_q. \tag{10}$$

The air gap torque is written as

$$T_e = \psi_{ad} i_q - \psi_{aq} i_d. \tag{11}$$

The corresponding terms cancel is

$$\begin{aligned} i_d &= \frac{X_{Tq}\left[\psi_{fd}\left(L_{ads}/L_{ads} + L_{fd}\right) - E_B \cos\delta\right] - R_T E_B \sin\delta}{D}, \\ i_q &= \frac{R_T\left[\psi_{fd}\left(L_{ads}/L_{ads} + L_{fd}\right) - E_B \cos\delta\right] + X_{Td} E_B \sin\delta}{D}, \end{aligned} \tag{12}$$

where

$$\begin{aligned} R_T &= R_a + R_E, \\ X_{Tq} &= X_E + \left(L_{aqs} + L_l\right) = X_E + X_{qs}, \\ X_{Td} &= X_E + \left(L'_{aqs} + L_l\right) = X_E + X'_{qs}, \\ D &= R_T^2 + X_{Tq} X_{Td}, \end{aligned} \tag{13}$$

where X_{qs} and X'_{ds} are the reactance saturation values. The reactance value is equal to the inductance value per unit. The perturbation values is

$$\begin{aligned} \Delta i_d &= m_1 \Delta\delta + m_2 \Delta\psi_{fd}, \\ \Delta i_q &= n_1 \Delta\delta + n_2 \Delta\psi_{fd}, \end{aligned} \tag{14}$$

where

$$\begin{aligned} m_1 &= \frac{E_B\left(X_{Tq}\sin\delta_0 - R_T\cos\delta_0\right)}{D}, \\ n_1 &= \frac{E_B\left(R_T\sin\delta_0 + X_{Tq}\cos\delta_0\right)}{D}, \\ m_2 &= \frac{X_{Tq}}{D}\frac{L_{ads}}{\left(L_{ads} + L_{fd}\right)}, \\ n_2 &= \frac{R_T}{D}\frac{L_{ads}}{\left(L_{ads} + L_{fd}\right)}. \end{aligned} \tag{15}$$

Then,

$$\begin{aligned} \Delta\psi_{ad} &= L'_{ads}\left(-\Delta i_d + \frac{\Delta\psi_{fd}}{L_{fd}}\right) = \left(\frac{1}{L_{fd}} - m_2\right) L'_{ads}\Delta\psi_{fd} - m_1 L'_{ads}\Delta\delta, \\ \Delta\psi_{aq} &= -L_{ads}\Delta i_q = -n_2 L_{ads}\Delta\psi_{fd} - n_1 L_{ads}\Delta\delta. \end{aligned} \tag{16}$$

Then,

$$\begin{aligned} \Delta i_{fd} &= \frac{\Delta\psi_{fd} - \Delta\psi_{ad}}{L_{fd}} = \frac{1}{L_{fd}}\left(1 - \frac{L'_{ads}}{L_{fd}} + m_2 L'_{ads}\right)\Delta\psi_{fd} + \frac{1}{L_{fd}} m_1 L'_{ads}\Delta\delta, \\ \Delta T_e &= \Delta\psi_{ad0}\Delta i_q + \Delta\psi_{ad}\Delta i_{q0} - \Delta\psi_{aq0}\Delta i_d - \Delta\psi_{aq}\Delta i_{d0}. \end{aligned} \tag{17}$$

Assume that

$$
\begin{aligned}
K_1 &= n_1\left(\psi_{ad0} + L_{aqs} i_{d0}\right) - m_1\left(\psi_{aq0} + L'_{aqs} i_{q0}\right),\\
K_2 &= n_2\left(\psi_{ad0} + L_{aqs} i_{d0}\right) - m_2\left(\psi_{aq0} + L'_{aqs} i_{q0}\right) + \frac{L'_{aqs}}{L_{fd}} i_{q0},\\
\Delta T_e &= K_1 \Delta\delta + K_2 \psi_{fd}.
\end{aligned}
\tag{18}
$$

The final system equation is written as

$$
\begin{bmatrix} \Delta\dot{\omega}_r \\ \Delta\dot{\delta} \\ \Delta\dot{\psi}_{fd} \end{bmatrix}
=
\begin{bmatrix}
-\dfrac{K_D}{2H} & -\dfrac{K_1}{2H} & -\dfrac{K_2}{2H} \\
\omega_0 & 0 & 0 \\
0 & -\dfrac{\omega_0 R_{fd}}{L_{fd}} m_1 L'_{ads} & -\dfrac{\omega_0 R_{fd}}{L_{fd}} \left[1 - \dfrac{L'_{ads}}{L_{fd}} + m_2 L'_{ads}\right]
\end{bmatrix}
\begin{bmatrix} \Delta\omega_r \\ \Delta\delta \\ \Delta\psi_{fd} \end{bmatrix}
+
\begin{bmatrix}
\dfrac{1}{2H} & 0 \\
0 & 0 \\
0 & \dfrac{\omega_0 R_{fd}}{L_{adu}}
\end{bmatrix}
\begin{bmatrix} \Delta T_m \\ \Delta E_{fd} \end{bmatrix}.
\tag{19}
$$

The prime-mover and exciter control are ΔT_m and ΔE_{fd}, respectively. If the air-gap torque output from the prime-mover and excitation voltage output from the exciter are constants, the value ΔT_m and ΔE_{fd} are zeros. If the final system equation is a classical generator model, both R_{fd} and R_a are equal to 0, $X_q = X'_d$. In the above equations, L_{ads} and L_{aqs} are the saturated values of mutual inductance L_{ad} and L_{aq}, respectively. L_{adu} and L_{aqu} are the unsaturated values of mutual inductance L_{ad} and L_{aq}, respectively. The initial static values of system variables are indicated by the subscript 0.

The variation of ψ_{fd} depends on the equation of the excitation circuit. Then,

$$
\Delta\psi_{fd} = \frac{K_3}{1 + pT_3}\left[\Delta E_{fd} - K_4 \Delta\delta\right],
\tag{20}
$$

where

$$
\begin{aligned}
K_3 &= \frac{L_{fd}}{\left[1 - L'_{ads}/L_{fd} + m_2 L'_{ads}\right] L_{adu}},\\
K_4 &= \frac{L_{adu}}{L_{fd}} m_1 L'_{ads},\\
T_3 &= K_3 T'_{d0} \frac{L_{adu}}{L_{ffd}}.
\end{aligned}
\tag{21}
$$

L_{ffd} is the rotor circuit self-inductance. The derivative operator p is replaced by the Laplace operator s as

$$
\Delta\psi_{fd} = \frac{K_3}{1 + sT_3}\left[\Delta E_{fd} - K_4 \Delta\delta\right].
\tag{22}
$$

Thus, a control block diagram with stable excitation voltage representation is obtained in Figure 4. If ΔE_{fd} is zeros, K_4 can be set as negative for large local load, which is supplied partly by generators and remote large system (Figure 4).

The values in parentheses are written in the following form as

$$
\begin{aligned}
\psi_{ad0} + L_{aqs} i_{d0} &= e_{q0} + R_a i_{q0} + X_{qs} i_{d0} = E_{q0},\\
\psi_{aq0} + L'_{aqs} i_{q0} &= -L_{aqs} i_{q0} + L'_{aqs} i_{q0} = -\left(X_q - X'_d\right) i_{q0}.
\end{aligned}
\tag{23}
$$

E_{q0} is the prefault value of the voltage after $R_a + jX_q$. The expanded form of K_1 constant is

$$
K_1 = \frac{E_B E_{q0}}{D}\left(R_T \sin\delta_0 + X_{Td} \cos\delta_0\right) + \frac{E_B i_{q0}}{D}\left(X_q - X'_d\right)\left(X_{Tq} \sin\delta_0 - R_T \cos\delta_0\right).
\tag{24}
$$

Similarly, the expansion of K_2, K_3, T_3, and K_4 is calculated as

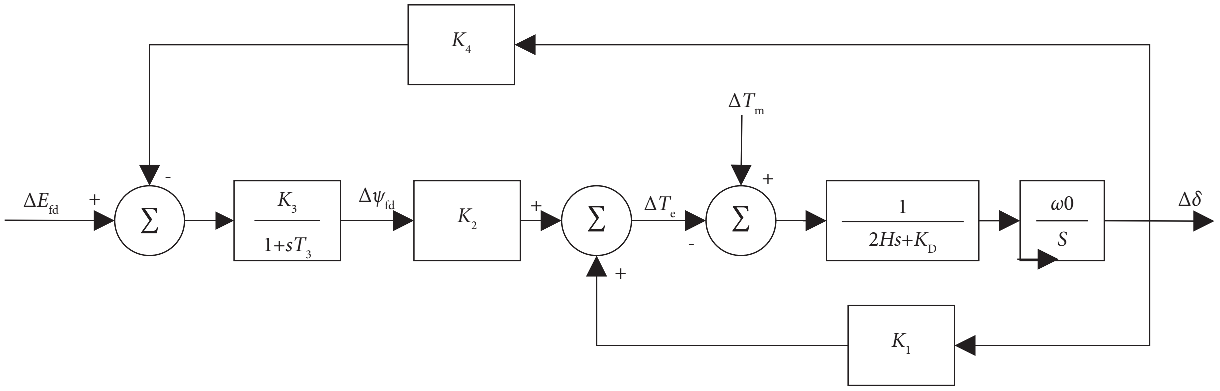

Figure 4: Control framework of single infinity system with constant excitation voltage.

$$K_2 = \frac{L_{ads}}{L_{ads}+L_{fd}}\left[\frac{R_T}{D}E_{q0} + \left(\frac{X_{Tq}\left(X_q - X_d'\right)}{D} + 1\right)i_{q0}\right], \tag{25}$$

$$\begin{aligned} K_3 &= \frac{L_{ads}+L_{fd}}{L_{adu}}\,\frac{1}{1 + X_{Tq}/D\left(X_q - X_d'\right)}, \\ T_3 &= \frac{T_{d0s}'}{X_{Tq}\left(X_q - X_d'\right)/D + 1}, \\ K_4 &= L_{adu}\frac{L_{ads}}{L_{ads}+L_{fd}}\,\frac{E_B}{D}\left(X_{Tq}\sin\delta_0 - R_T\cos\delta_0\right). \end{aligned} \tag{26}$$

If the influence of saturation is ignored, K_4 can be simplified to

$$K_4 = \frac{E_B}{D}\left(X_d - X_d'\right)\left(X_{Tq}\sin\delta_0 - R_T\cos\delta_0\right). \tag{27}$$

2.2. Model of Automatic Voltage Regulation. The input signal to the excitation system is the generator terminal voltage E_t. $\widetilde{E}_t$ is represented by the state variables $\Delta\omega_r$, $\Delta\delta$, and $\Delta\psi_{fd}$.

$$E_t^2 = e_d^2 + e_q^2. \tag{28}$$

In the case of small perturbations,

$$\left(E_{t0} + \Delta E_t\right)^2 = \left(e_{d0} + \Delta e_d\right)^2 + \left(e_{q0} + \Delta e_q\right)^2. \tag{29}$$

Neglecting the second-order term for all perturbation values, then

$$E_{t0}\Delta E_t = e_{d0}\Delta e_d + e_{q0}\Delta e_q. \tag{30}$$

Therefore,

$$\Delta E_t = \frac{e_{d0}}{E_{t0}}\Delta e_d + \frac{e_{q0}}{E_{t0}}\Delta e_q. \tag{31}$$

With the value of the disturbance, the stator voltage equations are written as

$$\begin{aligned} \Delta e_d &= -R_a\Delta i_d + L_l\Delta i_q - \Delta\psi_{aq}, \\ \Delta e_q &= -R_a\Delta i_q - L_l\Delta i_d + \Delta\psi_{ad}. \end{aligned} \tag{32}$$

Then,

$$\Delta E_t = K_5\Delta\delta + K_6\Delta\psi_{fd}, \tag{33}$$

where

$$\begin{aligned} K_5 &= \frac{e_{d0}}{\Delta E_{t0}}\left[-R_a m_1 + L_l n_1 + L_{aqs} n_1\right] + \frac{e_{q0}}{\Delta E_{t0}}\left[-R_a n_1 - L_l m_1 - L_{ads}' m_1\right], \\ K_6 &= \frac{e_{d0}}{\Delta E_{t0}}\left[-R_a m_2 + L_l n_2 + L_{aqs} n_2\right] + \frac{e_{q0}}{\Delta E_{t0}}\left[-R_a n_2 + L_l m_2 + L_{ads}'\left(\frac{1}{L_{fd}} - m_2\right)\right]. \end{aligned} \tag{34}$$

The model of the thyristor excitation system with automatic voltage regulation (AVR) is shown in Figure 5, where E_{FMAX} and E_{FMIN} are the upper and lower limits of the excitation output voltage, respectively; T_R is the time constant of the terminal voltage transducer; V_{ref} is the system reference voltage; and v_1 is the output of the terminal voltage transducer. The thyristor excitation system contains only the necessary connections for the specific system and uses a high-gain exciter. The limiting and protection circuits are omitted (Figure 5) because they do not affect the small-signal stability.

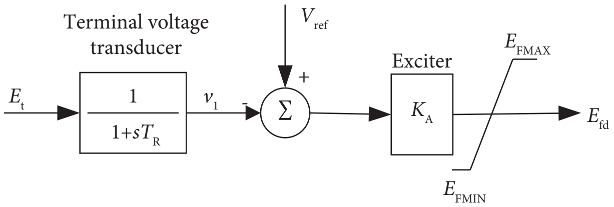

Figure 5: Thyristor excitation system with AVR.

Considering the effect of the excitation system, the equation of the excitation circuit is

$$p\Delta\psi_{fd} = -\frac{\omega_0 R_{fd}}{L_{fd}} m_1 L'_{ads}\Delta\delta - \frac{\omega_0 R_{fd}}{L_{fd}}\left[1 - \frac{L'_{ads}}{L_{fd}} + m_2 L'_{ads}\right]\Delta\psi_{fd} - \frac{\omega_0 R_{fd}}{L_{adu}} K_A \Delta v_1. \tag{35}$$

Because the exciter is a first-order model, the order of the whole system is increased by one order on top of the original one; the newly added state variables are Δv_1. Since $p\Delta\omega_r$ and $p\Delta\delta$ are not affected by the exciter, the entire state-space model of the power system is written in the form of the following vector-matrix:

$$\begin{bmatrix} \Delta\dot{\omega}_r \\ \Delta\dot{\delta} \\ \Delta\dot{\psi}_{fd} \\ \Delta\dot{v}_1 \end{bmatrix} = \begin{bmatrix} -\frac{K_D}{2H} & -\frac{K_1}{2H} & -\frac{K_2}{2H} & 0 \\ \omega_0 & 0 & 0 & 0 \\ 0 & -\frac{\omega_0 R_{fd}}{L_{fd}} m_1 L'_{ads} & -\frac{\omega_0 R_{fd}}{L_{fd}}\left[1 - \frac{L'_{ads}}{L_{fd}} + m_2 L'_{ads}\right] & -\frac{\omega_0 R_{fd}}{L_{adu}} K_A \\ 0 & \frac{K_5}{T_R} & \frac{K_6}{T_R} & -\frac{1}{T_R} \end{bmatrix} \begin{bmatrix} \Delta\omega_r \\ \Delta\delta \\ \Delta\psi_{fd} \\ \Delta v_1 \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Delta T_m. \tag{36}$$

If the mechanical torque input is constant, ΔT_m is 0. $G_{\text{ex}}(s)$ is the transfer function of the AVR and the exciter. $G_{\text{ex}}(s)$ is applied to any type of exciter, be expressed in terms of K_A as

$$G_{\text{ex}}(s) = K_A. \tag{37}$$

2.3. Model of Power System with Automatic Voltage Regulation and Power System Stabilizer. The PSS, which is an additional excitation control technique to suppress low-frequency oscillations of synchronous generators by introducing additional feedback signals, has been utilized to improve the stability of power systems. The control block diagram of the excitation system, including AVR and PSS, is shown in Figure 6. The PSS shown in Figure 6 includes three links: a phase compensation link, a signal filtering link, and an amplification link. The phase compensation link properly provides a phase lag characteristic to compensate for the phase lag between the exciter input and the air gap torque of the generator. Since the signal link is a high-pass filter with a large time constant T_{W}, the oscillating signal ω_{r} does not change as the oscillating signal passes through. The stabilizer gain K_{STAB} determines the magnitude of damping generated by PSS. Adding perturbation values to the signal-filtering module has

$$\Delta v_2 = \frac{pT_{\text{w}}}{1 + pT_{\text{w}}}\left(K_{\text{STAB}}\Delta\omega_{\text{r}}\right). \tag{38}$$

Hence,

$$p\Delta v_2 = K_{\text{STAB}}\, p\Delta\omega_{\text{r}} - \frac{1}{T_{\text{w}}}\Delta v_2. \tag{39}$$

Figure 6 shows that

$$\Delta E_{fd} = K_A\left(\Delta v_s - \Delta v_1\right). \tag{40}$$

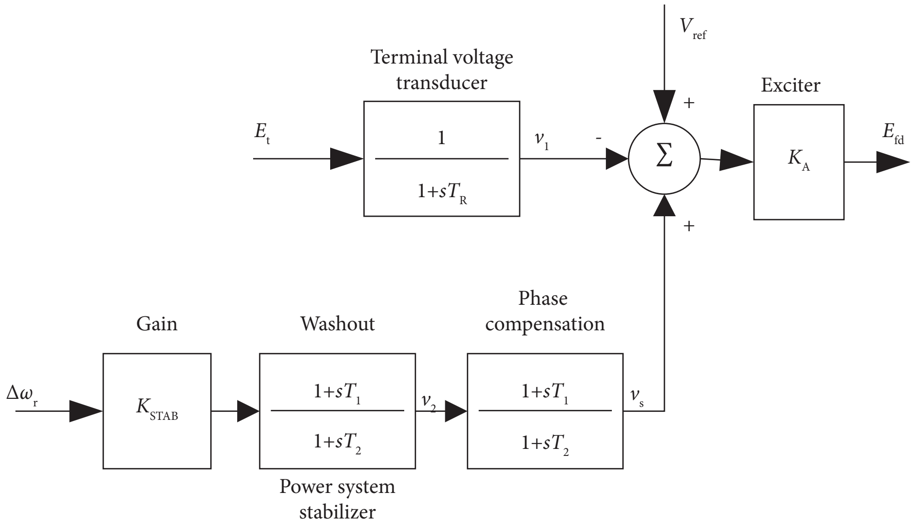

Figure 6: Thyristor excitation system including AVR and PSS.

After adding PSS, the whole state-space model is expressed as the following vector matrix if $\Delta T_m = 0$, then

$$\begin{bmatrix} \Delta\dot{\omega}_r \\ \Delta\dot{\delta} \\ \Delta\dot{\psi}_{fd} \\ \Delta\dot{v}_1 \\ \Delta\dot{v}_2 \\ \Delta\dot{v}_s \end{bmatrix} = \begin{bmatrix} -\frac{K_D}{2H} & -\frac{K_1}{2H} & -\frac{K_2}{2H} & 0 & 0 & 0 \\ \omega_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\omega_0 R_{fd}}{L_{fd}} m_1 L'_{ads} & -\frac{\omega_0 R_{fd}}{L_{fd}}\left[1-\frac{L'_{ads}}{L_{fd}}+m_2 L'_{ads}\right] & a_{34} & 0 & \frac{\omega_0 R_{fd}}{L_{adu}} K_A \\ 0 & \frac{K_5}{T_R} & \frac{K_6}{T_R} & -\frac{1}{T_R} & 0 & 0 \\ -K_{\text{STAB}}\frac{K_D}{2H} & -K_{\text{STAB}}\frac{K_1}{2H} & -\frac{K_2}{2H} K_{\text{STAB}} & 0 & -\frac{1}{T_W} & 0 \\ -K_{\text{STAB}}\frac{T_1}{T_2}\frac{K_D}{2H} & -K_{\text{STAB}}\frac{K_1}{2H}\frac{T_1}{T_2} & -K_{\text{STAB}}\frac{K_2}{2H}\frac{T_1}{T_2} & 0 & -\frac{T_1}{T_2}\frac{1}{T_W}+\frac{1}{T_2} & -\frac{1}{T_2} \end{bmatrix} \begin{bmatrix} \Delta\omega_r \\ \Delta\delta \\ \Delta\psi_{fd} \\ \Delta v_1 \\ \Delta v_2 \\ \Delta v_s \end{bmatrix}. \tag{41}$$

The control framework of power systems containing AVR and PSS is obtained, as shown in Figure 6.

3. Multilayer Deep Deterministic Policy Gradient

Reinforcement learning is an algorithm that performs actions based primarily on feedback from the environment. By continuously interacting with the environment, the agent continuously "tries and fails" with one or more learning strategies to maximize the gain and achieve a specific goal problem [23]. The interaction between the agent and the environment means that the agent observes the state s of the environment, performs an action a, changes the state of the environment, and returns a reward r and a new state s' to the agent [24].

The mathematical basis of reinforcement learning is a Markovian decision process (MDPs) [25]. An MDP usually consists of a state space, an action space, a state transfer matrix, a reward function, a policy function, and a discount factor. In an episode, write down all the rewards as: $R_1, \ldots, R_t, \ldots R_n$. Assuming a discount rate of $\gamma \in [0,1]$, the discounted return U_t can be defined as

$$U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \cdots + \gamma^{n-t} \cdot R_n, \tag{42}$$

at t, when the episode is not over, U_t is an unknown random variable whose randomness comes from all states and actions after the moment t. The action-value function is defined as

$$Q_\pi(s_t,a_t) = E\left[U_t \middle| S_t = s_t, A_t = a_t\right]. \tag{43}$$

The expectation in equation (43) eliminates all states $S_{t+1},\ldots,S_n$ with all actions $A_t,\ldots,A_n$ after the moment t. The optimal action-value function utilizing a maximization elimination strategy π is

$$Q_*(s_t,a_t) = \max_\pi Q_\pi(s_t,a_t), \forall s_t \in S, a_t \in A. \tag{44}$$

In equation (97), S is the set of all states, and A is the set of all actions.

To address the problem that DQN applied to continuous action spaces can suffer from dimensional catastrophe, the deterministic policy gradient (DPG) is proposed [26]. The DPG method is the most common reinforcement learning for doing continuous control actions. DDPG simply combines DQN and actor-critic. DDPG, which can also be described as a combination of DPG and DQN, can combine the successful structure of DQN with actor-critic to improve the stability and convergence of DDPG. Since DDPG is based on DPG and has deep learning integration, DDPG can characterize high-dimensional data. The DPG is based on deep Q-learning, which employs a neural network $\mu(s;\theta)$ to provide action and another neural network $q(s,a;w)$ to evaluate the performance of the actions for improving the accuracy of performed actions. $\mu(s;\theta)$ and $q(s,a;w)$ are called the strategy and value networks, respectively (Figure 7).

Collecting experience with behavior policy, assume behavior policy is

$$a = \mu(s;\theta_{\text{now}}) + \varepsilon, \tag{45}$$

where $\mu(s;\theta_{\text{now}})$ is the determined policy network; ε is the added noise, $\varepsilon \in R^d$. The behavior policy is implemented to control the interaction between the agent and the environment; the trajectory (s_t,a_t,r_t,s_{t+1}) of the agent is stored in the experience replay array; the collected experience is reused for training (Figure 8).

In the training process of the policy network, the policy network outputs an action a for a state s, and then the value network evaluates the action a output by the policy network to obtain the value of the evaluation $q(s,\mu(s;\theta);w)$. A higher evaluation of the value network means more accurate action given by the policy network. Hence, the objective function is defined as the expectation of the evaluation value.

$$J(\theta) = E_S[q(S,\mu(S;\theta);w)]. \tag{46}$$

The learning of policy network transforms is a problem of maximizing solution, i.e.,

$$\max_\theta J(\theta). \tag{47}$$

The gradient is calculated using one observation d of the random variable S at each iteration.

$$g_j = \nabla_\theta q\left(s_j,\mu\left(s_j;\theta\right);w\right), \tag{48}$$

where g_j is called the determined policy gradient, which is derived by applying the chain rule, as

$$\nabla_\theta q\left(s_j,\mu\left(s_j;\theta\right);w\right) = \nabla_\theta \mu\left(s_j;\theta\right)\cdot\nabla_a q\left(s_j,\hat{a}_j;w\right), \tag{49}$$

where $\hat{a}_j = \mu(s_j;\theta)$. A state s_j is randomly selected from the experience replay array at each iteration, $\hat{a}_j = \mu(s_j;\theta)$; gradient ascent is employed to update the parameters of the policy network, as

$$\theta \leftarrow \theta + \beta\cdot\nabla_\theta \mu\left(s_j;\theta\right)\cdot\nabla_a q\left(s_j,\hat{a}_j;w\right), \tag{50}$$

where β is the learning rate of policy networks. To bring the value network $q(s,a;w)$ closer to the true value function $Q_\pi(s,a)$, temporal-difference (TD) is utilized to train value networks for more accurately evaluating the actions of policy network output. A trajectory (s_j,a_j,r_j,s_{j+1}) of an agent is selected from the experience replay group at each iteration; the value network evaluates the output action by the policy network, as

$$\begin{cases} \hat{q}_j = q\left(s_j,a_j;w\right), \\ \hat{q}_{j+1} = q\left(s_{j+1},a_{j+1};w\right). \end{cases} \tag{51}$$

Calculate TD targets as

$$\hat{y}_j = r_j + \gamma\cdot\hat{q}_{j+1}. \tag{52}$$

Then, the loss function is

$$L(w) = \frac{1}{2}\left[q\left(s_j,a_j;w\right) - \hat{y}_j\right]^2. \tag{53}$$

Calculate gradients as

$$\nabla_w L(w) = \left(\hat{q}_j - \hat{y}_j\right)\cdot\nabla_w q\left(s_j,a_j;w\right). \tag{54}$$

Update the parameters of the value network with gradient descent as

$$w \leftarrow w - \alpha\cdot\nabla_w L(w). \tag{55}$$

After the loss function $L(w)$ is reduced, the prediction of the value network is closer to the target value function, where α is the learning rate of the value network.

The deterministic policy gradient suffers from the same overestimation problem as DQN, leading to difficulty in convergence during the training process. DPG is combined with deep learning as DDPG [26]. DDPG adds a policy target network and a value target network to the DPG; thus, two-goal networks are applied to calculate TD goals. The policy target and value target networks have the same structure as the policy network and value networks with different parameters. The DDPG policy network is updated in the same way as the DPG.

However, the parameters of the value network are updated differently. The evaluation of the trajectory of the agent at moment j is calculated by the value network; the evaluation of the trajectory at the next iteration is calculated

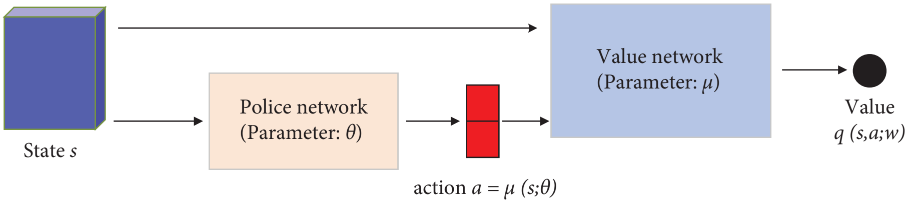

Figure 7: Diagram of DDPG.

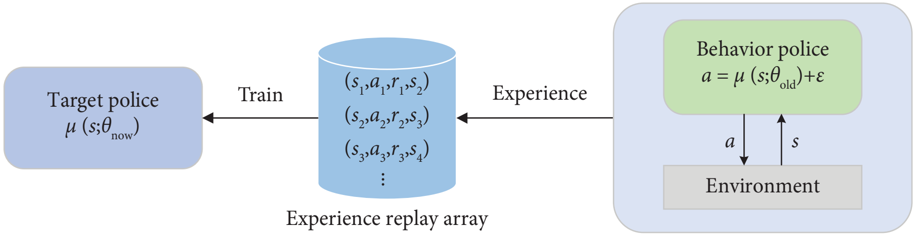

Figure 8: Separate experience gathering and policy updating.

by $\mu(s;\theta^-)$ and $q(s,a;w^-)$. Thus, the evaluation of the two moments is obtained as follows:

$$\begin{cases} \hat{q}_j = q\left(s_j, a_j; w\right) \\ \hat{q}_{j+1} = q\left(s_{j+1}, a_{j+1}'; w^-\right) \end{cases}, \tag{56}$$

where $a_{j+1}' = \mu(s_{j+1};\theta^-)$.

Update the parameters of the value network as

$$w \leftarrow w - \alpha \cdot \nabla_w L(w, w^-). \tag{57}$$

The target policy and target value networks are updated by a weighted average as follows, where τ is the hyperparameter (Figure 9):

$$\begin{aligned} \theta^- &\leftarrow \tau \cdot \theta + (1-\tau)\theta^-, \\ w^- &\leftarrow \tau \cdot w + (1-\tau)w^-. \end{aligned} \tag{58}$$

The MDDPG consists of multiple DDPGs, which contain two critic networks and two actor networks, (Figure 10). The MDDPG equally sums the actions by each DDPG to produce a new action a_t' with a given state s_t. The training steps of the MDDPG are shown in Algorithm 1.

In this study, two DDPGs applied to a power-stable stack are trained and updated simultaneously. In this study, the rlMultiAgentTrainingOptions() function in MATLAB is adopted in conjunction with the train() function to train these two agents simultaneously. Moreover, the proposed method is an integral control algorithm. This integral control algorithm requires the cooperation of two agents to complete an output action.

4. Results and Discussion

The simulation studies performed in this work are accomplished on a laptop with an AMD 8500H processor, 32 GB RAM, and 3060 GPU. To verify the feasibility and effectiveness of the proposed method, a disturbance input is designed, as shown in Figure 11. The designed disturbances fluctuated very sharply from 40 s to 50 s. The dramatically varying disturbance in Figure 11 is designed to verify whether the proposed method can stochastically adapt to complex disturbances to keep static safety and stability analysis of novel power systems.

The parameters of this novel power system are set as follows: $K_1 = 1.591$, $K_2 = 1.5$, $K_3 = 0.333$, $K_4 = 1.4187$, $K_5 = -0.12$, $K_6 = 0.3$, $K_D = 0$, $K_A = 200$, $K_{\text{STAB}} = 9.5$, $H = 3.0$, $T_1 = 0.154$, $T_2 = 0.033$, $T_3 = 1.91$, $T_W = 1.4$, and $T_R = 0.02$.

In this work, the proposed MDDPG method is compared with the traditional proportional-integral-derivative and the traditional Q-learning methods. For a fair comparison, a similar or the same parameter is adopted for the reinforcement learning family of methods.

The parameters of the proportional-integral-derivative (i.e., 0.7102, 25.8658, 0.0326) utilized in this study are optimized by a particle swarm optimization algorithm with a total population of 200 and the number of iterations of 200. The three parameters of this conventional algorithm are coupled with each other. These three parameters must be fully tuned in a wide range to obtain a high-performance control performance. The control parameters obtained by the optimization algorithm are superior when both the number of iterations and the population size of the chosen optimization algorithm exceed 100.

The parameters of the reinforcement learning series employed in this study are set as follows. The more the number of actions in the action matrix, the more training time is required to compute the memory, which may even exceed the computer memory. Although the smaller the number of actions in the action matrix, the smaller the computation memory, the faster the computation time, and the lower the accuracy. After extensive testing, the action matrix in this work is a 16-equivalent value between −0.1 and +0.1. The properties of setting the number of rows and

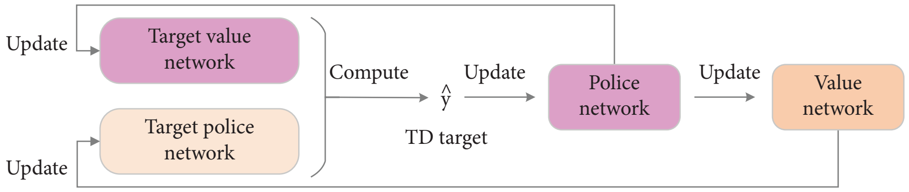

Figure 9: Relationship between four neural networks.

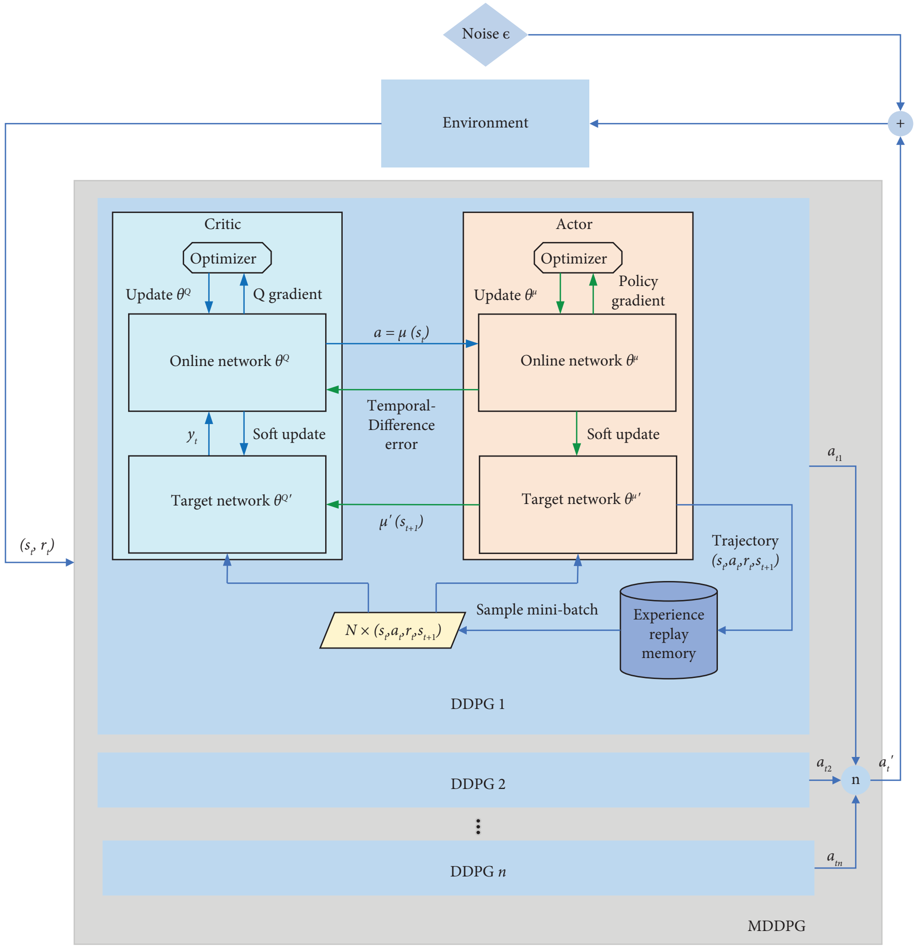

Figure 10: Structure of MDDPG.

columns of the Q-value matrix and the P matrix are similar to the characteristics of establishing the number of actions in the action matrix. Therefore, after numerous tests, both the Q-value matrix and the probability P matrix in this work are 16-row, 16-column matrices. Although higher learning rates imply faster convergence and inaccurate control actions, and lower learning rates imply slower convergence and longer training times, the proposed algorithm can characterize the input-output relationship of the system after a long period of online training iterations. Therefore, the learning rate, discount factor, and probability update rate are set to default values that are set by most references. The learning rate is 0.1. The discount factor is 0.05. The probability update rate is 0.9. The number of hidden layer units inside the actor and critic networks is set to [30 30].

The rotor angle deviations obtained by the compared algorithms are shown in Figure 12. The proposed MDDPG obtains the smallest rotor angle deviation. The reason why the Q-learning method based on the key-value pair type has not achieved a higher control performance than the proportional-integral-derivative is that the Q-learning method has too few action values. Although the number of proposed MDDPG actions is small, the MDDPG has strong prediction capability, which obtains better control performances.

The controller outputs given by the three comparison algorithms are shown in Figure 13. The conventional proportional-integral-derivative controller gives a smooth output curve. Reinforcement learning gives control actions that are too trial-and-error. The control commands given by the MDDPG proposed in this work appear to be irregular

```
Randomly initialize the parameters $\theta_t$ of policy network and $\theta_t$ of the target network.
Randomly initialize the parameters $\theta_t^-$ of the policy target network and $\theta_t^-$ of the target network.
Randomly initialize the experience replay matrix.
Execute the platform corresponding to the environment by the initial action in one step.
Obtain initial state $s_t$ from the environment.
For $i$ from 1 to maximum iteration $N$
    Obtain the actions $a_t$ from all policy networks through the received state $s_t$.
    Obtain new action $a_t'$ by summing the actions $a_t$ output from all policy networks and the agent performs the action $a_t'$ based on the received state $s_t$.
    Obtain reward value $r_t$ and next state $s_{t+1}$ from the environment.
    Deposit the quadratic array of trajectories $(s_t, a_t', r_t, s_{t+1})$ of the agent into the experience replay matrix.
    Update sampling priority.
    Randomly sample $M$ samples from the experience replay matrix and calculate the current target value $y_i$.
    Calculate TD error and TD target.
    Update the parameters $\theta_t$ and $w_t$ of policy and value networks by gradient ascent and descent, respectively.
    Set a hyperparameter $\tau$, update the parameters $\theta^-$ and $w^-$ of target policy and target value networks by weighted average.
End for
Save the trained model/networks.
```

Algorithm 1: Training steps of MDDPG.

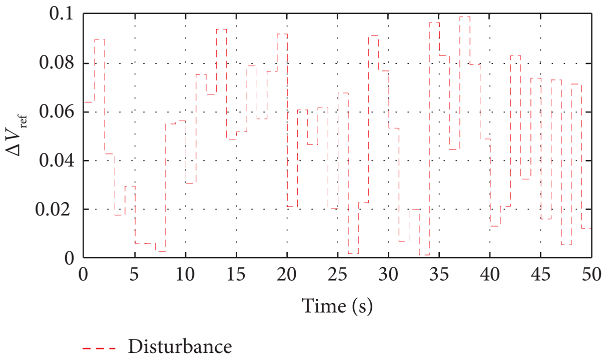

Figure 11: Curve charts of disturbance.

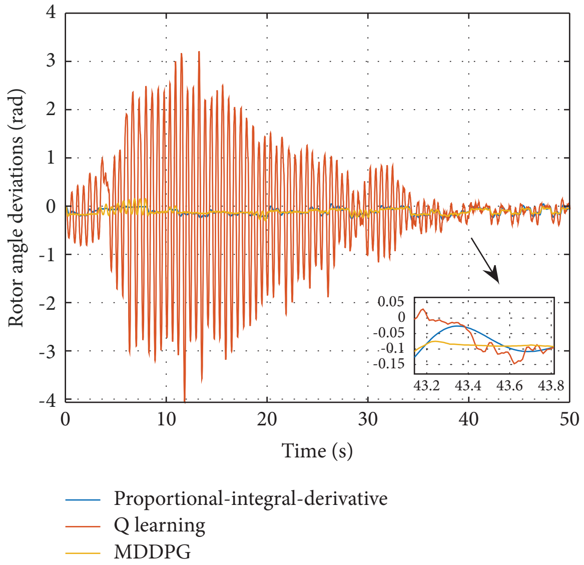

Figure 12: Curves of rotor angle deviations obtained by compared methods.

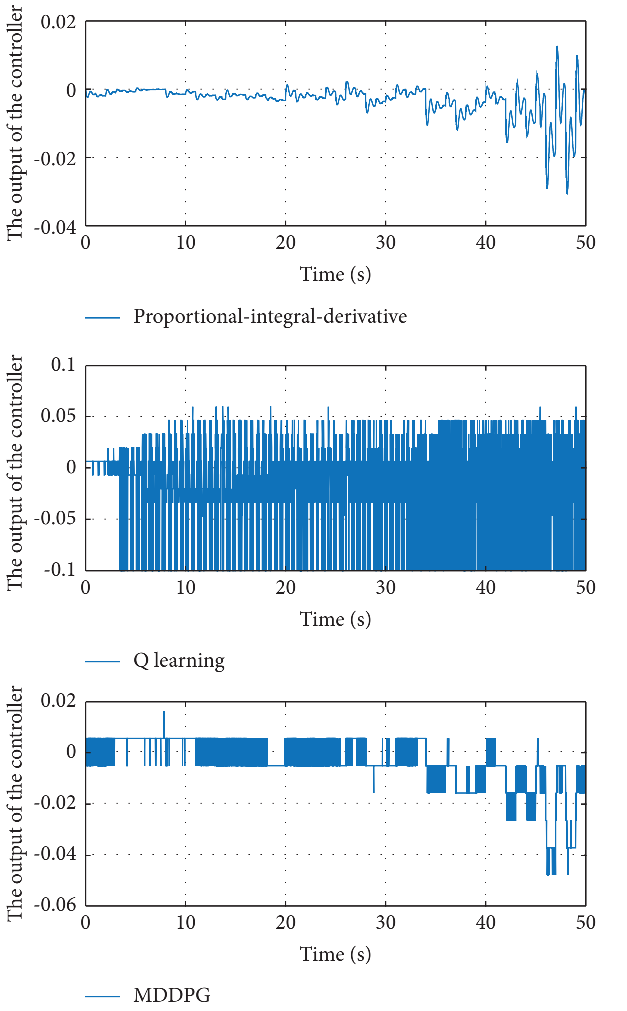

Figure 13: Controller output curves obtained by compared methods.

but can give better control results. The MDDPG proposed in this work gives actions between 40 and 50 s, which can eliminate the sharp disturbances.

The experiments of MDDPG for the angle stability control of power systems show that the control performance is more stable than other algorithms. The MDDPG processes more information and decomposes the high-dimensional input vector into multiple low-dimensional input vectors, effectively avoiding dimensional disasters. In addition, Figure 13 shows that (1) the conventional controller with very smooth and continuous control instructions obtains power angles with larger fluctuations in the end; a conventional controller with only three parameters is difficult to obtain the optimum in both steady-state values and convergence speed simultaneously. (2) Q-learning with strong random fluctuation can give trial-and-error signals with large fluctuation and a long convergence period. (3) The MDDPG that balances trial-and-error fluctuations and control performance can achieve smaller control errors than the traditional controller and Q-learning.

The method of principal component analysis could be considered to solve the coupling problems of the inputs of MDDPG. Exploring an MDDPG that can handle multidimensional information while reducing the computational memory and computational time of the system is an important direction.

The deficiencies of this proposed MDDPG are summarized as follows: (1) The MDDPG processes more information with more computation memory and longer computation time. (2) Meanwhile, MDDPG splits the high-dimensional vectors into several low-dimensional vectors as inputs, which weakens the coupling of input information. (3) This MDDPG has a total of eight networks that need to be trained and updated. The network number in this MDDPG is more than that of the normal deep reinforcement learning methods.

5. Conclusions

In this work, a reinforcement learning algorithm called MDDPG, which combines several DDPGs, is proposed to solve the rotor angle stability control of novel power systems. The test results verify the feasibility and effectiveness of the MDDPG. The primary characteristics of the methods are outlined as follows:

(1) The MDDPG combines multiple DDPGs and applies multiple deep neural networks with high adaptability, high fault tolerance, and self-organization capability. When the system is under different perturbations, an MDDPG can control the output of the system rotor angle stably with an error less than proportional-integral-derivative and Q learning.

(2) The MDDPG can transform the high-dimensional input into multiple low-dimensional inputs. The output action of MDDPG is deterministic, is the action superposition of each DDPG output, and provides accurate control continuously in real time with short systemic stability time.

(3) The MDDPG provides accurate control. In the listed example, the system rotor angle stability control error is smaller than in comparison with other algorithms.

Future work could be improved in the following three ways: first, the proposed MDDPG framework could be incorporated into other more effective deep reinforcement learning methods; second, the proposed multilayer framework could be reduced to a deep reinforcement learning method composed of multiple deep neural networks; and finally, the static safety and stability problem of the novel power system could be combined with the dynamic safety and stability problem to be solved by deep reinforcement learning simultaneously.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yun Long was responsible for conceptualization, funding acquisition, project administration, supervision, methodology, resources, writing review, editing, data curation, formal Analysis, investigation, software, validation, visualization, and writing the original draft. Youfei Lu, Hongwei Zhao, Renbo Wu, Tao Bao, and Jun Liu reviewed and edited the manuscript.

Acknowledgments

The authors appreciatively acknowledge the support of the Science and Technology Projects of the China Southern Power Grid (GZHKJXM20210041).

References

[1] L. Cai, J. Luo, M. Wang et al., "Pathways for municipalities to achieve carbon emission peak and carbon neutrality: a study based on the LEAP model," *Inside Energy*, vol. 262, Article ID 125435, 2023.

[2] D. Liu, Y. Wu, Y. Kang et al., "Multi-agent quantum-inspired deep reinforcement learning for real-time distributed generation control of 100% renewable energy systems," *Engineering Applications of Artificial Intelligence*, vol. 119, Article ID 105787, 2023.

[3] X. Fu and Y. Zhou, "Collaborative Optimization of PV greenhouses and clean energy systems in rural areas," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 1, pp. 642–656, 2023.

[4] L. Yin and Y. Wu, "Mode-decomposition memory reinforcement network strategy for smart generation control in multi-area power systems containing renewable energy," *Applied Energy*, vol. 307, Article ID 118266, 2022.

[5] M. R. Hamedi, M. Ghafory-Ashtiany, and M. Hosseini, "Hybrid simulation modeling framework for evaluation of

Thermal Power Plants seismic resilience in terms of power generation," *International Journal of Disaster Risk Reduction*, vol. 78, Article ID 103120, 2022.

[6] R. V. Yohanandhan, R. M. Elavarasan, R. Pugazhendhi, M. Premkumar, L. Mihet-Popa, and V. Terzija, "A holistic review on Cyber-Physical Power System (CPPS) testbeds for secure and sustainable electric power grid–Part–I: background on CPPS and necessity of CPPS testbeds," *International Journal of Electrical Power & Energy Systems*, vol. 136, Article ID 107718, 2022.

[7] J. Chen, K. Li, K. Li, P. S. Yu, and Z. Zeng, "Dynamic bicycle dispatching of dockless public bicycle-sharing systems using multi-objective reinforcement learning," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 5, no. 4, pp. 1–24, 2021.

[8] Y. Zhou, Q. Guo, H. Sun, Z. Yu, J. Wu, and L. Hao, "A novel data-driven approach for transient stability prediction of power systems considering the operational variability," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 379–394, 2019.

[9] X. Fu, Q. Guo, and H. Sun, "Statistical machine learning model for stochastic optimal planning of distribution networks considering a dynamic correlation and dimension reduction," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 2904–2917, 2020.

[10] X. Fu and H. Niu, "Key technologies and applications of agricultural energy internet for agricultural planting and fisheries industry," *Information Processing in Agriculture*, 2022.

[11] L. Yin and B. Zhang, "Time series generative adversarial network controller for long-term smart generation control of microgrids," *Applied Energy*, vol. 281, Article ID 116069, 2021.

[12] K. Han, K. Yang, and L. Yin, "Lightweight actor-critic generative adversarial networks for real-time smart generation control of microgrids," *Applied Energy*, vol. 317, Article ID 119163, 2022.

[13] L. Yin, Z. Sun, F. Gao, and H. Liu, "Deep forest regression for short-term load forecasting of power systems," *IEEE Access*, vol. 8, pp. 49090–49099, 2020.

[14] R. Kamalraj, S. Neelakandan, M. Ranjith Kumar, V. Chandra Shekhar Rao, R. Anand, and H. Singh, "Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm," *Measurement*, vol. 183, Article ID 109804, 2021.

[15] L. Yin and B. Zhang, "Relaxed deep generative adversarial networks for real-time economic smart generation dispatch and control of integrated energy systems," *Applied Energy*, vol. 330, Article ID 120300, 2023.

[16] L. Yin and Y. Li, "Fuzzy vector reinforcement learning algorithm for generation control of power systems considering flywheel energy storage," *Applied Soft Computing*, vol. 125, Article ID 109149, 2022.

[17] J. Li, "A multi-objective energy coordinative and management policy for solid oxide fuel cell using triune brain large-scale multi-agent deep deterministic policy gradient," *Applied Energy*, vol. 324, Article ID 119313, 2022.

[18] C. Samende, J. Cao, and Z. Fan, "Multi-agent deep deterministic policy gradient algorithm for peer-to-peer energy trading considering distribution network constraints," *Applied Energy*, vol. 317, Article ID 119123, 2022.

[19] Z. Xu, C. Li, and Y. Yang, "Fault diagnosis of rolling bearings using an improved multi-scale convolutional neural network with feature attention mechanism," *ISA Transactions*, vol. 110, pp. 379–393, 2021.

[20] L. Yin and S. Li, "Hybrid metaheuristic multi-layer reinforcement learning approach for two-level energy management strategy framework of multi-microgrid systems," *Engineering Applications of Artificial Intelligence*, vol. 104, Article ID 104326, 2021.

[21] X. Fu, "Statistical machine learning model for capacitor planning considering uncertainties in photovoltaic power," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, p. 5, 2022.

[22] A. Chen, Y. Ba, X. Luo, H. Huang, L. Meng, and J. Shi, "Strategy for grid low-frequency oscillation suppression via VSC-HVDC linked wind farms," *Energy Reports*, vol. 8, pp. 1287–1295, 2022.

[23] J. Li, D. Pang, Y. Zheng, X. Guan, and X. Le, "A flexible manufacturing assembly system with deep reinforcement learning," *Control Engineering Practice*, vol. 118, Article ID 104957, 2022.

[24] M. Eppe, C. Gumbsch, M. Kerzel, P. D. H. Nguyen, M. V. Butz, and S. Wermter, "Intelligent problem-solving as integrated hierarchical reinforcement learning," *Nature Machine Intelligence*, vol. 4, no. 1, pp. 11–20, 2022.

[25] H. Kurniawati, "Partially observable Markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 253–277, 2022.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, https://arxiv.org/abs/1509.02971.