Hindawi
Science and Technology of Nuclear Installations
Volume 2022, Article ID 2941705, 13 pages
https://doi.org/10.1155/2022/2941705

Research Article

Design of Control System of Once-Through Steam Generator Based on Proximal Policy Optimization Algorithm

Cheng Li,[1,2] **Ren Yu**,[1] **Wenmin Yu**,[1] **and Tianshu Wang**[1]

[1]*Naval University of Engineering, Wuhan 430033, China*
[2]*China Nuclear Power Operation Technology Corporation, Ltd, Wuhan 430000, China*

Correspondence should be addressed to Cheng Li; 296379210@qq.com and Ren Yu; 18071068480@163.com

Received 14 November 2021; Revised 9 March 2022; Accepted 7 April 2022; Published 20 May 2022

Academic Editor: Guglielmo Lomonaco

Because of the characteristics of the small water volume of OTSG, it is hard to control the outlet steam pressure when the load is changed or disturbed. This study is devoted to the control of the once-through steam generator (OTSG). A double-layer controller based on the PPO algorithm is proposed to control the outlet steam pressure of OTSG. The bottom layer is the PID controller; it directly regulates the OTSG feed water valve and then controls the steam pressure. The top layer of the controller is the agent based on the PPO algorithm, which is responsible for optimizing the parameters of the PID in real time to obtain better control performance. The agent chooses PID parameters as actions to the environment, and then, the reward value is obtained through the reward function of the environment which enables online learning of the agent. Compared with the PID controller, the simulation experiment result shows that the method not only has a good control performance but also has a good anti-interference ability.

1. Introduction

The once-through steam generator (OTSG) is a key part of the nuclear power plant, which is used to produce superheated steam without dehumidification devices. It has advantages of simple structure, small volume, good static characteristics, maneuver performance, and so on and can improve the thermal efficiency of the equipment. Due to the advantages, OTSG is often taken into account for small- and middle-sized nuclear power plants [1].

At present, the research on the control of the OTSG mainly focuses on the pressure control in the secondary loop; that is, the outlet steam quality control is realized by controlling the steam pressure at the outlet. Due to the strong coupling characteristics of the OTSG, the outlet pressure control is difficult [2]. In order to keep the steam outlet pressure constant, Zhang et al. addressed a scheme based on PID for adjusting the secondary feedwater flow rate with steam pressure deviation signal and steam flow signal [3]. Cheng et al. [4] proposed a distributed and multi-input and output coupling artificial immune control strategy and applied the strategy to the pressure control of the OTSG, which can effectively improve the dynamic operating characteristics of the pressure and related parameters of the OTSG. Chen et al. [5] introduced a T-S fuzzy neural control principle into the water supply control system of the OTSG.

The above studies have achieved many good results in simulation experiments and practical applications, which benefit from the development of diversified control algorithms and computer technology. However, the above methods also have drawbacks, such as the need for a large amount of training data or accurate mathematical models. Many methods need to establish a relatively accurate system model and need to design accurate parameters of the controller. When the system cannot be modeled completely or the environment changes greatly, the performance of the controller will degrade to some extent. For systems with unclear or completely unknown mathematical models, the emergence of intelligent control methods with self-learning ability can provide ideas for solving those problems.

Reinforcement learning (RL) is a kind of machine learning, the basic idea of RL is to explore the optimal strategy through the interaction between agent and environment and to maximize the return [6]. Classical RL, such

as Q-learning, discretizes the action and state space and uses the Q-table to solve the problem [7]. However, classical RL is difficult to discretize the continuous action and state space in the actual control problem; moreover, the high-dimensional continuous state space and action space increase the calculation burden. Fortunately, due to the rise of deep learning, deep neural networks have been introduced into RL as value function approximators in recent years. Deep reinforcement learning (DRL) solves the curse of dimensions problem by introducing neural networks to the algorithm. Timothy et al. proposed the Deep Deterministic Policy Gradient (DDPG) algorithm to solve the control problem with continuous action space [8]. DRL became widely known when Google's AlphaGo defeated Lee Sedol, one of the world's top Go players.

DRL not only has the excellent data processing ability of deep learning but also has the excellent decision ability of reinforcement learning, which has developed into an important part of the machine learning field. Up to now, DRL has been used in robots, HVAC, UAV, energy, and other control fields. In order to enable full participation of high-performance units controlled by different dispatching centers in the performance-based frequency regulation market, Li et al. used an effective exploration-based multiagent deep deterministic policy gradient (EE-MADDPG) algorithm for the grid-area coordinated load frequency control (GAC-LFC) [9]. He proposed another method based on the imitation guided-exploration multiagent twin-delayed deep deterministic policy gradient (IGE-MATD3) algorithm to address the coordination problems between AGC controllers in multiarea power systems [10]. Designing a controller for the attitude control of the moving mass-actuated unmanned aerial vehicle (MAUAV) faces severe challenges due to the strong nonlinearity and coupling of its dynamics. Qiu et al. proposed an attitude controller based on deep reinforcement learning for the (MAUAV). It directly maps the states to the needed deflection of the actuators and is an end-to-end controller [11]. Deng et al. proposed a novel optimal heating, ventilation, and air conditioning (HVAC) control method combining active building environment change detection and deep Q network (DQN). This method aims to disentangle the nonstationarity by actively identifying the change points of building environments and learning effective control strategies for corresponding building environments [12]. Zhang et al. adopted a reinforcement learning algorithm to optimize the task sequence allocation scheme in assembly processes of the human-robot collaborative [13]. A data-driven approach that leverages deep reinforcement-learning techniques to intelligently learn effective strategies for state diagnosis of safety functions is proposed by JaeKwan Park. The approach shows that it has the potential to assist human operators in monitoring the safety functions of nuclear facilities [14]. At present, RL is not widely used in nuclear power plants, but good results have been achieved. Park et al. propose an automatic control method for plant heat-up mode using deep reinforcement-learning technology as a basic study for plant automation [15]. A multilayer perception (MLP)-based reinforcement learning control (RLC) is applied to the optimization of thermal power response for a high temperature gas-cooled reactor-based nuclear steam supply system (NSSS) [16].

As an intelligent algorithm, DRL has been paid more attention in recent years. Its self-learning and model-free characteristics provide a new idea to solve the control problem of the OTSG. The present study investigates the feasibility of applying the DRL technique to the OTSG. The random policy search method is one of the model-free methods in reinforcement learning. Its representative method, PPO (Proximal Policy Optimization), can make the rewards monotonic nondecreasing, that is, policies are always updated for the better. The PPO algorithm constantly explores through the interaction with the environment to obtain the optimal policy without any prior knowledge.

In this study, a two-layer control scheme for the OTSG based on PPO is proposed. On the basis of the bottom PID controller, the upper layer which applies the PPO algorithm is designed to realize the self-learning of the parameters of the PID controller for the OTSG. The contributions of this study are as follows:

(i) A two-layer control scheme based on RL for the OTSG is proposed

(ii) PPO is designed to realize the self-learning of the parameters of the PID controller

(iii) The novel part of this work is to verify the feasibility and validity of the RL method for solving the control problem of OTSG in this study

The study is organized as follows. Based on the previous research study, the innovation points of this study are described in Section 1. The nonlinear mathematical model of the OTSG is introduced in Section 2. Section 3 outlines the control framework for OTSG based on the PPO algorithm. The accuracy and effectiveness of the controller are evaluated in Section 4. Conclusions are given in Section 5.

2. Nonlinear Mathematical Model of the OTSG

The OTSG is a type of steam generator which applies double sides to transfer heat. The primary fluid flows from top to bottom both in the internal part of the inner pipe and the external part of the outer pipe, and the secondary fluid flows from bottom to top in the annulus channel between the inner pipe and the outer pipe and then flows out the superheated steam.

2.1. Model Simplification and Assumptions. OTSG can be divided into three regions: subcooled, nucleate boiling, and superheat regions [17]. The heat flux between water of centric tube and outside annuli tube and that of annulus channel is assumed to be equal, and then, the steam generator's model is built by lumped parameters with moving boundary. The temperature and enthalpy, the outlet parameters of each section of the primary and secondary loops, are considered as lumped parameters [18]. The input and output of the mathematical model are shown in Figure 1.

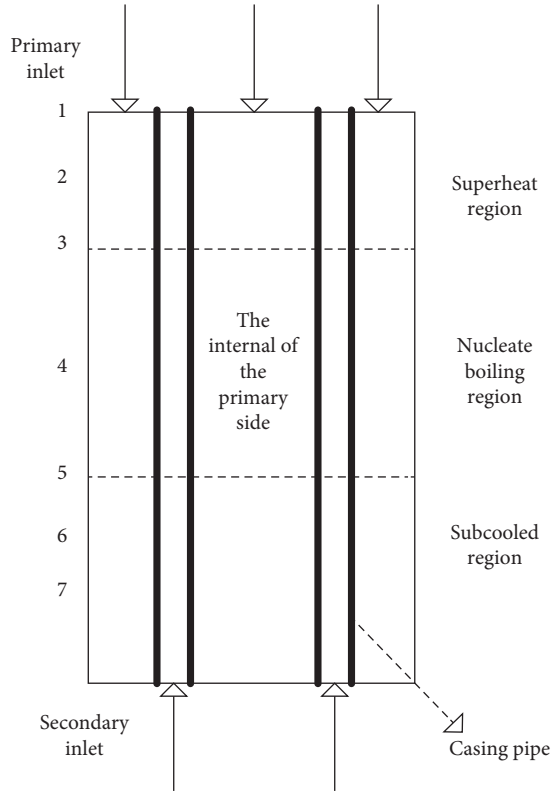

Figure 1: The schematic diagram of OTSG.

The assumptions of the model are as follows:

(1) The weight, specific heat, and heat transfer coefficient of the primary fluid change slowly in each region

(2) The weight, specific heat, and thermal conductivity of the metal are constant in each region

(3) The physical characteristics and heat conduction coefficient of the secondary fluid change linearly in each region

(a) Subcooled region: $h \le hf$ (hf is the enthalpy of saturated water)
Energy conservation equation in the primary subcooled region is

$$\frac{\mathrm{d}\left(\rho_p l_6 A_p h_{p7}\right)}{\mathrm{d}t} = w_{p5}h_{p5} - w_{p7}h_{p7} - Q_{p6}. \quad (1)$$

Energy conservation equation in the secondary subcooled region is

$$\frac{\mathrm{d}\left(\rho_{s6} l_6 A_s h_{s5}\right)}{\mathrm{d}t} = w_{s7}h_{s7} - w_{s5}h_{s5} + Q_{p6}. \quad (2)$$

Mass conservation equation in the secondary subcooled region is

$$\frac{\mathrm{d}\left(\rho_{s6} l_6 A_s\right)}{\mathrm{d}t} = w_{s7} - w_{s5}. \quad (3)$$

(b) Nucleate boiling region: $0 < x < 1$ (x is dryness)
Energy conservation equation in the primary nucleate boiling region is

$$\frac{\mathrm{d}\left(\rho_p l_4 A_p h_{p5}\right)}{\mathrm{d}t} = w_{p3}h_{p3} - w_{p5}h_{p5} - Q_{p4}. \quad (4)$$

Energy conservation equation in the secondary nucleate boiling region is

$$\frac{\mathrm{d}\left(\overline{\rho_{s4}} l_4 A_s h_{s3}\right)}{\mathrm{d}t} = w_{s5}h_{s5} - w_{s3}h_{s3} + Q_{p4}. \quad (5)$$

Mass conservation equation in the secondary nucleate boiling region is

$$\frac{\mathrm{d}\left(\overline{\rho_{s4}} l_4 A_s\right)}{\mathrm{d}t} = w_{s5} - w_{s3}. \quad (6)$$

(c) Superheat region: the secondary saturates steam to the outlet.

Energy conservation equation in the primary superheat region is

$$\frac{\mathrm{d}\left(\rho_p l_2 A_p h_{p3}\right)}{\mathrm{d}t} = w_{p1}h_{p1} - w_{p3}h_{p3} - Q_{p2}. \quad (7)$$

Energy conservation equation in the secondary superheat region is

$$\frac{\mathrm{d}\left(\overline{\rho_{s2}} l_2 A_s h_{s1}\right)}{\mathrm{d}t} = w_{s3}h_{s3} - w_{s1}h_{s1} + Q_{p2}. \quad (8)$$

Mass conservation equation in the secondary superheat region is

$$\frac{\mathrm{d}\left(\overline{\rho_{s2}} l_2 A_s\right)}{\mathrm{d}t} = w_{s3} - w_{s1}, \quad (9)$$

where Q is heat transfer, l is effective length, h is the enthalpy of each cross-section, w is flow, ρ is density, A is efficient flow area, and P and s represent primary and secondary loops individually.

3. RL Controller Design for OTSG

3.1. Reinforcement Learning. Reinforcement learning is an unsupervised learning method. Through repeated interactions with the dynamic environment, the agent learns to select the optimal or near-optimal action to achieve its long-term goal [19]. The basic framework of reinforcement learning is shown in Figure 2.

Markov decision process (MDP) is an interactive learning framework as well as a mathematical description of reinforcement learning problems. The goal of reinforcement learning is to learn a policy to maximize the expected reward in which the object starts from its initial state s under a certain policy π:

$$J = E_{s\sim\rho, a\sim\pi}[R_1]. \quad (10)$$

The process can be described by a five-tuple $\{S, A, P, R, \gamma\}$, where S is the set of states, A is the set of actions, P is the state transition matrix, R is the reward function, and γ is the discount coefficient.

The reward is defined as the discount accumulated reward:

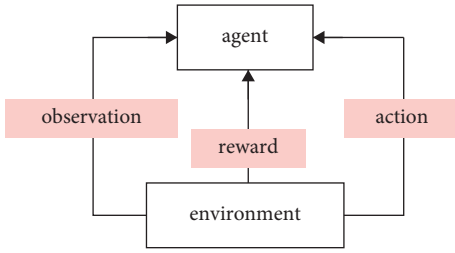

Figure 2: The framework of reinforcement learning.

$$R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i). \tag{11}$$

The state value function $V^\pi(s)$ represents the discount accumulation rewards that are obtained from the environment when executing policy π at state s:

$$V^\pi(s_t) = E_{s\sim\rho, a\sim\pi}[R_t|s_t]. \tag{12}$$

The state action value function $Q^\pi(s,a)$ represents the accumulation rewards brought by taking action at state s and then executing the policy π:

$$Q^\pi(s,a) = E_{s\sim\rho, a\sim\pi}[R_t|s_t, a_t]. \tag{13}$$

To solve reinforcement learning problems, the methods mainly include dynamic programming (DP), Monte Carlo (MC), and temporal difference (TD). DP is suitable for solving model-based reinforcement learning problems, while model-free reinforcement learning problems need to be solved by MC or TD. MC does not make full use of the MDP structure of reinforcement learning, which makes the efficiency low. TD combines the ideas of DP and MC, which can achieve more efficiency in model-free learning [20].

The TD defines that the current state action value function is estimated by the state action value function at the next moment:

$$Q^\pi(s,a) = E_{r,s_{t+1}\sim E}\lfloor r(s_t, a_t) + \gamma E_{a_{t+1}\sim\pi}[Q\pi(s_{t+1}, a_{t+1})] \rfloor. \tag{14}$$

The error of temporal difference can be defined as

$$\delta_t = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) - Q^\pi(s,a). \tag{15}$$

When approximating the value function of state and action by the neural network, the value function is a function about parameter θ; then, the loss function of the network can be defined as

$$L(\theta) = E_{s\sim\rho, a\sim\pi}\lfloor (y_t - Q(s_t, a_t|\theta))^2 \rfloor, \tag{16}$$

where

$$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1}|\theta). \tag{17}$$

3.2. PPO Algorithm. Since the traditional policy gradient algorithm is greatly affected by the step size, the too big step size will affect the final learning effect. In response to this problem, the PPO algorithm, a new type of policy gradient algorithm, is proposed by OpenAI. The PPO algorithm proposes a new objective function that can be updated in small batches through multiple training steps, thereby solving the problem of step size selection in traditional policy gradient algorithms [21–23].

The optimization goal of reinforcement learning is to maximize the reward. In the policy gradient algorithm, the parameter θ of the objective function is updated as

$$L(\theta) = E[\log \pi(a_t|s_t;\theta) A_t(s_t, a_t)], \tag{18}$$

where $A_t(s_t, a_t)$ is the superiority function under the current policy:

$$A_t(s_t, a_t) = Q_t(s_t, a_t) - V_t(s_t). \tag{19}$$

The parameter θ is updated by policy gradient algorithm as follows:

$$\theta_{t+1} = \theta_t + a\nabla_\theta L(\theta_t). \tag{20}$$

The PPO algorithm overcomes the problem that the policy gradient algorithm is difficult to select an appropriate step size, ensuring that the policy model is monotonically improved during the optimization of the model. The objective function is modified as

$$L(\theta) = E\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t\right]. \tag{21}$$

The Kullback–Leibler (KL) divergence of the old and new policies satisfies the following constraints:

$$E\lfloor KL\lfloor \pi_{\theta_{\text{old}}}(a_t|s_t), \pi_\theta(a_t|s_t) \rfloor \rfloor \le \delta, \tag{22}$$

where KL divergence is used to measure the difference degree between the two distributions. The larger the value is, the greater the difference between the two distributions is, which can stabilize the training process.

The constraint term is introduced into the objective function as a penalty term in the PPO algorithm; that is, the objective function is

$$L(\theta) = E\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t - \beta KL\left[\pi_{\theta_{\text{old}}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]. \tag{23}$$

It is found that the truncation function clip instead of KL divergence is used to constrain $r_t(\theta)$ to prevent the large difference between the old and new policies reach better results. The ratio of the old and new policies is

$$r_t(\theta) \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \tag{24}$$

The objective function is

$$L(\theta) = E[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)], \tag{25}$$

where ε is a hyperparameter. The PPO algorithm avoids the mutation of policy by truncation function or limited KL divergence and enhances the training effect in the process of parameter updating.

3.3. Controller Design Based on PPO Algorithm for OTSG. PID control algorithm is widely used in industrial control because of its simple structure and good robustness. The control parameters of traditional PID are largely obtained according to experience. However, due to the uncertainties in practical application, PID control is often difficult to achieve the optimal. In order to solve the above problem, this study proposes a PID controller for OTSG based on the PPO algorithm, as shown in Figure 3.

The double-layer controller proposed in this study adopts a two-stage control structure, and the bottom controller adopts the PID method; it directly regulates the OTSG feed water valve and then controls the steam pressure. The upper controller adopts the intelligent agent controller based on the PPO algorithm and is responsible for the online adjustment of K_p, K_i, and K_d of the PID controller. In the control process, the bottom controller and the top controller work together to adjust the control strategy in real time according to the state of the system and realize the intelligent autonomous control.

The whole control system can be divided into two layers; the bottom layer is the PID controller; it directly regulates the OTSG feed water valve to adjust the flow and then control the steam pressure. The top layer of the control system is the agent which is based on PPO, which explores the optimal policy for the PID controller through the interaction with the environment. The agent receives states and rewards from the environment and sends actions to the environment. The PPO algorithm is used as the learning method of the agent in this study. PPO creates a replay buffer to store historical experiences and then randomly sample transitions from it and feed those samples to update actor and critic networks. The replay buffer helps the agent to be able to learn previous experiences and improve the efficiency of sample utilization. Random sampling can break the correlation between samples and make the learning process of agents more stable.

PPO uses 3 neural networks, namely, actor-new network, actor-old network, and critic network; the role of each network is given below:

(1) Actor-new network: it selects an action according to state s; the action is used to interact with the environment (OTSG) to generate the next state $s\_$ and reward r

(2) Actor-old network: the parameters of the actor-old network are periodically copied from the actor-new network before each batch size step, which is to prevent the update step too big

(3) Critic network: it is responsible for the iterative update of critic network parameters and calculating the state value function v, which represents the cumulative discount returns when executing the current policy

The pseudocode of PPO is given in Algorithm 1.

The initialization parameters of the PID controller in the bottom layer can be adjusted by Ziegler–Nichols law, and the upper PPO algorithm adjusts PID parameters on this basis. The core of the PPO algorithm is to design appropriate state and action space and reward function. The state space is the representation of the environment, the action space is the reasonable description of the action of the agent, and the reward function can correctly evaluate the control effect of PID parameter optimization. The following sections describe this in detail.

3.4. Closed-Loop Stability Analysis. The general PID control system is linear, and its stability problem has been solved. However, compared with the PID control system based on reinforcement learning, since the PID parameters change during the operation of the system, the system is nonlinear, and its stability is affected. How to ensure the stability of the system under the condition of parameter changes is a problem to be considered in the design of the PID control system based on reinforcement learning.

It can be seen from Figures 4–6 that the functional relationship between the PID control and pressure of the OTSG can be approximated as

$$A\dot{x} + Bx + C = Df, \tag{26}$$

where A, B, C, and D are constant coefficients, x is pressure, and f is feedwater flow.

The feedwater flow is regulated by the PID controller:

$$f = K_p e_w + K_d \dot{e_w} + K_i \int e_w \mathrm{d}t, \tag{27}$$

where e_w is the difference between the pressure and the set point and K_p, K_i, and K_d are the input parameters of the PID controller.

Substitute (27) into (26):

$$A\dot{x} + Bx + C = DK_p e_w + DK_d \dot{e_w} + DK_i \int e_w \mathrm{d}t. \tag{28}$$

Considering that a certain equilibrium state is the initial state of the system after the system is disturbed:

$$\begin{aligned} &-A\dot{e_w} - Be_w + C = DK_p e_w + DK_d \dot{e_w} + DK_i \int e_w \mathrm{d}t, \\ \text{or}\quad &\dot{e_w} + a_1 e_w + a_0 \int e_w \mathrm{d}t = C', \end{aligned} \tag{29}$$

where $a_1 = (Dk_p + B)/(Dk_d + A)$, $a_0 = Dk_i/(Dk_d + A)$, and $C' = C/(Dk_d + A)$.

Set $x_1(t) = \int e_w \mathrm{d}t$, $x_2(t) = e_w$; then, equation (29) can be expressed as the equation of state:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} C'. \tag{30}$$

For the system represented by equation (30), the necessary and sufficient conditions for its stability are

$$\begin{cases} a_1 > 0, \\ a_0 > 0. \end{cases} \tag{31}$$

Then,

$$\begin{cases} k_i > 0, \\ Dk_p + B > 0. \end{cases} \tag{32}$$

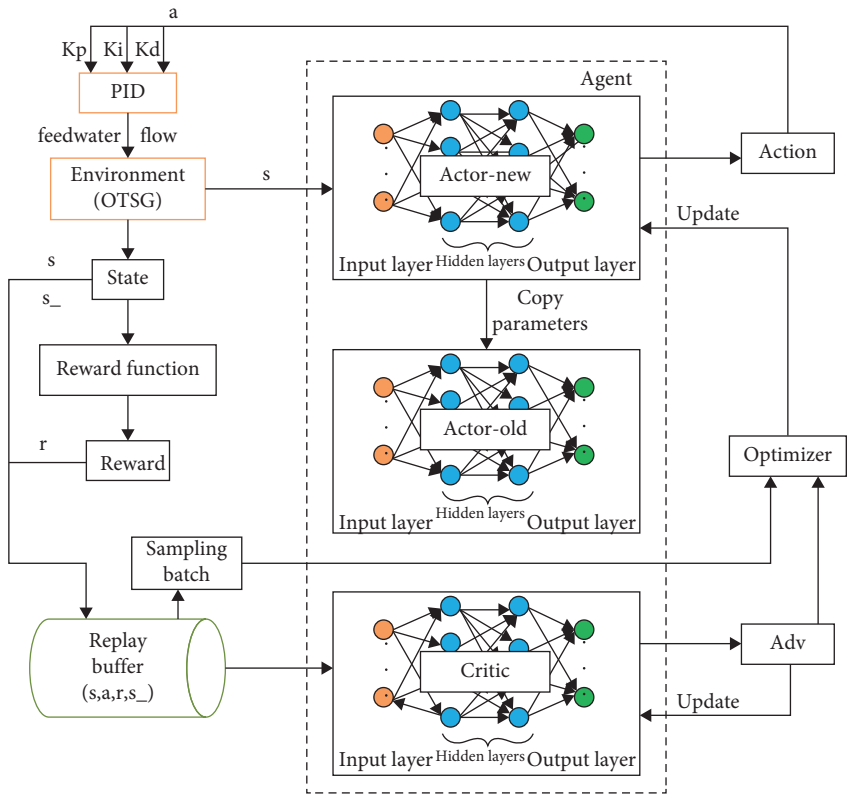

Figure 3: The framework of control structure based on reinforcement learning.

(1) Input: actor-new network $\pi_\theta(a_t|s_t)$, actor-old network $\pi_{\theta_{\text{old}}}(a_t|s_t)$
(2) Initial Replay Buffer R
(3) for $k = 1$, N do
(4) for $t = 1$, T do
(5) Execute actor-new network $\pi_\theta(a_t|s_t)$ in the environment to obtain the data (s, a, r, $s\_$).
(6) R.save (s, a, r, $s\_$)
(7) end for
(8) for $t = 1$, T do
(9) Compute advantage estimates based on the critic network
(10) end for
(11) for $j = 1$, K do
(12) R.sample (M)
(13) Update parameter θ according to the target function L^{clip} or L^{clip}
(14) end for
(15) $\theta_{\text{old}} \longleftarrow \theta$
(16) end for

Algorithm 1: PPO.

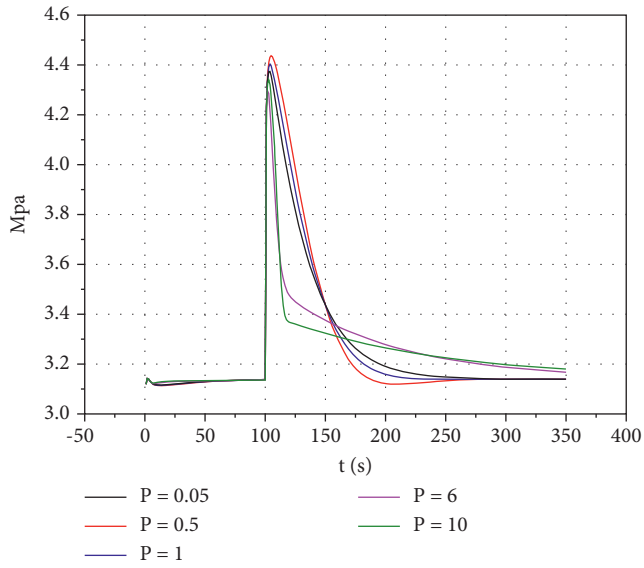

Figure 4: The changes of pressure at different P when I and D are fixed.

The value k_d thus seen has no effect on the stability of the equation. Sufficient conditions for system stability can be expressed as

$$\begin{cases} k_p > -\dfrac{B}{D}, \\ k_i > 0, \\ k_d > 0. \end{cases} \tag{33}$$

For the control system proposed above, the controller will adjust PID parameters online according to the system operation. Real-time parameters are represented by

$$\begin{cases} k_p = k_{p0} + \Delta k_p, \\ k_i = k_{i0} + \Delta k_i, \\ k_d = k_{d0} + \Delta k_d, \end{cases} \tag{34}$$

where k_{p0}, k_{i0}, and k_{d0} are the initial values of PID parameters and $\Delta k_p, \Delta k_i$, and Δk_d are the adjustment.

Substitute (34) into (33); then,

$$\begin{cases} \Delta k_p > -k_{p0} - \dfrac{B}{D}, \\ \Delta k_i > -k_{i0}, \\ \Delta k_d > -k_{d0}. \end{cases} \tag{35}$$

It can be seen that the stability of the system can be guaranteed as long as the regulation quantity of the controlled system is limited to a certain range.

4. Experiments and Results

OTSG simulation model is built with Matlab, the reinforcement learning control algorithm program is developed with Python, and the data exchange between Python program and Matlab simulation model through Socket communication. The simulation experiment and performance analysis of the OTSG pressure control is carried out to verify the effectiveness of the above scheme.

4.1. Design of State Space. According to the structure of the steam generator, the water volume in the secondary loop of the OTSG is small. When the load changes, the steam pressure is easy to fluctuate. If the water supply cannot keep up with the pace of changes at this time, equipment in the secondary loop will have an impact. Therefore, in order to represent the dynamic characteristic of the OTSG and advantageous for the observation, we choose the steam pressure as the state space; therefore, this study selects the parameters of state space including the steam generator outlet pressure, the current pressure deviation $e(t)$ (deviation between the current outlet pressure and set pressure), and the deviation value in last time $e(t-1)$.

4.2. Design of Action Space

4.2.1. Parameter Sensitivity Analysis. Before selecting the action space and formulating the control strategy, the sensitivity analysis of the control parameters to the dynamic characteristics of the OTSG is carried out. Directly take the PID controller parameters as the main control parameters. Analyze the relationship between the parameters P, I, D, and the OTSG pressure. The research is set up as three experiments of power reduction, two control parameters are fixed, respectively, and the other is adjusted to analyze the dynamic characteristics of the OTSG.

The increase of P makes the system responsive, the adjustment speed is faster, and the steady-state error can be reduced. As can be seen from Figure 4 that if the P gets too large, the overshoot will increase and the adjustment time will lengthen and too large P will even make the closed-loop system unstable.

The integral function of the controller is set up to eliminate the redundancy of the control system. As long as the deviation exists, the output of the integrating control will change; that is, the integration will always work, and the integration will stop only if the deviation does not exist. From Figure 5, we can see that if the integration time is small, the integration speed is large and the integration effect is strong. Conversely, the integration time is large, the integration effect is weak.

The output variation of differential action is proportional to the differential time and the speed of deviation change and has nothing to do with the deviation. The greater the speed of deviation change, the longer the differential time and then the greater the output variation of differential action. Figure 6 shows that proper differential control can reduce overshoot and increase system stability.

4.2.2. Design of Action Space. According to the above analysis results, PID control parameters are designed as the action space of the agent, and the dimension is 3; that is,

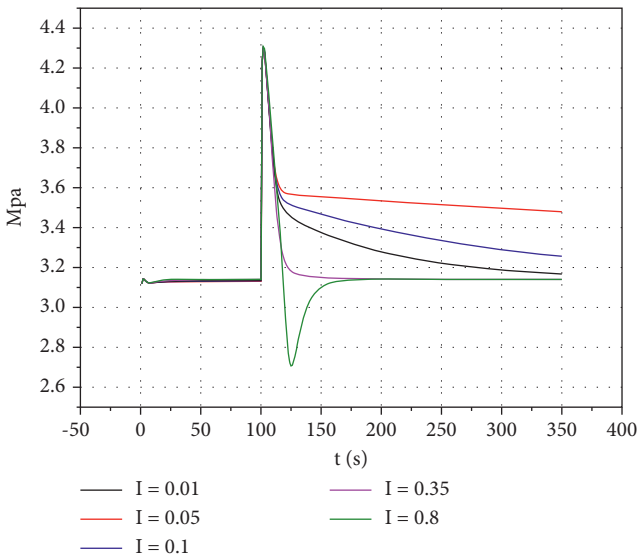

Figure 5: The changes of pressure at different I when P and D are fixed.

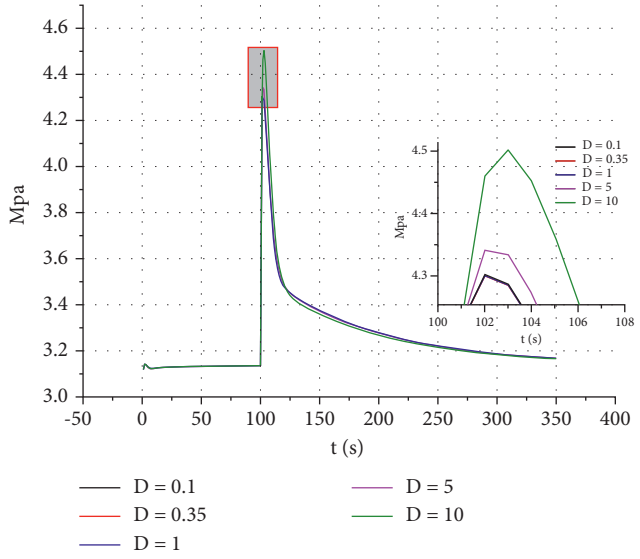

Figure 6: The changes of pressure at different D when P and I are fixed.

action = $[K_p, K_i, K_d]$, and the range of the set values are $K_p = [0, 10]$, $K_i = [0, 0.5]$, and $K_d = [0, 0.5]$.

4.3. Design of Reward Function. The design of the reward function is one of the core problems of the reinforcement learning algorithm, which directly determines whether an agent can achieve the expected goal. The primary problem faced by the design of the reward function is the sparse reward function. The agent only gets a reward when reaches the target value. This sparse reward function is the most common kind of reward, but it often makes the algorithm difficult to converge. The solution is usually to use reward shaping; that is, corresponding rewards are given at each step while approaching the goal instead of giving the final reward only when the episode is terminated. Aiming at the goal of pressure control of the OTSG, this study makes a comparison between the sparse reward and reward-shaping functions:

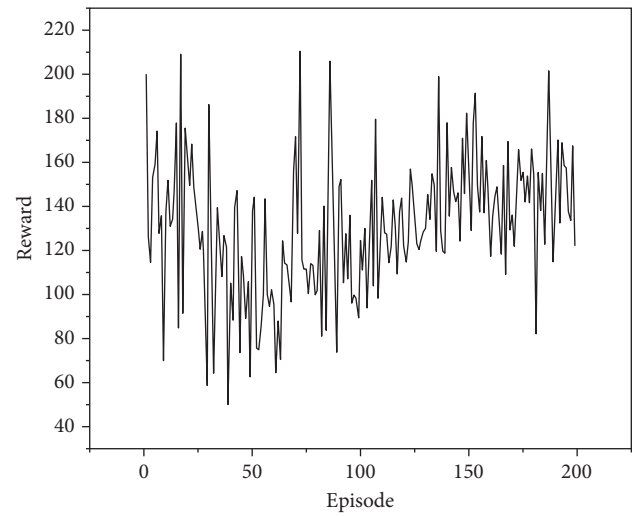

Figure 7: Training effect of sparse reward function.

(1) Sparse reward: in each episode, the reward value r is obtained only when the pressure reaches the set-point; otherwise, the reward is 0:

$$r_t = \begin{cases} 0, & \text{if } |e| \geq 0.01, \\ r, & \text{if } |e| < 0.01. \end{cases} \tag{36}$$

(2) Reward shaping: in each episode, the distance between the pressure and the set value is regarded as the punishment item at each step. After reaching the target, reward 1 can be given:

$$r_t = \begin{cases} -\text{abs}(e), & \text{if } |e| \geq 0.01, \\ 1, & \text{if } |e| < 0.01. \end{cases} \tag{37}$$

After training and testing of the above two reward functions, the training effect diagrams are as follows.

The horizontal axis represents the number of training episodes, and the vertical axis represents the cumulative rewards during the whole episode. It can be seen from Figure 7 that the agent does not obtain steadily increasing rewards through training; that is, the training with sparse rewards does not converge and does not achieve the expected effect. This is because it is difficult for the agent to achieve the target state by random actions, and thus, it is difficult to get the final reward.

In Figure 8, it can be seen that more reward settings of reward-shaping function are used to guide the agent, and the agent can search for the optimal action according to the feedback reward value in each step, that is, PID parameter, so that the action with high reward value can be selected more quickly after training. Therefore, the design of the reward function determines the convergence ability of the reinforcement learning algorithm.

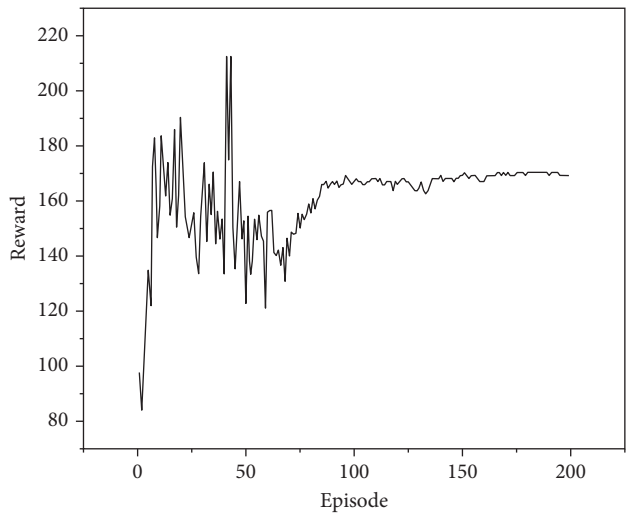

Figure 8: Training effect of reward shaping function.

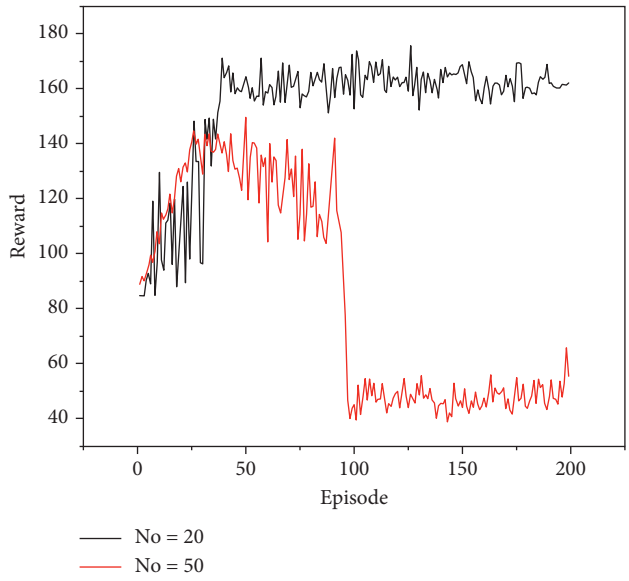

Figure 9: Training effects of critic with different numbers of neurons in hidden layers.

4.4. Design of Algorithm Parameter. PPO algorithm has many hyperparameters and is sensitive to the parameter setting. Settings for PPO algorithm parameters are divided into two parts: parameter settings of neural network and hyperparameter settings of the algorithm. In order to explore the influence of different parameters on the training effect, this study has carried out several experiments on parameter selection.

4.4.1. Settings of Neural Network Parameters. Because the PPO algorithm is based on the framework of Actor-Critic, in which the Actor uses the policy function to generate actions and Critic uses the value function to evaluate the performance of the Actor, the algorithm sets two neural networks. Therefore, the parameters of the two neural networks need to be designed. On the basis of other fixed parameters, the study gives some examples in the setting of Critic network parameters. The comparison curve of the cumulative reward with the different numbers of neurons in hidden layers is shown in Figure 9. The training with 20 neurons converges, but when the number of neurons increases to 50, the reward value curve first gradually increases and then drops off in a cliff, and the training effect deteriorates sharply. The reason is that gradient explosion occurs with more neurons, leading to a poor training effect.

Therefore, after repeated training and verification, the settings of the neural network parameters in this study are shown in Table 1.

Table 1: Settings of neural network parameter.

Parameter	Actor	Critic
Number of neurons in the input layer	3	3
Number of neurons in the hidden layer	20	20
Number of neurons in the output layer	3	1
Activation function	Tanh&softpus	Relu

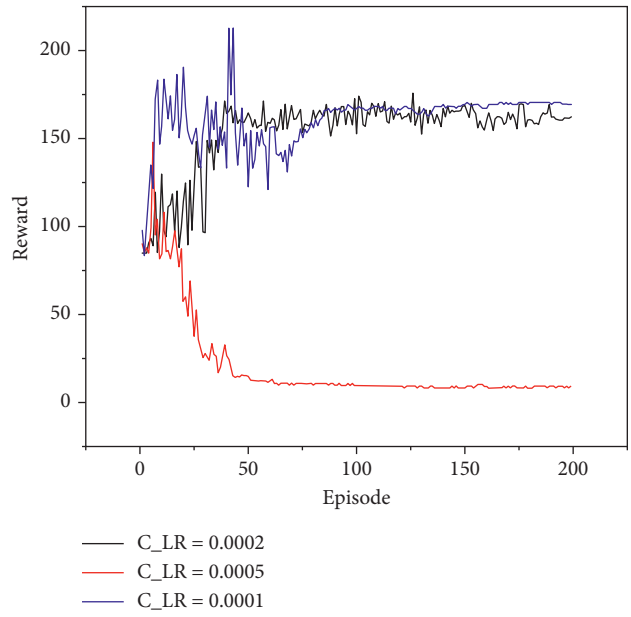

Figure 10: Training effects of critic with different learning rates.

4.4.2. Settings for Hyperparameters of PPO Algorithm. Hyperparameters refer to the parameters' set before the training starts. In the hyperparameter design of the PPO algorithm, the learning rate of the critic neural network is selected as the representative to illustrate the importance of the parameter set to the algorithm. The learning rate of the neural networks will affect the training time and stability.

As can be seen from Figure 10, the comparison of curves with the learning rates of 0.0001 and 0.0002 shows that the learning rate has no significant influence on the final training effect, but the curve with a learning rate of 0.0002 reaches the stable maximum reward value faster than the curve with a learning rate of 0.0001, and the stability is relatively strong. It can be seen that the higher the learning rate is, the faster the agent can learn "knowledge." However, comparing the curves with a learning rate of 0.0005, it can be found that the

Table 2: Settings of hyperparameter for PPO algorithm.

Parameter	Value
Learning rate of actor network	0.0001
Learning rate of critic network	0.0002
Discount factor	0.9
Truncation constant	0.2
Batch size	25
Max episode	200
Max step	250

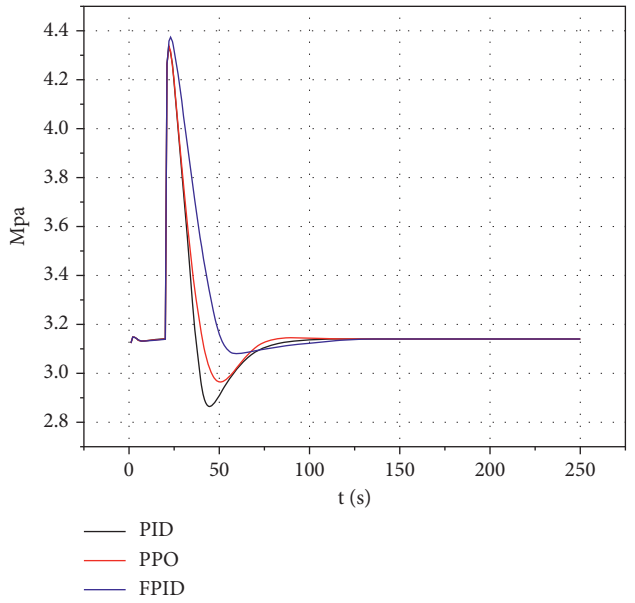

Figure 11: Simulation curve of steam pressure from 100%FP to 70%FP.

Table 3: Performance indicators of the three algorithms when the load decreases from 100% to 70%.

Algorithm	Steady time (s)	Overshoot (%)
PID	148	26.9838
FPID	116	29.0041
PPO	124	26.7704

greater the learning rate, the weaker the stability of training, and there is even a rapid deterioration coming out on training, so the learning rate should not be too large.

Therefore, the parameters of the PPO algorithm selected in this study are as follows (Table 2).

4.5. Comparative Analysis of Simulation Results. In order to test the performance of the trained controller, this study carries out transient tests, anti-interference tests, and tracking tests for OTSG and compares the control effect of the controller with PID and fuzzy PID controllers (FPID), respectively.

4.5.1. Transient Test. In order to test the performance of the trained proposed controller based on the PPO algorithm, the reducing and increasing load tests are carried out, respectively.

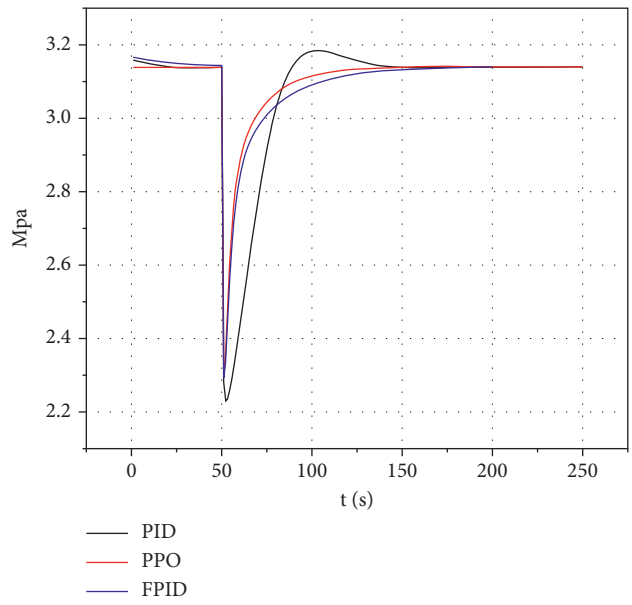

Figure 12: Simulation curve of steam pressure from 70%FP to 100%FP.

Table 4: Performance indicators of the three algorithms when the load increases from 70% to 100%.

Algorithm	Steady time (s)	Overshoot (%)
PID	106	39.3051
FPID	87	38.0808
PPO	61	37.982

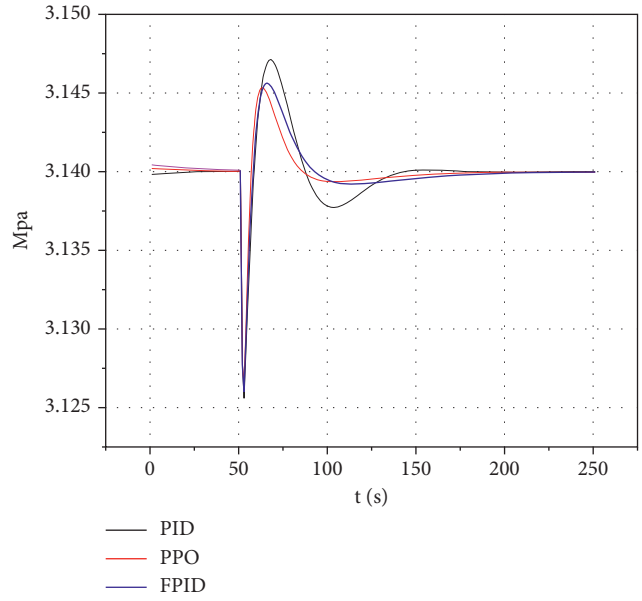

Figure 13: Simulation curve of steam pressure with feed water temperature step up to 80°C.

When the load is reduced from 100% to 70%, the steam outlet valve opening step decreases and the steam pressure rises. When the valve opening is stable, the feedwater flow

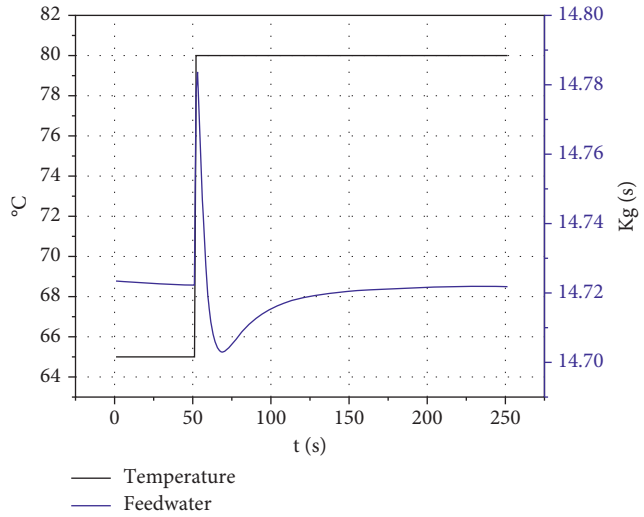

Figure 14: Simulation curves of feedwater flow and feedwater temperature.

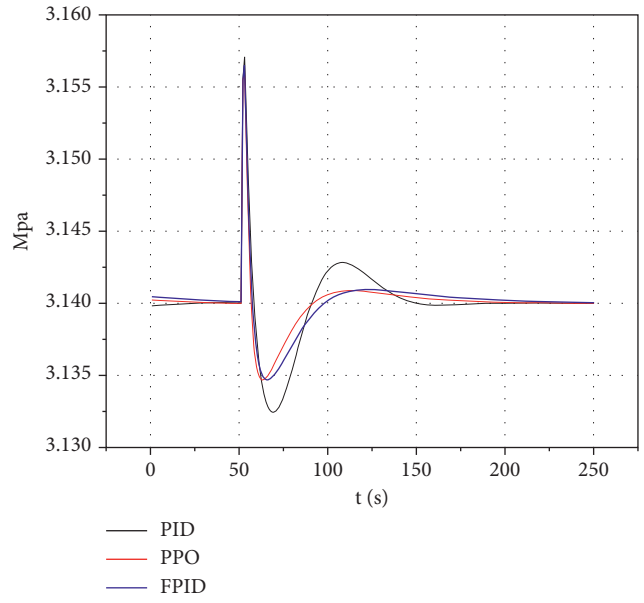

Figure 15: Simulation curve of steam pressure with feed water temperature steps down to 50°C.

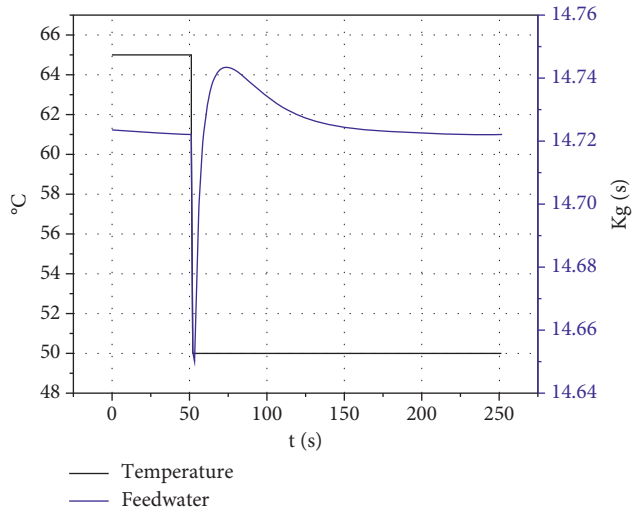

Figure 16: Simulation curves of feedwater flow and feedwater temperature.

Table 5: Performance indicators of the three algorithms when the feedwater temperature step rises to 80°C.

Algorithm	Steady time (s)	Overshoot (%)
PID	100	0.4582
FPID	71	0.445
PPO	50	0.4321

Table 6: Performance indicators of the three algorithms when the feedwater temperature steps down to 50°C.

Algorithm	Steady time (s)	Overshoot (%)
PID	87	0.5429
FPID	74	0.5254
PPO	45	0.508

Table 7: Performance indicators of the three algorithms when set point steps up from 3.14 MPa to 3.15 MPa.

Algorithm	Steady time (s)	Overshoot (%)
PID	92	0.3468
FPID	145	0.318471
PPO	81	0.3171

gradually decreases with the valve opening changes and the steam pressure drops and tends to be stable. Figure 11 shows the steam pressure change curve when the load decreases from 100% to 70% under the two control algorithms. It can be seen that the controller with PPO algorithms not only has a smaller overshoot than the PID controller but also stabilizes faster. It can be concluded from Table 3 that, compared with FPID, FPID stabilization time is faster, but the overshoot 29.0041% is larger than PID and PPO. Figure 12 and Table 4 show the simulation curve of steam pressure when the load increases from 70% to 100%. PPO control not only reduces the overshoot but also improves the response speed of the system.

4.5.2. Anti-Interference Test. In order to verify the capability of the proposed controller, the feedwater temperature disturbance test is carried out. In the test process, the step disturbance of feed water temperature was added at 50 s. The test results are shown in Figures 10–13 and Tables 5 and 6. It can be seen from the figures that the three methods have excellent anti-interference ability and can quickly restore the water level to its normal state (Figure 14).

When the feedwater temperature rises, the steam pressure in the secondary loop rises rapidly, and the heat absorption from the pipe wall increases, so the pressure gradually decreases. When the feedwater temperature drops, the steam pressure in the secondary loop decreases rapidly,

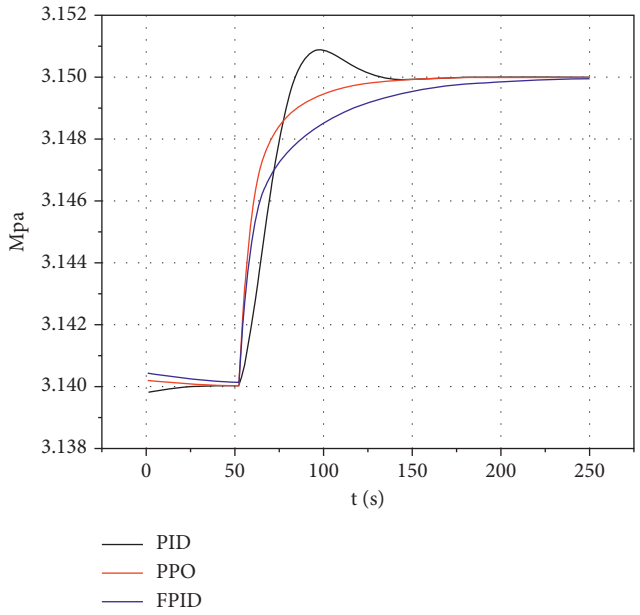

Figure 17: Simulation curve of steam pressure step up to 3.15 MPa.

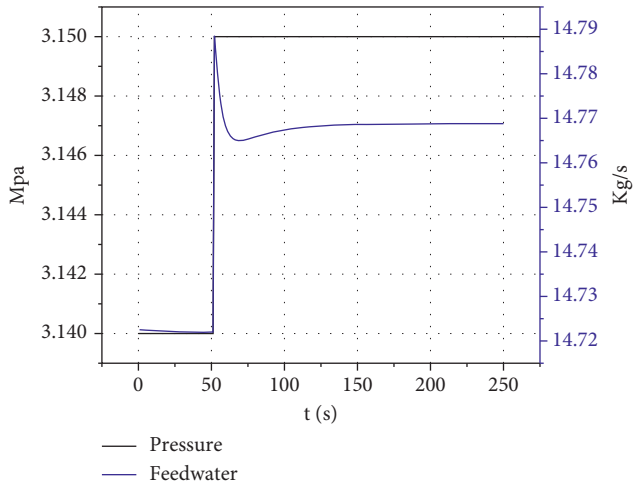

Figure 18: Simulation curves of pressure and feedwater flow.

and its heat absorption from the pipe wall also drops, so the pressure gradually increases. Figures 13 and 15, respectively, show the curve of steam pressure change when the feedwater temperature step rises to 80°C and the steep drops to 50°C. The results show that the controller designed in this study has some advantages over the PID and FPID controllers in terms of response time, overshoot, and control error. From the simulation results, the controller based on reinforcement learning can well guarantee the high stability of the OTSG (Figure 16 and Tables 5–6).

4.5.3. Tracking Test. In order to test the response of the controller under step function, the pressure set point steps up from 3.14 MPa to 3.15 MPa at 10 s during the test. Figure 17 shows a comparison of the three methods at the full power level. The three methods can effectively adjust the pressure. Compared with the other two methods, the proposed method has a faster response speed and lower overshoot (Table 7 and Figure 18).

5. Conclusions

In this study, a PPO algorithm of reinforcement learning is applied for the control of OTSG. A double-layer pressure control structure of OTSG is designed in this study, which realizes the parameter adjustment policy of online learning in the upper layer and the adaptive adjustment of parameters of the PID controller in the bottom layer. The results of simulations show that the controller based on the PPO algorithm proposed in this study can realize the self-tuning of PID parameters under all kinds of working conditions and has the advantages of fast response speed and strong adaptive ability.

However, our method is not perfect and there are some limitations. First, the convergence of the algorithm depends on the setting of the reward function. The reward function needs to be set artificially according to different objects, and the algorithm will not be able to converge with an unreasonable reward function. Second, the hyperparameters of the PPO algorithm need to be regulated relying on experience or trial and error to get better performance. In future work, we need to improve the portability of the algorithm when applying our method to practical work so that the algorithm performs equally well in different situations.

Data Availability

The design data of the once-through steam generator used to support the findings of this study were supplied by China Nuclear Power Operation Technology Corporation under license and so cannot be made freely available. Requests for access to these data should be made to the primary author. And the parameters used in the algorithm proposed are listed in detail within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Yao, G. Chen, K. Lu et al., "Study on the systematic thermal-hydraulic characteristics of helical coil once-through steam generator," *Annals of Nuclear Energy*, vol. 154, Article ID 108096, 2021.

[2] G. Zhao, Y. Zhao, and J. Liu, "Integral control strategy between the casing once-through steam generator and the turbin," *Energy Conservation Technology*, vol. 38, no. 2, pp. 162–166, 2020.

[3] Y. Zhang, M. Zheng, Z. Ma, and J. Wu, "Dynamic modeling ,simulation and control of helical coiled once-through steam generator," *Applied Science and Technology*, vol. 47, no. 6, pp. 71–77, 2020.

[4] S. Cheng, L. Cheng, M. Peng, and X. Liu, "Research of pressure control based on artificial immune control of once-

through steam generator," *Nuclear Power Engineering*, vol. 36, no. 3, pp. 62–65, 2015.

[5] Z. Chen, L. Liao, L. Liu, and W. Li, "Study on application of T-S fuzzy neural method in once-through steam generator feedwater control," *Nuclear Power Engineering*, vol. 33, no. 4, pp. 20–23 + 33, 2012.

[6] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[7] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3/4, pp. 279–292, 1992.

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

[9] J. Li, J. Geng, and T. Yu, "Grid-area coordinated load frequency control strategy using large-scale multi-agent deep reinforcement learning," *Energy Reports*, vol. 8, pp. 255–274, 2022.

[10] J. Li, T. Yu, and X. Zhang, "Coordinated automatic generation control of interconnected power system with imitation guided exploration multi-agent deep reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 136, Article ID 107471, 2022.

[11] X. Qiu, C. Gao, K. Wang, and W. Jing, "Attitude control of a moving MassA-ctuated UAV based on deep reinforcement learning," *Journal of Aerospace Engineering*, vol. 35, no. 2, 2022.

[12] X. Deng, Y. Zhang, and H. Qi, "Towards optimal HVAC control in non-stationary building environments combining active change detection and deep reinforcement learning," *Building and Environment*, vol. 211, 2022.

[13] R. Zhang, Q. Lv, J. Li, J. Bao, T. Liu, and S. Liu, "A reinforcement learning method for human-robot collaboration in assembly tasks," *Robotics and Computer-Integrated Manufacturing*, vol. 73, 2022.

[14] J. K. Park, T. K. Kim, and S. H. Seong, "Providing support to operators for monitoring safety functions using reinforcement learning," *Progress in Nuclear Energy*, vol. 118, no. C, Article ID 103123, 2020.

[15] J. K. Park, T. K. Kim, S. SeungHwan, and S. R. Koo, "Control automation in the heat-up mode of a nuclear power plant using reinforcement learning," *Progress in Nuclear Energy*, vol. 145, 2022.

[16] Z. Dong, X. Huang, Y. Dong, and Z. Zhang, "Multilayer perception based reinforcement learning supervisory control of energy systems with application to a nuclear steam supply system," *Applied Energy*, vol. 259, no. C, Article ID 114193, 2020.

[17] I. I. Belyakov, M. A. Kvetnyi, D. A. Loginov, and S. I. Mochan, "Static instability of once-through steam generators with convective heating," *Soviet Atomic Energy*, vol. 56, no. 5, pp. 347–350, 1984.

[18] M. Osakabe, "Thermal-hydraulic study of integrated steam generator in PWR," *Journal of Nuclear Science and Technology*, vol. 26, no. 2, pp. 286–294, 1989.

[19] Z. Wang, Z. Shi, Y. Li, and J. Tu, "The optimization of path planning for multi-robot system using Boltzmann Policy based Q-learning algorithm," in *Proceedings of the 2013 IEEE International Conference on Robotics and Biomimeties(ROBIO)*, pp. 1199–1204, Shenzhen, China, December 2013.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 2018.

[21] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *Journal of Artificial Intelligence Research*, vol. 15, no. 1, pp. 319–350, 2001.

[22] Y. Duan, Xi. Chen, and R. Houthooft, "Benchmarking deep reinforcement learning for continuous control," in *Proceedings of the International Conference on Machine Learning(ICML)*, New York, NY, USA, June 2016.

[23] Y. H. Wu, Z. C. Yu, C. Y. Li, M. J. He, B. Hua, and Z. M. Chen, "Reinforcement learning in dual-arm trajectory planning for a free-floating space robot," *Aerospace Science and Technology*, vol. 98, Article ID 105657, 2020.