Nuclear Engineering and Technology 54 (2022) 3283–3292

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Fault-tolerant control system for once-through steam generator based on reinforcement learning algorithm

Cheng Li [a, b], Ren Yu [a, *], Wenmin Yu [a], Tianshu Wang [a]

[a] *Naval University of Engineering, Wuhan, 430033, China*
[b] *China Nuclear Power Operation Technology Corporation LTD, Wuhan, 430000, China*

A R T I C L E I N F O

Article history:
Received 16 September 2021
Received in revised form
17 April 2022
Accepted 18 April 2022
Available online 27 April 2022

Keywords:
Once-through steam generator
Reinforcement learning
DQN algorithm
Incremental action

A B S T R A C T

Based on the Deep Q-Network(DQN) algorithm of reinforcement learning, an active fault-tolerance method with incremental action is proposed for the control system with sensor faults of the once-through steam generator(OTSG). In this paper, we first establish the OTSG model as the interaction environment for the agent of reinforcement learning. The reinforcement learning agent chooses an action according to the system state obtained by the pressure sensor, the incremental action can gradually approach the optimal strategy for the current fault, and then the agent updates the network by different rewards obtained in the interaction process. In this way, we can transform the active fault tolerant control process of the OTSG to the reinforcement learning agent's decision-making process. The comparison experiments compared with the traditional reinforcement learning algorithm(RL) with fixed strategies show that the active fault-tolerant controller designed in this paper can accurately and rapidly control under sensor faults so that the pressure of the OTSG can be stabilized near the set-point value, and the OTSG can run normally and stably.

1. Introduction

The once-through steam generator(OTSG) is a key part of the nuclear power plant, the research on the control of the OTSG mainly focus 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 [1]. The OTSG control system has a large number of sensors, due to equipment fatigue and changes in internal and external conditions, sensors may appear precision decline, indicator drift, or complete failure in the long-term high temperature and high-pressure operation process. This may cause the operation of the nuclear power plant to deviate from the optimal state, and affect the unit's economy, or even the safety of the whole plant [2]. At present, most nuclear power plants use hardware redundancy to deal with the fault tolerance of equipment, which not only requires a large amount of money and manpower input but also brings safety and reliability problems in the design and construction stage. However, the development of fault-tolerant control technology has greatly improved the defects caused by hardware redundancy [3]. Therefore, it is of great practical significance to study the fault-tolerant control method of the OTSG control system with faults for improving the safety, reliability, and economy of the nuclear power plants.

To overcome the faults of sensors, actuators, and other components, scholars at home and abroad have made a lot of efforts in the research of fault diagnosis and fault-tolerant control. In 1986, the concept of fault-tolerant control was formally proposed by the national science foundation and IEEE control systems society at the control symposium held at Santa Clara University. In 1993, the first comprehensive paper on fault-tolerant control appeared in the world, comprehensively describing the problems and basic solutions of fault-tolerant control technology [4]. In 2004, the work on fault diagnosis and fault-tolerant control compiled by Mogens Blanke et al. provided a relatively complete theoretical basis for fault-tolerant control [5]. For the controlled system under failure, the method of reliable stabilization control can be adapted to control the target system by connecting multiple compensation controllers in parallel simultaneously, to maintain the stability of the system when the system is under failure [6]. In addition, Gopinathan and Boskovic et al. put forward an idea of integrity control, which can still ensure the stability and security of the system when sensor or actuator failure occurs in the system [7]. Methods above are passive fault-tolerant control which does not need online monitoring and evaluation when the system fails and has certain robustness, but the design of passive fault-tolerant control system needs prediction of the possibility of all kinds of

* Corresponding author.
E-mail address: 18071068480@163.com (R. Yu).

https://doi.org/10.1016/j.net.2022.04.014

faults, and the implementation is usually very complex. The performance of the passive fault-tolerant control system cannot be optimized because it is also too conservative.

Different from passive fault-tolerant control, active fault-tolerant control ensures the stable operation of the system through online fault evaluation and control signal reconstruction. Most active fault tolerant control systems need to acquire online fault information in real-time through fault detection and diagnosis unit(FDD) and adjust the parameters or structure of the control system according to the fault information, and then reconstruct the control system, which enables the system to run normally after failure [8].

In recent years, with the rise of big data and artificial intelligence, scholars at home and abroad have begun to pay attention to fault-tolerant control based on intelligent learning methods. Zhanshan Wang et al. [9] studied the fault-tolerant control problem of multiple-input multiple-output of nonlinear discrete systems based on reinforcement learning method, sought the optimal control signal and new cost function after failure by utilizing the approximation ability of the neural network. For actuator and sensor faults of n-order nonlinear systems, Faezeh Farivar et al. [10] combined reinforcement learning with traditional control theory and designed a fault-tolerant controller based on robust control and reinforcement learning, which constructed evaluation methods and action strategies using neural networks.

In this paper, an active fault-tolerant control scheme based on the Deep Q-Network(DQN) algorithm of reinforcement learning is proposed for the OTSG control system with sensor faults, and incremental action for the reinforcement learning method is innovatively proposed. Compared with the model-based active fault-tolerant control method, fault-tolerant control based on reinforcement learning can extract fault features according to the system state, and give the optimal strategy under the current condition at the same time, so that the system can maintain stability.

Because of the uncertainty of the control system failure, the effects of the fault-tolerant control systems based on the deterministic strategy of reinforcement learning are discouraging. This paper proposes the incremental action instead of deterministic strategy which reconstructs the controller by incremental iteration in the fault-tolerant control system, so as to achieve the optimal solution to implement the fault-tolerant control.

2. Non-linear mathematical model of the OTSG

2.1. Model simplification and assumptions

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 inner tube of the casing pipe and the shell part of the outer tube, and the secondary fluid flows from bottom to top in the annulus channel of the casing pipe and then flows out the superheated steam. The OTSG can be divided into three regions: subcooled, nucleate boiling, superheat regions [11]. The input and output of the mathematical model are as Fig. 1

The mass, energy, and state equations were established by using lumped parameter method. The outlet parameters were selected as lumped parameters to reflect the variation of medium parameters in the whole pipe section, and the average inlet and outlet parameters were selected as lumped parameters to reflect the average medium parameters in the whole pipe section. The temperature and enthalpy, the outlet parameters of each section of the primary and secondary loops are considered as lumped parameters [12]. Considering the variation of gaseous density with pressure, the density of the boiling region can be illustrated by the average density of the inlet and outlet, and the whole density of the superheated region can be substituted by the inlet density.

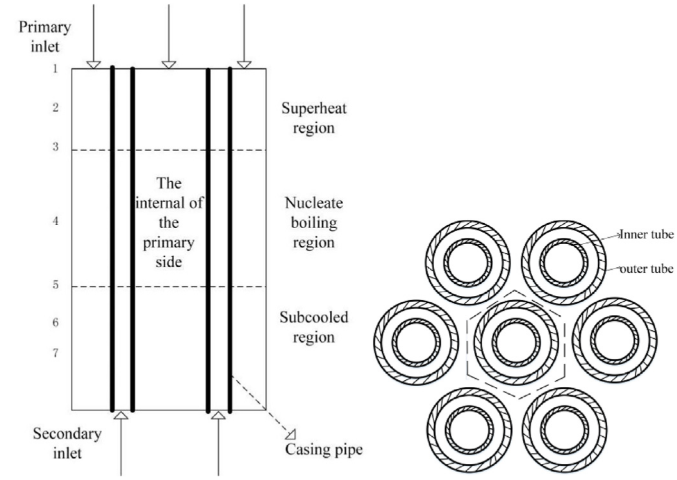

Fig. 1. The schematic diagram of the OTSG.

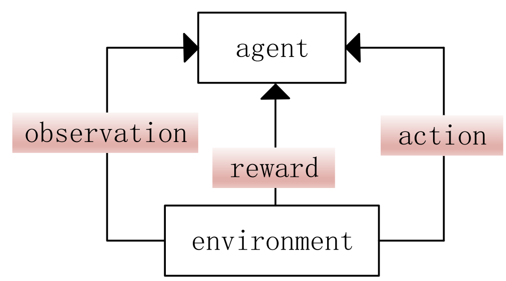

Fig. 2. The framework of reinforcement learning.

The assumptions of the model are as follows :

(1) The steam generator mathematical model is assumed to one-dimensional model.
(2) The subcooled region, nucleate boiling region, superheat region are regarded as independent heat exchangers.
(3) The density of the whole liquid flow region is considered to be a constant, and the average density of the gas flow region is considered to change with pressure.
(4) The heat transfer coefficient is treated as a constant.
(5) The heat fluxes in the inner tube, the shell part of the outer tube, and the annulus channel of the casing pipe are assumed to be equal.

The mathematical model of the OTSG is shown below:

(1) Subcooled region: $h \le hf$ (hf is the enthalpy of saturated water)

Energy conservation equation in the primary subcooled region:

$$\frac{d(\rho_p l_6 A_p h_{p7})}{dt} = w_{p5}h_{p5} - w_{p7}h_{p7} - Q_{p6} \tag{1}$$

Energy conservation equation in the secondary subcooled region:

$$\frac{d(\rho_{s6} l_6 A_s h_{s5})}{dt} = w_{s7}h_{s7} - w_{s5}h_{s5} + Q_{p6} \tag{2}$$

Mass conservation equation in the secondary subcooled region:

$$\frac{d(\rho_{s6} l_6 A_s)}{dt} = w_{s7} - w_{s5} \tag{3}$$

(2) Nucleate boiling region: $0 < x < 1$ (x is dryness)

Energy conservation equation in the primary nucleate boiling region:

$$\frac{d(\rho_p l_4 A_p h_{p5})}{dt} = w_{p3} h_{p3} - w_{p5} h_{p5} - Q_{p4} \tag{4}$$

Energy conservation equation in the secondary nucleate boiling region:

$$\frac{d(\overline{\rho_{s4}} l_4 A_s h_{s3})}{dt} = w_{s5} h_{s5} - w_{s3} h_{s3} + Q_{p4} \tag{5}$$

Mass conservation equation in the secondary nucleate boiling region:

$$\frac{d(\overline{\rho_{s4}} l_4 A_s)}{dt} = w_{s5} - w_{s3} \tag{6}$$

(3) Superheat region: the secondary saturates steam to the outlet

Energy conservation equation in the primary superheat region:

$$\frac{d(\rho_p l_2 A_p h_{p3})}{dt} = w_{p1} h_{p1} - w_{p3} h_{p3} - Q_{p2} \tag{7}$$

Energy conservation equation in the secondary superheat region:

$$\frac{d(\overline{\rho_{s2}} l_2 A_s h_{s1})}{dt} = w_{s3} h_{s3} - w_{s1} h_{s1} + Q_{p2} \tag{8}$$

Mass conservation equation in the secondary superheat region:

$$\frac{d(\overline{\rho_{s2}} l_2 A_s)}{dt} = w_{s3} - w_{s1} \tag{9}$$

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. p and s represent primary and secondary loops individually.

3. Design of active fault-tolerant controller for the OTSG based on reinforcement learning

The core of the reinforcement learning 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 effect of fault-tolerant control. This section will define the state space of the active fault-tolerant controller, design the reward function, and introduce the incremental action.

3.1. DQN algorithm

Reinforcement learning is a process in which an agent interacts with the environment continuously to obtain different reward values for training. Fig. 2 shows the interaction process of reinforcement learning. The Q-learning algorithm of reinforcement learning is a value iterative algorithm, which calculates each Q value and updates the Q value table according to the prior knowledge learned by the agent [13]. However, when the state of the environment becomes complex, the state-action space, namely the Q value table, will become very complex, leading to the problem of "dimension disaster", which makes the model cannot be calculated. Fortunately, the emergence of deep reinforcement learning can effectively solve this problem.

The representative algorithm of deep learning is the DQN. DQN algorithm combines the advantages of deep neural networks and Q-learning. The neural network is responsible for modeling the Q value table which realizes the representation of all the state-action values. Q-learning is modeled by Markov decision which is represented by the current state, action, reward, strategy, and next action [14]. DQN improves the relevance and efficiency of samples by setting the experience replay and improves the stability of updates by regularly updating the target Q network. The core of the DQN algorithm includes three points: objective function, network setting, and experience replay:

(1) Objective function: The objective function of DQN is constructed by Q-learning, and the formula is shown below:

$$Q'(s,a) \leftarrow Q(s,a) + \alpha[r + \max Q(s',a') - Q(s,a)] \tag{14}$$

where, (s,a) represents the current state and action, (s′,a') represents the state and action at the next moment, Q(s,a) represents the current state-action value, Q′(s,a) represents the updated state-action value, and r represents the action reward at the state s.

The target state-action value function can be expressed by the Behrman equation as follows:

$$y' = r + \gamma \max Q(s',a',\theta) \tag{15}$$

where y' represents the target Q value.

The loss function is the mean square error loss function:

$$L(\theta) = E\left[(y' - Q(s,a,\theta))^2\right] \tag{16}$$

where θ is the weight parameter of the neural network model.

(2) Network setting. DQN evaluates the current state-action value function through the target network and Q network. The target network estimates the Q value at the next moment and solves the problem of the "dimension disaster" of the Q value table. Q network uses stochastic gradient descent to update network weight θ, and the gradient descent algorithm formula is as follows:

$$\Delta\theta = E[y' - Q(s,a,\theta_1)]\nabla_\theta Q(s,a,\theta_1) \tag{17}$$

(3) Experience replay. Experience replay solves the problem of sample relevance and efficiency utilization of sample. When an agent interacts with the environment, samples can be obtained from the environment, and the sample data can be stored in the experience pool. We can train the neural network by randomly selecting a small batch of samples from the experience pool at a time. The main purpose of

Table 1
Reconstruction scheme of fault-tolerant controller based on reinforcement learning.

Reconstruction scheme	Scheme state	Scheme illustration
A_0	[0]	The system is normal and no action
A_1	[-0.001]	Negative action to compensate sensor fault
A_2	[0.001]	Forward action to compensate sensor fault
A_3	[-0.0001]	Negative action to compensate sensor fault
A_4	[0.0001]	Forward action to compensate sensor fault

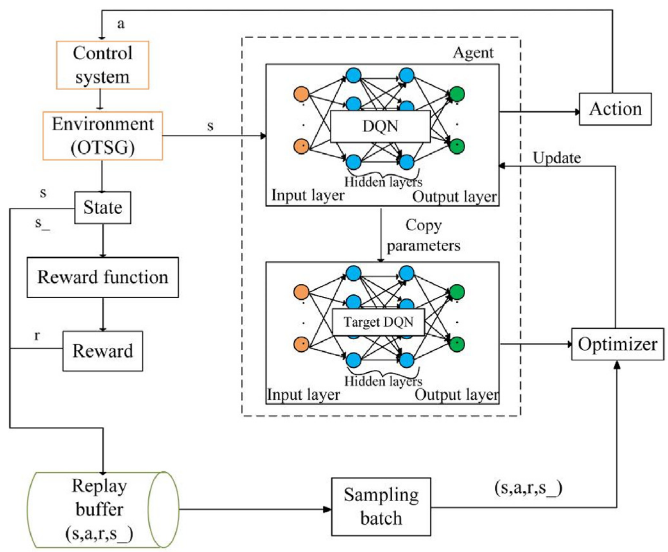

Fig. 3. The framework of the fault-tolerant controller based on reinforcement learning.

experience replay is to improve learning efficiency by reusing samples.

3.2. Design of state-space

The state-space parameters are selected from the control system, which can reflect the current operating state of the control system. The fault-tolerant controller extracts information through the state space parameters and takes them as an important basis for fault-tolerant decision-making. 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 oscillates easily. If the water supply cannot keep up with the pace of changes at this time, the equipment in the secondary loop will have an impact [15]. Therefore, in order to represent the dynamic characteristic of the OTSG, we choose the steam pressure as the state space. This article 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 deviation value in last time$e$ (t-1).

Table 2
Main parameters of the OTSG.

Operating parameters at Full power	
Inlet pressure at primary side	15.2MPa
Total coolant flow rate	12.45 kg/s
Coolant inlet temperature	298°C
Coolant outlet temperature	265°C
Steam pressure	3.14 MPa
Feedwater temperature	65°C
Feedwater flow	0.783 kg/s
Full capacity	3.0MW

3.3. Design of reward function

The reward function can be regarded as the score to evaluate the actions generated by the fault-tolerant controller, and the score is the loss index of training. The optimal action can be determined by minimizing the loss, so the reward function directly determines whether the agent can achieve the expected goal. 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.

$$r_t = \begin{cases} -abs(e), if\,|e| \geq 0.01 \\ 1, if\,|e| < 0.01 \end{cases} \tag{18}$$

3.4. Design of incremental action

Traditional reinforcement learning methods often adopt deterministic strategies in non-controlled fields. Google DeepMind team developed an artificial intelligence chess agent by using the deep reinforcement learning method, and the strategy adopted is the deterministic strategy[16].

The fault of the control system is unpredictable and the fault-tolerant controller based on deterministic strategy will be limited. In this paper, the DQN algorithm of reinforcement learning based on incremental actions is proposed to achieve optimal actions. Moreover, the method enriches the diversity of actions and gives full play to the self-learning characteristics of reinforcement learning agents.

3.4.1. Action selection method

In this paper, the ε-greedy factor is introduced into the action selection of the fault-tolerant controller, and the ε-greedy factor represents the probability of random strategy selection. The ε-greedy factor is annealed from 1 to 0 during the training process. In the initial exploration stage, the controller has a high probability of random strategy selection. As the action strategy is iterated to the later stage, the controller will choose the optimal strategy which has a high probability. At the end of the training, when the controller's decision is fully mature, the ε-greedy factor is annealed to zero. The ε -greedy strategy is expressed as follows:

$$a = \begin{cases} \text{random action } a \in A,\ d < \varepsilon \\ \underset{a}{\operatorname{argmin}} \widehat{Q}(s, a) \end{cases} \tag{19}$$

where: a is the action of the controller, that is, the incremental adjustment value of the feedwater flow for the OTSG; A is the action set, where the elements are all possible actions selected by the controller. In contrast to the state-action value function Q(s, a), Qˆ(s, a) is the estimate of the state-action value in the current state s and action a.

3.4.2. *Incremental action*

The output of the fault-tolerant controller based on reinforcement learning is the reconstruction scheme of the control system. Reconstruction schemes can be listed through enumeration. Different reconstruction schemes represent incremental actions with different steps. During the training process, controller actions are incrementally superposed and updated, so as to obtain the optimal strategy.

$$\Omega = \{A_0, A_1, A_2, A_3, A_4\}$$

Ω is reconstruction scheme set for the control system, as shown in Table 1, the controller action is incremental, and divided into reverse action and positive action. The controller is fault-tolerant by the compensation method. This article design specific strategies for the once-through steam generator sensor fault, considering the controller training convergence speed and the accuracy of the optimal strategy, the incremental action take steps ±0.0001 and ± 0.001.

4. Active fault-tolerant controller structure and algorithm implementation

The fault-tolerant controller structure is based on DQN as shown in Fig. 3. The DQN algorithm sets up a dual network, the Q network, and the target network. The target Q value remains unchanged for a while by introducing the target network, which reduces the relevance between the current Q value and target Q value to a certain extent, thereby reducing the possibility of loss value oscillation at training, to improve the stability of the algorithm.

The agent receives states and rewards from the environment and sends actions to the environment., and explores the optimal action for the fault-tolerant controller through the interaction with the environment. DQN creates a replay buffer to store historical experiences and then randomly selects samples from it and feeds those samples to update the Q network. 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.

The DQN uses 2 neural networks, namely Q network and target Q network, the role of each network are as below:

(1) Q network: selects an action according to the state s, the agent interacts with the environment(OTSG) to generate the next state s_ and reward r. (s, a, r, s_) is stored in the replay buffer. When the number of samples stored in the experience pool reaches a certain amount, small-batch samples are taken out to train the network.

(2) Target network: generates the target Q value. Before each batch size step, the parameters are regularly copied from the Q network to the target network to provide a stable training target for the Q network, so that the estimation error can be controlled and the possibility of model oscillation and divergence can be solved to a certain extent.

The pseudocode of DQN is given in Algorithm 1.

Algorithm1. DQN

Algorithm1 DQN

1:Input: state-space parameter s, Q network parameter θ, and target Q network parameter θ'

2:Output: increment action for the feedwater flow of OTSG

3:Initial: initial s, replay buffer D with size n, Q network with weight θ, target network with weight θ', θ'= θ

4:for episode=1,N do

5: for t=1,T do

6: Input the initial state space parameter s_t

7: Select a random action with a certain probability ε, otherwise select the optimal action $a_t = \max_a Q'((s_t), a; \theta)$

8: Execute the action a_t in the environment to obtain the reward r_t and the next moment state s_{t+1}

9: Save (s_t, a_t, r_t, s_{t+1}) to the replay buffer D

10: Randomly select a small batch of samples from the replay buffer D to update the Q network

11: Calculate $y_j(^{a}|^{s}) = \begin{cases} r_j, \text{for terminal} \\ r_j + \gamma \max Q(s_j + 1, a'; \theta), \text{for non} - \text{terminal} \end{cases}$

12: The loss function is $L(\theta) = E\left[\; Q(s, a\; \theta))^2 \right]$

13: Execute θ'= θ every M steps

14: end for

15: end for

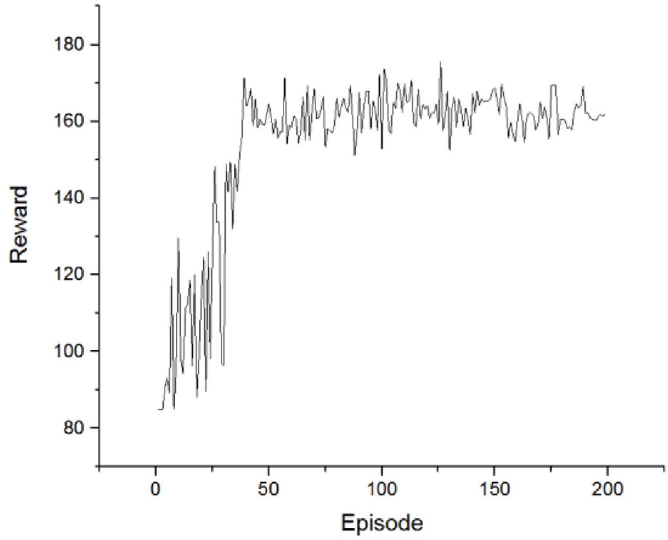

Fig. 4. Training effects of DQN.

5. Experiments and results

The 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.

The main parameters of the OTSG are shown in Table 2.

Fig. 4 shows the reward curve of the active fault-tolerant control based on reinforcement learning in the training process. The horizontal axis represents the number of training episodes, and the vertical axis represents the total amount of reward earned during the episode. The reward function is defined in formula (18), when the pressure deviation is less than the set value, the reward can be 1.

As can be seen from the reward curve, the agent can be guided by more reward settings to find the optimal action according to the reward of real-time feedback, so that it can choose the action with high reward value more quickly after training and learning, and gradually converge.

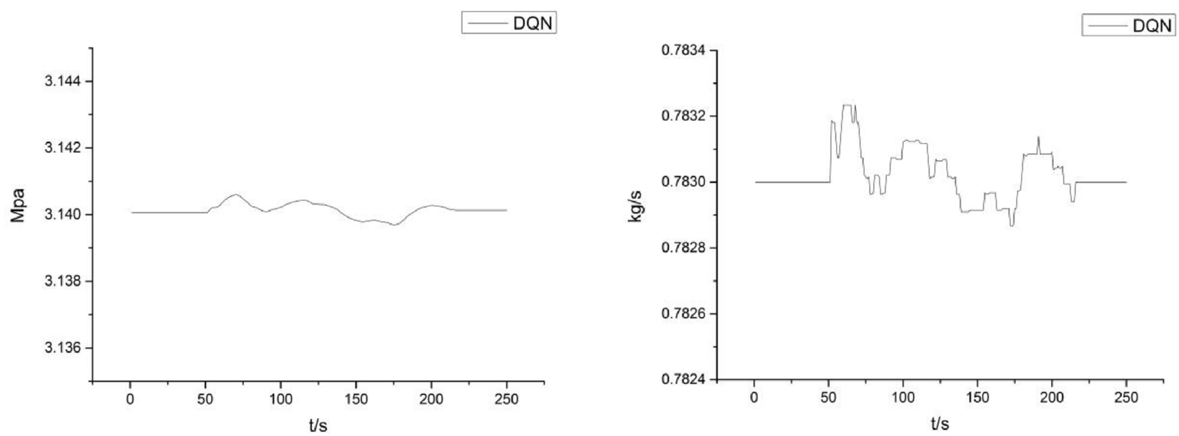

Fig. 5. The pressure and feedwater curves with DQN control under constant deviation fault of the pressure sensor.

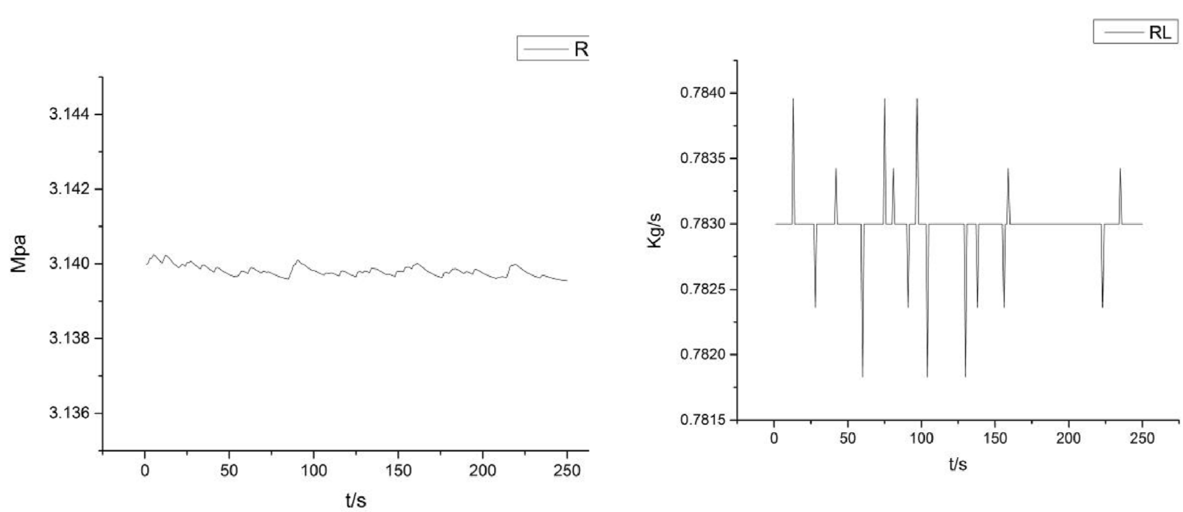

Fig. 6. The pressure and feedwater curves with RL control under constant deviation fault of the pressure sensor.

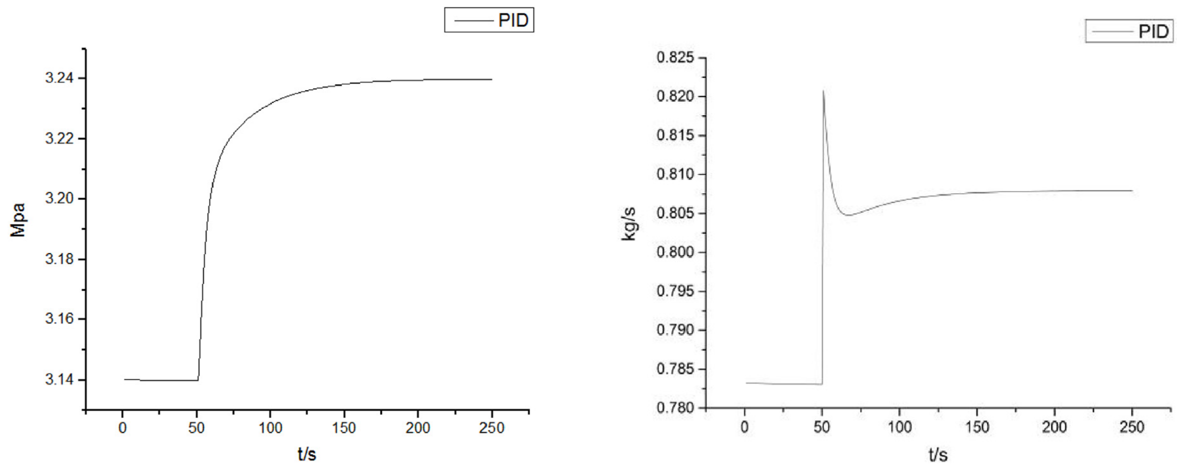

Fig. 7. The pressure and feedwater curves without fault tolerance control under constant deviation fault of the pressure sensor.

In order to verify the effect of the fault-tolerant controller for the OTSG, the active fault-tolerant controller after training is tested under three kinds of faults of the pressure sensor: constant deviation, constant gain and stuck.

To compare the proposed intelligent fault-tolerant control method, the traditional reinforcement learning algorithm(RL) with fixed strategies is selected as the comparison method. At the same time, in order to explain the performance of the system without fault-tolerant control, PID control without fault-tolerant control is selected, that is, sensor fault information is not provided to PID (PID

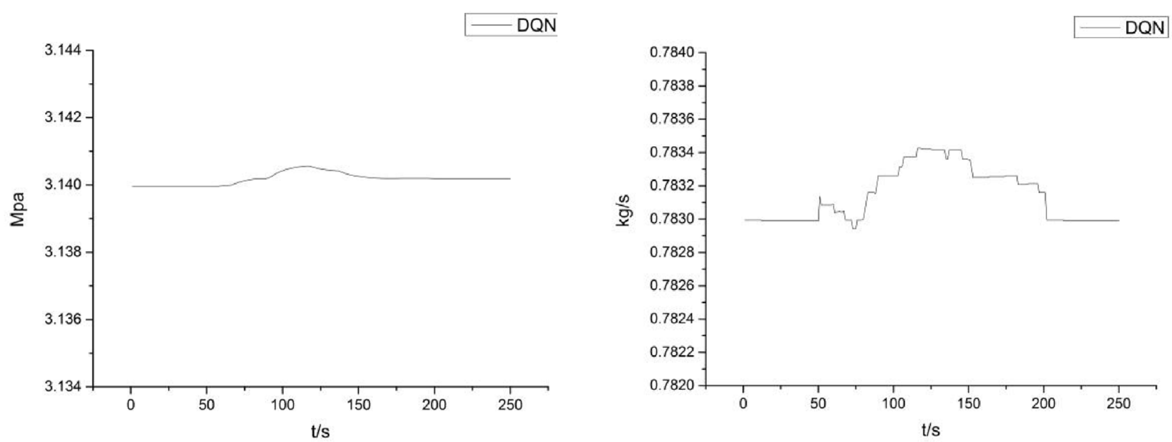

Fig. 8. The pressure and feedwater curves with DQN control under constant gain fault of the pressure sensor.

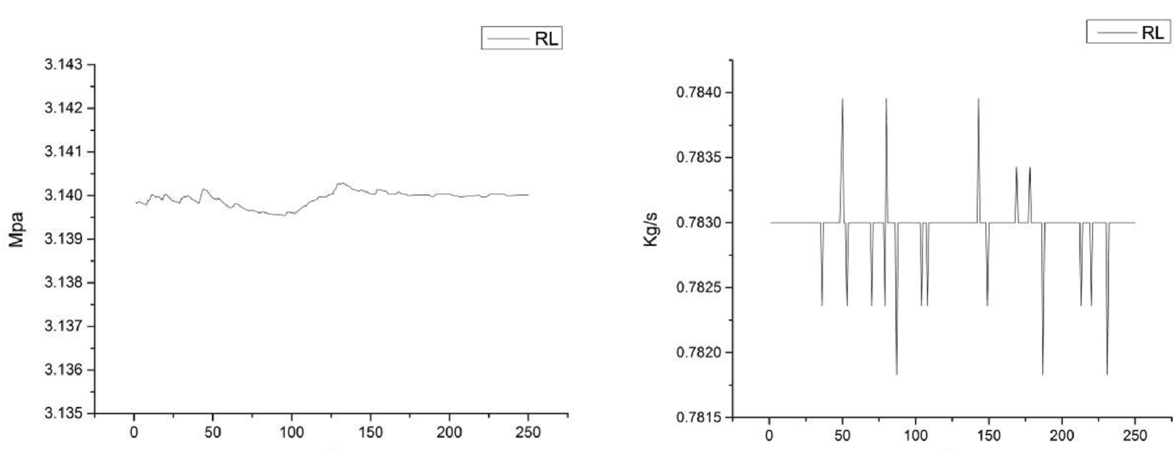

Fig. 9. The pressure and feedwater curves with RL control under constant gain fault of the pressure sensor.

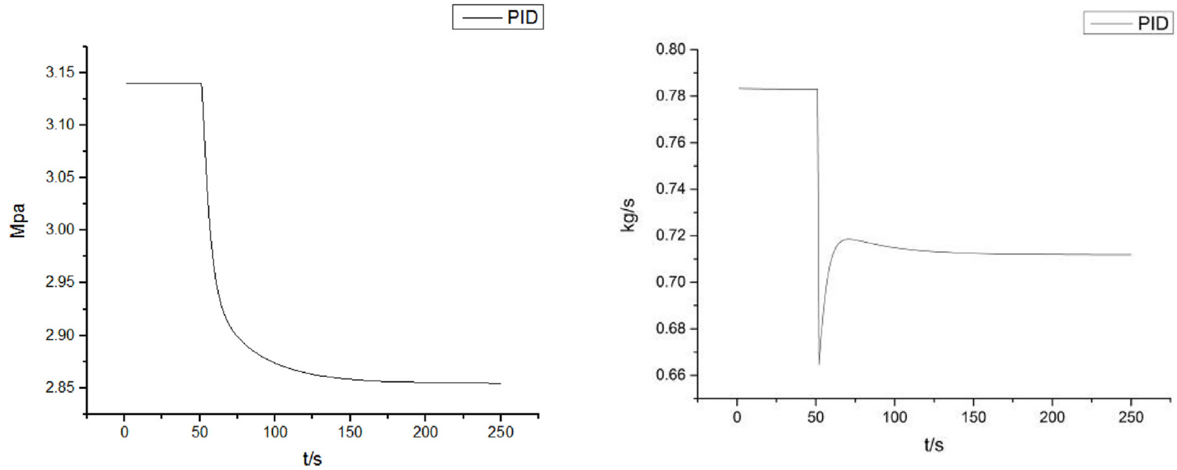

Fig. 10. The pressure and feedwater curves without fault tolerance control under constant gain fault of the pressure sensor.

can also achieve fault-tolerant control when providing fault information, but it is not the focus of this paper).

5.1. Experiment of the sensor under constant deviation fault

When the OTSG is stable running for 50s, the constant deviation fault of the pressure sensor is added with the fault degree

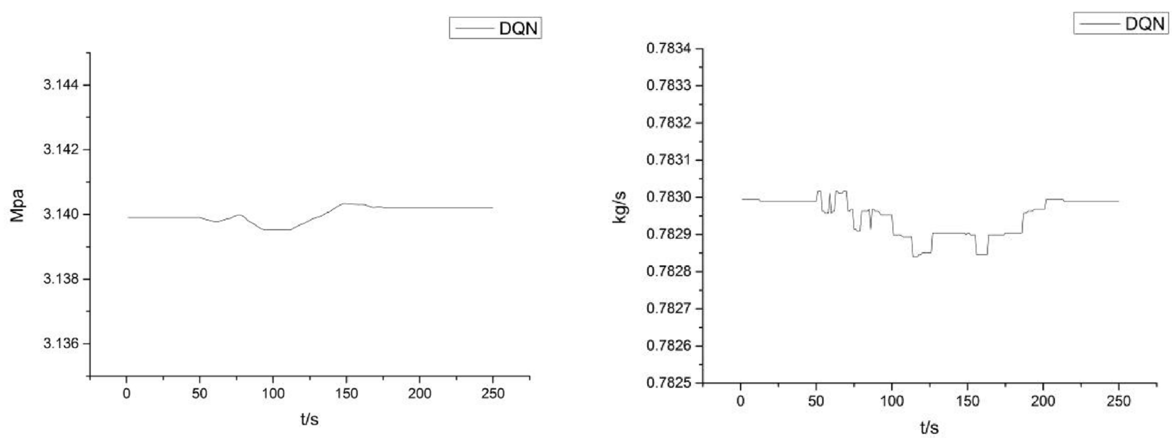

Fig. 11. The pressure and feedwater curves with DQN control under stuck fault of the pressure sensor.

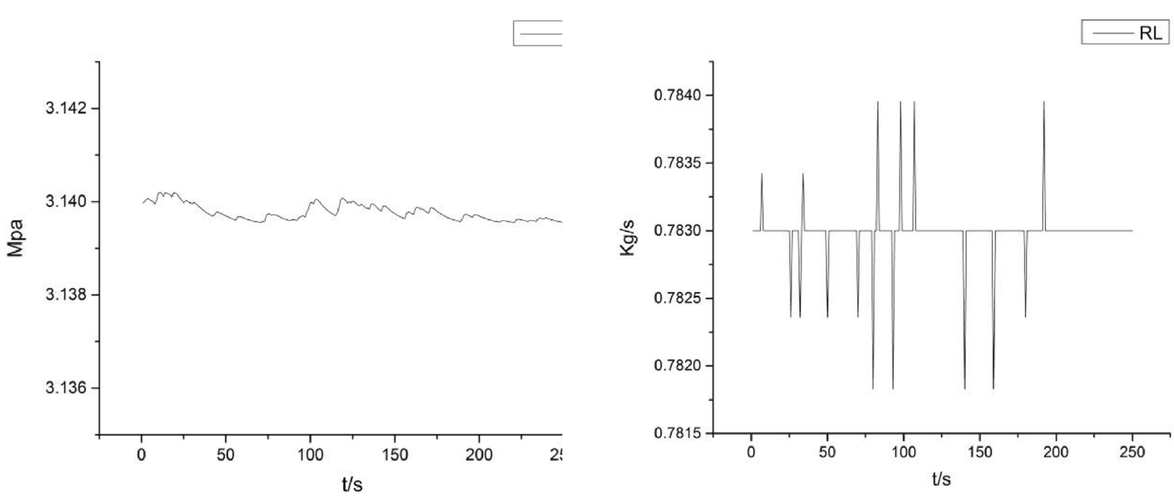

Fig. 12. The pressure and feedwater curves with RL control under stuck fault of the pressure sensor.

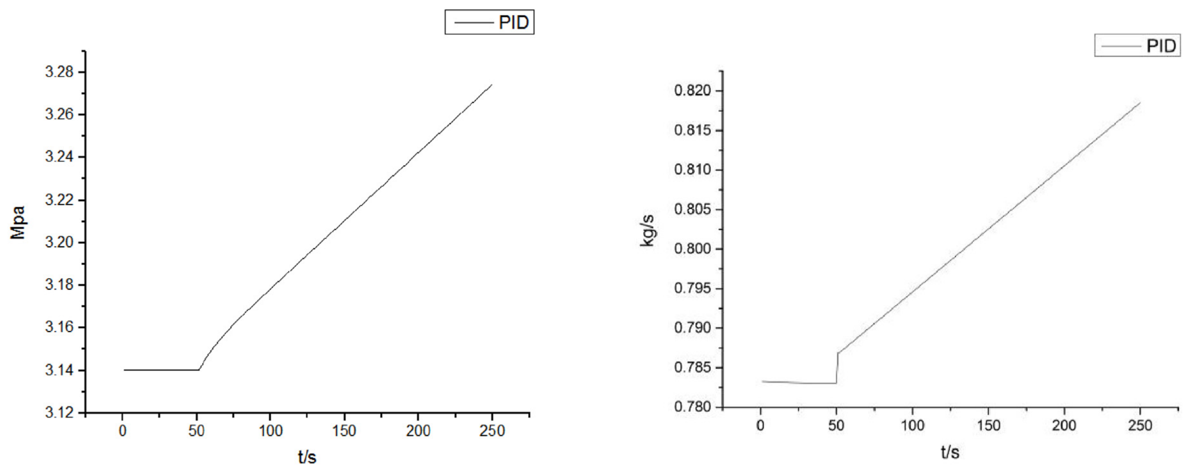

Fig. 13. The pressure and feedwater curves without fault tolerance control under stuck fault of the pressure sensor.

of −0.1Mpa. In order to verify the effect of the active fault-tolerant controller based on reinforcement learning with incremental action, the RL with fixed strategies is selected as the comparison in this paper. The fault tolerance control effect of the system with DQN, RL, PID(without fault tolerance control) are shown in Figs. 5–7 below.

As can be seen from the figures above, due to the dynamic characteristics of the OTSG, when the constant deviation fault occurs to the pressure sensor, the pressure without fault tolerance control fluctuates rapidly in a short time and finally stabilizes near the deviation value. As shown in Fig. 7, compared with the method in this paper, the traditional reinforcement learning algorithm(RL) can only select the fixed strategies due to the limitation of deterministic strategies, resulting in a large error between the control effect and the setpoint, and the fluctuation range of the curve is large. From Fig. 5, the fault-tolerant controller based on DQN can quickly respond to the sensor fault, which makes the pressure change can be contained in the transition period, the fluctuation range of pressure and feedwater is relatively small, and the pressure can be stabilized at the set value quickly.

5.2. Experiment of the sensor under constant gain fault

The gain fault is also a kind of fault that occurs frequently. Gain fault shows that the measured value is a certain proportion of the real value, and usually, the measured value is greater than the real value. When the OTSG is stable running for 50s, the constant gain fault of the pressure sensor is added with the fault degree of 1.1. The control effect of the system is shown in Figs. 8–10 below.

For Fig. 8, this is similar to pressure change when the pressure sensor is under constant deviation fault and the fault-tolerant controller based on DQN can quickly regulate pressure to the setpoint with a small fluctuation range. Compared with the pressure based on RL control in Fig. 9, it can be seen that although RL can be stabilized, the fluctuation range of the pressure is large. As can see from Fig. 10, when the failure occurs, the control system without fault tolerance can't control the pressure in the set value, the control system almost failed.

5.3. Sensor stuck fault experiment

When the OTSG is stable running for 50s, the stuck fault of the pressure sensor is added with the fault degree of 3.13Mpa. The control effect of the system is shown in Figs. 11–13 below.

It can be seen from the figures above that the fault-tolerant control based on DQN can judge the sensor fault and stabilize the pressure value near the set value in a short time. Although the pressure controlled by RL can be stabilized around the setpoint, the pressure curve is not stable and fluctuates widely. The control system without fault tolerance has no effect after the pressure sensor is stuck, and the pressure is in a divergent state, which will eventually lead to the collapse of the control system.

6. Conclusions

Based on the DQN algorithm of reinforcement learning, this paper proposes an active fault-tolerant control method for the OTSG and innovatively proposes incremental action. The results of Matlab and Python co-simulation show that the fault-tolerant control algorithm proposed in this paper can maintain the pressure near the set value rapidly when the pressure sensor fails, and the comparison effect is obvious. The main innovations of this article are as follows:

(1) The fault-tolerant controller based on reinforcement learning adopts the Q network to extract characteristics of the state, and sense the fault information, and then adjust the controller. Compared with the traditional fault-tolerant control method, this method is a kind of active fault-tolerant control method based on data, which simplify the fault detection system, system modeling, and the design of the controller;

(2) For uncertain faults, a reinforcement learning controller with incremental actions is proposed, which breaks through the limitation of the deterministic strategy in traditional reinforcement learning algorithm, and achieves the optimal fault-tolerant strategy for the current system.

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 hyper-parameters of the DQN 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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Peiwei Sun, Xianshan Zhang, Modeling and simulation of feedwater control system with multiple once-through steam generators in a sodium-cooled fast reactor (II): control system simulation, Ann. Nucl. Energy 149 (2020), https://doi.org/10.1016/j.anucene.2020.107780.
[2] wei Li, minjun Peng, yongkuo Liu, Research on condition monitoring technique of sensors in NPPs based on PCA, Nucl. Power Eng. 39 (1) (2018) 136–139, https://doi.org/10.13832/j.jnpe.2018.01.0136.
[3] Simani Silvio, Zattoni Elena, Advanced control design and fault diagnosis, Energies 14 (18) (2021) 5699, https://doi.org/10.3390/en14185699.
[4] Alain Yetendje, Maria Seron, José De Doná, Robust multisensor fault tolerant model-following MPC design for constrained systems, Int. J. Appl. Math. Comput. Sci. 22 (1) (2012) 211–223, https://doi.org/10.2478/v10006-012-0016-7.
[5] Mogens Blanke, Michel Kinnaert, Lunze Jan, et al., Diagnosis and Fault-Tolerant Control, Springle Press, 2004.
[6] Wu Hn, H.Y. Zhang, Reliable fuzzy control for continuous-time nonlinear systems with actuator failures, IEEE Trans. Fuzzy Syst. 14 (5) (2006) 609–618, https://doi.org/10.1109/TFUZZ.2006.876723.
[7] M. Gopinathan, J.D. Boskovic, R.K. Mehra, et al., A Multiple Model Predictive Scheme for Fault-Tolerant Flight Control Design, in: IEEE Conference on Decision & Control, IEEE, Tampa, FL, USA, 1998, pp. 1376–1381.
[8] A. Paoli, M. Sartini, L. Stephane, Active fault tolerant control of discrete event systems using online diagnostics, Automatica 47 (4) (2011) 639–649, https://doi.org/10.1016/j.automatica.2011.01.007.
[9] Z. Wang, L. Liu, H. Zhang, et al., Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm, IEEE Trans. Syst. Man Cybern. Syst. 46 (5) (2016) 611–622, https://doi.org/10.1109/TSMC.2015.2478885.
[10] Farivar Faezeh, Ahmadabadi Majid Nili, Continuous reinforcement learning to robust fault tolerant control for a class of unknown nonlinear systems, Appl. Soft Comput. 37 (2015) 702–714, https://doi.org/10.1016/j.asoc.2015.08.049.
[11] I.I. Belyakov, et al., Static instability of once-through steam generators with convective heating, Sov. Atom. Energy 56 (5) (1984) 347–350.
[12] M. Osakabe, Thermal-hydraulic study of integrated steam generator in PWR, J. Nucl. Sci. Technol. 26 (2) (1989) 66–74.
[13] Ellen Namalomba, Hu Feihu, Haijie Shi, Agent based simulation of centralized electricity transaction market using bi-level and Q-learning algorithm approach, Int. J. Electr. Power Energy Syst. 134 (2022), https://doi.org/10.1016/J.IJEPES.2021.107415.
[14] Liu Wentao, Su Shuai, Tao Tang, Xi Wang, A DQN-based intelligent control method for heavy haul trains on long steep downhill section, Transport. Res. Part C 129 (2021), https://doi.org/10.1016/J.TRC.2021.103249.
[15] Li Haipeng, Huang Xiaojin, Zhang Liangju, A lumped parameter dynamic model of the helical coiled once-through steam generator with movable boundaries, Nucl. Eng. Des. 238 (7) (2008) 1657–1663, https://doi.org/10.1016/j.nucengdes.2008.01.009.
[16] D. Silver, A. Huang, C.J. Maddison, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489, https://doi.org/10.1038/nature16961.