Tie-line Power Adjustment Method Based on Proximal Policy Optimization Algorithm

Jinxu Hou¹, Zhihong Yu¹, Qingping Zheng²*, Huating Xu³ and Shufang Li²
1China Electric Power Research Institute, Beijing, 100192, China
2Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing, 100876, China
3College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China
*zhengqp@bupt.edu.cn

Abstract. The tie-line power adjustment is an essential part of the power system operation state calculation. Various existing algorithms to solve the tie-line power adjustment problem are mainly implemented by introducing tie-line power equation constraints into conventional power flow calculations. Such methods have low calculation efficiency, not enough automation, and are prone to non-convergence in the power flow calculation. In this paper, the tie-line power adjustment problem is formulated as a Markov decision process, and the proximal policy optimization algorithm is introduced to optimize the decision policy. In order to enhance the effectiveness of the proposed method, a new deep neural network structure suitable for the proximal policy optimization algorithm is designed. The proposed method is verified with the IEEE 39-bus system.

1. Introduction
Adjusting the tie-line power to the target value in the power system operation state calculation is of great significance for studying the maximum transmission capacity of the tie-line, studying the power exchange capacity of the power grid, and checking the power generation plan. In recent years, the scale of China’s power grid has expanded significantly, and the grid structure is becoming more and more complicated with the integration of renewable energy [1]. As a result, the tie-line power adjustment in the operation state calculation is becoming more and more challenging and time consuming [2].

At present, the commonly used method to solve the problem of tie-line power adjustment is mainly completed manually and implemented by introducing tie-line power equation constraints into conventional power flow calculations. Yao et al. [3] presented a new power flow algorithm based on tie-line power equivalence constraints after disturbance, considering the necessary inter-area power constrains’ need. Zhao et al. [4] proposed a new approach to compute distributed transfer capability. The proposed method is suitable for distributed implementation, especially in a deregulated environment. Ding et al. [5] put forward an algorithm of power flow for the bulk interconnected grid with transmission interfaces power constraints based on the Newton method, which named interfaces controlling method. It can solve equations of power flow in the grid and regulate transmission interfaces’ power to appointed value synchronously. From the viewpoint of time-domain simulation, Zhao et al. [6] proposed a new method that turns a series of problems to solve nonlinear power flow equations into the time-domain simulation problem under smaller and slower perturbations to obtain a
steady-state operation point. Hu et al. [7] considered the trade power constraints of transmission interface and proposed a model and algorithm of interconnection optimal power flow. Yan et al. [8] considered both frequency characteristics and constraints of active power transmitted via tie-line and proposed a power flow model.

Even though the methods above contribute a lot to the tie-line power adjustment from different perspectives, they cannot complete the tie-line power adjustment without manual interventions. Thanks to the massive data accumulation of smart grids and the vigorous development of artificial intelligence technology, artificial intelligence methods for solving complex optimization problems in power systems have attracted widespread attention [9]. For example, deep reinforcement learning (DRL) has been used to solve some power system control problems, such as microgrid hybrid energy storage [10-11], demand response [12-13], grid emergency control [14], automatic power generation control [15].

In this paper, we formulated the tie-line power adjustment as a finite Markov decision process (MDP). Then we introduced the proximal policy optimization algorithm (PPO) [16] to solve the formulated MDP. Furthermore, a new deep neural network (DNN) structure is proposed to improve the stability of model training.

2. Problem formulation

The tie-line power adjustment is a decision-making process and suitable for being formulated as an MDP. PPO algorithm is a typical DRL algorithm, widely used to solve complex decision-making problems. The schematic diagram of DRL is shown in Fig.1. The main characters of DRL are the agent (in this paper, the agent is PPO) and the environment (in this paper, the environment is the power system). The environment is the world that the agent lives in and interacts with. At every time step \( t \) of interaction, the agent sees a observation of the state \( s_t \) of the world, and then decides on an action \( a_t \) to take. The environment changes when the agent acts on it, but may also change on its own. The agent also perceives a reward \( r_t \) signal from the environment, a number that tells it how good or bad the current world state is. The goal of the agent is to maximize its cumulative reward. DRL methods are ways that the agent can learn behaviours to achieve its goal.

![Figure 1. The schematic diagram of DRL](image)

2.1. Environment state

A state \( s_t \) is a complete description of the state of the environment. In this paper, the state \( s_t \) at time step \( t \) is defined as in equation (1).

\[
s_t = [P_{\text{cw}}, P_{\text{c1}}, P_{\text{c2}}, \ldots, P_{\text{cm}}, m]
\]  
(1)
Where $m$ is a number to distinguish different tie-line; $P_{tar}^m$ is the target power of tie-line $m$; $P_G$ represents the adjustable generator’s injected power; $k$ represents the number of adjustable generators not include the slack bus generator.

2.2. Reward function

The reward function $r_t$ is critically important in DRL. It depends on the current state of the environment, the action just taken, and the next state of the environment. Due to the diversity of different scenarios, there has not been a general principle for reward function designing in DRL [17-18]. In the tie-line power adjustment scene, the goal is to make the power of tie-line $m$ reaches its target value. Therefore, we define the reward function as in equation (2).

$$ r_t = -|P - P_{tar}^m| $$

Where $P$ represents the current power of tie-line $m$. After the action $a_t$ is executed, if the power of tie-line $m$ is closer to the target value $P_{tar}^m$, the reward value is higher, and vice versa. In addition, one of the advantages of the reward function is that the agent can get reward feedback at every time step of the interaction with the environment, which can avoid the situation that the reward is too sparse and the training is difficult to converge.

2.3. Action

The set of all valid actions in a given environment is often called the action space. In the power system environment, the action is to adjust the injected power of generators. The action space is continuous. The complete action execution process needs to map the action $a_t \in [-1, 1]$ to the range of generators’ injected power. In the actual tie-line power adjustment process, operators usually give priority to adjusting generators with higher sensitivity. Although this process needs the experience of the operators themselves, it is beneficial for completing tasks quickly. Similarly, in the field of DRL, human experience can also be introduced to improve training efficiency [19-20]. This paper introduces a generator screening and power compensation mechanism (GS&PC) [21], which can generate different action mapping spaces for different tie-line and target power. The mechanism is divide into two stages. The first stage is data preparation that helps find the sensitive and insensitive generators to narrow the search space. The second stage is the dynamic mapping that helps execute the specific adjustments on the injected power of generators. The second stage consists of the active mapping and the passive mapping. At the active mapping, those generators with higher sensitivity will be adjusted. The passive mapping compensates for the power fluctuation and guarantees the convergence of the power flow calculation.

3. The agent (proximal policy optimization algorithm, PPO)

Traditional reinforcement learning (RL) methods have limited ability to represent policy and can only be applied in some low-dimensional scenarios. DRL is the combination of deep learning (DL) and RL. It aims at realizing the optimization objective of RL with the operation mechanism of DL to advance toward general artificial intelligence. DRL can solve complex decision-making processes and end-to-end control problems.

DRL methods are consist of two categories: value function (VF) based DRL and policy gradient (PG) based DRL [22]. PPO belongs to the latter, uses a DNN to approximate the policy. The key idea underlying policy gradient is to push up the probabilities of actions that lead to higher returns and push down the probabilities of actions that lead to a lower return until the agent arrives at the optimal policy.

The policy gradient parameterize policy $\pi$ as $\pi_{\theta}$, and the DNN parameters $\theta$ are updated as in equation (3).
\[ \theta_{k+1} = \theta_k + \alpha \nabla_{\theta} J(\pi_{\theta}) \bigg|_{\theta_k} \]  

(3)

Where \( \nabla_{\theta} J(\pi_{\theta}) \) represents the policy gradient, and \( \alpha \) is the learning rate. The commonly used approximate policy gradient is as in equation (4).

\[ \nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_t \left[ \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{A}_t \right] \]

(4)

\[ \hat{A}_t = r_t + \gamma V(s_{t+1}) - V(s_t) \]

(5)

\[ V(s_t) = r_t + \gamma r_{t+1} + \gamma^2 s_{t+2} + \ldots + \gamma^{T-t} r_T \]

(6)

Where \( \gamma \) is the discount factor balancing the immediate and the future reward, \( T \) determines how many the future reward will be considered, \( \hat{A}_t \) is the approximate advantage function of action \( a_t \) at step \( t \), \( \mathbb{E}_t \) represents the averaging over a small batch of samples.

3.1. Approximate objective function

The approximate objective function form corresponding to the approximate policy gradient \( \nabla_{\theta} J(\pi_{\theta}) \) is as in equation (7).

\[ L^{PG}(\theta) = \mathbb{E}_t \left[ \log \pi_{\theta}(a_t | s_t) \hat{A}_t \right] \]

(7)

The disadvantage of the policy gradient based DRL method is that the learning rate \( \alpha \) is challenging to set. When the learning rate \( \alpha \) is inappropriate, the decision policy will get worse and worse. PPO uses a few tricks to avoid this problem and keep new policies close to old. PPO doesn’t have a KL-divergence term in the objective function and doesn’t have a constraint at all. As in equation (8), instead relies on specialized clipping in the objective function to remove incentives for the new policy to get far from the old policy.

\[ L^{clip}(\theta) = \mathbb{E}_t \left[ \min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon) \hat{A}_t) \right] \]

(8)

\[ r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)} \]

(9)

Where \( \theta_{old} \) are the DNN parameters before the update, \( \varepsilon \) is a (small) hyperparameter which roughly says how far away the new policy is allowed to go from the old.

3.2. Actor-critic framework

In this paper, we propose a new DNN structure for PPO. As shown in Fig.2, it consists of two parts of fully-connected layers: the actor and the critic. The actor network is used to approximate the policy, and the critic network is used to approximate the value function. They share part of the DNN structure and parameters. In order to optimize the decision policy and ensuring the accuracy of the value function approximation at the same time, we use an approximate objective function that combines \( L^{clip}(\theta) \) and a value function error term \( L^{vf}(\theta) \) [16]. As in equation (11), this approximate objective function can further be augmented by adding an entropy bonus to ensure sufficient exploration [23].

\[ L^{vf}(\theta) = (V(s_t) - V_{target})^2 \]

(10)

\[ L^{clip+vf}(\theta) = \mathbb{E}_t \left[ L^{clip}(\theta) - c_1 L^{vf}(\theta) + c_2 S[\pi_{\theta}](s_t) \right] \]

(11)

Where \( c_1, c_2 \) are hyperparameters, and \( S \) denotes an entropy bonus.
4. Simulations
We evaluate the proposed method on the IEEE 39-bus system. As shown in Fig.3, the power system contains 10 generators, 2 key transmission sections. The first key transmission section (contains 3 tie-lines) tie-line power range is initialized to $[200\text{MW}, 1400\text{MW}]$; the initial tie-line power is 828MW. Similarly, the second key transmission section (contains 1 tie-line) tie-line power range is initialized to $[-200\text{MW}, 400\text{MW}]$; the initial tie-line power is 37MW. Furthermore, All the generator’s injected power is initialized to 1100MW. For the sake of simplification, in this paper, we only consider the active power adjustment.

4.1. General experimental setup and parameters initialization
After many experiments, the proposed new DNN structure is set to have 8 fully-connected layers. The actor network and critic network share the first 3 layers, and the dimensions of the first 3 layers are (10, 400, 300). Besides, in the actor network, there are 3 independent layers (2 hidden layers, 1 output layer), and the dimensions of each layer are (300, 100, 1). Similarly, in the critic network, there has the same structure as the actor network. In addition, the activation function in every hidden layer is a hyperbolic tangent function (tanh).

The training process of the agent starts by initializing the DNN parameters and experience replay buffer. First, we initialize the maximum capacity $C$ of the experience replay buffer and the maximum
training time steps $T_{\text{max}}$. After that, the agent sees a observation of the state $s_t$, and then take an action $a_t$, perceives a reward signal $r_t$ from the environment. We store the experience data $(s_t, a_t, r_t, s_{t+1})$ to the experience replay buffer. This process will continue until the power of target tie-line close to the target value in an acceptable range ($\pm 10$ MW) and the power flow calculation is convergent and the output power of the slack bus generator is within its available range. The process can also be executed synchronously by multiple threads to speed up experience data accumulation. Next, if time step $t = T$, all experience data are taken out and used to update the DNN parameters with the minibatch stochastic gradient descent method. The experience data accumulation and parameters update process will continue to loop until the time step $t = T_{\text{max}}$.

According to lots of experiments, the batch size is set to 192, and all circuit hyperparameters are set as follows: $\gamma = 0.99$, $\alpha = 0.00025$, $\epsilon = 0.2$, $c_1 = 0.5$, $c_2 = 0.01$, $C = 128$, $T_{\text{max}} \gg C$. The computer used for the simulation contains one i7-8700k CPU and one GTX 1080Ti GPU. The power system environment is built with pandapower [24], and the PPO algorithm is built with Tensorflow; all programs are written in python.

4.2. Experimental results

On the IEEE 39-bus system, we evaluate the PPO with GS&PC and the PPO without GS&PC. The accuracy curves of both methods on the complete training process are shown in Fig.4. The accuracy represents that the trained agent gives how many correct adjustment policies on a certain number of targets and indicates the agent training’s convergence. As shown in Fig.4, the PPO with GS&PC can gradually converge with the parameters update. When the parameters update times reaches 1100 times, the accuracy rate reaches 100%. On the contrary, the method without GS&PC does not converge under the same number of parameters update times, and the accuracy rate is not significantly improved. Experimental results show that adding GS&PC can effectively reduce agent training difficulty, shorten training time, and improve stability.

The trained agent is verified on the whole target ranges of two key transmission sections for the rightness. The test results on the key transmission sections 1 and 2 are shown in Fig. 5 and 6, respectively. The x-axis represents the target tie-line power of each key transmission section; The green line represents the achieved tie-line power of each key transmission section; the orange line represents the percentage of the generator injected power. The dotted line represents the generator on the slack bus (Gen. 10).

It can be seen from Fig. 5 and 6 that within the tie-line power adjustment range, the agent can give the correct adjustment policy for any given target value. The adjustment results all meet the accuracy.
requirements. Simultaneously, under different tie-lines and different target value settings, the injected power of the generator on the slack bus can be maintained within its available range, and the overall power flow of the system has not changed significantly. In total, the trained agent can automatically and flexibly adjust the tie-line power to avoid the tedious manual operation process.

5. Conclusion
In this paper, the tie-line power adjustment problem is formulated as an MDP and introduced a DRL based PPO algorithm to solve the formulated problem. The proposed DRL based method uses an actor-critic framework with a new DNN structure to improve training efficiency and stability. Finally, the proposed method is performed on the IEEE 39-bus system. The experimental results show that the proposed method is capable of adjusting the tie-line power automatically with only the target tie-line information. Furthermore, the proposed method does not rely on the expert experience and has little effect on the overall power flow. In the future, the algorithm can be further improved by using an imitation learning algorithm.

Acknowledgments
This work was supported by Adaptive Generation and Interactive Adjustment Technology of Power Grid Operation State Based on Online Data: 5442XT190011.

References
[1] Du, E, et al. (2018) The Role of Concentrating Solar Power Toward High Renewable Energy Penetrated Power Systems. IEEE Trans. on Power Systems, vol. 33, no. 6, pp. 6630-6641.
[2] Yan, J, Zhang, H, Liu, Y, Han, S, Li, L, and Lu, Z. (2018) Forecasting the High Penetration of Wind Power on Multiple Scales Using Multi-to-Multi Mapping. IEEE Trans. on Power Systems, 33(3), 3276-3284
[3] Yao, Y, Hang, N. (2006) Power flow studies of multi-swing bus based on tie-line power equivalence constraints. Power System Protection and Control, 34(1), 31-33.
[4] Zhao, J, Wei, W, Wang, C. (2008) Distributed transfer capability computation of electric power system based on decomposed system network. Proceedings of the CSEE, 28(07), 3-8.
[5] Ding, P, Zhou, X, Yan, J, et al. (2010) Algorithm of power flow for bulk interconnected grid with transmission interface power constraints. Proceedings of the CSEE, 30(10), 8-15.
[6] Zhao, M, An, N, Shi, H, et al. (2012) Algorithm research on automatic calculation of large-scale interconnected power system’s total transfer capability. Power System Technology, 36(11), 179-1839.
[7] Hu, X, Yan, W, Zhao, L, et al. (2013) Model and algorithm of interconnection optimal power flow for interconnected power grid. Automation of Electric Power Systems, 37(3), 47-53.
[8] Yan, W, Mao, Y, Wang, C, et al. (2015) A power flow model considering frequency characteristic and constraint of active power transmitted via tie line. Power System Technology, 39(1), 236-241.
[9] Zhang D, Han X, Deng C. (2018) Review on the research and practice of deep learning and reinforcement learning in smart grids. CSEE Journal of Power and Energy Systems, 4(3), 362-370.

[10] Zhang Z, Qiu C, Zhang D, et al. (2019) A coordinated control method for hybrid energy storage system in microgrid based on deep reinforcement learning. Power System Technology, 43(6), 1914-1921.

[11] Mbuwir B, Ruelens F, Spiessens F, et al. (2017) Reinforcement learning-based battery energy management in a solar microgrid. Energy-Open, 2(4), 36.

[12] Wen Z, O’Neill D, Maer H. (2015) Optimal demand response using device-based reinforcement learning. IEEE Transactions on Smart Grid, 6(5), 2312-2324.

[13] Li D, Jayaweera S K. (2014) Machine-learning aided optimal customer decisions for an interactive smart grid. IEEE Systems Journal, 9(4), 1529-1540.

[14] Liu W, Zhang D, Wang X, et al. (2018) A decision making strategy for generating unit tripping under emergency circumstances based on deep reinforcement learning. Proceedings of the CSEE, 38(1), 109-119.

[15] Xi L, Yu L, Fu Y, et al. (2019) Automatic generation control based on deep reinforcement learning with exploration awareness. Proceedings of the CSEE, 39(14), 4150-4160.

[16] Schulman J, Wolski F, Dhariwal P, Radford A, & Klimov O. (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[17] Dewey D. (2014) Reinforcement Learning and The Reward Engineering Principle. In: AAAI Spring Symposium Series. Stanford, CA, USA, pp. 1–8.

[18] Hadfield-Menell D, et al. (2017) Inverse reward design. In: Advances in neural information processing systems. pp. 6765-6774.

[19] Silver D, et al. (2016) Mastering the game of go with deep neural networks and tree search. Nature, vol. 529, no. 7587, pp. 484-489.

[20] Fettes Q, et al. (2019) Dynamic Voltage and Frequency Scaling in NoCs with Supervised and Reinforcement Learning Techniques. IEEE Trans on Computers, vol. 68, no. 3, pp. 375-389.

[21] Xu H, Yu Z, Zheng Q, Hou J, Wei Y, & Zhang Z. (2019) Deep Reinforcement Learning-Based Tie-Line Power Adjustment Method for Power System Operation State Calculation. IEEE Access, 7, 156160-156174.

[22] Liu J, Gao F, Luo X. (2019) Survey of deep reinforcement learning based on value function and policy gradient. Chinese Journal of Computers, 42(6), 1406-1438.

[23] Mnih V, Badia A. P, Mirza M, Graves A, Lillicrap T. P, Harley T, Silver D, and Kavukcuoglu K. (2016) Asynchronous methods for deep reinforcement learning. In: arXiv preprint arXiv, 1602.01783.

[24] Pandapower 2.0. [Online]. Available: http://www.pandapower.org/.