This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TSTE.2022.3156426, IEEE Transactions on Sustainable Energy

Self-Dispatch of Wind-Storage Integrated System: A Deep Reinforcement Learning Approach

Xiangyu Wei, Yue Xiang, *Senior Member, IEEE,* Junlong Li, Xin Zhang, Senior *Member, IEEE*

***Abstract*—The uncertainty of wind power and electricity price restrict the profitability of wind-storage integrated system (WSS) participating in real-time market (RTM). This paper presents a self-dispatch model for WSS based on deep reinforcement learning (DRL). The designed model is able to learn the integrated bidding and charging policy of WSS from the historical data. Besides, the maximum entropy and distributed prioritized experience replay frame, known as Ape-X, is used in this model. The Ape-X decouples the acting and learning in training by a central shared replay memory to enhance the efficiency and performance of the DRL procedures. Besides, the maximum entropy framework enables the designed agent to explore various optimal possibilities, thus the learned policy is more stable considering the uncertainty of wind power and electricity price. Compared with traditional methods, this model brings more benefits to wind farms while ensuring robustness.**

***Index terms*—wind farm; energy storage system; electricity market; deep reinforcement learning; distributed prioritized experience replay; maximum entropy.**

I. Introduction

The growing penetration of wind power in power system triggers the decline of the power system stability. To address this problem, incentive policies have been introduced around the world [1], especially in China [2]. These policies require new wind farms to install energy storage systems (ESSs) with 10%-30% of wind farms' installed capacities. The installation of ESS effectively alleviates the uncertainty and intermittence of wind power generation, and also provides new control strategies for the wind farms' operation [3]. Meanwhile, the market-oriented reform of the electric power industry and carbon neutrality policy is proceeding step by step in China, which requires wind farms, the main source of clean energy, to participate in the electricity market [4].

For wind farms participating in the electricity market, the uncertainty of wind power may cause the deviation between the bidding power and generated power, which leads to the penalty fee from the market. The integration of ESS can effectively alleviate the unbalance quantity. In addition, the ESS can also obtain profits for wind farms via ESS arbitrage. Correspondingly, wind farms may bid less in low-price periods and more in high-price periods as a response to the operation of ESS. However, the uncertainty of electricity prices makes the bidding policy and ESS operation full of risks.

Existing investigations of wind-storage correlated self-dispatch were mainly developed with optimization methods under uncertainty, such as stochastic programming and robust optimization [5]. However, the prediction accuracy is limited by the chaos of atmospheric and human behaviour, and it's difficult for robust optimization to maximize the benefit for the error of wind power and electricity price forecast.

Deep reinforcement learning is a novel solution for wind-storage correlated self-dispatch. The self-dispatch process can be modelled as a Markov decision process (MDP) and further solved. Ref. [6] utilized the expected state-action-reward-state-action (SARSA) algorithm with clustering to learn the ESS operation policy. However, this method is incapable of complex scenarios due to the limit of the Q-table. Ref. [7] realized hour-ahead control of ESS based on the Rainbow method. But this method cannot be utilized in continuous action space because of its value-based mechanism. On the other hand, the deep deterministic policy gradient (DDPG) algorithm [8] is popular in recent years. But for the limited sample efficiency and high sensitivity to hyperparameters, this method is not stable enough and difficult to converge for the self-dispatch in wind farms. Therefore, it is necessary to develop an efficient wind-storage correlated self-dispatch method.

To address the above problems, this letter proposes a DRL-based model for wind-storage correlated self-dispatch. The soft actor-critic (SAC) algorithm with Ape-X is adopted, aiming to improve the benefit of wind farms participating in the electricity market and enhance the robustness. The Ape-X frame effectively improves the stability and convergence of the learned policy. Besides, since the maximum entropy mechanism of SAC can avoid the local optimal solution, the learned policy is more robust.

II. Proposed Self-Dispatch Methodology

To formulate the wind-storage correlated self-dispatch model, the Ape-X SAC algorithm is introduced firstly, then the external environment is modelled in detail.

A. Ape-X SAC algorithm

Based on the maximum entropy formulation, the strong anti-interference ability and stable performance make SAC an important breakthrough in DRL [9]. However, same as the DDPG, SAC is still limited by sample efficiency while dealing with the coupled uncertainty of wind power and electricity price. This is because the action of actors synchronizes with the training of the policy network in SAC. Ape-X is a viable

This work was supported by the National Natural Science Foundation of China under Grants U2166211 and 52177103. *(Corresponding author: Yue Xiang.)*

Xiangyu Wei and Yue Xiang are with the College of Electrical Engineering, Sichuan University, Chengdu 610065, China (email: xiangyu_wei@163.com, xiang@scu.edu.cn).

Junlong Li is with the Department of Electronic and Electrical Engineering, University of Bath, Bath BA2 7AY, UK (email: jl3466@bath.ac.uk).

Xin Zhang is with the Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK (email: xin.sam.zhang@gmail.com)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TSTE.2022.3156426, IEEE Transactions on Sustainable Energy

solution to address this problem [10]. Ape-X decouples actors' action and policy network training via a central shared replay buffer. This enables multi actors to act in the environment parallelly and centralizes the explored transitions in the replay buffer. Thus, the learner can centrally learn the transitions by their priority. The architecture of the proposed Ape-X SAC model is shown in Fig. 1.

Different from the traditional SAC, the multi actors in this framework are not responsible for learning. Correspondingly, they explore the environment based on the shared policy network, which is derived from the learner periodically. To construct the transition data efficiently, each actor owns an independent local circuit buffer, where the constructed transitions are first stored in. However, since a great number of transitions has already been generated by multi actors, not all the transitions are able to be uploaded to the central buffer. Thus, the prioritized experience replay is introduced to rank the transitions, by comparing the absolute time difference (TD) error $|\delta_{i,t}|$ of transition $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t}')$ at step t.

$$\delta_{i,t} = |r_{i,t} + \max_{a_{i,t}} \gamma V(s_{i,t}', a_{i,t}') - V(s_{i,t}, a_{i,t})| \tag{1}$$

where, $r_{i,t}$ is the reward, γ is the discount factor, $V(s_{i,t}, a_{i,t})$ is the action $a_{i,t}$ at state $s_{i,t}$.

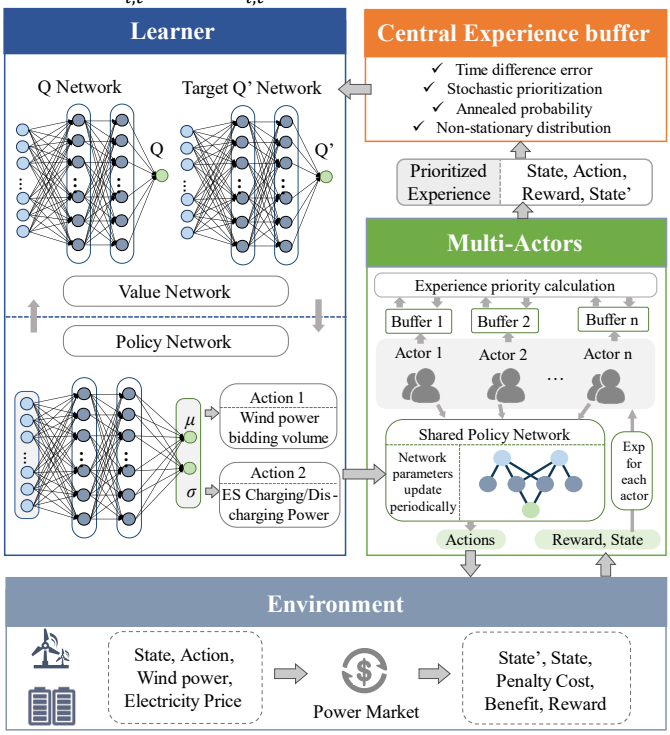

Fig.1.The architecture of the Ape-X SAC model

After being uploaded to the central experience buffer periodically, the prioritization of transitions needs to be recalculated for the difference of the network. The probability $P(i)$ of transition i to be learned is calculated based on the stochastic prioritization.

$$P(i) = \frac{p_i^\alpha}{\sum_N p_n^\alpha}, \; p_i = \frac{1}{rank(i)} \tag{2}$$

where α is the hyperparameter, $rank(i)$ is the rank by the absolute TD error, N is the number of transitions in the buffer. For the bias method is introduced to alter the converged results, the importance-sample weights ω_i is utilized to anneal the bias.

$$\omega_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\beta \tag{3}$$

where β is the hyperparameter.

According to the annealed probability, the learner selects the transitions from the experience buffer. The exploration of actors is completed by the central processing unit (CPU). Thus, the computing power of the graphics processing unit (GPU) will not be shared by actors. For Ape-X SAC, SAC tries to maximize the expected sum of rewards and the entropy [11] of a policy π:

$$J(\pi) = \sum_{t=0}^{T} E_{(s_t,a_t) \sim \rho_\pi}[r(s_t,a_t) + \alpha H(\pi(\cdot \mid s_t))] \tag{4}$$

where s_t is the state at step t, a_t is the action at step t, $r(s_t,a_t)$ presents the reward of action a_t at state s_t, ρ_π is the state-action marginal of the trajectory distribution introduced by π, α is the hyper-parameter that balances the entropy term against the reward, $H(\cdot)$ is the entropy, $H(X) = -\sum_{x_i \in X} P(x_i) \log P(x_i)$, $\pi(\cdot \mid s_t)$ is the policy in the state s_t.

There are 3 main functions that need to be trained in SAC: State value function $J_V(\psi)$, soft Q-function $J_Q(\theta)$, and policy function $J_\pi(\phi)$. Among them, the policy is modelled as a Gaussian distribution, and its mean vector and covariance matrix are given by the neural network. Therefore, the information projection is needed for the soft policy improvement, and it's defined by relative entropy. The policy parameters can be obtained by formula (5):

$$J_\pi(\phi) = E_{S_t \sim D}\left[D_{KL}\left(\pi_\phi(\cdot \mid s_t) \parallel \frac{\exp(Q_\theta(s_t,\cdot))}{Z_\theta(s_t)}\right)\right] \tag{5}$$

where, D is the distribution of the previously sampled states and actions, $\exp(Q_\theta(s_t,\cdot))$ is the exponential of the Q-function $Q_\theta(s_t,\cdot)$, $D_{KL}(Q \parallel P)$ is the KL divergence, or relative entropy, $Z_\theta(s_t)$ is the partition function, it normalizes the distribution.

Reparametrize the policy by the neural network, $a_t = f_\phi(\epsilon_t; s_t)$, then formula (5) can be re-written as formula (6):

$$J_\pi(\phi) = E_{S_t \sim D, \varepsilon_t \sim \mathrm{N}(0,1)}[\log \pi_\phi(f_\phi(\varepsilon_t; s_t) \mid s_t) - Q_\theta(s_t, f_\phi(\varepsilon_t; s_t))] \tag{6}$$

where ε_t is an input noise vector, $\pi_\emptyset$ is defined implicitly in terms of f_ϕ. Therefore, the gradient of formula (6) can be further approximated [9].

B. External environment

The setting of the external environment is crucial for DRL, especially considering the uncertainty of wind power and electricity price. The real-time electricity market is set as an hour-ahead market. At hour t, the self-dispatch system should determine the hour-ahead operation scheme of ESS and bidding volume for the period from the hour t+1 to hour t+2, which includes 4-time slots with the length of 15 minutes. The reward of action a at time t on day j is set as:

$$R_{s,a,j} = B_{a,t}^{\mathrm{rt}} + J_{a,t} + \omega_1(\mu P_{\mathrm{ES},j}^{\mathrm{c.sum}} - \tau V_{ES}^{max}) + \omega_2 |V_{\mathrm{ES},j}^{\mathrm{end}} - V_{\mathrm{ES},j}^{\mathrm{in}}| \tag{6}$$

$$B_{a,t}^{\mathrm{rt}} = Pr_t\left(P_{\mathrm{To},t} - |P_{\mathrm{To},t} - P_{\mathrm{bid},t}| * \varepsilon(P_{\mathrm{To},t} - P_{\mathrm{bid},t})\right) \tag{7}$$

$$J_{a,t} = \omega_3\left(|P_{\mathrm{bid},t} - P_{\mathrm{To},t}| * \varepsilon(P_{\mathrm{bid},t} - P_{\mathrm{To},t})\right) \tag{8}$$

$$P_{\mathrm{To},t} = P_{\mathrm{real},t} + \mu P_{\mathrm{Charge},t}^{real} \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TSTE.2022.3156426, IEEE Transactions on Sustainable Energy

where, $B_{a,t}^{\mathrm{rt}}$ is the benefit from the real-time market of action a at time t, ω_1 and ω_2 are the penalty coefficients, $P_{\mathrm{ES},j}^{\mathrm{c.sum}}$ is the gross charging volume of ESS on day j, V_{ES}^{max} is the max state of charge (SoC) of ESS, τ is the max daily charge-discharge cycle, $V_{\mathrm{ES},j}^{\mathrm{end}}$ is the SoC of ESS after the last action on day j, $V_{\mathrm{ES},j}^{\mathrm{in}}$ is the initial SoC of ESS on day j, $J_{a,t}$ is the penalty cost for the real power deviations from the bidding volume of time t, $J_t = 0$ when the real power surplus, but the surplus power needs to be curtailed, μ is the charging/discharging efficiency, Pr_t is the electricity price at time t, $P_{\mathrm{To},t}$ is the total power of WSS at time t, $P_{\mathrm{Charge},t}^{real}$ is the real charging power at time t, $\varepsilon(\cdot)$ is the unit step function, $P_{\mathrm{bid},t}$ is the bidding power at time t for the next hour, $P_{\mathrm{real},t}$ is the wind power output at time t, ω_3 is the real-time balanced electricity price.

The state-space S_t and action space A_t is set as:

$$S_t = \{T_t, P_{\mathrm{WP},t}^{\mathrm{Fcst}}, Pr_t^{\mathrm{Fcst}}, W_{\mathrm{WT},t}, V_{\mathrm{ES},t}, \varepsilon_{\mathrm{WP},t}\} \quad (7)$$

$$A_t = \{P_{\mathrm{Charge},t+1}, P_{\mathrm{Charge},t+1}^{pre}\} \quad (8)$$

where T is the time, $P_{\mathrm{WP},t}^{\mathrm{Fcst}}$ is the forecasted wind power output at time t for the next 8 quarters and 9 hours, Pr_t^{Fcst} is the forecasted electricity price at time t for the next 8 quarters and 24 hours, $W_{\mathrm{WT},t}$ is the historical weather data recorded by the WSS, $V_{\mathrm{ES},t}$ is the SoC of ESS at time t, $\varepsilon_{\mathrm{WP},t}$ is the wind power forecasting error for the last 8 quarters, $P_{\mathrm{Charge},t+1}^{pre}$ is the planned charging power at time t for the next hour, $P_{\mathrm{bid},t+1}$ is the bidding power for the next hour.

Additionally, some tricks are applied to enhance the performance. For example, 1) The 24 forecasted hourly electricity price data is compressed to 8 to streamline the action space; 2) The historical forecasting error is added to remedy the forecasting error; 3) The agent will act four times to form a complete action, aiming at improving the convergence.

III. Case Study

In this section, the proposed self-dispatch model is performed on a 99 MW wind farm with 30MW ESS located in northern China. One-year data is scaled and used for validation. The max charge/discharge rate of ESS is 6MW, and the efficiency is 95%. Due to the deviation of the hour-ahead plan, ESS is designed to be able to modify the charging/discharging plan in real time, and the modification margin is 3MW for each time step. The forecast data is generated by XGboost [12], which is testified efficient in time series forecast. The mean absolute percentage error (MAPE) of the ultra-short/short term wind power forecast is 4.372% and 8.761%. As for the ultra-short/short term electricity price, they are 5.561% and 6.746% respectively.

TABLE I
Parameters Setting of the Proposed Model

Parameters	Values
Structure of policy network	[512, 256, 256, 128]
Structure of value network	[512, 512, 256, 256]
Learning rate	3×10^{-6}
Batch size	4096
Central experience buffer size	2×10^{6}
Number of actors	16
Penalty coefficient ω_1 and ω_2	-1000 ¥/MWh
Real-time balanced price ω_3	$-2.5\times Pr_t$
Number of max charging cycles/day	1.0

The proposed approach is trained for 10000 episodes, 96 steps (1 day) per episode to learn the optimal self-dispatch policy. To evaluate the performance of the self-dispatch policy learned from training data, comparative tests are carried out on the test set, where 30 days of data are used.

The results of different methods are compared in Table II. The results demonstrate the effectiveness of the proposed model in enhancing the benefit of the ESS. Meanwhile, the learned policy is more robust. The deviation between the hour-ahead dispatch plan and real-time power generation is much less than the other methods, which can mostly be covered by the real-time dispatch of ESS.

TABLE II
Benefit Comparison Of Different Algorithms

Methods	Average Benefit (¥)	Average deviation (MW/step)	Max deviation (MW/step)
Original	204763.0	0.578	4.719
DDPG	186231.2	0.886	6.041
SAC	204935.1	0.504	5.700
Ape-X SAC	210398.5	0.244	2.560
Ideal model	213294.8	0	0

In addition, compared with the average benefit, ¥186231.2 in the test set earned by DDPG, that of the SAC is almost 10% higher, which proved the effectiveness of the entropy mechanism introduced in SAC. The great potential of DDPG can not be denied because it's almost omnipotent theoretically. But in this case, the optimal parameters are hard to find. As for the Ape-X SAC, the average benefit earned by the proposed model reached ¥210398.5, which is 2.7% higher than the traditional SAC model. The experiment results verify the role of distributed prioritized experience replay frame in alleviating the uncertainty brought by the coupled uncertainty of wind power and electricity market.

To further elaborate on the performance of the proposed model, a windy day in the test set is selected. The electricity price and charging plan are shown in Fig. 2. From Fig. 2, it can be found that the learned policy is able to formulate the charging/discharging plan considering the long-term benefit in the electricity market. Furthermore, for the deviation between forecasted wind power and real power, the charging plan can not make the maximum benefit from the market, even with the real-time dispatch.

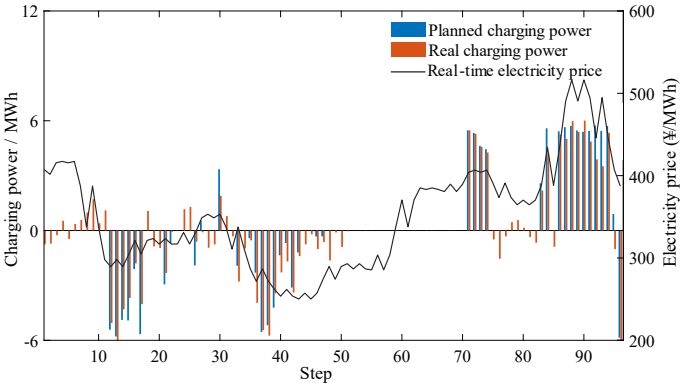

Fig.2. The electricity price and charging plan in one day

The self-dispatch result is shown in Fig. 3. Due to the consideration of the historical forecasting error, the learned

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TSTE.2022.3156426, IEEE Transactions on Sustainable Energy

policy is more accurate. For example, most of the forecasted wind power in step (0, 25) is less than the real-time power generation and the error is similar. Obviously, the algorithm captures this feature, so the bidding power is extremely close to the real-time power generation. Furthermore, due to the high penalty fee of the insufficient power supply, the learned policy tends to bid less to prevent the additional cost from the market, although it may lead to the wind curtail. Fortunately, the amount of curtailed power is under control. Meanwhile, the robustness of the learned policy gets proved.

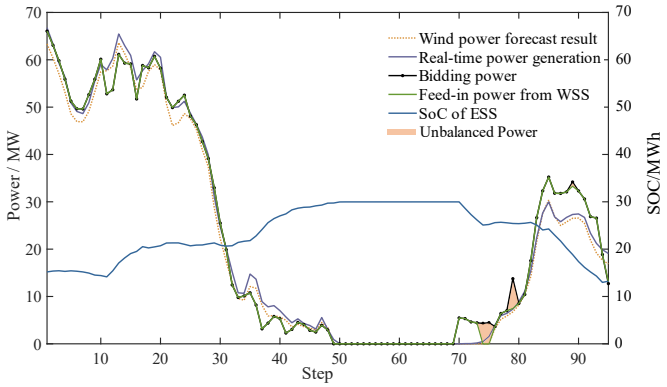

Fig.3. The self-dispatch result in one day

IV. Conclusion

This letter presents a self-dispatch model of WSS based on Ape-X SAC. The designed algorithm is able to learn the self-dispatch strategies from historical data through actor-critic networks. The use of distributed prioritized experience replay frame and entropy enables the designed agent to explore various optimal possibilities, thus the learned policy is more stable when considering the coupled uncertainty of wind power and electricity price. Comparative results illustrate that the learned policy earns 1.36% less than the optimal policy, which is much better than the others.

Reference

[1] H Zhang, J Yang, X Ren, et al. "How to accommodate curtailed wind power: A comparative analysis between the US, Germany, India and China," Energy Strategy Reviews, Vol. 32: 100538, Nov. 2020.

[2] China Energy Storage Alliance, "2020 Energy Storage Industry Summary: A New Stage in Large-scale Development," [online] Available: http://en.cnesa.org/latest-news.

[3] H. Ding, P. Pinson, Z. Hu and Y. Song, "Integrated Bidding and Operating Strategies for Wind-Storage Systems," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 163-172, Jan. 2016.

[4] Z. Zhang, Y. Zhang, Q. Huang and W. Lee, "Market-oriented optimal dispatching strategy for a wind farm with a multiple-stage hybrid energy storage system," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 4, pp. 417-424, 2018.

[5] A. A. Thatte, L. Xie, D. E. Viassolo and S. Singh, "Risk Measure Based Robust Bidding Strategy for Arbitrage Using a Wind Farm and Energy Storage," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2191-2199, Dec. 2013.

[6] E. Oh and H. Wang, "Reinforcement-Learning-Based Energy Storage System Operation Strategies to Manage Wind Power Forecast Uncertainty," *IEEE Access*, vol. 8, pp. 20965-20976, 2020.

[7] J. Yang, M. Yang, M. Wang, et al. "A Deep Reinforcement Learning Method for Managing Wind Farm Uncertainties through Energy Storage System Control and External Reserve Purchasing," *International Journal of Electrical Power & Energy Systems,* vol. 119, 105928, Feb. 2020.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al., "Continuous control with deep reinforcement learning," *Computer Science*, vol. 8, no. 6, pp. A187, 2015.

[9] Tuomas H, Aurick Z, Pieter A, et al. (2018) "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *International Conference on Machine Learning*, 2018.

[10] Horgan D, Quan J, Budden D, et al. (2018) "Distributed prioritized experience replay," arXiv preprint arXiv:1803.00933, Apr. 2018, [online] Available: http://arxiv.org/abs/1803.00933.

[11] T. Haarnoja, H. Tang, P. Abbeel and S. Levine, "Reinforcement Learning with Deep Energy-based Policies," *International Conference on Machine Learning*, 2017.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.