One-Step Two-Critic Deep Reinforcement Learning for Inverter-based Volt-Var Control in Active Distribution Networks

Qiong Liu, Ye Guo, Lirong Deng, Haotian Liu, Dongyu Li, Hongbin Sun, and Wenqi Huang

Abstract—A one-step two-critic deep reinforcement learning (OSTC-DRL) approach for inverter-based volt-var control (IB-VVC) in active distribution networks is proposed in this paper. Firstly, considering IB-VVC can be formulated as a single-period optimization problem, we formulate the IB-VVC as a one-step Markov decision process rather than the standard Markov decision process, which simplifies the DRL learning task. Then we design the one-step actor-critic DRL scheme which is a simplified version of recent DRL algorithms, and it avoids the issue of Q value overestimation successfully. Furthermore, considering two objectives of VVC: minimizing power loss and eliminating voltage violation, we utilize two critics to approximate the rewards of two objectives separately. It simplifies the approximation tasks of each critic, and avoids the interaction effect between two objectives in the learning process of critic. The OSTC-DRL approach integrates the one-step actor-critic DRL scheme and the two-critic technology. Based on the OSTC-DRL, we design two centralized DRL algorithms. Further, we extend the OSTC-DRL to multi-agent DRL and design two multi-agent DRL algorithms. Simulations demonstrate that the proposed OSTC-DRL has a faster convergence rate and a better control performance, and the multi-agent OSTC-DRL works well for decentralized IB-VVC problems.

Index Terms—Volt-Var control, deep reinforcement learning, active distribution network.

I. INTRODUCTION

Recently, the active distribution networks are incorporating more distribution generations, such as wind and solar power. The high penetration of distribution generations poses new challenges to voltage regulation and network loss minimization problems. To tackle the problem, Volt-Var control (VVC) is integrated into ADNs to minimize power loss and eliminate voltage violations by controlling the reactive power resources. Since most DGs are inverter-based energy resources (IB-ERS) that are possible to provide reactive power support rapidly, it is of crucial importance to utilize these resources to achieve the VVC [1], [2].

The recent methods for inverter-based VVC (IB-VVC) problems can be divided into model-based and data-driven. Model-based methods solve the IB-VVC problems based on a reliable physical model of the ADN. However, a reliable physical model is hard to acquire for electric utilities because of the complexity and interconnectivity of the electric infrastructure [3]. As an alternative solution, data-driven methods learn optimal actions from measurements data directly. In data-driven methods, deep reinforcement learning (DRL) methods is intensively studied in recent years and have shown remarkable performance in games [4], robotics [5], as well as the power system [6]. Meanwhile, DRL algorithms can achieve real-time decision-making by shifting the computational expense from online optimization to offline training.

Generally, actor-critic DRL algorithms are practical methods to deal with IB-VVC problems [7], [8], [9]. Actor-critic algorithms train both the state-action value network (also named as critic) and the policy network (also named as actor). They reduce the variance, improve data efficiency, and accelerate the learning process [10]. However, the performance of recent DRL algorithms is worse than the model-based optimization methods with an accurate model, especially in minimizing power loss [7], [11]. The learning process is slow [12], and the convergence tends to be unstable [13]. The reasons for the problems can be analyzed from the general DRL field and the property of VVC problems.

From the perspective of the general DRL field, the intrinsic reason for the above problems of DRL is the unavoidable estimation error of critic networks including overestimation or underestimation [14]. The overestimation error is accumulated when using the temporal difference learning method to train critic networks [15]. It may worsen the performance of the policy network or even lead to divergent behavior. The underestimated bias will not be explicitly accumulated through the policy update but still degrade performance [16], [17]. Deep deterministic policy gradient (DDPG) uses a replay buffer and soft target updates to alleviate the overestimation of critics [18]. Twin delayed deep deterministic policy gradient (TD3) [15] and soft actor-critic (SAC) [19] mitigates the overestimation bias and its accumulation successfully by the technology of clipped double-Q learning. Nevertheless, clipped double-Q learning leads to an underestimation bias, which also degrades performance [16], [17]. For VVC problems, if we apply TD3 or SAC algorithms directly, we may encounter similar
From the property of VVC problems, two objectives of VVC: minimizing power loss and eliminating voltage violations, hinder the performance of DRL algorithms further. Generally, the reward of DRL is designed as the combination of power loss and voltage violation rate [20], [12]. The weight ratio between power losses and voltage violations needs to be tuned carefully, otherwise, the performance of DRL would degrade [13], [21]. To alleviate the problem, a Lagrangian relaxation method is introduced to tune the weight ratio between power losses and voltage violations in the reward design [21]. To simply the learning task, the reward is the negative of voltage violation when voltage violation appears, and the negative of the power loss when no voltage violation appears [13]. The critic network only needs to learn the penalty of voltage violation when voltage violation appears. The above reward design methods still mix the two objectives and hinder the approximation process of critic networks. To address the problem, Ref. [22] designs two critic networks for power loss and voltage violation separately, but the entropy regularization term mixed in the critic networks in the learning process degrades the accuracy of critics.

Since the estimation error of critic networks is unavoidable in the process of learning from data, [23], [24] derive the critic based on the power flow model. It avoids the inaccuracy estimation of critic value directly. However, an accurate ADN model is needed for obtaining a reliable VVC policy.

Given the literature discussion above, we notice that two problems degrade the performance of DRL for IB-VVC in ADNs. 1) IB-VVC can be simplified as a single-period optimization problem, whereas recent DRL solves it as a multi-period optimization problem. It encounters overestimated or underestimated Q value issues. 2) Two objectives of VVC have completely different properties, whereas recent DRL uses one critic to approximate the two objectives. It degrades the approximation performance. To address the two problems, we propose a one-step two-critic DRL (OSTC) approach for IB-VVC in ADNs. For the first problem, we formulate the IB-VVC as a one-step Markov decision process (MDP), also named as the contextual bandit problem [25], and design a one-step DRL scheme to solve the one-step MDP. For the second problem, we use two critic networks to approximate the rewards of power loss and voltage violation separately. Based on the OSTC-DRL approach, we design two kinds of centralized DRL algorithms which are OSTC with a deterministic policy (OSTC-DP) derived from DDPG and OSTC soft actor-critic (OSTC-SAC) derived from SAC. For decentralized VVC, we extend the OSTC-DRL approach to a multi-agent OSTC-DRL approach and design two multi-agent DRL algorithms which are multi-agent OSTC-DP and multi-agent OSTC-SAC. Our proposed approach is simple, stable, and efficient. Compared with the existing DRL-based VVC algorithms, the main contributions of this paper and the technical advancements are summarized as follows:

1. We propose an OSTC-DRL approach for IB-VVC in ADNs that accelerates the convergence rate and improves control performance by simplifying the recent DRL algorithm [18], [19] and decreasing the approximation difficulties of each critic. It is a simple yet effective DRL approach for IB-VVC.

2. In the OSTC-DRL approach, we use the critic to learn the recent reward rather than the infinity discounted accumulated reward, which simplifies the learning process of DRL considerably, and avoids the overestimation of Q value and its accumulation in the learning process successfully.

3. In the OSTC-DRL approach, we utilize the two critics to learn the two objectives separately. It decreases the approximation difficulties of each critic, thus having a faster convergence rate and a higher approximation accuracy.

4. We extend the OSTC-DRL approach to the multi-agent OSTC-DRL approach for decentralized IB-VVC. Multi-agent OSTC-DRL algorithms have a very similar performance to the centralized OSTC algorithms even when each actor executes based on the local one bus information. The associated codes in this paper will be shared on Github.

II. PROBLEM FORMULATION

A. IB-VVC Problem Formulation

IB-ERs can provide high-speed reactive power support and have a large capacity in high penetration areas. Without considering the degradation of IB-ERs, IB-VVC is a single-period optimization problem [26], [27]. The objective of VVC is to minimize the power loss while satisfying the voltage constraints by optimizing the output of controllable devices for each time $t$,

$$\min_{x_t, u_t} r_p(x_t, u_t, D_t, p, A)$$

s.t. $f(x_t, u_t, D_t, p, A) = 0$

where $r_p$ is power loss, $x_t$ is the state variable vector of the ADN like active power injection $P_t$, reactive power injection $Q_t$, and voltage magnitude $V_t$, $u_t$ is a vector of the control variables which are reactive power produced by and static var compensators (SVCs) and IB-ERs, $D_t$ is the vector of uncontrollable variables which are power generations of distributed energy resources and power loads, $p$ is the parameters of an ADN, $A$ is the incidence matrix of an ADN, $f$ is the power flow equation, $\bar{u}, \underline{u}$ are the lower and upper bounds of controllable variable, and $\bar{h}_v, \underline{h}_v$ are the lower and upper bounds of voltage.

When the power flow model is known accurately, we can use model-based optimization methods to solve the problem [2], [28]. However, when the power flow model cannot be accurately known, DRL methods may be a potential method to deal with the problem.

B. Formulating the IB-VVC as a One-Step Markov Decision Process

DRL algorithms learn the policy to maximize its cumulative reward from the data generated by interacting with ADNs.
Generally, the interaction data is formulated as an MDP. MDP is designed for formulating long horizon decision problems or multi-period optimization problems, but the IB-VVC problem is a single-period optimization problem. Using MDP to formulate the problem is feasible but it increases the complexity of optimization tasks.

For a single-period problem, it would be better to formulate the problem as one-step MDP, also known as the contextual bandit. One-step MDP is a simple version of MDP. The one-step MDP is defined by a tuple \((S, A, R)\). At each time step \(t\), the agent observes a state \(s \in S\), and selects actions \(a \in A\) with respect to its policy \(\pi : S \rightarrow A\), receiving a reward \(r \in R\). The next state \(s_{t+1}\) is not included in the process.

The detailed process of the IB-VVC problem is shown in Fig. 1. In practice, there costs a short time to execute after calculating the action. We define the execution time \(t\) for the IB-VVC problems, state space, action space, and reward function are defined for one-step MDP as follows:

1) State: The state is \(s_t = (P_t, Q_t, V_t, Q_{G,t}, Q_{C,t})\). \(Q_{G,t}, Q_{C,t}\) is the reactive power outputs of all controllable IB-ERs and SVCs at time \(t\). Since the paper focus on the IB-VVC, for simply the problem, we assume the incidence matrix \(A\) is not changed in the process, so the information of \(A\) is not added into the state [1]. [20]. Compared with \(s_t = (P_t, Q_t, V_t)\) in [20], adding the previous action in \(Q_{G,t}, Q_{C,t}\) is to reflect the state of ADNs completely, otherwise, the state would degrade to partial observation, because only \(Q_t\) cannot deduce the uncontrollable variables \(D_t\) and the controllable variables \(Q_{G,t}, Q_{C,t}\). There are also another two kinds of state settings \(s_t = (P_t, Q_t, V_t, T_{tap,t})\) [20], \(s_t = (V_t)\) [13] [9], where \(T_{tap}\) is the switch position of capacity banks or transformers. The design of the state is still an open question, and maybe there needs concrete literature to discuss the essential difference between them from the representation learning theory in the future.

2) Action: The action is \(a_t = (Q_{G,t}, Q_{C,t})\), where \(Q_{G,t}, Q_{C,t}\) are the reactive power outputs of all IB-ERs and SVCs, and the range are \(Q_{G,t} \leq \sqrt{S_G^2 - P_G^2}\) and \(Q_C \leq Q_{C,t} \leq Q_C^{max}\) [20], [30]. \(P_G\) is the upper limit of active power generation of IB-ERs, and \(Q_C\) and \(Q_C^{max}\) the upper and bottom limit of reactive power generation of SACs.

3) Reward: We assume the new state \(s_t\) can be obtained after executing action \(a_t\) immediately. Reward is calculated based on \(s_t\). VVC problems have two objectives: minimizing active power loss, and eliminating voltage violation. Hence, the reward consists of two terms: the negative of active power loss \(r_{p,t}\), and the negative of voltage violation rate \(r_{v,t}\). \(r_{p,t}\) is defined as:

\[
r_{p,t} = -\sum P_t'
\]

Similar to [13], [24], \(r_{v,t}\) is defined as

\[
r_{v,t} = -\sum \left[ \max \left( V_t - \bar{V}, 0 \right) + \max \left( \bar{V} - V_t, 0 \right) \right]
\]

There are also another two kinds of \(r_{v,t}\) setting that are \(r_{v,t} = -\sum \left[ \max \left( V_t - \bar{V}, 0 \right)^2 + \max \left( \bar{V} - V_t, 0 \right)^2 \right]\) [20], and \(r_{v,t} = \sum \left[ \left( \left| V_t \right| > \bar{V} \right) + \left( \left| V_t \right| < \bar{V} \right) \right]\) [21]. \(r_{v,t}\) in [20] is similar to mean squared error loss in deep learning, but the violation value is weakened when the value is very small. For example, if \(r_{v,t} = 0.001\), the mean squared error is \(10^{-6}\) that nearly be omitted compared with the value of \(r_{p,t}\). \(r_{v,t}\) in [21] is similar to zero-one loss in deep learning which may be not easy to learn for critic networks because of its discontinuous property.

III. ONE-STEP TWO-CRITIC DRL

This section introduces the details of the OSTC-DRL approach. We first propose a one-step DRL scheme to learn from the one-step MDP data, and then utilize the two-critic to learn the reward of power loss and voltage violation separately. Fig. 2 shows the proposed one-step two-critic DRL approach.

A. One-Step DRL

The objective of one-step DRL is learning a policy \(\pi\) that maximizes the expected rewards \(J(\pi)\),

\[
\pi^* = \arg\max_{\pi} J(\pi) \quad s.t. \quad J(\pi) = E_{a \sim \pi} (r(s,a)),
\]

where \(r = r_p + c_v v_t\), \(c_v\) is a constant. It is different from the ordinary DRL algorithm that the policy tries to maximize.
the expected infinite-horizon discounted accumulated reward \( E_{a \sim \pi} (\sum_{t=0}^{\infty} \gamma^t r_t(s, a)) \), where \( \gamma < 1 \) is a discount factor.

In actor-critic methods, we use deep neural networks to parametrize the state-action value function \( Q(s, a) \), known as the critic, and the policy \( \pi \), known as the actor. For one-step DRL, the critic \( Q(s, a) \) is the expected reward for state \( s \) and action \( a \):

\[
Q^\pi(s, a) = E_{a \sim \pi} [r | s, a].
\]

(5)

The aim of the actor is selecting an action to maximize its critic value \( Q(s, a) \):

\[
\pi(a|s) = \arg \max_a Q(s, \pi(a|s)).
\]

(6)

In practical implementation, both the critic \( Q(s, a) \) and the actor \( \pi(a|s) \) are approximated by the neural networks \( Q_\phi \) and \( \pi_\theta \) with parameters of \( \phi \) and \( \theta \). \( Q_\phi(s, a) \) is learned by minimizing the MSE loss:

\[
L_Q(\phi) = \frac{1}{|B|} \sum_{(s, a, r) \in B} (Q_\phi(s, a) - r)^2,
\]

(7)

where \( B \) is a mini-batches data \((s, a, r)\) sampling from the reply buffer, and \(|B|\) is the number of samples in the batch.

The actor \( \pi_\theta \) is learned by minimizing the loss function:

\[
L_\pi(\theta) = \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \mu_\theta(s)).
\]

(8)

The proposed one-step DRL method is effective because of the following reasons:

1. The one-step DRL is a specific method to learn from the one-step MDP data.
2. The objective of one-step DRL is simpler. One-step DRL learns the policy to maximize the recent reward \( r_t \) rather than the infinity discounted accumulated reward \( \sum_{t=0}^{\infty} \gamma^t r_t \).
3. The one-step DRL avoids the overestimation issue of critic. In existing off-policy actor-critic DRL algorithms, the estimation error is unavoidable, and the overestimation error would be accumulated when the critic network is learned by using temporal difference learning: \( y_t = r_t + \gamma Q_\phi(s_{t+1}, a_{t+1}), a_{t+1} \sim \pi_\theta(s_{t+1}) \). The overestimation error is accumulated because of the term \( Q_\phi(s_{t+1}, a_{t+1}) \). TD3 or SAC have used clipped double-Q learning to remedy it, but it leads to the underestimation of the Q-function. One-step DRL, specifically designed for IB-VVC, do not need to use the term \( Q_\phi(s_{t+1}, a_{t+1}) \), thus avoiding the overestimation issue completely.

Remark 1: The one-step DRL scheme is the simplest version of DDPG or TD3. Setting \( \gamma = 0 \) for DDPG or TD3 would derive the scheme directly. After setting \( \gamma = 0 \), the target network, clipped double-Q learning, delayed policy updates, target policy smoothing, and the soft updating of the target network are not necessary for DDPG or TD3. Compared with DDPG with 4 neural networks, TD3 with 6 neural networks, the scheme only needs 2 neural networks. However, the scheme would have better performance because it suits the IB-VVC problem specifically.

B. Two-Critic scheme for One-Step DRL

VVC has two objectives: minimizing power loss, and eliminating voltage violation. For multi-objectives, the existing DRL integrates the multi-objective into one objective and uses one critic network to learn it. Neural networks indeed have the universe approximation capability to approximate any continuous function with arbitrarily small error with enough number of hidden nodes theoretically [31]. However, using one critic network to learn the integration of two objectives maybe degrade its approximation accuracy and convergence speed in the learning process because of the following properties of VVC problems:

1) The reward for active power loss \( r_p \) and the reward for voltage violations rate \( r_v \) have different properties. As shown in Fig. 3 the relationship between reactive power injection \( Q_G \) and voltage \( V \) at ADNs is close to linear, whereas the relationship between reactive power injection \( Q_G \) and power loss \( P_{loss} \) is strongly nonlinear. If we use one critic network to approximate the reward containing two objectives, the critic network may mix the two functions. In addition, the data in the reply buffer is dynamic and does not satisfy the independent-identical-distribution condition, and thus it increases the difficulty further.
2) Eliminating voltage violation is more important than minimizing power loss, so generally a large weight \( c_v \) is set to penalize voltage violation. This may result in the critic network approximating more on the reward for voltage violations and less on the reward for active power loss.
3) The difficulty of learning the function of power loss and
voltage violation tends to be different, so the convergence time may not be synchronous. The non-synchronized convergence behavior of two objectives in a critic neural maybe needs more time and data to converge.

To address the above three reasons, instead of using one critic network to approximate the integration of two objectives, we use two critic networks to approximate the two objectives separately. The two-critic scheme decreases the approximation difficulties of each critic, thus having a fast convergence rate and better control performance. Correspondingly, the reward stored in MDP is designed as \( r = [r_p, r_v]^T \).

The critic for power loss \( Q_p(s, a) \) and voltage violation \( Q_v(s, a) \) are

\[
\begin{align*}
Q_p(s, a) &= E_{a \sim \pi} [r_p | s, a] \\
Q_v(s, a) &= E_{a \sim \pi} [r_v | s, a].
\end{align*}
\]

(9)

In real application, the two critics \( Q_p(s, a) \) and \( Q_v(s, a) \) approximated by the two neural networks \( Q_{\phi_p} \) and \( Q_{\phi_v} \) with parameters of \( \phi_p \) and \( \phi_v \). \( Q_{\phi_p}(s, a) \) and \( Q_{\phi_v}(s, a) \) are learned by minimizing the MSE losses,

\[
\begin{align*}
L_Q(\phi_p) &= \frac{1}{|B|} \sum_{(s,a,r_p) \in B} (Q_{\phi_p}(s, a) - r_p)^2 \\
L_Q(\phi_v) &= \frac{1}{|B|} \sum_{(s,a,r_v) \in B} (Q_{\phi_v}(s, a) - c_v r_v)^2.
\end{align*}
\]

(10)

The network actor is updated by maximizing the loss function,

\[
L_\pi(\theta) = \frac{1}{|B|} \sum_{s \in B} (Q_{\phi_p}(s, \pi_\theta(s)) + Q_{\phi_v}(s, \pi_\theta(s))).
\]

(11)

The OSTC-DRL approach integrates a one-step DRL scheme and a two-critic scheme. OSTC-DRL is compatible well with any off-policy actor-critic algorithms. We design an OSTC-DP derived from DDPG to show the approach is compatible well with deterministic policies. Algorithm 1 provides the detail of the OSTC-DP. We also design OSTC-SAC derived from SAC to show the approach is compatible well with stochastic policies. To obtain OSTC-SAC, we need to make little modifications in steps 2 and 9 in Algorithm 1. In step 2, we need to replace the deterministic policy \( \pi_\theta(s) = \text{tanh}(\mu_\theta(s)) \) as the stochastic policy \( \pi_\theta(s, \xi) = \text{tanh}(\mu_\theta(s) + \sigma_\theta(s) \odot \xi), \xi \sim \mathcal{N}(0, I) \), where \( \mu_\theta, \sigma_\theta \) are neural networks. In step 9, we need to add the entropy regularization term, \( \nabla_{\theta} \frac{1}{|B|} \sum_{s \in B} (Q_{\phi_p}(s, \pi_\theta(s)) + c_v Q_{\phi_v}(s, \pi_\theta(s)) - \alpha \log \pi_\theta(\pi_\theta(s) | s)) \).

Algorithm 1 One-Step Two-Critic with Deterministic Policy (OSTC-DP) algorithm

**Input:** Initial policy parameters \( \theta \), Q-function parameters \( \phi_p, \phi_v \), empty replay buffer \( D \);

1: repeat
2: Observe state \( s \), and select action \( a = clip(\pi_\theta(s) + \epsilon, a_{\text{Low}}, a_{\text{High}}) \), where \( \epsilon \sim \mathcal{N} \);
3: Execute \( a \) in the environment, and obtain reward \( r_p, r_v \);
4: Store \( (s, a, r_p, r_v) \) in replay buffer;
5: if it’s time to update then
6: for \( j \) in range (how many updates) do
7: Randomly sample a batch of transitions, \( B = (s, a, r_p, r_v) \) from \( D \);
8: Update \( Q_p \) and \( Q_v \) by one step of gradient descent using
9: \[
\begin{align*}
\nabla_{\phi_p} \frac{1}{|B|} & \sum_{(s,a,r_p) \in B} (Q_{\phi_p}(s, a) - r_p)^2 \\
\nabla_{\phi_v} \frac{1}{|B|} & \sum_{(s,a,r_v) \in B} (Q_{\phi_v}(s, a) - r_v)^2.
\end{align*}
\]

9: Update policy by one step of gradient ascent using
10: \[
\nabla_{\theta} \frac{1}{|B|} \sum_{s \in B} (Q_{\phi_p}(s, \pi_\theta(s)) + c_v Q_{\phi_v}(s, \pi_\theta(s)))
\]
12: until Convergence

IV. EXTENDING OSTC-DRL TO MULTI-AGENT OSTC-DRL FOR DECENTRALIZED IB-VVC

OSTC-DRL is a centralized approach that requires the real-time full information of ADNs. However, the centralized approach requires massive real-time communication and is fragile for single-point failure. For ADNs that each sub-area only can acquire the local information in real-time and upload the information to the center at a slow rate, we extend the OSTC-DRL to multi-agent OSTC-DRL for decentralized IB-VVC. It is based on a centralized training decentralized execution approach. In off-policy DRL algorithms, both actor networks and critic networks are trained by the data sampling from the data buffer, so the interaction data stored in the data buffer with a delayed time has little influence in the training stage. In executing stage, each actor-network of the sub-area just requires real-time local observation to make the decision. The framework of multi-agent OSTC-DRL is shown in Fig. 4.

A. Formulating VVC as One-step Markov Game

Markov game is an extension of MDP for formulating a multi-agent system. Similar to the one-step MDP, we formulate the IB-VVC as a one-step Markov game that which multiple agents interact with the same environment. We use a tuple \((S, [O_i]_n, [A_i]_n, R)\) to describe a Markov game with \( n \) agent. \( s \in S \) is the full state of environment. \( o_i \in O_i \) and \( a_i \in A_i \) are the local observations and actions for each agent \( i \). \( [r_p, r_v]^T \in R \) are the reward functions relating power loss and voltage violation rate that are defined as \( S \times A_1 \times \ldots \times A_N \times \rightarrow R_p, R_v \). The goal of each agent is to maximize the expected reward \( J = E_{a_{i,\sim \pi_i(a_i)}} (r_{p}(s, a_1, a_2, \ldots, a_n) + c_v r_v(s, a_1, a_2, \ldots, a_n)) \).

The definition of state space, local observation space, action space, and reward function are following:
1) State: The state is the same as the centralized version of OSTC-DRL, which is set as $s_t = (P_t, Q_t, V_t, Q_{G,t}, Q_{C,t})$.

2) Local observation: The local observations of each agent is $o_{i,t} = (P_{i,t}, Q_{i,t}, V_{i,t}, a_{i,t-1})$, where $P_{i,t}, Q_{i,t}$ is the vector of nodal active/reactive power injections of $i^{th}$ area, $V_{i,t}$ is the vector of voltage magnitudes of $i^{th}$ area, and $a_{i,t-1}$ is the vector of actions of $i^{th}$ area. The selection of local observation space depends on the measurement conditions of each sub-area. It can be the measurement of one sub-area and its neighbor areas, or just one bus of the IB-ER or SVC installed.

3) Action: The action of each agent is $a_{i,t} = \{Q_{G,i,t'}, Q_{C,i,t'}\}$, where $Q_{G,i,t'}, Q_{C,i,t'}$ are the controllable reactive power resources in $i^{th}$ area, including PV inverters and SVCs.

4) Reward: Reward is the same as the centralized version of OSTC. The reward of power loss $r_{p,t}$ is

$$r_{p,t} = -\sum P_{i,t}.$$  

The reward of voltage violation rate $r_{v,t}$ is

$$r_{v,t} = -\sum \left[ \max (V_t - \bar{V}, 0) + \max (\bar{V} - V_t, 0) \right].$$

B. Multi-Agent OSTC

As shown in Fig. 4, Multi-agent OSTC-DRL contains two stages: centralized training and decentralized execution. In the centralized training stage, the off-policy Multi-agent OSTC-DRL algorithm samples data from the data buffer to train actors and critics. The decentralized execution stage generates interaction data and then the data is stored in the data buffer. The data storage can be at a slow rate. The execution of actors is based on real-time local measurements. It depends on the communication system between each sub-areas of ADNs. Since the data buffer can obtain the full state of ADNs and actions of each actor with the slow rate, one critic $Q(s, a_1, \ldots, a_n)$ is enough to represent the state-action value of all actors. Similar to OSTC, we use two critics to approximate the state-action function for power loss $Q_p(s, a_1, \ldots, a_n)$ and the state-action value for voltage violation $Q_v(s, a_1, \ldots, a_n)$ separately,

$$Q_p^\rho(s, a_1, \ldots, a_n) = E_{a_i \sim \pi_i} \left[ r_p | s, a_1, \ldots, a_n \right]$$

and

$$Q_v^\rho(s, a_1, \ldots, a_n) = E_{a_i \sim \pi_i} \left[ r_v | s, a_1, \ldots, a_n \right].$$

The neural network $Q_{\phi_p}(s, a_1, \ldots, a_n)$ and $Q_{\phi_v}(s, a_1, \ldots, a_n)$ are learned by minimizing the MSE loss with stochastic gradient descent,

$$\nabla_{\phi_p} \frac{1}{|B|} \sum_{(s,a_1,\ldots,a_n,r_p) \in B} \left( Q_{\phi_p}(s, a_1, \ldots, a_n) - r_p \right)^2$$

and

$$\nabla_{\phi_v} \frac{1}{|B|} \sum_{(s,a_1,\ldots,a_n,r_v) \in B} \left( Q_{\phi_v}(s, a_1, \ldots, a_n) - r_v \right)^2.$$

The network actors are updated with stochastic gradient ascent with respect to the sum of $Q_{\phi_p}$ and $Q_{\phi_v}$,

$$\nabla_{\theta_1,\ldots,\theta_n} \frac{1}{|B|} \sum_{s \in B} \left( Q_{\phi_p}(s, \pi_{\theta_1}(o_1), \ldots, \pi_{\theta_n}(o_n)) + c_v Q_{\phi_v}(s, \pi_{\theta_1}(o_1), \ldots, \pi_{\theta_n}(o_n)) \right).$$

Algorithm 2 shows the details of Multi-agent OSTC-DP. In step 6 of Algorithm 2, we need to collect the $(s, a_1, \ldots, a_n)$ of the ADNs at the same time. It is accessible because recently measurement devices can add a timestamp to the measured data, and then the data can be updated to the center with a slow time rate or a constant time interval decay. Of course, those data may be non-synchronized, additional methods can be used to preprocess the non-synchronized data.

To obtain Multi-agent OSTC-SAC, we need to make little modifications in steps 3 and 10 in Algorithm 2. In step 3, we need replace the policy as the stochastic policy.
\[ \pi_{\theta_i}(o_i, \xi) = \tanh(\mu_{\theta_i}(o_i) + \sigma_{\theta_i}(o_i) \odot \xi_i), \quad \xi_i \sim \mathcal{N}(0, I), \]
where \( \mu_{\theta_i}, \sigma_{\theta_i} \) are neural networks. In step 10, we need add the entropy regularization term, \( \nabla_{\theta_1, ..., \theta_n} \frac{1}{|B|} \sum_{s \in B} \left( Q_{\theta_a}(s, \pi_{\theta_1}(o_1), ..., \pi_{\theta_n}(o_n)) + c_v Q_{\phi_v}(s, \pi_{\theta_1}(o_1), ..., \pi_{\theta_n}(o_n)) - \sum_{i}^{n} \alpha \log \pi_{\theta_i}(\pi(\theta_i) | s) \right) \). We name the algorithm derived from OSTC-SAC as Multi-agent OSTC-SAC.

Multi-agent OSTC-DP and Multi-agent OSTC-SAC are flexible for the measurement conditions of ADNs. The algorithms work well on the measurement conditions that each actor executes based on the local single bus information or sub-area information.

V. SIMULATION

Numerical simulation is conducted on IEEE 33-bus [33] and 69-bus [34] distribution test systems to demonstrate the advantages of the proposed OSTC-DRL approach. In the 33-bus system, 3 IB-ERs of 2.5 MVA capacity and 1.5 MW active power are connected to buses 18, 22, 25, respectively, and 1 SVC of 2 MVA is connected to bus 33. In the 69-bus system, 4 IB-ERs of 2.5 MVA capacity and 1.5 MW active power are connected to buses 6, 24, 45, 58 respectively, and 1 SVC of 2 MVA is connected to bus 14. All load and generation levels are multiplied with the fluctuation ratio [20] and a 20% uniform distribution noise to reflect the variance. Each day has 96 data. The voltage limits for all buses are set to be [0.95, 1.05]. The algorithms are implemented in Python. The balanced power flow is solved by Pandapower [35] to simulate ADNs, and the implementation of the DRL algorithms uses PyTorch.

A. Simulation for the Centralized OSTC-DRL Approach

We perform simulations to understand the contribution of each component: one-step and two-critic, and show the superiority of the proposed centralized OSTC-DRL approach with both deterministic policies and stochastic policies. We design 3 classes of simulation experiments.

| Table I | Parameter Setting for the Reinforcement Learning Algorithm |
|-----------------|-----------------|
| Algo. | Parameter | Value |
| optimizer | Adam |
| Activation function | ReLU |
| Number of hidden layers | 2 |
| Actor hidden layer neurons | {512, 512} |
| Critic hidden layer neurons | {512, 512} |
| Batch size | 128 |
| Replay buffer size | 3 \times 10^3 |
| The deterministic policy | Exploration Policy | \mathcal{N}(0.0,1) |
| Entropy target | \text{Temperature learning rate} | 3 \times 10^{-4} |
| The stochastic policy | Gaussian distribution |
| Table II | Parameter Setting for the Reinforcement Learning Algorithm |
|-----------------|-----------------|
| Table III | Parameter Setting for the Reinforcement Learning Algorithm |
| Table IV | Parameter Setting for the Reinforcement Learning Algorithm |

![Fig. 5. Testing results of the training stage for the 33-bus system.](image1)

**Deterministic policy:** 1) DDPG [36]; 2) One-step with deterministic policy (OS-DP) derived from DDPG; 3) Two-critic with deterministic policy (TC-DP) derived from DDPG; 4) One-step two-critic with deterministic policy (OSTC-DP) derived from DDPG;

**Stochastic policy:** 5) SAC [5]; 6) One-step with stochastic policy (OS-SAC) derived from SAC; 7) Two-critic with stochastic policy (TC-SAC) derived from SAC; 8) One-step two-critic with stochastic policy (OSTC-SAC) derived from SAC.

**Model-based:** 9) Model-based optimization method with accurate power flow model. Model-based optimization is solved by recalling PandaPower with the interior point solver.

The result of model-based optimization can be seen as the optimal result, which is a baseline result for the performance of DRL algorithms. We train the DRL agent using 300 days of data. The parameter setting for four DRL algorithms is provided in Table I. The discount factor \( \gamma \) for DDPG, TC-DP, SAC, TC-SAC is 0.9. In the training process, we test the DRL algorithm in the same environment at each step.

The testing results in the training process are shown in Figs. [5] [6]. The reward, power loss, and voltage violation rate
TABLE II
QUANTIFIED INDICES OF THE BENCHMARKS IN THE FINAL 50 EPISODES FOR CENTRALIZED DRL ALGORITHMS

| Algorithm          | Reward   | $P_{\text{loss}}$/MW | VVR/p.u.² |
|--------------------|----------|-----------------------|-----------|
|                    | 33-bus   | 69-bus                | 33-bus    | 69-bus    | 33-bus    | 69-bus    |
| Model-based MBO¹   | −4.199   | −3.957                | 4.199     | 3.957     | 0         | 0         |
| Deterministic      |          |                       |           |           |           |           |
| policy             | DDPG     | −4.898                | −4.701    | 4.898     | 4.622     | 4.882e-6  | 1.580e-3  |
|                    | OS-DP    | −4.867                | −4.867    | 4.867     | 4.778     | 1.129e-5  | 1.782e-3  |
|                    | TC-DP    | −4.289                | −4.217    | 4.288     | 4.140     | 2.642e-5  | 1.533e-3  |
|                    | OSTC-DP  | −4.270                | −4.161    | 4.268     | 4.104     | 3.988e-5  | 1.137e-3  |
| Stochastic         | SAC      | −4.548                | −4.491    | 4.548     | 4.471     | 0         | 3.859e-4  |
| policy             | OS-SAC   | −4.368                | −4.383    | 4.357     | 4.339     | 2.00e-4   | 8.800e-4  |
|                    | TC-SAC   | −4.384                | −4.275    | 4.373     | 4.205     | 2.184e-4  | 1.396e-3  |
|                    | OSTC-SAC | −4.272                | −4.191    | 4.268     | 4.121     | 9.205e-5  | 1.402e-3  |

¹ MBO is the model-based optimization method using an accurate ADN model, which can be seen as the optimal result. However, the accurate ADN model is not available in real applications.
² VVR is the daily accumulation voltage violation rate.

First, the proposed OSTC-DP and OSTC-SAC converge faster compared with DDPG and SAC [see the learning trajectory from days 20-50].

Second, the proposed OSTC-DP and OSTC-SAC achieve higher rewards considerably compared with DDPG and SAC [see the learning trajectory from days 250-300]. OSTC-DP and OSTC-SAC also have smaller power losses.

The voltage violation rates of all DRL algorithms have a small fluctuation around zeros. Increasing the voltage violation penalty will decrease the voltage violation rate, but the method just can alleviate the issues and cannot avoid voltage violation completely. DRL algorithms learn by trial and error, so in the training trajectory, DRL algorithms must trail both sides of the voltage boundary many times to find the optimal solution. The alternative way to address the voltage violation rates issues is tightening the voltage limits[9]. For example, for the normal voltage operation interval [0.95, 1.04], we set the objective voltage interval for DRL algorithms is [0.955, 1.045]. Even there are slight voltage violations for the interval [0.955, 1.045] in the learning process, it is no voltage violation for the interval [0.95, 1.05].

To show the super-performance of the proposed OSTC-DRL approach quantitatively, Table II gives the quantified results of the 9 methods. We use the accuracy equation $Acc = (R_i - R_{\text{MBO}})/R_{\text{MBO}}$ where $i$ represent OSTC-DP, DDPG, OSTC-SAC, or SAC, and MBO represents model based optimization method using an accurate power flow model. From the perspective of reward, the accuracy of OSTC-DP, DDPG, OSTC-SAC, SAC are 1.694%, 16.65%, 1.742%, and 8.304% in the 33-bus case, 5.133%, 18.79%, 5.891%, 13.47% in the 69-bus system. We can see that the accuracy of OSTC-DP is 9.831 times as DDPG in the 33-bus system, and 3.660 times as DDPG in the 69-bus system. The accuracy of OSTC-SAC is 4.766 times as SAC in the 33-bus system, and 2.286 times as SAC in the 69-bus system. The results show that the OSTC-DRL approach can improve the performance of DRL algorithms considerably for both deterministic policies and stochastic policies. The reason for the performance of the 33-bus system over the corresponding 69-bus system may be that the task on the 69-bus system is more complex than the

Fig. 6. Testing results of the training stage for the 69-bus system.

in those figures are the daily accumulation values. To show the superiority of the proposed OSTC-DRL approach over the traditional DRL approaches clearly, we omit the results of OS-DP, TC-DP, OS-SAC, and TC-SAC. We make two observations from Figs. 5, 6.
The contribution of two-critic

Achieve the best performance among the algorithms.

IEEE 69-bus system. The contribution of the component of except that compares the results of DDPG and OS-DP for the

Double-Q learning, and entropy regularization bring positive

Convergence to a bad local optimum. However, entropy regular-

Overestimation of Q value [5], [15] while bringing an

Remark 2: Comparing the performance between deterministic policies (DDPG, OS-DP, TC-DP, and OSTC-DP) and stochastic policies (SAC, OS-SAC, TC-SAC, OSTC-SAC) is not the scope of the paper, but we discuss the difference between the two class of algorithms in the remark. Different from DDPG, SAC has extra three components: clipped double-

33-bus system.

To show the contribution of the component of one-step and two-critic, Fig. 7 gives the ablation study results. The contribution of the component of one-step is shown in Fig. 7(a) One-step improves the performance of those algorithms except that compares the results of DDPG and OS-DP for the IEEE 69-bus system. The contribution of the component of the two-critic is shown in Fig. 7(b) Two-critic improves the performance of all of those algorithms. The contribution of the combination of the one-step and the two-critic is shown in Fig. 7(c) The algorithms with the OSTC-DRL approach achieve the best performance among the algorithms.

Remark 2: Comparing the performance between deterministic policies (DDPG, OS-DP, TC-DP, and OSTC-DP) and stochastic policies (SAC, OS-SAC, TC-SAC, OSTC-SAC) is not the scope of the paper, but we discuss the difference between the two class of algorithms in the remark. Different from DDPG, SAC has extra three components: clipped double-Q learning, entropy regularization for Q updating, and entropy regularization for actor updating. OS-SAC and OSTC-SAC only have clipped double-Q learning and entropy regularization for actor updating. "clipped double-Q learning" mitigates the overestimation of Q value [5], [15] while bringing an underestimation bias. Two entropy regularization components accelerate learning and prevent the policy from prematurely converging to a bad local optimum. However, entropy regularization may bring additional regularization errors when the estimation accuracy of the Q function is high enough. When the estimation of Q value does not have high accuracy, clipped double-Q learning, and entropy regularization bring positive influence and lead SAC and OS-SAC over DDPG and OS-DP. However, when the estimation of Q value has high accuracy, clipped double-Q learning, and entropy regularization bring negative influence and lead TC-DP and OSTC-DP over TC-SAC and OSTC-SAC.

B. Simulation for the Multi-Agent OSTC-DP Algorithm

The proposed multi-agent algorithms are designed for measurement conditions of ADNs in that all the measurements are uploaded to the center at a slow rate and each agent only can obtain its sub-area measurements in real-time. We test the multi-agent algorithms on the two measurement conditions: 1) each agent can obtain the local one bus information, 2) each agent can obtain the sub-area bus information. This setting is to show the multi-agent OSTC-DRL is flexible to different measurement situations.

For the first measurement conditions, the 33-bus system is divided into 4 sub-areas and the 69-bus test system is divided into 5 sub-areas. Each subarea contains one bus in which the controllable devices have been installed. For the second measurement condition, we divide the distribution networks into 4 sub-areas for both case 33 and 69 systems. In the 33-bus test system, the sub-areas are [7, 8, ..., 18], [19, 20, 21, 22], [23, 24, 25] and [26, 27, ..., 33]. In the 69-bus test system, the sub-areas are [2, 3, ..., 10], [11, 12, ..., 26], [36, 37, ..., 45] and [53, 54, ..., 64]. The partitioning is flexible. Some buses can belong to two partitions concurrently, or not belong to any partitioning.

Correspondingly, we extend OSTC-DP to multi-agent OSTC-DP-local and multi-agent OSTC-DP-sub for two measurement conditions. Also, we extend OSTC-SAC to multi-agent OSTC-SAC-local and multi-agent OSTC-SAC-sub. The parameters of DRL algorithms are the same as the corresponding subsection except for the number of actors. We test the performance of DRL algorithms at each step in the training process. Table II shows the qualified indices of the 6 algorithms in the final 50 episodes.

After enough time to learn, for deterministic policies, multi-agent OSTC-DP-sub achieves a similar performance as OSTC-DP, while the performance of multi-agent OSTC-DP-local is slightly worse than the performance of OSTC-DP. For stochastic policies, the rank of the performance of three algorithms from high to low is OSTC-SAC, multi-agent OSTC-SAC-sub, and multi-agent OSTC-SAC-local. Those results show that the
multi-agent OSTC-DP algorithm is robust for the information obtained by each actor in the execution stage, even when each actor just can obtain its local one bus information. It is reasonable because the voltage information of one bus is influenced by other buses, so it can reflect the global information partially. Meanwhile, the partial information degrades the control performance of DRL slightly.

VI. CONCLUSION

This paper proposed an OSTC-DRL approach for IB-VVC in ADNs. Based on the OSTC-DRL approach, we designed two DRL algorithms that are OSTC-DP and OSTC-SAC. We also extended the approach to the multi-agent OSTC-DRL approach for decentralized IB-VVC problems. We designed multi-agent OSTC-DP and multi-agent OSTC-SAC algorithms and they work well on two measurement conditions. The DRL with the OSTC-DRL approach outperforms the state-of-the-art DRL algorithms because of two reasons: 1) the algorithm focuses on solving one-step MDP problem which is consistent with the IB-VVC; 2) The two-critic technology simplifies the approximation task of each critic, thus leading to a better approximation performance. Simulation results show the contributions of one-step and two-critic separately and the OSTC-DRL approach has improved the VVC performance considerably compared with the state-of-the-art DRL algorithm for IB-VVC in ADNs. After extending to the multi-agent DRL algorithm, multi-agent OSTC-DRL algorithms achieve nearly equal or slight degradation performance as the OSTC-DRL algorithms.

The proposed OSTC-DRL approach focus on single-period optimization problems and the action space is continuous. However, for ADNs embedded with capacity banks, on-load tap changers, and storage devices, the actions contain both continuous and discrete, and the optimization task should consider the long horizontal process. Therefore, we would extend our algorithm to mixed-integer multi-period optimization problems in future works.

REFERENCES

[1] H. Liu, C. Zhang, Q. Chai, K. Meng, Q. Guo, and Z. Y. Dong, “Robust regional coordination of inverter-based volt/var control via multi-agent deep reinforcement learning,” IEEE Trans. on Smart Grid, vol. 12, no. 6, pp. 5420–5433, Nov. 2021.

[2] Y. Tang, K. Dvijotham, and S. Low, “Real-time optimal power flow,” IEEE Trans. on Smart Grid, vol. 8, no. 6, pp. 2963–2973, Nov. 2017.

[3] R. Albert, I. Albert, and G. L. Nakarado, “Structural vulnerability of the north american power grid,” Phys. Rev. E, vol. 69, p. 025103, Feb 2004.

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lancot, L. Sifre, D. Kumaran, T. Graepel et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” Science, vol. 362, no. 6419, pp. 1140–1144, 2018.

[5] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, F. Abbeel et al., “Soft actor-critic algorithms and applications,” arXiv preprint arXiv:1812.05905, 2018.

[6] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O’Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Pantiaci, and C. Romero, “Learning to run a power network challenge: a retrospective analysis,” in NeurIPS 2020 Competition and Demonstration Track. PMLR, 2021, pp. 112–132.

[7] H. Liu and W. Wu, “Bi-level off-policy reinforcement learning for volt/var control involving continuous and discrete devices,” arXiv preprint arXiv:2104.05902, 2021.

[8] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, “A multi-agent deep reinforcement learning based voltage regulation using coordinated PV inverters,” IEEE Trans. on Power Syst., vol. 35, no. 5, pp. 4120–4123, Sept. 2020.

[9] P. Kou, D. Liang, C. Wang, Z. Wu, and L. Gao, “Safe deep reinforcement learning-based constrained optimal control scheme for active distribution networks,” Appl. Energy, vol. 264, p. 114772, Apr. 2020.

[10] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[11] F. O. Schaal, “Learning to run a power network challenge: a retrospective analysis,” in NeurIPS 2020 Competition and Demonstration Track. PMLR, 2021, pp. 112–132.

[12] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

TABLE III

QUANTIFIED INDICES OF THE BENCHMARKS IN THE FINAL 50 EPISODES FOR MULTI-AGENT DRL ALGORITHMS

| Algorithm              | Reward | $P_{loss}$/MW | VVR/p.u. |
|------------------------|--------|---------------|----------|
|                        | 33-bus | 69-bus        | 33-bus   | 69-bus   |
| Deterministic policy   | 4.270  | -1.161        | 4.268    | 4.104    |
| MA-OSTC-DP-sub         | 4.264  | -1.142        | 4.262    | 4.115    |
| MA-OSTC-DP-local       | 4.301  | -3.397        | 4.288    | 4.269    |
| MA-OSTC-sub           | 4.272  | -1.191        | 4.268    | 4.121    |
| MA-OSTC-SAC-sub       | 4.343  | -4.234        | 4.317    | 4.128    |
| MA-OSTC-SAC-local     | 4.370  | -4.422        | 4.331    | 4.282    |

1 “MA” means multi-agent, “sub” represents the sub-area bus information, “MA-OSTC-DP-sub” means that the MA-OSTC-DP-sub algorithm works on that each actor executes based on the sub-area bus information.

2 “local” represents the local one bus information. “MA-OSTC-DP-sub” means that the MA-OSTC-DP-sub algorithm works on that each actor executes based on the local one bus information.

...
actor,” in *Int. Conf. Mach. Learn. (ICML)*. Stockholm, Sweden: PMLR, 2018, pp. 1861–1870.

[20] H. Liu and W. Wu, “Two-stage deep reinforcement learning for inverter-based volt-var control in active distribution networks,” *IEEE Trans. on Smart Grid*, vol. 12, no. 3, pp. 2037–2047, May 2021.

[21] W. Wang, N. Yu, Y. Gao, and J. Shi, “Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems,” *IEEE Trans. on Smart Grid*, vol. 11, no. 4, pp. 3008–3018, July 2020.

[22] H. Liu and W. Wu, “Online multi-agent reinforcement learning for decentralized inverter-based volt-var control,” *IEEE Trans. on Smart Grid*, vol. 12, no. 4, pp. 2980–2990, July 2021.

[23] Z. Yan and Y. Xu, “Real-time optimal power flow: A lagrangian based deep reinforcement learning approach,” *IEEE Trans. on Power Syst.*, vol. 35, no. 4, pp. 3270–3273, July 2020.

[24] W. Wang, N. Yu, Y. Gao, and J. Shi, “Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems,” *IEEE Trans. on Smart Grid*, vol. 11, no. 4, pp. 3008–3018, July 2020.

[25] H. Liu and W. Wu, “Online multi-agent reinforcement learning for decentralized inverter-based volt-var control,” *IEEE Trans. on Smart Grid*, vol. 12, no. 4, pp. 2980–2990, July 2021.

[26] E. Dall’Anese, S. V. Dhople, and G. B. Giannakis, “Photovoltaic inverter controllers seeking ac optimal power flow solutions,” *IEEE Trans. on Power Syst.*, vol. 31, no. 4, pp. 2809–2823, July 2015.

[27] H. J. Liu, W. Shi, and H. Zhu, “Distributed voltage control in distribution networks: Online and robust implementations,” *IEEE Trans. on Smart Grid*, vol. 9, no. 6, pp. 6106–6117, Nov. 2018.

[28] A. Singhal, V. Ajjarpur, J. Fuller, and J. Hansen, “Real-time local volt/var control under external disturbances with high PV penetration,” *IEEE Trans. on Smart Grid*, vol. 10, no. 4, pp. 3849–3859, July 2018.

[29] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, “Two-timescale voltage control in distribution grids using deep reinforcement learning,” *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2313–2323, May 2020.

[30] V. Kekatos, G. Wang, A. J. Conejo, and G. B. Giannakis, “Stochastic reactive power management in microgrids with renewables,” *IEEE Trans. on Power Syst.*, vol. 30, no. 6, pp. 3386–3395, Nov. 2015.

[31] T. Chen and H. Chen, “Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems,” *IEEE Trans. Neural Netw.*, vol. 6, no. 4, pp. 911–917, Jul. 1995.

[32] A. Alimardani, F. Therrien, D. Atanackovic, J. Jatskevich, and E. Vahidi, “Distribution system state estimation based on nonsynchronized smart meters,” *IEEE Trans. on Smart Grid*, vol. 6, no. 6, pp. 2919–2928, Nov. 2015.

[33] M. E. Baran and F. F. Wu, “Network reconfiguration in distribution systems for loss reduction and load balancing,” *IEEE Power Energy Mag.*, vol. 9, no. 4, pp. 101–102, Apr. 1989.

[34] D. Dae, “Optimal placement of capacitors in radial distribution system using a fuzzy-GA method,” *Int. J. Electr. Power Energy Syst.*, vol. 30, no. 6-7, pp. 361–367, July 2008.

[35] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, “pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems,” *IEEE Trans. on Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.

[36] D. Silver, G. Levess, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in *Int. Conf. Mach. Learn. (ICML)*. Beijing, China: PMLR, 2014, pp. 387–395.