Volt-VAR Control in Power Distribution Systems with Deep Reinforcement Learning

Wei Wang, Nanpeng Yu, Jie Shi, and Yuanqi Gao
Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, CA
Email: {wwang031, nyu, jshi005, ygao024}@ucr.edu

***Abstract*—Volt-VAR control (VVC) plays an important role in enhancing energy efficiency, power quality, and reliability of electric power distribution systems by coordinating the operations of equipment such as voltage regulators, on-load tap changers, and capacitor banks. VVC not only keeps voltages in the distribution system within desirable ranges but also reduces system operation costs, which include network losses and equipment depreciation from wear and tear. In this paper, the deep reinforcement learning approach is taken to learn a VVC policy, which minimizes the total operation costs while satisfying the physical operation constraints. The VVC problem is formulated as a constrained Markov decision process and solved by two policy gradient methods, trust region policy optimization and constrained policy optimization. Numerical study results based on IEEE 4-bus and 13-bus distribution test feeders show that the policy gradient methods are capable of learning near-optimal solutions and determining control actions much faster than the optimization-based approaches.**

***Index Terms*—Reinforcement learning, Volt-VAR control, constrained Markov decision process, policy gradient methods.**

I. Introduction

As the penetration level of distributed energy resources (DERs) continues to rise in power distribution systems, it is increasingly difficult to keep the voltages along the feeders within the desired range. The voltage profile highly impacts the electricity service quality for end users. Both over-voltage and under-voltage conditions could reduce energy efficiency, cause equipment malfunction, and damage customers' electrical appliances. Equipped with remote control and monitoring devices, electric utilities started adopting Volt-VAR control (VVC) to maintain voltages within allowable range, manage power factor, and reduce operation costs. These control objectives can be achieved by coordinating the operations of various equipment such as voltage regulators, on-load tap changers, switchable capacitor banks, and smart inverters.

Although successful field demonstrations of VVC have been reported by many electric utilities, there are still many barriers to the wide-spread adoption of the technology. One of the most significant barriers is the lack of robust distribution network topology and parameter information, which are required in optimization based VVC approaches. In particular, inaccurate distribution secondary systems' information [1]–[3] makes it difficult for VVC to ensure that customers' voltages will stay within the acceptable range. To overcome the drawbacks of optimization-based approaches, we develop a data-driven deep reinforcement learning based approach to solve the VVC problem.

The existing algorithms for VVC can be divided into two categories: optimization-based approach and reinforcement learning based approach. The optimization-based approach to solve the VVC problem has been well researched. The VVC problem is formulated as a deterministic optimization problem with different extensions [4]–[7]. Voltage-dependent load model is introduced in [4]. Continuous controllable reactive power source is considered in [5]. The interaction between the Volt-VAR optimizer and prosumers is incorporated in a game theory model [6]. Considering the uncertainties of DERs, the VVC problem is formulated as a robust optimization problem [8], [9]. Both papers propose a two-stage coordination scheme for the VVC, which consists of the less-frequent control for on-load tap changers and the more-frequent control for smart inverters. Model predictive control (MPC) based VVC is studied in [10], [11] to reduce real power losses and voltage fluctuation [10] and preserve the life of controllable equipment by penalizing the number of tap changes [11]. In the optimization-based approach, the VVC problem is typically formulated as a mixed-integer conic programming (MICP) or mixed-integer nonlinear programming problem. The computational complexity of the solution algorithms for these NP-hard problems increases exponentially with the distribution network size and the number of controllable devices. Thus, the optimization-based approach does not scale well for real-time application of VVC.

The reinforcement learning approach is capable of making control decisions online based on off-line trained models. In particular, Q-learning based algorithms are developed for the VVC problem [12]–[14]. The tabular Q-learning method is adopted to solve the VVC problem [12]. A tabular Q-learning method is proposed to solve the optimal reactive power dispatch problem [13], where the global reward is obtained with a consensus-based global information discovery algorithm. In [14], separate Q-values of on-load tap changers are approximated sequentially by radial kernel functions. So far, all reinforcement learning based algorithms developed for the VVC problem are action-value methods [15], [16]. They learn the values of actions and then select actions based on estimated action values.

In this paper, we adopt a different reinforcement learning approach called policy gradient methods [17]–[20] to solve the VVC problem. Policy gradient methods directly learn a parameterized control policy that can select actions without using a value function. Policy gradient methods have two advantages over action-value methods. First, the VVC policy may be a simpler function to approximate than the action-value function. Second, continuous policy parameterization yields stronger convergence guarantees for policy-gradient methods than the ϵ-greedy action selection for action-value methods [21]. Compared to the optimization-based approaches, our proposed algorithm has better scalability and does not require accurate and complete physical model of the distribution network.

The existing reinforcement learning based VVC works allow controllers to freely explore any control actions during learning. However, certain control actions will lead to severe voltage violations in the distribution feeder. To enable safe exploration for controllers, we adopt the constrained policy optimization [20] algorithm, which statistically guarantees every control policy during learning will satisfy operational constraints in the form of expectation.

The remainder of the paper is organized as follows. Section II presents the formulations of the VVC problem as an optimization problem and as a constrained Markov decision process (CMDP) problem. Section III describes how to leverage policy gradient methods to solve the VVC problem. Section IV shows the numerical results, which demonstrate the performance of our proposed reinforcement learning based VVC algorithms. Section V concludes the paper.

II. Problem Formulation

In this section, we first formulate the VVC problem as an optimization problem and then as a CMDP problem.

A. Volt-VAR Control Formulated as an Optimization Problem

VVC algorithm aims at minimizing the total system losses and equipment operation costs while satisfying voltage constraints. In this formulation, we assume the voltage regulators, on-load tap changers and capacitor banks are the primary control knobs. Then, the VVC problem can be formulated as an optimization problem as follows [22]:

$$\begin{aligned}\min C_p[\sum_{t=1}^{T} P_{loss}^t] &+ C_r \sum_{t=1}^{T}\sum_{j=1}^{N_r} |Tap_j^r(t) - Tap_j^r(t-1)| \\ &+ C_l \sum_{t=1}^{T}\sum_{j=1}^{N_l} |Tap_j^l(t) - Tap_j^l(t-1)| \\ &+ C_c \sum_{t=1}^{T}\sum_{j=1}^{N_c} |Tap_j^c(t) - Tap_j^c(t-1)| \qquad (1)\end{aligned}$$

s.t.

$$f_{PB}(\boldsymbol{PG}^t, \boldsymbol{QG}^t, \boldsymbol{PD}^t, \boldsymbol{QD}^t, \boldsymbol{TAP}_t, \boldsymbol{u}^t, \boldsymbol{l}^t) = 0, \forall t \qquad (2)$$

$$f_{OL}(\boldsymbol{PF}^t, \boldsymbol{QF}^t, \boldsymbol{TAP}_t, \boldsymbol{u}^t, \boldsymbol{l}^t) = 0, \forall t \qquad (3)$$

$${PF_{ij}^t}^2 + {QF_{ij}^t}^2 = l_{ij}^t u_i^t, \forall i, j \in \mathcal{N}, (i,j) \in E, t \qquad (4)$$

$$\underline{u} \leq u_i^t \leq \overline{u}, \forall i \in \mathcal{N}, t \qquad (5)$$

The objective function (1) minimizes the total operation costs, which include the costs associated with line losses and the switching costs of voltage regulators, on-load tap changers, and capacitor banks. The switching cost is assumed to be proportional to the absolute number of tap changes between consecutive hours. P_{loss}^t denotes the total real line losses at hour t. C_p, C_r, C_l, and C_c are the cost coefficients for the real power loss, the tap changes of voltage regulators, on-load tap changers, and capacitor banks respectively. N_r, N_l, and N_c are the total numbers of voltage regulators, on-load tap changers, and capacitor banks. $Tap_j^r(t)$, $Tap_j^l(t)$, and $Tap_j^c(t)$ denote the tap position of the j-th voltage regulator, on-load tap changer, and capacitor bank at hour t. T is the operation horizon of the VVC algorithm.

The formulation of constraints leverages the DistFlow equations [23]. The decision variables of the DistFlow formulation are the vector ($\boldsymbol{u}^t$) of u_i^t for all the nodes ($\mathcal{N}$), the vector ($\boldsymbol{l}^t$) of l_{ij}^t for all the lines (E), and the vector ($\boldsymbol{TAP}_t$) of tap positions for all the devices. u_i^t denotes the square of voltage magnitude of node i at hour t. l_{ij}^t denotes the square of current magnitude of the line connecting node i and j at hour t.

The set of power balance constraints in the DistFlow is represented by (2), where $\boldsymbol{PG}^t$, $\boldsymbol{QG}^t$, $\boldsymbol{PD}^t$, and $\boldsymbol{QD}^t$ denote the vector of nodal real and reactive power generations and demands at hour t. The constraints corresponding to the Ohm's law is represented by (3), where $\boldsymbol{PF}^t$ and $\boldsymbol{QF}^t$ denote the vector of real and reactive power flows at hour t. Equality constraint (4) is the only nonlinear constraint in the DistFlow formulation, which can be relaxed as a second order cone [23]. PF_{ij}^t and QF_{ij}^t are the real and reactive power flow on the line connecting node i and j at hour t. E and $\mathcal{N}$ denote the set of edges and nodes in the distribution feeder. Equation (5) represents the nodal voltage constraints, where $\underline{u}$ and $\overline{u}$ are the lower and upper limits for the square of voltage magnitude. The detailed formulations for the operating constraints can be found in [22], where binary variables are introduced to represent the tap positions. The optimization problem shown above is a MICP problem.

Finally, to account for generation and load uncertainties, the VVC problem can be formulated as a MPC [10]. The optimization problem shown above can be solved on a rolling basis based on the updated load and generation forecasts.

B. Volt-VAR Control Formulated as a Constrained Markov Decision Process

In the Markov decision process (MDP), the grid operator or controller is denoted by an agent. This agent and the distribution grid interact at each of a sequence of discrete time steps $t = 0, 1, 2, \ldots$. At each time step t, the agent receives the system's state $s_t \in \mathcal{S}$, and selects a control action $a_t \in \mathcal{A}(s)$. One time step later, the agent receives a numerical reward $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$, and finds itself in a new state s_{t+1}. The probability of receiving a reward and observing a new state depends on the preceding state and control action as $P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_0, a_0, ..., s_t, a_t)$.

In the context of the VVC, the state is defined as $s = [\boldsymbol{P}, \boldsymbol{Q}, \mathcal{T}, t]$, where $\boldsymbol{P}, \boldsymbol{Q}, \mathcal{T}$ and t denote the nodal real and reactive power injections, the current tap positions, and the time step. The action taken by a VVC agent is changing the tap positions of controllable devices to $\mathcal{T}'$. The size of the action space is $\Pi_{i=1}^{N_s} n_i$, where $N_s = N_r + N_l + N_c$ is the number of controllable devices and n_i denotes the number of tap positions of device i. The reward received by the controller $R(s_t, a_t, s_{t+1})$ for taking action a_t at state s_t and reaching state s_{t+1} is defined as the negative of the system operational costs, which include the costs associated with real power losses and equipment operations.

$$\begin{aligned} &R(s_t, a_t, s_{t+1}) \\ &\quad = -\Big[C_p P_{loss}^t + C_r \sum_{j=1}^{N_r} |Tap_j^r(t+1) - Tap_j^r(t)| \\ &\qquad + C_l \sum_{j=1}^{N_l} |Tap_j^l(t+1) - Tap_j^l(t)| \\ &\qquad + C_c \sum_{j=1}^{N_c} |Tap_j^c(t+1) - Tap_j^c(t)|\Big] \end{aligned} \tag{6}$$

The goal of an agent is to find a control policy π that maximizes the expected discounted return defined as:

$$J(\pi) = \mathop{E}_{\tau\sim\pi}[\sum_{t=0}^{T} G(\tau)] \tag{7}$$

where control policy π is a mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$ for a deterministic policy and a mapping from states to probabilities of selecting each possible action for a probabilistic policy. τ is a trajectory or sequence of states and actions, $\{s_0, a_0, s_1, a_1, ..., s_{T-1}, a_{T-1}, s_T\}$. $G(\tau)$ is the discounted return along a trajectory. $G(\tau) = \sum_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})$, where $\gamma \in (0, 1)$ is the discount factor.

Two important functions, action-value function and state-value function for policy π are defined as follows [21]:

$$Q^\pi(s, a) = \mathop{E}_{\tau\sim\pi}[G(\tau)|s_0 = s, a_0 = a] \tag{8}$$

$$V^\pi(s) = \mathop{E}_{\tau\sim\pi}[G(\tau)|s_0 = s] \tag{9}$$

The action-value function $Q^\pi(s, a)$ represents the expected return starting with state s, taking action a, and following π thereafter. The state-value function $V^\pi(s)$ represents the expected return starting with state s and thereafter following policy π.

To enforce the voltage constraints, we augment the MDP with a set of cost functions $R_C(s_t, a_t, s_{t+1})$. For the VVC problem, it is defined as the number of voltage violations across all nodes, i.e.,

$$R_C(s_t, a_t, s_{t+1}) = \sum_{i=1}^{N} [\mathbb{1}(|v_i^{t+1}| > \overline{v}) + \mathbb{1}(|v_i^{t+1}| < \underline{v})] \tag{10}$$

where $\mathbb{1}(\cdot)$ is the indicator function; v_i^{t+1} is the voltage of node i at hour $t+1$; $\overline{v}$ and $\underline{v}$ are the upper and lower limits for voltage magnitudes. Additional operating constraints such as the line flow limits could be incorporated in a similar manner.

Now the expected discounted return of policy π with respect to the cost function can be defined as

$$J_C(\pi) = \mathop{E}_{\tau\sim\pi}[\sum_{t=0}^{T} \gamma^t R_C(s_t, a_t, s_{t+1})] \tag{11}$$

The final CMDP formulation for the VVC problem is:

$$\max_{\pi} J(\pi) \tag{12}$$

s.t.

$$J_C(\pi) \leq \overline{J} \tag{13}$$

where $\overline{J}$ is the limit for the expected discounted return of the cost function associated with the voltage constraints.

III. Technical Methods

So far all reinforcement learning algorithms adopted to solve the VVC problem have been action-value methods, which approximate the action-value functions through learning and then select actions based on the estimated action-value functions. In this paper, we consider policy gradient methods, which learn a parameterized control policy that directly selects actions without consulting a value function [21]. Typically, an approximate policy is parameterized according to the soft-max in action preferences, which makes approaching deterministic policy easier and finding stochastic policy feasible [21]. Both of these goals can not be achieved by the ϵ-greedy action selection in the action-value methods. Another notable advantage of the policy gradient methods over the action-value methods is that the control policy functions may be easier to approximate than action-value functions in many applications such as the VVC problem.

In this section, we first introduce the preliminaries of the policy gradient methods. Then two state-of-the-art policy gradient methods based on trust region algorithms [18], [20] are adopted to solve the VVC problem. Finally, the design of neural networks to approximate the policy and value functions in the two algorithms will be discussed.

A. Preliminaries of policy gradient method

Policy gradient methods learn a parameterized control policy π_θ that maximizes the performance measure $\hat{J}(\pi_\theta)$ by updating the parameter θ iteratively as follows:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \hat{J}(\theta_k) \tag{14}$$

According to the policy gradient theorem [21], the gradient can be derived as

$$\nabla_\theta \hat{J}(\theta) = \mathop{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\Psi_t] \tag{15}$$

where Ψ may have various forms including the action-value function $Q^{\pi_\theta}(s, a)$ and the advantage function $A^{\pi_\theta}(s, a)$.

The advantage function, which quantifies the improvement by taking action a in state s compared to randomly selecting an action according to policy π_θ and following π_θ afterwards, is defined as

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \tag{16}$$

Two policy gradient methods, trust region policy optimization (TRPO) and constrained policy optimization (CPO), that use the advantage function are presented in the following subsections. We will discuss how to adopt them to solve the VVC problem formulated as MDP and CMDP. The implementation details of these two algorithms can be found in [18], [20].

B. Trust Region Policy Optimization

The TRPO algorithm originally proposed in [18] provides a theoretical guarantee of monotonic improvement of the control policy at each policy iteration step.

The design of the policy iteration procedure is based on the lower bound [20] of the performance improvement of policy $\pi_{\theta'}$ over policy π_θ:

$$\begin{aligned} J(\pi_{\theta'}) - J(\pi_\theta) \geq & \frac{1}{1-\gamma} \mathop{E}_{\substack{s\sim\eta^{\pi_\theta}\\ a\sim\pi_{\theta'}}} \Big[A^{\pi_\theta}(s,a) \\ & - \frac{\gamma\xi^{\pi_{\theta'}}}{1-\gamma}\sqrt{2KL(\pi_{\theta'}||\pi_\theta)[s]}\Big] \end{aligned} \tag{17}$$

where $\xi^{\pi_{\theta'}} = \max_s |E_{a\sim\pi_{\theta'}}[A^{\pi_\theta}(s,a)]|$. $KL(\pi_{\theta'}||\pi_\theta)[s]$ is the KL-divergence between policy $\pi_{\theta'}$ and π_θ at state s. η^{π_θ} is the discounted future state distribution, $\eta^{\pi_\theta}(s) = (1-\gamma)\sum_{t=0}^{T}\gamma^t P(s_t = s|\pi_\theta)$. $P(s_t = s|\pi_\theta)$ denotes the probability of state s appearing at time t under policy π_θ.

Thus, we can update the policy parameters iteratively by maximizing the expected advantage with a small step size δ:

$$\pi_{\theta_{k+1}} = arg\max_{\pi_\theta} \mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_\theta}} [A^{\pi_{\theta_k}}(s,a)] \tag{18}$$

$$s.t. \mathop{E}_{s\sim\eta^{\pi_{\theta_k}}} \big[KL(\pi_\theta, \pi_{\theta_k})[s]\big] \leq \delta \tag{19}$$

If π_{θ_k} is a feasible solution, the maximum expected advantage is non-negative. With a small enough δ, monotonic policy improvement is guaranteed according to (17). The optimization problem (18) and (19) can be solved by linearizing the objective function and quadratically approximating the KL-divergence around θ_k.

The final iterative TRPO algorithm to solve the VVC problem is shown in Algorithm 1.

To adopt the TRPO algorithm for the VVC problem, the reward function is augmented with a penalty term associated with the voltage violations:

$$R'(s_t, a_t, s_{t+1}) = R(s_t, a_t, s_{t+1}) - C_V R_C(s_t, a_t, s_{t+1}) \tag{20}$$

where C_V is the penalty factor for voltage violations.

Algorithm 1 TRPO for VVC

```
Initialize parameters for policy and value function, $\theta_0$, $\phi_0$
for k = 0,1,2,... do
    Generate sample trajectories $Tr_k = \{\tau\}$ with $\pi_{\theta_k}$ through power flow simulations
    Calculate the discounted return for the objective $\hat{G}_t$ after each time step $t$ along the trajectories
    Estimate the advantage for the objective $\hat{A}_t$ based on the value function $V_{\phi_k}$
    Obtain $\pi_{\theta^*_{k+1}}$ by solving (18) and (19)
    Update the parameters $\phi_k$ of the value function neural network with $\hat{G}_t$ as labels
end for
```

C. Constrained Policy Optimization

To directly solve the VVC problem formulated as a CMDP, the CPO algorithm, which guarantees approximate constraints satisfaction, can be leveraged [20]. The theoretical guarantee of the constraint satisfaction can be shown with the upper bound [20] of the performance improvement associated with constraints of policy $\pi_{\theta'}$ compared to policy π_θ:

$$\begin{aligned} J_C(\pi_{\theta'}) - J_C(\pi_\theta) \leq & \frac{1}{1-\gamma} \mathop{E}_{\substack{s\sim\eta^{\pi_\theta}\\ a\sim\pi_{\theta'}}} \Big[A_C^{\pi_\theta}(s,a) \\ & + \frac{\gamma\xi_C^{\pi_{\theta'}}}{1-\gamma}\sqrt{2KL(\pi_{\theta'}||\pi_\theta)[s]}\Big] \end{aligned} \tag{21}$$

where $\xi_C^{\pi_{\theta'}} = \max_s |E_{a\sim\pi_{\theta'}}[A_C^{\pi_\theta}(s,a)]|$ and $A_C^{\pi_\theta}(s,a)$ is the corresponding advantage function for the constraint. According to (21), the constraint at each updating step is specified as:

$$J_C(\pi_{\theta_k}) + \frac{1}{1-\gamma} \mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_\theta}} [A_C^{\pi_{\theta_k}}(s,a)] \leq \overline{J} \tag{22}$$

The policy update for CMDP can be found by solving (18), (19), and (22). Therefore, with a small enough δ, the constraint satisfaction is almost guaranteed at step $k+1$ if we start from a feasible solution π_{θ_k} according to (21). The worst-case constraint violation at step $k+1$ is:

$$\overline{J} - J_C(\pi_{\theta_{k+1}}) \leq \frac{\sqrt{2\delta}\gamma\xi^{\pi_{\theta_{k+1}}}}{(1-\gamma)^2} \tag{23}$$

Similarly, to solve the optimization problem, (22) should be linearized around θ_k. At the beginning of the training process, a feasible solution can be recovered by solving the following problem subject to (19):

$$\min_{\pi_\theta} \mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_\theta}} [A_C^{\pi_{\theta_k}}(s,a)] \tag{24}$$

The final CPO algorithm to solve the VVC problem is shown in Algorithm 2.

D. Value and Policy Networks

Both the objective function (18) and the expectation of the advantage function associated with the constraint in (22) can

Algorithm 2 CPO for VVC

```
Initialize parameters for policy and value functions, $\theta_0$, $\phi_0^1$, and $\phi_0^2$
for k = 0,1,2,... do
    Generate sample trajectories $Tr_k = \{\tau\}$ with $\pi_{\theta_k}$ through power flow simulations
    Calculate the discounted returns $\hat{G}_t^1$, $\hat{G}_t^2$ for the objective function and the constraint after each time step $t$ along the trajectories
    Estimate the advantages for the objective $\hat{A}_t^1$ and the constraint $\hat{A}_t^2$, based on the value functions $V_{\phi_k^1}$ and $V_{\phi_k^2}$.
    if the problem (18), (19) and (22) is feasible then
        Obtain the optimal solution $\pi_{\theta_{k+1}^*}$
    else
        Obtain the solution $\pi_{\theta_{k+1}^*}$ by solving (19) and (24)
    end if
    Update the parameters $\phi_k^1$ and $\phi_k^2$ of the value function neural networks with $\hat{G}_t^1$ and $\hat{G}_t^2$ as labels
end for
```

be calculated with only the state-value function and the policy function as follows:

$$\mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_\theta}}[A^{\pi_{\theta_k}}(s,a)] = \mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_{\theta_k}}}[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A^{\pi_{\theta_k}}(s,a)] =$$
$$\mathop{E}_{\substack{s\sim\eta^{\pi_{\theta_k}}\\ a\sim\pi_{\theta_k}}}[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}(R(s,a,s') + \gamma V^{\pi_{\theta_k}}(s') - V^{\pi_{\theta_k}}(s))] \quad (25)$$

Therefore, we only need to design neural networks to approximate the state-value function and the policy function. The state-value function V_ϕ corresponding to the augmented reward in Algorithm 1 is parameterized with ϕ. The state-value functions corresponding to the reward V_{ϕ_1} and the constraint V_{ϕ_2} in Algorithm 2 are parameterized with ϕ_1 and ϕ_2. The inputs of all the value networks are states. The output is the expected discounted return.

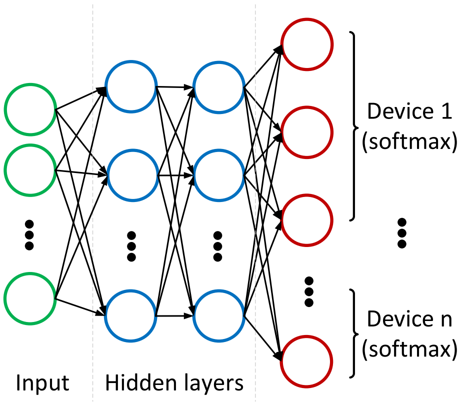

Fig. 1. Structure of the policy network

The policy function π_θ is approximated by a neural network with parameter θ. The structure of the policy network is shown as in Fig. 1. The inputs are the states and the outputs are the probabilities of selecting various actions, which represent the switch status of the devices. The size of the output layer is $\sum_{i=1}^{N_s} n_i$, where N_s and n_i are the number of devices and the number of tap positions for device i. The probability distribution P_i of the actions for device i, is obtained from the subset of the output neurons with size n_i. A softmax activation function is applied to each subset of the output neurons corresponding to a device. The final probability distribution of the tap combinations across all devices is calculated with $P = \Pi_{i=1}^{N_s} P_i$. Thus, in our proposed methods the network size only increases linearly with N_s.

IV. Numerical Study

A. Simulation Setup

The numerical studies are conducted on the IEEE 4-bus and 13-bus distribution test feeders [24]. The real-world smart meter data of an electric utility is used as the nodal load in the simulation environment to generate power flow solutions. The length of historical data is about six months. One week of data during the summer peak are used for the out-of-sample test and the rest are used for training. The length of the VVC optimization horizon or an episode in reinforcement learning is one week. The load time series data is scaled and allocated to each node according to the load profile of the standard test case. Each test feeder has three switching devices: a voltage regulator, an on-load tap changer, and a capacitor bank. Both the voltage regulator and the on-load tap changer have 11 tap positions with turns ratios between 0.95 and 1.05. The capacitor bank can be switched on and off remotely and the number of 'tap positions' is treated to be 2. The size of the action space for each test case is $11 \times 11 \times 2 = 242$. In the 4-bus test feeder, the capacitor bank is placed at node 4. In the 13-bus test feeder, the capacitor bank is placed at node 675. The nominal capacity of the capacitor banks is $200kW$. Initially, the turns ratios of the voltage regulators and on-load tap changers are 1, while the capacitor banks are switched off. The electricity price C_p is assumed to be $\$40/MWh$. The switching costs of the devices C_r, C_l, and C_c are set at $0.1 per tap change.

B. Benchmarking Algorithms

The MPC-based optimization algorithm is chosen as the first benchmark. The control horizon is at 24 hours. The ARIMA [25] model is used to forecast the load during the control horizon. The MICP problem formulated in Section II-A is solved on a rolling basis at each step of MPC. MOSEK and GUROBI are used to solve the MICP problem. The second benchmark is set up by replacing the load forecast with actual load data in the MPC framework. The last benchmark represents the baseline where all switching devices are kept at their initial positions.

C. Policy Gradient Methods

In the TRPO and CPO algorithms, both the value and policy neural networks have two hidden layers with 64 and 32 neurons respectively. The tanh activation function is used in all the hidden layers. The linear and softmax activation functions are used for the output layers of the state-value and the policy networks. In the TRPO algorithm, the reward function is augmented by a penalty cost for voltage constraint violations. The penalty coefficient C_V is \$1 per voltage violation per node. The terminal state is chosen as the last hour of a week for both algorithms.

D. Performance Comparison

The control performances of CPO, TRPO, and MPC-based approaches are evaluated in this subsection. Both the CPO algorithm and the TRPO algorithm are trained for 500 iterations. Each training iteration consists of 298 episodic trajectories, which correspond to about 50,000 samples. Over the training episodes, we record the average discounted return (ADR), which includes the costs associated with the line losses, tap changes, and the penalty of voltage violations. As shown in Fig. 2, the CPO algorithm starts to outperform the TRPO algorithm after about 200 training iterations for the 4-bus test case. For the 13-bus test case, the CPO algorithm always outperform the TRPO algorithm. At the end of the training process, the improvements of episodic returns for both algorithms become saturated.

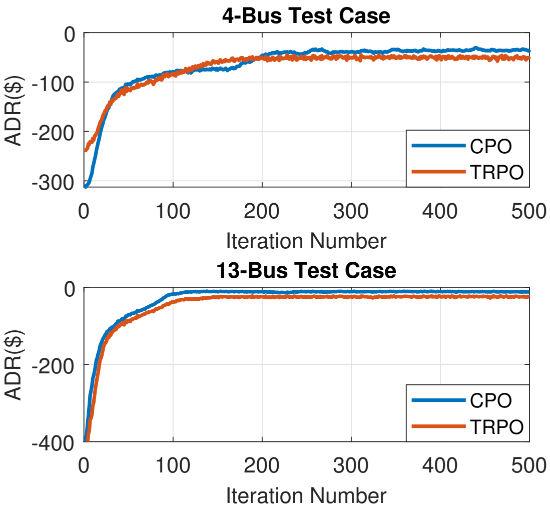

Fig. 2. Training performance of the reinforcement learning algorithms

The total operation cost (OC), the number of tap changes (# of TC), the number of voltage violations (# of VV), and the accumulated per unit voltage violation (AVV) over the test week are recorded in Table I for all the reinforcement algorithms and the benchmark algorithms. The operation cost includes the costs associated with the line losses and the tap changes. The accumulated per unit voltage violation is calculated as $\sum_i^N \sum_t [\max(0, |v_i^t| - \overline{v}) + \max(0, \underline{v} - |v_i^t|)]$.

TABLE I
PERFORMANCE COMPARISON OF VOLT-VAR CONTROL ALGORITHMS

	Algorithm	OC (\$)	# of TC	# of VV	AVV (per unit)
4-bus test case	Baseline	150.13	0	91	2.748
	MPC (Actual)	111.44	18	0	0
	MPC (Forecast)	111.89	20	0	0
	CPO	115.01	9	5	0.044
	TRPO	120.05	3	16	0.286
13-bus test case	Baseline	77.88	0	268	2.673
	MPC (Actual)	58.05	6	0	0
	MPC (Forecast)	58.44	6	0	0
	CPO	58.92	6	0	0
	TRPO	61.29	3	2	0.004

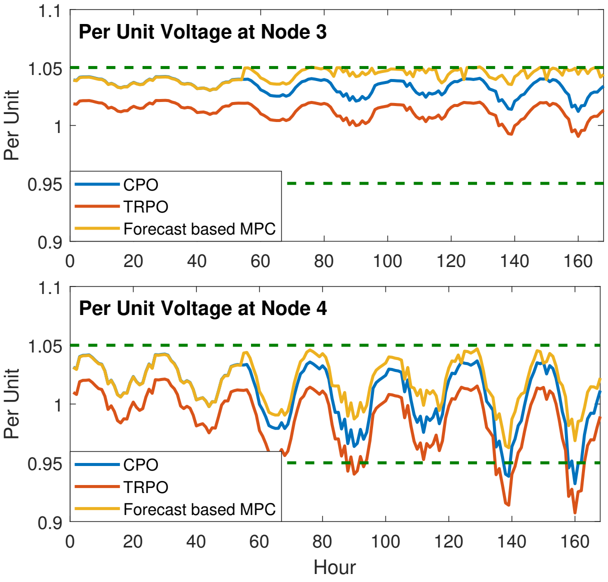

Fig. 3. Comparison of voltage profiles on the 4-bus test feeder

The MPC with actual load represents the global optimal solution. As shown in Table I, the CPO algorithm is capable of achieving a near-optimal operational cost and is nearly constraint-satisfying. The CPO algorithm yields a lower operation cost compared to the TRPO algorithm. The per unit voltages at node 3 and 4 of the 4-bus test feeder are depicted in Fig. 3. It can be seen that the voltage solutions at node 3 of the MPC-based approach with forecasted load hit the upper bound a few times. This is common for optimization approaches as the optimal solutions are likely to be boundary points. By following the CPO algorithm, the voltage profiles at node 4 nearly stay in bounds all the time except for 5 minor violations. The CPO algorithm outperforms the TRPO algorithm by approximately satisfying the voltage constraints all the time.

The average and the maximum computation time of the MPC-based algorithms with different solvers and the policy gradient methods to determine the tap positions at each hour are provided in Table II. Without parallel computing

TABLE II
COMPUTATION TIME OF VOLT-VAR CONTROL ALGORITHMS

	Algorithm	Average Time (s)	Maximum Time (s)
4-bus test case	MPC (GUROBI)	10.43	90.28
	MPC (MOSEK)	346.80	3904.22
	TRPO/CPO	$< 10^{-3}$	$< 10^{-3}$
13-bus test case	MPC (GUROBI)	4.69	8.57
	MPC (MOSEK)	53.83	328.98
	TRPO/CPO	$< 10^{-3}$	$< 10^{-3}$

(MOSEK), the computation time of the MPC-based algorithm could exceed 1 hour in the worst case on an entry level DELL desktop. On the other hand, once trained the policy gradient methods have a much faster execution speed, which makes them suitable for online applications. Moreover, the MPC-based algorithms require accurate and complete topology model and parameters of the distribution network, which are not often available.

V. CONCLUSION

In this paper, the Volt-VAR control problem is modeled as a CMDP and solved with policy gradient methods for the first time. The constrained policy optimization algorithm is adopted to enable safe exploration for the controller. Both policy and state-value functions are approximated by neural networks. The structure of the policy network is tailored to achieve better scalability for the Volt-VAR control problem. The performance of the policy gradient methods and benchmarking algorithms are validated with the IEEE 4-bus and 13-bus test feeders. The results show that the constrained policy optimization algorithm can achieve near-optimal solutions with negligible voltage violations. Compared to the conventional optimization based approach, the proposed reinforcement learning algorithm is better suited for online VVC tasks where accurate and complete distribution network models are not available.

REFERENCES

[1] W. Wang, N. Yu, B. Foggo, J. Davis, and J. Li, "Phase identification in electric power distribution systems by clustering of smart meter data," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 259–265.

[2] B. Foggo and N. Yu, "A comprehensive evaluation of supervised machine learning for the phase identification problem," *World Acad. Sci. Eng. Technol. Int. J. Comput. Syst. Eng*, vol. 12, no. 6, 2018.

[3] W. Wang, N. Yu, and Z. Lu, "Advanced metering infrastructure data driven phase identification in smart grid," *GREEN 2017 Forward*, pp. 16–23, 2017.

[4] H. Ahmadi, J. R. Mart, and H. W. Dommel, "A framework for Volt-VAR optimization in distribution systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1473–1483, May 2015.

[5] P. Li, H. Ji, C. Wang, J. Zhao, G. Song, F. Ding, and J. Wu, "Coordinated control method of voltage and reactive power for active distribution networks based on soft open point," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1430–1442, Oct. 2017.

[6] M. H. K. Tushar and C. Assi, "Volt-VAR control through joint optimization of capacitor bank switching, renewable energy, and home appliances," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4077–4086, Sept. 2018.

[7] M. B. Liu, C. A. Canizares, and W. Huang, "Reactive power and voltage control in distribution systems with limited switching operations," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 889–899, May 2009.

[8] Y. Xu, Z. Y. Dong, R. Zhang, and D. J. Hill, "Multi-timescale coordinated voltage/VAR control of high renewable-penetrated distribution systems," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4398–4408, Nov. 2017.

[9] W. Zheng, W. Wu, B. Zhang, and Y. Wang, "Robust reactive power optimisation and voltage control method for active distribution networks via dual time-scale coordination," *IET Generation, Transmission Distribution*, vol. 11, no. 6, pp. 1461–1471, May 2017.

[10] Z. Wang, J. Wang, B. Chen, M. M. Begovic, and Y. He, "MPC-based voltage/VAR optimization for distribution circuits with distributed generators and exponential load models," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2412–2420, Sept. 2014.

[11] M. Falahi, K. Butler-Purry, and M. Ehsani, "Dynamic reactive power control of islanded microgrids," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3649–3657, Nov. 2013.

[12] J. G. Vlachogiannis and N. D. Hatziargyriou, "Reinforcement learning for reactive power control," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1317–1325, Aug. 2004.

[13] Y. Xu, W. Zhang, W. Liu, and F. Ferrese, "Multiagent-based reinforcement learning for optimal reactive power dispatch," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1742–1751, Nov. 2012.

[14] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *arXiv*, July 2018. [Online]. Available: https://arxiv.org/abs/1807.10997

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[16] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *AAAI*, Feb. 2016, pp. 2094–2100.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv*, Sept. 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[18] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *ICML*, vol. 37, 2015, pp. 1889–1897.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv*, July 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *ICML*, vol. 70, Aug. 2017, pp. 22–31.

[21] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[22] F. U. Nazir, B. C. Pal, and R. A. Jabr, "A two-stage chance constrained Volt/VAR control scheme for active distribution networks with nodal power uncertainties," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 314–325, Jan. 2019.

[23] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.

[24] W. H. Kersting, "Radial distribution test feeders," in *IEEE Power Engineering Society Winter Meeting*, vol. 2, Jan. 2001, pp. 908–912.

[25] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2213–2219, Nov. 2007.