Abstract—Inverter-based distributed energy resources provide the possibility for fast time-scale voltage control by quickly adjusting their reactive power. The power-electronic interfaces allow these resources to realize almost arbitrary control law, but designing these decentralized controllers is nontrivial. Reinforcement learning (RL) approaches are becoming increasingly popular to search for policy parameterized by neural networks. It is difficult, however, to enforce that the learned controllers are safe, in the sense that they may introduce instabilities into the system.

This paper proposes a safe learning approach for voltage control. We prove that the system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. The set of Lipschitz bound is optimized to enlarge the search space for neural network controllers. We explicitly engineer the structure of neural network controllers such that they satisfy the Lipschitz constraints by design. A decentralized RL framework is constructed to train local neural network controller at each bus in a model-free setting.

Index Terms—Safe RL, Voltage control, Stability, Decentralized Learning

I. INTRODUCTION

Distributed energy resources (DERs) such as rooftop solar PV, electric vehicles and battery storage are growing at an increasing pace. For example, solar capacity had almost 50% yearly growth in 2021 [1], which is by far the fastest among all renewable resources. Most of these growth are occurring in the distribution network, the low voltage network that connects customers to substations.

High variability of solar PV and sudden change in load due to electric vehicles and storage can lead to large voltage fluctuations. These fluctuations occur at timescales much faster than the conventional mechanical control devices such as tap-changing transformers. Instead, power electronic devices allow flexible and frequent control actions without degrading lifetime. Consequently, there have been growing interests to use the power electronic inverters on the DERs themselves to provide voltage control [2]–[6].

Since most distribution networks are not yet equipped with real-time communication infrastructure, voltage control strategies should use local measurements available at each bus. More specifically, controllers need to operate at an iterative fashion [7], [8], successively updating their control actions based on each measurement. Designing such decentralized controller is a nontrivial problem. Linear controllers can be far from optimal, even for quadratic costs. Therefore, neural networks have been used to parametrize the controllers to fully utilize the capabilities of the inverters [9]–[11].
This paper presents a decentralized safe learning method, which guarantees the learned neural network would maintain the stability of iterative voltage control dynamics. We prove that the system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. We optimize the set of Lipschitz bounds to enlarge the search space of controllers. On this basis, we propose to engineer the structure of neural network controllers such that they can satisfy the Lipschitz constraints by design. A decentralized RL framework is constructed to train neural network controller locally at each bus with policy gradient algorithm. The structure of the proposed approach is illustrated in Fig. 1.

Case studies show that the controllers learned with stability constraints outperform those with linear controllers and unconstrained neural network controllers. Interestingly, we also observe good learning convergence of the controllers in a model-free setting, even though they interact through the underlying distribution network. Code and data are available at https://github.com/Safe-RL-Power-Systems-Control/Voltage-Control.

The paper is organized as follows. Section II introduces the model and the optimal voltage control problem. Section III gives the main theorems governing the structure of a stabilizing controller and derives the optimal Lipschitz bounds. Section IV illustrates the decentralized safe RL framework for training a stabilizing neural network controller locally at each bus. Section V shows the simulation results and Section VI concludes the paper.

II. MODEL

A standard requirement for distribution network is that voltages should deviate no more than 5% from their rated values at all buses [16]. For example, if the rated voltage is 110 V, then the actual voltages should be in the interval from 104.5 V to 115.5 V. For simplicity, we normalize the units such that the reference value for voltage is 1 p.u. For a power network with N buses, let \( v \) be the voltage vector where \( v_i \) is the voltage at bus \( i \). Let \( p \) be active power and \( q \) be reactive power. The voltage of the system follows the LinDistFlow model:

\[
v = Rp + Xq + 1
\]  

where \( 1 \) is the all one’s vector and \( R \) and \( X \) are positive definite matrices describing the network [7]. The active power depends on external environment and is uncertain and variable. The reactive power comes from phase offsets and is controlled, subject to some actuation constraints [2].

This work focuses on optimizing the control of \( q \) through inverter-based resources. The aim of voltage regulation is to control \( q \) such that \( v \) is close to its reference value. Due to the lack of communication in many distribution systems, \( q \) needs to be successively updated based on the local voltage measurements. Denote \( u_i(v_i) \) as the control law for each bus \( i = 1, \cdots, N \), which is a mapping from the voltage to reactive power. Let \( v_i,t \) be the local voltage at the bus \( i \) at the \( t \)-th iteration step, and denote \( u_i = (u_i(v_{1,t}), \cdots, u_i(v_{N,t})) \). We update \( q \) and \( v \) iteratively as

\[
q_{t+1} = q_t - u_t, \quad (2a)
\]

\[
v_{t+1} = Rp + X(q_t - u_t) + 1, \quad (2b)
\]

A. Optimal voltage control

Our objective is to optimize the \( u_i \) to minimize cost in \( v \) and \( q \) defined as \( C(u) \), subject to the iterative update rule and the saturation limit on \( u_i \). The optimization problem is

\[
\min_u C(u) \quad (3a)
\]

\[
s.t. \quad q_{t+1} = q_t - u_t, \quad (3b)
\]

\[
v_{t+1} = Rp + X(q_t - u_t) + 1, \quad (3c)
\]

\[
u \leq u_i \leq \bar{u}, \quad (3d)
\]

\[
u_i \text{ is stabilizing} \quad (3e)
\]

where constraints (3b)–(3e) hold for the iteration step \( t \) from 0 to \( T \). The cost typically trades off between driving voltage to the reference value and the control effort. The deviation of voltage can typically be quantified as two-norm, one-norm or infinity-norm of the sequence of \( v_t \) [4], [17], [18]. The control effort depends on the type of resources and can be both quadratic [19], [20] and non-quadratic ones [17], [18], [21]. For example, control effort from batteries is commonly defined as one-norm of actions since charging/discharging power affects cycle-depth linearly [18], [21]. The proposed safe RL approach works for all types of cost functions listed above. The lower and upper bound for the control action at bus \( i \) are \( u_i \) and \( \bar{u}_i \), respectively.

The controllers \( u \) are conventionally designed to be linear (up to a thresholding by \( |3d| \)), which does not leverage the capability of inverter-based resources in implementing almost arbitrary control laws [22]. To design a flexible non-linear control law for inverter-based resources, we parameterize each controller \( u_i(v_i) \) as a neural network with weight \( \theta_i \), sometimes written as \( u_{\theta_i}(v_i) \).

However, there remain two challenges. First, due to the lack of communication, neural network controller needs to be trained decentralized in each bus with local observations of voltages. Second, even if the controller is optimized and implemented locally, they need to stabilize the system defined by (3b) and (3e). In the next sections, we show how to design the local neural network controllers that guarantee the stability of this system, and how to train the controllers through decentralized reinforcement learning.

In this paper, we assume that the topology and parameter information of the distribution system is available. That is, we know \( X \), but there is no real-time communication between the buses. This assumption comes from the fact that \( X \) (and \( R \)) can be estimated using smart meter data collected over a period of time [23]–[25], where the communication rate can be quite slow (e.g., once per day [26]). Therefore, design of the controllers \( u \) can depend on \( X \), but the dependence must be determined offline.
III. STABILIZING CONTROLLER

In this section, we derive the properties of a stabilizing local controller from the Lyapunov stability theory and standard nonlinear system theory. We engineer the structure of neural network to satisfy these structure properties and thus guarantee the stability of the system.

A. Reduced-order system

We can simplify the dynamics in (2) by shifting the origin of the system. Denote \( \Delta v_t \) as the difference between voltage and its reference value \( \mathbb{1} \) at time \( t \). Then we have

\[
\Delta v_t = v_t - \mathbb{1} = R p + X q_t \\
= R p + X(q_{t-1} - u_t) \\
= (R p + X q_{t-1}) - X u_t \\
= \Delta v_{t-1} - X u_t
\]



B. Structure property of a stabilizing controller

The structure property of a stabilizing controller is obtained from Theorem 1. It shows that as long as each controller \( u_i \) satisfies the Lipschitz constraints, the system is guaranteed to be locally exponentially stable.

**Theorem 1.** Suppose a vector \( k = (k_1, \ldots, k_N) \) satisfies \( 0 \prec \text{diag}(k) \prec 2X^{-1} \). Then if the derivative of controller satisfies \( u_i(0) = 0 \) and \( 0 \prec \frac{du_i(t)}{dt} \prec k_i \) for all \( i = 1, \ldots, N \), the equilibrium point \( v = 0 \) of the dynamic system in (4) is locally exponentially stable.

**Proof.** The Jacobian of the state transition dynamics in (4) is

\[
J(v) = I - X \nabla v u
\]

where \( \nabla v u \) is the gradient of control action \( u \) with respect to \( v \) defined as

\[
\nabla v u = \begin{bmatrix}
\frac{du_1(v_1)}{dv_1} \\
\vdots \\
\frac{du_N(v_N)}{dv_N}
\end{bmatrix}
\]

To guarantee an exponentially stable system around the equilibrium, the goal is to show that all the eigenvalues of \( J(v) \) have magnitude less than 1. To this end, we first show that the eigenvalues of \( J(v) \) are the same as that of \( I - (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2} \).

Let \( (\lambda, w) \) be an eigenpair for \( I - X (\nabla v u) \) \( w = \lambda w \). Then, we have

\[
(I - (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2}) (\nabla v u) \frac{1}{2} w \\
= (\nabla v u) \frac{1}{2} w - (\nabla v u) \frac{1}{2} X (\nabla v u) w \\
= (\nabla v u) \frac{1}{2} (I - X (\nabla v u)) w \\
= \lambda (\nabla v u) \frac{1}{2} w
\]

Therefore, \( (\lambda, (\nabla v u) \frac{1}{2} w) \) is an eigenpair for \( I - (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2} \). To prove that the eigenvalue of \( J(v) \) to be strictly smaller than 1, it suffices to show that

\[
-1 < I - (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2} \prec 1
\]

By picking the controller \( u \) such that \( 0 \prec \frac{du_i(t)}{dt} \prec k_i \) for all \( i = 1, \ldots, N \) and \( 0 \prec \text{diag}(k) \prec 2X^{-1} \), we have \( 0 \prec \nabla v u \prec 2X^{-1} \) and thus \( (\nabla v u) \frac{1}{2} \prec \frac{1}{2} X \). Since \( \nabla v u \succ 0 \) is diagonal, we then have \( (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2} \prec 2I \) and thus

\[
-1 < I - (\nabla v u) \frac{1}{2} X (\nabla v u) \frac{1}{2} \prec 1
\]

The right side inequality holds because \( \lambda \succ 0 \).

C. Optimizing search space for neural network controllers

Note that all the feasible stabilizing \( u \) are in a convex set described by \( S = \{ \nabla v u \prec \nabla v u \prec 2X^{-1} \} \). Since there is no communication between buses during training, each \( \frac{du_i(t)}{dt} \) needs to be bounded by a separate \( k_i \) for bus \( i = 1, \ldots, N \). Therefore, the search space for neural network controllers is constrained by the selection of \( k \). A uniform bound \( k_i = \frac{1}{\lambda_{\text{max}}(X)} \) can be found in literatures [7], but it might be too conservative since \( S \) may be much larger than the region described by \( D = \{ \nabla v u \prec \nabla v u \prec \frac{2}{\lambda_{\text{max}}(X)} \} \).

Here we show an illustration on a three-bus system (with the first bus as the feeder) where \( X = \begin{bmatrix} 2 & 0.20 & -0.16 \\ 0.16 & 0.97 & 0.20 \end{bmatrix} \). For different Lipschitz bounds on controllers, feasible regions for \( \nabla v u \) are shown in Fig. 2.
The volume is not a convex function in \( k \), but we can apply a simple log trick and solve the following optimization problem:

\[
\max_k \sum_{i=1}^{N} \frac{v_i}{w_i \log(k_i)} \\
\text{s.t. } 0 < k_i < 2X^{-1}
\]  

(8a)

(8b)

where \( w_1, \cdots, w_N \) are the coefficients to represent the relative importance of buses. For example, if bus \( j \) has none or very limited capacity for voltage regulation, \( w_j \) is set to be small. If bus \( j \) is the source node of a branch, \( w_j \) can be set to be larger to speed up the convergence of voltage at the source node and thus help the convergence of following branches. In reality, this set of coefficients can be adjusted according to the optimization results. For the controller \( u_i \) that the derivative \( \frac{du_i(v_i)}{dv_i} \) is far from being bounded by \( k_i \), the coefficient \( w_i \) can be adjusted to be smaller to give larger searching freedom to other buses.

D. Design of stabilizing neural network controllers

From Theorems [1] the structural property of locally exponentially stabilizing controllers is derived in Corollary [1]. We aim to engineer the neural networks to satisfy these structural property in Corollary [1] by design.

Corollary 1. The condition for a locally exponentially stabilizing controller in Theorem [2] is equivalent to:

1) \( u_\theta(v_i) \) has the same sign as \( v_i \)
2) \( \frac{du_\theta(v_i)}{dv_i} \) is monotonically increasing
3) \( \frac{du_\theta(v_i)}{dv_i} < k_i. \)

The first two requirements are equivalent to designing a monotonically increasing function through the origin. This is constructed by decomposing the function into a positive and a negative part as \( f_i(v_i) = f_i^+(v_i) + f_i^-(v_i) \), where \( f_i^+(v_i) \) is monotonically increasing for \( v_i > 0 \) and zero when \( v_i \leq 0; f_i^-(v_i) \) is monotonically increasing for \( v_i < 0 \) and zero when \( v_i \geq 0 \). To this end, we formulate the controller with a stacked-ReLU structure shown in Fig. 3 which is developed in [27]. This design is a piecewise linear function where the slope of each piece is equal to the summation of weights in activated neurons. Then the requirement 3) can be satisfied by directly thresholding the slope. The neural network controller is constructed as

\[
u_i(v_i) = s_i \text{ReLU}(v_i + b_i) + z_i \text{ReLU}(-v_i + d_i)
\]  

(9a)

where \( 0 < \sum_{j=1}^{i} s_i^j < k_i, \forall l = 1, 2, \cdots, m \)

(9b)

\[-k_i < \sum_{j=1}^{i} z_i^j < 0, \forall l = 1, 2, \cdots, m \]

(9c)

\[b_i^l = 0, b_i^l \leq b_i^{l-1}, \forall l = 2, 3, \cdots, m \]

(9d)

\[d_i^l = 0, d_i^l \leq d_i^{l-1}, \forall l = 2, 3, \cdots, m \]

(9e)

where \( m \) is the number of hidden units and \( 1 \in \mathbb{R}^{m} \) is the all 1’s column vector. Variables \( s_i = [s_i^1, s_i^2, \cdots, s_i^m] \) and \( z_i = [z_i^1, z_i^2, \cdots, z_i^m] \) are the weight vector of bus \( i \); \( b_i = [b_i^1, b_i^2, \cdots, b_i^m]^T \) and \( d_i = [d_i^1, d_i^2, \cdots, d_i^m]^T \) are the corresponding bias vector. The variables to be trained are weights \( \theta = \{s, b, z, d\} \) in (9). The saturation limits can be satisfied by hard thresholding the output of the neural network.

IV. DECENTRALIZED SAFE REINFORCEMENT LEARNING

In this section, we construct a decentralized reinforcement learning framework to optimize neural network controller in each bus locally with observation of trajectories.

Most reinforcement learning algorithms, including Q-learning, actor-critic and DDPG, rely on learning a value function (Q-function) satisfying the Bellman equations. Q-function assumes an infinite-horizon formulation where the states follow a stationary probability distribution, which is generally not true for the voltage control problem in this paper. Instead, REINFORCE policy gradient algorithm adopts the log probability trick and avoids learning the value function [12]. Therefore, we use REINFORCE policy gradient algorithm to obtain sampled gradient for updating the weights of neural network controllers.

Notably, there are natural noises in the system coming from the changes in active power \( p \), which enable us to implement REINFORCE policy gradient with equivalent stochastic policy. Specifically, we assume that the distribution of noise on the system can be estimated. By incorporating noise term into control action, each action \( u_{i,t} \) comes from an equivalent stochastic policy with probability distribution \( \pi_\theta(u_{i,t}|v_{i,t}) \).

The gradient for updating weights of neural network controller at bus \( i \) is obtained by

\[
\nabla J(\theta) = \mathbb{E} \left[ \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|v_{i,t}) \sum_{l=1}^{T} C_i(u_{i,t}) \right]
\]  

(10)
The pseudo-code for the decentralized RL framework is given in Algorithm 1. Each bus $i$ has its local RL agent for training in a batch-updating style. Let $H$ be the number of batches. At each episode, each agent collects trajectory $\{v_{i,1}^h, u_{i,1}^h, \cdots, v_{i,T}^h, u_{i,T}^h\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(v_{i,t}^h)$ for $h = 1, \cdots, H$. Adam algorithm is adopted to update weights of neural network controllers with gradient computed through batch average of $\{c_i^h\}$.

**Algorithm 1 Decentralized Reinforcement Learning algorithm with Policy Gradient**

**Require:** Learning rate $\alpha$, batch size $H$, trajectory length $T$, number of episodes $E$

**Input:** Initial weights $\theta$ for control network

1: for episode $= 1$ to $E$ do
2: for agent $i = 1$ to $N$ do
3: Collect trajectories $\{v_{i,1}^h, u_{i,1}^h, \cdots, v_{i,T}^h, u_{i,T}^h\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(v_{i,t}^h)$ for $h = 1, \cdots, H$
4: Compute the gradient $\nabla J_i(\theta_i) = \frac{1}{H} \sum_{h=1}^H \nabla \log p_\theta (u_{i,t}^h|v_{i,t}^h)c_i^h$
5: Update weights in the neural network by passing $J_i(\theta_i)$ to Adam optimizer: $\theta_i \leftarrow \theta_i - \alpha \nabla J_i(\theta_i)$
6: end for
7: end for

V. NUMERICAL RESULTS

We verify the performance of the proposed safe RL approach on IEEE 33-bus test feeders [28]. We first show that unconstrained neural network controllers learned by RL might lead to an unstable system, while the controllers trained by safe RL approach are guaranteed to stabilize the system. Then, we show that the proposed decentralized RL framework can learn flexible non-linear controllers for different buses that outperform conventional linear control law.

A. Simulation setup

The cost function that each controller collectively optimizes is $C(u) = \sum_{i=1}^T (||v_i||_1 + \gamma ||u_i||_1)$, where $\gamma$ acts as a trade-off parameter and is set to be 0.01. The base unit for power and voltage is 100kVA and 12.66kV, respectively. The bound on action $u$ is generated to be uniformly distributed in $[0.01, 0.05]$. We use TensorFlow 2.0 framework to build the reinforcement learning environment. The episode number, batch size and the number of neurons are 500, 500, 20, respectively. Parameters of neural network controllers are updated using Adam with learning rate initialized to be 0.003 and decayed every 100 steps with a base of 0.6. We compare the performance of neural network controller designed with and without the safe RL approach, as well as conventional linear controller. All of them are trained using the decentralized RL framework.

B. Necessity of the stabilizing requirement

Intuitively speaking, if a controller achieves a low loss function after training converges, one might hope that it naturally leads to a stabilizing controller since the trajectory does not blow up to a high cost. Fig. 4 shows the dynamics of voltage deviation under the neural network controllers trained with and without the safe RL approach. The one without safe RL approach is unstable and leads to very large state oscillations (Fig. 4(b)). In contrast, the controller with safe RL approach shows good performance in Fig. 4(a). Therefore, explicitly constraining the controller structure is necessary.

C. Performance comparison

To investigate the convergence of the safe RL approach, Fig. 6(a) shows the normalized cost on the test set along episodes for training of neural network controllers and linear controllers. All the losses converge, with the proposed neural network controllers achieving the lowest cost. Fig. 6(b) shows the cost on selected buses along the episodes of training. It is interesting to observe that training the controllers in a decentralized fashion did not impact convergence or performance. Namely, during training, $u_i$ is updated based only on the trajectory of $v_i$, even though the control action impacts the voltage at all neighboring buses.

The control law for neural network controller learned with safe RL, without safe RL approach and linear controller with optimal linear coefficient are shown in Fig. 5. The neural network controllers learn flexible non-linear control laws for different buses, with the slope of controller obtained by safe RL approach bounded by Lipschitz constraints. Fig. 7 illustrates the dynamics of voltage deviation...
The uniform Lipschitz bound 2 \lambda_{max}(X) was obtained in (8) and is used to bound the cost by approximately 5.26% compared to safe RL with the optimal Lipschitz bound. The neural network controller trained by safe RL approach reduces the average cost by approximately 5.26%, 18.18%, respectively.

In the test set with random initial states, the distribution of cost in selected buses is shown in Fig. 8. The average cost during training of neural network controller and linear controller. Neural network controller designed with safe RL approach achieves lower cost than conventional linear controller. All learning trajectories converge well in the decentralized model-free setting, even though they interact through the underlying distribution network.

The exponential stability of the system is guaranteed by controllers constrained by Lipschitz bounds, which are optimized to enlarge the search space. The neural network controllers are parameterized by a staked ReLU neural network to satisfy stabilizing constraints implicitly. Each bus updates weights locally with the decentralized RL framework. Case studies show that RL without stability constraints can lead to unstable controllers, while the proposed safe learning approach will lead to a stabilizing controller. The neural network controllers outperform conventional linear controllers by speeding up the convergence of voltages to reference values with relatively low control effort. Rigorously analysing and comparing decentralized and centralized training is an important future direction for us.

VI. CONCLUSIONS

This paper proposes a safe RL approach for optimal voltage control. The exponential stability of the system is guaranteed by controllers constrained by Lipschitz bounds, which are optimized to enlarge the search space. The neural network controllers are parameterized by a staked ReLU neural network to satisfy stabilizing constraints implicitly. Each bus updates weights locally with the decentralized RL framework. Case studies show that RL without stability constraints can lead to unstable controllers, while the proposed safe learning approach will lead to a stabilizing controller. The neural network controllers outperform conventional linear controllers by speeding up the convergence of voltages to reference values with relatively low control effort. Rigorously analysing and comparing decentralized and centralized training is an important future direction for us.

REFERENCES

[1] M. Davis, “Solar market insight report,” Wood Mackenzie Power and Renewables, 2021.
[2] K. Turitsyn, P. Sule, S. Backhaus, and M. Cherkov, “Options for control of reactive power by distributed photovoltaic generators,” Proceedings of the IEEE, vol. 99, no. 6, pp. 1063–1073, 2011.
[3] H.-G. Yeh, D. F. Gayme, and S. H. Low, “Adaptive var control for distribution circuits with photovoltaic generators,” IEEE Transactions on Power Systems, vol. 27, no. 3, pp. 1656–1663, 2012.
[4] B. Zhang, A. Y. Lam, A. D. Domínguez-García, and D. Tse, “An optimal and distributed method for voltage regulation in power distribution systems,” IEEE Transactions on Power Systems, vol. 30, no. 4, pp. 1714–1726, 2014.
[5] N. Li, G. Qu, and M. Dahleh, “Real-time decentralized voltage control in distribution networks,” in 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2014, pp. 582–588.
[6] S. Bolognani and S. Zampieri, “A distributed control strategy for reactive power compensation in smart microgrids,” IEEE Transactions on Automatic Control, vol. 58, no. 11, pp. 2818–2833, 2013.
[7] H. Zhu and H. J. Liu, “Fast local voltage control under limited reactive power: Optimality and stability analysis,” IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3794–3803, 2015.
[8] G. Qu and N. Li, “Optimal distributed feedback voltage control under limited reactive power,” IEEE Transactions on Power Systems, vol. 35, no. 1, pp. 315–331, 2019.
[9] Y.-Y. Hsu and F.-C. Lu, “A combined artificial neural network-fuzzy dynamic programming approach to reactive power/voltage control in a distribution substation,” IEEE transactions on Power Systems, vol. 13, no. 4, pp. 1265–1271, 1998.

[10] S. Toma, T. Senjyu, Y. Miyazato, A. Yona, K. Tanaka, and C.-H. Kim, “Decentralized voltage control in distribution system using neural network,” in 2008 IEEE 2nd International Power and Energy Conference. IEEE, 2008, pp. 1557–1562.

[11] X. Shen, H. Wang, J. Li, Q. Su, and L. Gao, “Distributed secondary voltage control of islanded microgrids based on rbf-neural-network sliding-mode technique,” IEEE Access, vol. 7, pp. 65 616–65 623, 2019.

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

[13] N. E. M. Association et al., American National Standard for Electric Power Systems and Equipment-Voltage Ratings (60 Hertz). National Electrical Manufacturers Association, 1996.

[14] Y. Gao, W. Wang, and N. Yu, “Consensus multi-agent reinforcement learning for volt-var control in power distribution networks,” IEEE Transactions on Smart Grid, 2021.