



Decentralized safe reinforcement learning for inverter-based voltage control[☆]

Wenqi Cui^{*}, Jiayi Li, Baosen Zhang

Electrical and Computer Engineering, University of Washington, Seattle, WA, United States of America

ARTICLE INFO

Keywords:

Safe RL
Voltage control
Stability
Decentralized learning

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.

1. 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 parametrise the controllers to fully utilize the capabilities of the inverters [9–11].

Reinforcement learning algorithms are proposed to train the neural network controllers with trajectory measurements. This provides the advantages of updating neural networks in a model-free setting, i.e., eliminating the requirement on system parameters and communications [12]. Many algorithms, such as deep Q learning [13], actor-critic [14], DDPG [15], have been applied to the control of tap-changing transformers or inverter based resources. Since the control actions are taken in an iterative fashion, it creates a dynamical system, whose transition depends on the actions and the underlying physical distribution network. The key constraint on the controllers is that they do not destabilize the system. However, most works neglect the stability requirement and currently this stability condition is checked through simulations [13,14]. Considering that voltage control is implemented locally without real-time communication, formal guarantees on stability are required in practice.

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 authors are partially supported by NSF, United States of America grants ECCS-1942326, CNS-1931718 and the Washington Clean Energy Institute, United States of America.

^{*} Corresponding author.

E-mail addresses: wengqicui@uw.edu (W. Cui), ljiy9712@uw.edu (J. Li), zhangbao@uw.edu (B. Zhang).

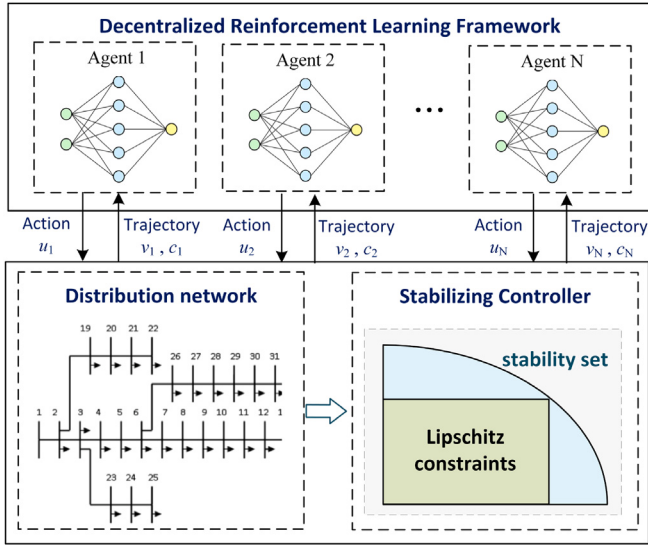


Fig. 1. Proposed decentralized safe RL approach for optimal voltage control. We prove that the system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. The neural network controllers are engineered to satisfy these Lipschitz constraints by design, and is updated from local trajectories with a decentralized RL framework.

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 2 introduces the model and the optimal voltage control problem. Section 3 gives the main theorems governing the structure of a stabilizing controller and derives the optimal Lipschitz bounds. Section 4 illustrates the decentralized safe RL framework for training a stabilizing neural network controller locally at each bus. Section 5 shows the simulation results and Section 6 concludes the paper.

2. 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 $\mathbf{v}$ be the voltage vector where v_i is the voltage at bus i . Let $\mathbf{p}$ be active power and $\mathbf{q}$ be reactive power. The voltage of the system follows the LinDistFlow model:

$$\mathbf{v} = \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + \mathbb{1} \quad (1)$$

where $\mathbb{1}$ is the all one's vector and $\mathbf{R}$ and $\mathbf{X}$ 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 controllable, subject to some actuation constraints [2].

This work focuses on optimizing the control of $\mathbf{q}$ through inverter-based resources. The aim of voltage regulation is to control $\mathbf{q}$ such that $\mathbf{v}$ is close to its reference value. Due to the lack of communication in many distribution systems, $\mathbf{q}$ needs to be successively updated based on the local voltage measurements. Denote $u_i(v_i)$ as the control law for each bus $i = 1, \dots, N$, 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 $\mathbf{u}_t = (u_1(v_{1,t}), \dots, u_N(v_{N,t}))$. We update $\mathbf{q}$ and $\mathbf{v}$ iteratively as

$$\mathbf{q}_{t+1} = \mathbf{q}_t - \mathbf{u}_t, \quad (2a)$$

$$\mathbf{v}_{t+1} = \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}_t - \mathbf{u}_t) + \mathbb{1}, \quad (2b)$$

2.1. Optimal voltage control

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

$$\min_{\mathbf{u}} C(\mathbf{u}) \quad (3a)$$

$$\text{s.t. } \mathbf{q}_{t+1} = \mathbf{q}_t - \mathbf{u}_t \quad (3b)$$

$$\mathbf{v}_{t+1} = \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}_t - \mathbf{u}_t) + \mathbb{1} \quad (3c)$$

$$\underline{\mathbf{u}}_t \leq \mathbf{u}_t \leq \bar{\mathbf{u}}_t \quad (3d)$$

$$\mathbf{u}_t \text{ is stabilizing} \quad (3e)$$

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 $\mathbf{v}_t$ [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 $\underline{u}_{t,i}$ and $\bar{u}_{t,i}$, respectively. The subscript t signifies that these bounds can be time-varying as active power changes.

The controllers $\mathbf{u}$ 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 parametrise each controller $u_i(v_i)$ as a neural network with weight θ_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 decentralizedly in each bus with local observations of voltages. Second, even if the controller is optimized and implemented locally, they need to be “safe” in the sense that the controller stabilizes the entire system, as defined by (3b) and (3c). 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 $\mathbf{X}$, but there is no real-time communication between the buses. This assumption comes from the fact that $\mathbf{X}$ (and $\mathbf{R}$) 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 can depend on $\mathbf{X}$, but the dependence must be determined offline. The system parameters are not required for real-time training and implementation.

3. 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.

3.1. Reduced-order system

We can simplify the dynamics in (2) by shifting the origin of the system. Denote $\hat{v}_t$ as the difference between voltage and its reference value $\mathbb{1}$ at time t . We assume that the active power p remains constant during one iteration period. Then we have

$$\begin{aligned}\hat{v}_t &= v_t - \mathbb{1} \\ &= R p + X q_t \\ &= R p + X(q_{t-1} - u_{t-1}) \\ &= (R p + X q_{t-1}) - X u_{t-1} \\ &= \hat{v}_{t-1} - X u_{t-1}\end{aligned}\quad (4)$$

Therefore, instead of the iteration with both q_t and v_t being the state variables, it suffices to study the dynamics in $\hat{v}_t$.

3.2. 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 $k = (k_1, \dots, k_N)$ satisfies $0 < \text{diag}(k) < 2X^{-1}$. Then if the derivative of controller satisfies $u_i(0) = 0$ and $0 < \frac{du_i(\hat{v}_i)}{d\hat{v}_i} < k_i$ for all $i = 1, \dots, N$, the equilibrium point $\hat{v} = 0$ of the dynamic system in (4) is locally exponentially stable.

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

$$J(\hat{v}) = I - X \nabla_{\hat{v}} u \quad (5)$$

where $\nabla_{\hat{v}} u$ is the gradient of control action u with respect to $\hat{v}$ defined as

$$\nabla_{\hat{v}} u = \begin{bmatrix} \frac{du_1(\hat{v}_1)}{d\hat{v}_1} & & \\ & \ddots & \\ & & \frac{du_N(\hat{v}_N)}{d\hat{v}_N} \end{bmatrix}. \quad (6)$$

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

Let (λ, w) be an eigenpair for $I - X(\nabla_{\hat{v}} u)$. That is, $(I - X(\nabla_{\hat{v}} u))w = \lambda w$. Then, we have

$$\begin{aligned}& (I - (\nabla_{\hat{v}} u)^{\frac{1}{2}} X (\nabla_{\hat{v}} u)^{\frac{1}{2}}) (\nabla_{\hat{v}} u)^{\frac{1}{2}} w \\ &= (\nabla_{\hat{v}} u)^{\frac{1}{2}} w - (\nabla_{\hat{v}} u)^{\frac{1}{2}} X (\nabla_{\hat{v}} u)^{\frac{1}{2}} w \\ &= (\nabla_{\hat{v}} u)^{\frac{1}{2}} (I - X(\nabla_{\hat{v}} u)) w \\ &= \lambda (\nabla_{\hat{v}} u)^{\frac{1}{2}} w\end{aligned}\quad (7)$$

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

By picking the controller u such that $0 < \frac{du_i(\hat{v}_i)}{d\hat{v}_i} < k_i$ for all $i = 1, \dots, N$ and $0 < \text{diag}(k) < 2X^{-1}$, we have $0 < \nabla_{\hat{v}} u < 2X^{-1}$ and thus $(\nabla_{\hat{v}} u)^{-1} > \frac{1}{2}X$. Since $\nabla_{\hat{v}} u > 0$ is diagonal, we then have $(\nabla_{\hat{v}} u)^{\frac{1}{2}} X (\nabla_{\hat{v}} u)^{\frac{1}{2}} < 2I$ and thus $-I < I - (\nabla_{\hat{v}} u)^{\frac{1}{2}} X (\nabla_{\hat{v}} u)^{\frac{1}{2}} < I$. The right side inequality holds because $X > 0$. $\square$

3.3. Optimizing search space for neural network controllers

Note that all the feasible stabilizing u are in a convex set described by $S = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < 2X^{-1}\}$. Since there is no communication between buses during training, each $\frac{du_i(\hat{v}_i)}{d\hat{v}_i}$ needs to be bounded by a

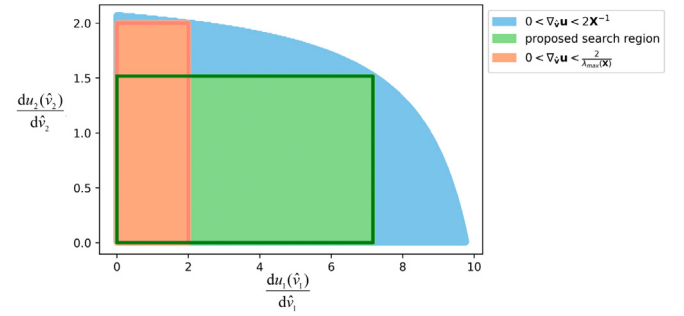


Fig. 2. Feasible search space comparisons for controllers. The blue area is the set of all feasible u in $S = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < 2X^{-1}\}$. The orange area is the search space with uniform Lipschitz bounds defined as $D = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < \frac{2}{\lambda_{\max}(X)} I\}$, which is the largest square within blue region but is only a very small subset of S . With each controller being trained independently, it is natural to consider some larger non-uniform search space such as the green area.

separate k_i for bus $i = 1, \dots, N$. Therefore, the search space for neural network controllers is constrained by the selection of k . A uniform bound $k_i = \frac{2}{\lambda_{\max}(X)}$ can be found in literatures [7], but it might be too conservative since S may be much larger than the region described by $D = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < \frac{2}{\lambda_{\max}(X)} I\}$.

Here we show an illustration on a three-bus system (with the first bus as the feeder) where $X = \begin{bmatrix} 0.20 & -0.16 \\ -0.16 & 0.97 \end{bmatrix}$. For different Lipschitz bounds on controllers, feasible regions for $\nabla_{\hat{v}} u$ are shown in Fig. 2.

The blue area demonstrates the space of controllers constrained by $S = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < 2X^{-1}\}$. The orange area is the space defined by $D = \{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < \frac{2}{\lambda_{\max}(X)} I\}$, which is the largest square within blue region but is only a very small subset of blue area for S . Note that the axes are scaled so the orange one does not look like a square. With each controller being trained independently, it is natural to consider some larger non-uniform search space such as the green area by choosing different k . We may choose a k^* such that the search space $\{\nabla_{\hat{v}} u | 0 < \nabla_{\hat{v}} u < k^*\}$ is the largest rectangular volume inside blue space, denoted as $\prod_{i=1}^N k_i$.

The volume is not a convex function in k , but we can apply a simple log trick and solve the following optimization problem:

$$\max_k \sum_{i=1}^N \beta_i \log(k_i) \quad (8a)$$

$$\text{s.t. } 0 < \begin{bmatrix} k_1 & & \\ & \ddots & \\ & & k_N \end{bmatrix} < 2X^{-1} \quad (8b)$$

where $\beta_1, \dots, \beta_N$ are the coefficients to represent the relative importance of buses. For example, if bus j has none or very limited capacity for voltage regulation, β_j is set to be small. If bus j is the source node of a branch, β_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 practice, this set of coefficients can be adjusted according to the solutions of the optimization problem and the training of controllers. For the controller u_i whose derivative $\frac{du_i(\hat{v}_i)}{d\hat{v}_i}$ is far from being bounded by k_i , its coefficient β_i can be adjusted to be smaller to encourage larger control action at the other buses. We envision that the system operator has the capability to communicate with each bus at a slower timescale (e.g., once a day) and collect the above information. Accordingly, the operator adjusts coefficients β_i , solves (8) and issues the bounds k_i to each bus at this the slower timescale.

3.4. Design of stabilizing neural network controllers

From Theorem 1, the structural property of locally exponentially stabilizing controllers is derived in Corollary 1. We aim to engineer the

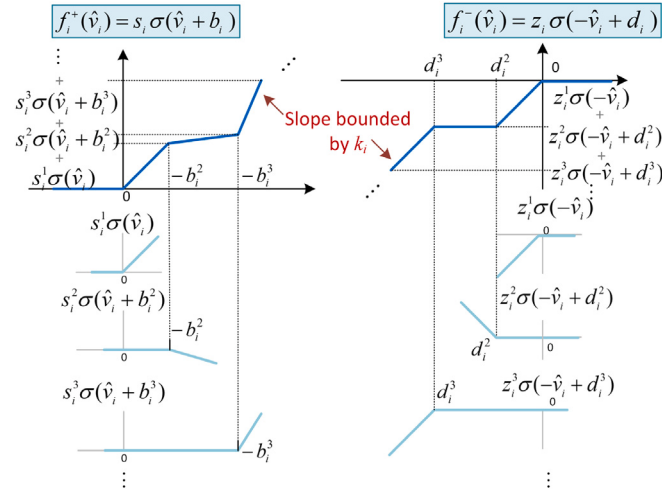


Fig. 3. Stacked ReLU neural network to formulate a controller satisfying the stabilizing constraint.

neural networks to satisfy these structural property in Corollary 1 by design.

Corollary 1. The condition for a locally exponentially stabilizing controller in Theorems 1 is equivalent to:

- (1) $u_{\theta_i}(\hat{v}_i)$ has the same sign as $\hat{v}_i$
- (2) $u_{\theta_i}(\hat{v}_i)$ is monotonically increasing
- (3) $\frac{du_{\theta_i}(\hat{v}_i)}{d\hat{v}_i} < k_i$.

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 $f_i(\hat{v}_i) = f_i^+(\hat{v}_i) + f_i^-(\hat{v}_i)$, where $f_i^+(\hat{v}_i)$ is monotonically increasing for $\hat{v}_i > 0$ and zero when $\hat{v}_i \leq 0$; $f_i^-(\hat{v}_i)$ is monotonically increasing for $\hat{v}_i < 0$ and zero when $\hat{v}_i \geq 0$. 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 (9)

$$u_i(\hat{v}_i) = s_i \text{ReLU}(\mathbb{1}\hat{v}_i + b_i) + z_i \text{ReLU}(-\mathbb{1}\hat{v}_i + d_i) \quad (9a)$$

$$\text{where } 0 < \sum_{j=1}^l s_i^j < k_i, \quad \forall l = 1, 2, \dots, m \quad (9b)$$

$$-k_i < \sum_{j=1}^l z_i^j < 0, \quad \forall l = 1, 2, \dots, m \quad (9c)$$

$$b_i^1 = 0, b_i^l \leq b_i^{(l-1)}, \quad \forall l = 2, 3, \dots, m \quad (9d)$$

$$d_i^1 = 0, d_i^l \leq d_i^{(l-1)}, \quad \forall l = 2, 3, \dots, m \quad (9e)$$

where m is the number of neurons and $\mathbb{1} \in \mathbb{R}^m$ is the all 1's column vector. Variables $s_i = [s_i^1 \ s_i^2 \ \dots \ s_i^m]$ and $z_i = [z_i^1 \ z_i^2 \ \dots \ z_i^m]$ are the weight vector of bus i ; $b_i = [b_i^1 \ b_i^2 \ \dots \ b_i^m]^T$ and $d_i = [d_i^1 \ d_i^2 \ \dots \ d_i^m]^T$ 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. Note that (9) is a single-layer neuron network and m determines the number of pieces for the piece-wise linear function. We tune m according to the testing performances of the controllers and we find that $m = 20$ is generally enough in most settings.

After one iteration, the Lipschitz constraints guarantee that $\|\hat{v}_i\| \leq \|\hat{v}_{i-1}\|$ and thus the magnitude of voltage deviation will not be worse

after a control action. If there is an abrupt change in active power, the voltage will experience an abrupt change and its magnitude may be larger than before. Therefore, the “safety” in our proposed method is in the sense of stability, where the voltage deviations would go to zero if active power changes relatively slowly. In addition, this guarantees the neural network controller does not degrade the voltage performance compared to an uncontrolled system.

4. 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. By having the constraints in (9), the neural network controller is guaranteed to stabilize the system.

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 $\pi_{\theta}(u_{i,t}|\hat{v}_{i,t})$. The gradient for updating weights of neural network controller at bus i is obtained by [12]

$$\nabla J(\theta) = \mathbb{E}[\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(u_{i,t}|\hat{v}_{i,t}) \sum_{t=1}^T C_i(u_{i,t})] \quad (10)$$

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 $\{\hat{v}_{i,1}^h, u_{i,1}^h, \dots, \hat{v}_{i,T}^h, u_{i,T}^h\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(u_{i,t}^h)$ for $h = 1, \dots, H$. Adam algorithm is adopted to update weights of neural network controllers with gradient computed through batch average of (10). We would also like to emphasize that any model-free RL algorithms can be readily utilized in our framework by replacing the REINFORCE algorithm. We use the standard REINFORCE algorithm in this paper to illustrate our contributions: the design of stabilizing neural network controllers and training them in a decentralized manner.

Algorithm 1 Decentralized Reinforcement Learning algorithm with Policy Gradient

Require: Learning rate α , batch size H , trajectory length T , number of episodes E

Input: Initial weights θ for control network

1: **for** episode = 1 to E **do**

2: **for** agent $i = 1$ to N **do**

3: Collect trajectories $\{\hat{v}_{i,1}^h, u_{i,1}^h, \dots, \hat{v}_{i,T}^h, u_{i,T}^h\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(u_{i,t}^h)$ for $h = 1, \dots, H$

4: Compute the gradient $\nabla J_i(\theta_i) = \frac{1}{H} \sum_{h=1}^H \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(u_{i,t}^h|\hat{v}_{i,t}^h) c_i^h$

5: Update weights in the neural network by passing $J_i(\theta_i)$ to Adam optimizer: $\theta_i \leftarrow \theta_i - \alpha_i \nabla J_i(\theta_i)$

6: **end for**

7: **end for**

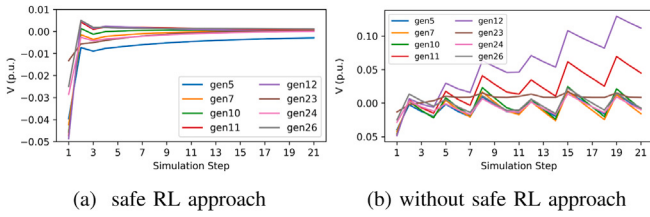


Fig. 4. Dynamics of voltage deviation for safe RL approach(left) and without safe RL approach(right). The controller designed without the safe RL approach leads to unstable trajectories.

5. 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.

5.1. Simulation setup

The cost function that each controller collectively optimizes is $C(u) = \sum_{i=1}^T (\|\hat{v}_i\|_1 + \gamma \|u_i\|_1)$, where γ acts as a trade-off parameter and is set to be 0.01. The base unit for power and voltage is 100 kVA and 12.66 kV, respectively. The bound on action $\bar{u}$ is generated to be uniformly distributed in $[0.01, 0.05]$. We assume other voltage regulation equipment, such as tap-changing transformers and discrete switching capacitor backs, operate at much slower timescales than the inverters. Therefore, in the simulations, we only consider the operation of the inverters, as learned by an agent running RL algorithms [4,29]. 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.

5.2. 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.

5.3. 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

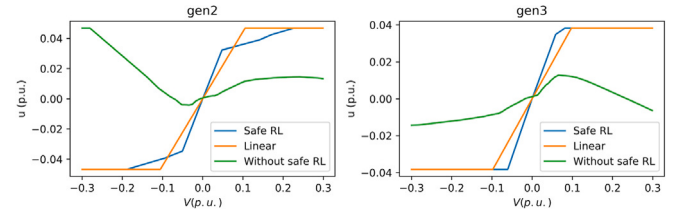


Fig. 5. Voltage control law obtained by linear controller with optimal linear coefficient, neural network controllers designed with safe RL approach and without safe RL approach. The neural network controllers learn flexible non-linear control laws for different buses, with the slope of controller obtained by safe RL approach bounded by Lipschitz constraints.

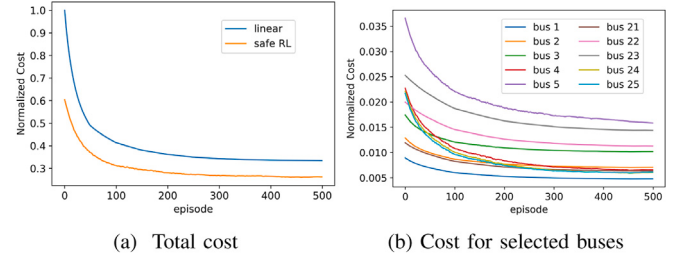


Fig. 6. Normalized cost on test set along the episode of training. (a) Total cost during training of neural network controller and linear controller. Neural network controller designed with safe RL approach achieves lower cost than conventional linear controller. (b) Cost on selected generator buses during the training of neural network controller. All learning trajectories converge well in the decentralized model-free setting, even though they interact through the underlying distribution network.

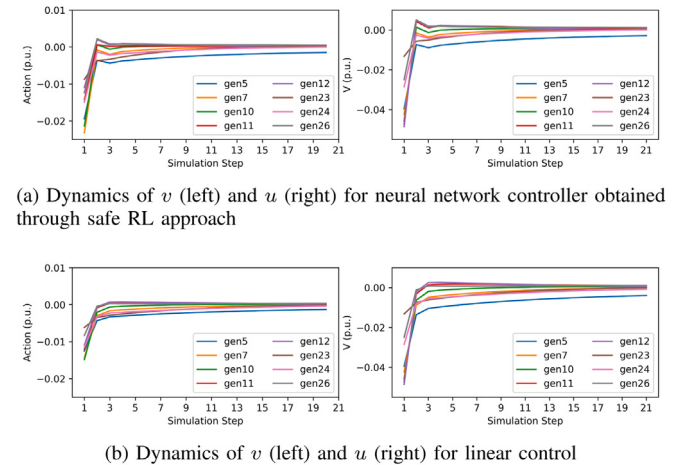


Fig. 7. Dynamics of the voltage deviation $\hat{v}$ and the control action u in selected generator buses corresponding to (a) neural network controller trained with safe RL approach (b) Linear controller obtained by the same decentralized RL algorithm. The neural network controller generally leads to faster decay of voltage deviation.

$\hat{v}_i$, 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 law for different generators, with the safe RL approach guaranteeing a stabilizing controller by bounding the slope with Lipschitz constraints. Fig. 7 illustrates the dynamics of voltage deviation $\hat{v}$ and corresponding control action u under optimal linear controller and neural network controller trained by safe RL approach. The neural network controller generally leads to faster decay of voltage deviation.

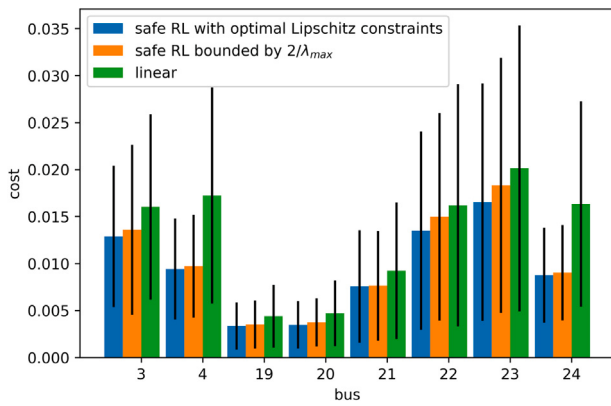


Fig. 8. Distribution of cost in selected generator buses with random initial states corresponding to safe RL with proposed optimal Lipschitz constraints, safe RL bounded by $\frac{2}{\lambda_{max}(X)}$ and optimal linear control. Compared to uniform bound $\frac{2}{\lambda_{max}(X)}$ and linear controller, the proposed 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 costs of the linear controller, the neural network controller bounded by $\frac{2}{\lambda_{max}(X)}$, and the neural network controller with optimal Lipschitz bound obtained in (8) are 0.44, 0.38 and 0.36, respectively. Therefore, the proposed approach can learn a stabilizing controller that reduces the cost by approximately 18.18% compared to conventional linear control law. Moreover, safe RL with the optimal Lipschitz bound also reduces the cost by approximately 5.26% compared to safe RL with the uniform Lipschitz bound $\frac{2}{\lambda_{max}(X)}$.

6. 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 stacked 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 analyzing the difference between decentralized and centralized training, their convergence behaviors and generalizing the proposed methods to nonlinear power flow models are important future directions for us.

CRediT authorship contribution statement

Wenqi Cui: Conceptualization, Methodology, Validation, Writing. **Jiayi Li:** Conceptualization, Methodology, Validation, Writing. **Baosen Zhang:** Conceptualization, Methodology, Writing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] M. Davis, Solar market insight report, Wood Mackenzie Power Renew. (2021).
- [2] K. Turitsyn, P. Sulc, S. Backhaus, M. Chertkov, Options for control of reactive power by distributed photovoltaic generators, *Proc. IEEE* 99 (6) (2011) 1063–1073.
- [3] H.-G. Yeh, D.F. Gayme, S.H. Low, Adaptive VAR control for distribution circuits with photovoltaic generators, *IEEE Trans. Power Syst.* 27 (3) (2012) 1656–1663.
- [4] B. Zhang, A.Y. Lam, A.D. Domínguez-García, D. Tse, An optimal and distributed method for voltage regulation in power distribution systems, *IEEE Trans. Power Syst.* 30 (4) (2014) 1714–1726.
- [5] N. Li, G. Qu, 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, S. Zampieri, A distributed control strategy for reactive power compensation in smart microgrids, *IEEE Trans. Automat. Control* 58 (11) (2013) 2818–2833.
- [7] H. Zhu, H.J. Liu, Fast local voltage control under limited reactive power: Optimality and stability analysis, *IEEE Trans. Power Syst.* 31 (5) (2015) 3794–3803.
- [8] G. Qu, N. Li, Optimal distributed feedback voltage control under limited reactive power, *IEEE Trans. Power Syst.* 35 (1) (2019) 315–331.
- [9] Y.-Y. Hsu, F.-C. Lu, A combined artificial neural network-fuzzy dynamic programming approach to reactive power/voltage control in a distribution substation, *IEEE Trans. Power Syst.* 13 (4) (1998) 1265–1271.
- [10] S. Toma, T. Senjyu, Y. Miyazato, A. Yona, K. Tanaka, 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, L. Gao, Distributed secondary voltage control of islanded microgrids based on RBF-neural-network sliding-mode technique, *IEEE Access* 7 (2019) 65616–65623.
- [12] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
- [13] Q. Yang, G. Wang, A. Sadeghi, G.B. Giannakis, J. Sun, Two-timescale voltage control in distribution grids using deep reinforcement learning, *IEEE Trans. Smart Grid* 11 (3) (2019) 2313–2323.
- [14] Y. Gao, W. Wang, N. Yu, Consensus multi-agent reinforcement learning for volt-var control in power distribution networks, *IEEE Trans. Smart Grid* (2021).
- [15] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, Z. Yi, Deep-reinforcement-learning-based autonomous voltage control for power grid operations, *IEEE Trans. Power Syst.* 35 (1) (2019) 814–817.
- [16] National Electrical Manufacturers Association, et al., American National Standard for Electric Power Systems and Equipment-Voltage Ratings (60 Hertz), National Electrical Manufacturers Association, 1996.
- [17] A. Vaccaro, G. Velotto, A.F. Zobaa, A decentralized and cooperative architecture for optimal voltage regulation in smart grids, *IEEE Trans. Ind. Electron.* 58 (10) (2011) 4593–4602.
- [18] M. Jafari, T.O. Olowu, A.I. Sarwat, Optimal smart inverters volt-var curve selection with a multi-objective volt-var optimization using evolutionary algorithm approach, in: 2018 North American Power Symposium (NAPS), IEEE, 2018, pp. 1–6.
- [19] C. Zhao, U. Topcu, N. Li, S. Low, Design and stability of load-side primary frequency control in power systems, *IEEE Trans. Automat. Control* 59 (5) (2014) 1177–1189.
- [20] E. Mallada, C. Zhao, S. Low, Optimal load-side control for frequency regulation in smart grids, *IEEE Trans. Automat. Control* 62 (12) (2017) 6294–6309.
- [21] Y. Shi, B. Xu, D. Wang, B. Zhang, Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains, *IEEE Trans. Power Syst.* 33 (3) (2017) 2882–2894.
- [22] B.B. Johnson, S.V. Dhople, A.O. Hamadeh, P.T. Krein, Synchronization of parallel single-phase inverters with virtual oscillator control, *IEEE Trans. Power Electron.* 29 (11) (2013) 6124–6138.
- [23] J. Yu, Y. Weng, R. Rajagopal, PaToPaEM: A data-driven parameter and topology joint estimation framework for time-varying system in distribution grids, *IEEE Trans. Power Syst.* 34 (3) (2018) 1682–1692.
- [24] J. Zhang, Y. Wang, Y. Weng, N. Zhang, Topology identification and line parameter estimation for non-PMU distribution network: A numerical method, *IEEE Trans. Smart Grid* 11 (5) (2020) 4440–4453.
- [25] S. Lin, H. Zhu, Data-driven modeling for distribution grids under partial observability, in: 2021 North American Power Symposium (NAPS), IEEE, 2021, pp. 1–6.
- [26] R. Kavet, Characterization of radiofrequency emissions from two models of wireless SmartMeters, in: Project Manager Electric Power Research Institute, EPRI, 2011.
- [27] W. Cui, Y. Jiang, B. Zhang, Reinforcement learning for optimal primary frequency control: A Lyapunov approach, *IEEE Trans. Power Syst.* (2022).
- [28] M.E. Baran, F.F. Wu, Network reconfiguration in distribution systems for loss reduction and load balancing, *IEEE Power Eng. Rev.* 9 (4) (1989) 101–102.
- [29] B.A. Robbins, C.N. Hadjicostis, A.D. Domínguez-García, A two-stage distributed architecture for voltage control in power distribution systems, *IEEE Trans. Power Syst.* 28 (2) (2012) 1470–1482.