Decentralized Safe Reinforcement Learning for Voltage Control

Wenqi Cui, Jiayi Li and Baosen Zhang

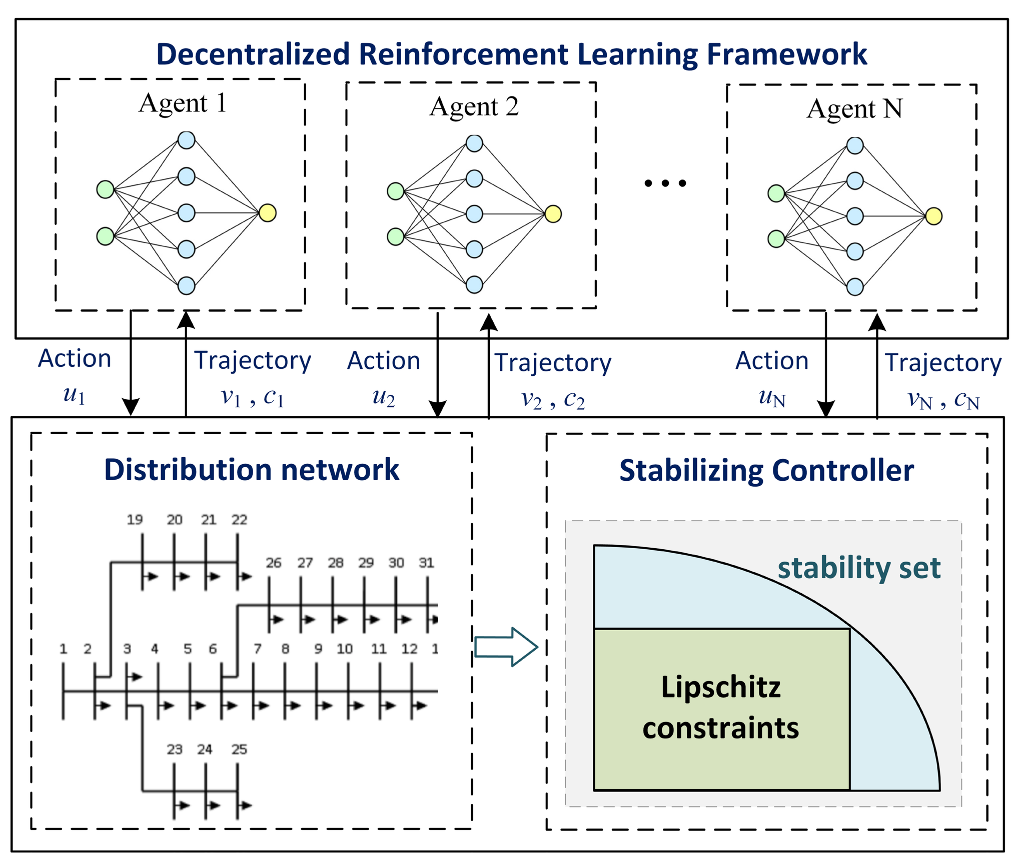
Voltage Control Problems and Challenges

- Distributed energy resources (DERs) such as rooftop solar PV, electric vehicles and battery storage are growing at an increasing pace.
- High variability of solar PV and sudden change in load due to electric vehicles and storage can lead to large voltage fluctuations. Linear controllers can be far from optimal, even for quadratic costs.
- Power electronic devices allow flexible and frequent control actions without degrading lifetime. Neural networks have been used to parametrize the controllers to fully utilize the capabilities of the inverters.
- Reinforcement learning algorithms are proposed to train the neural network controllers in a model-free setting. However, most works neglect the stability requirement and currently this stability condition is checked through simulations.

Our Contribution

A decentralized safe learning method that 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 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.



Model for Optimal Voltage Control

- Let v be the voltage. Let p be active power and q be reactive power. The voltage of the system follows the LinDistFlow model. The aim of voltage regulation is to control q such that v is close to its reference value.
- This work focuses on optimizing the control of q through inverter-based resources. Due to the lack of communication, q needs to be successively updated based on the local voltage measurements.
- At the t-th iteration step, $u_t(v_t)$ is the control law that maps voltage to reactive power.
- Our objective is to optimize the u_t to minimize cost in v_t and q_t defined as $C(u_t)$

$$\min_{\mathbf{u}} \; C(\boldsymbol{u})$$
$$\text{s.t.} \; \boldsymbol{q}_{t+1} = \boldsymbol{q}_t - \boldsymbol{u}_t$$
$$\boldsymbol{v}_{t+1} = \mathbf{R}\boldsymbol{p} + \mathbf{X}(\boldsymbol{q}_t - \boldsymbol{u}_t) + \mathbb{1}$$
$$\underline{\boldsymbol{u}} \leq \boldsymbol{u}_t \leq \overline{\boldsymbol{u}}$$
$$\boldsymbol{u}_t \text{ is stabilizing}$$

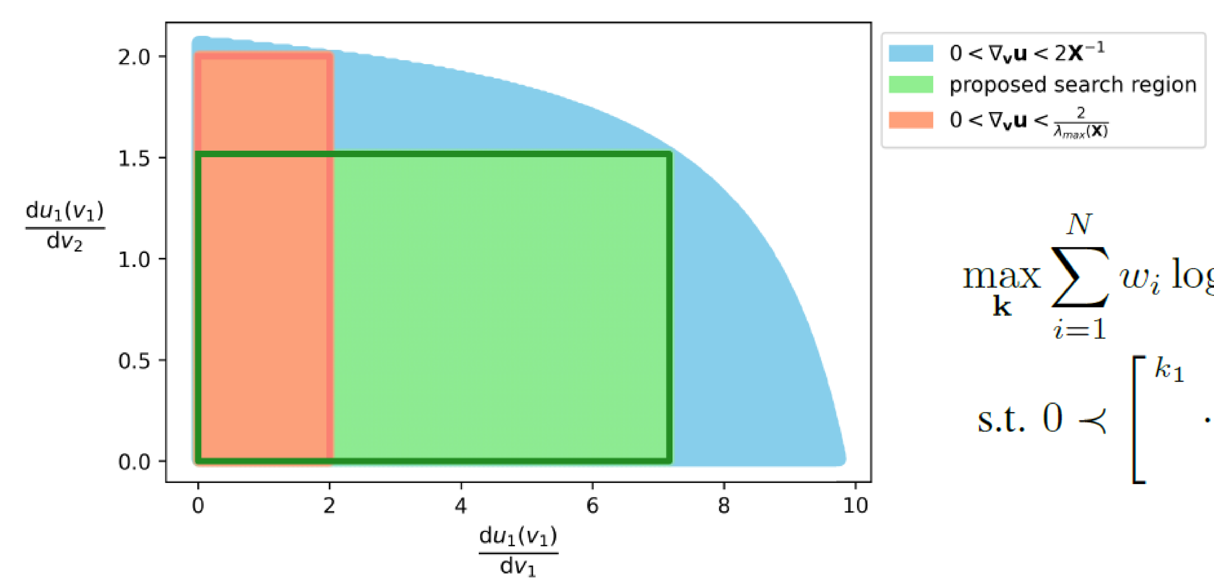
Iterative Update from LinDistFlow

Key Challenge

Lipschitz Constraints for Exponential Stability

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

Theorem 1 shows that as long as each controller u_i satisfies the Lipschitz constraints in the convex set, the system is guaranteed to be locally exponentially stable. We optimize the set of Lipschitz bounds to enlarge the search space of controllers.



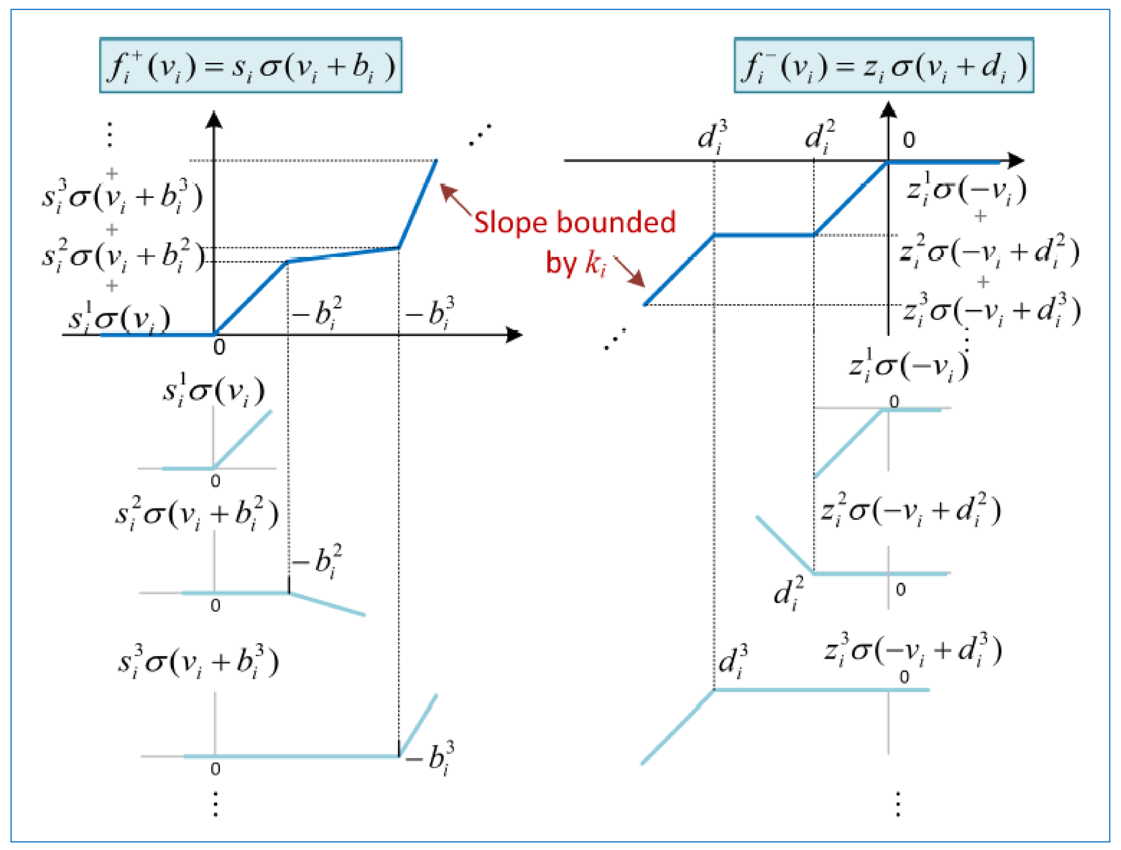
$$\max_{\mathbf{k}} \sum_{i=1}^{N} w_i \log(k_i)$$
$$\text{s.t. } 0 \prec \begin{bmatrix} k_1 & & \\ & \ddots & \\ & & k_N \end{bmatrix} \prec 2\mathbf{X}^{-1}$$

Neural Network Controller

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

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

- The first two requirements are equivalent to designing a monotonically increasing function through the origin.
- To this end, we formulate the controller with a stacked-ReLU structure shown in the right side.
- Then the requirement 3) can be satisfied by directly thresholding the slope.



A Decentralized RL Framework

- 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.

Algorithm 1 Decentralized Reinforcement Learning algorithm with Policy Gradient

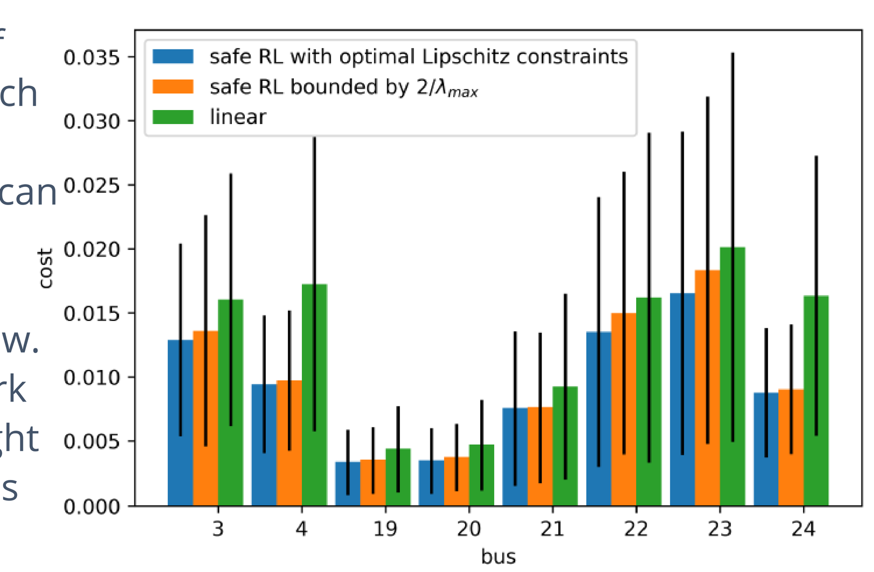
Require: Learning rate α, batch size H, trajectory length T, number of episodes E

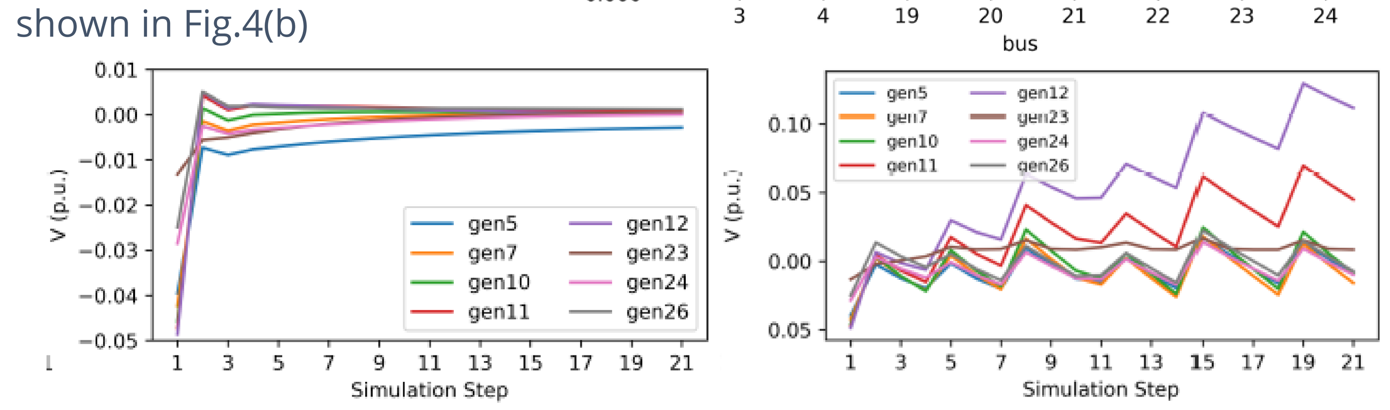
Input: Initial weights θ for control network

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

Numerical Results

- We verify the performance of the proposed safe RL approach on IEEE 33-bus test feeders
- Decentralized RL framework can learn flexible non-linear controllers that outperform conventional linear control law.
- Unconstrained neural network controllers learned by RL might lead to an unstable system, as shown in Fig.4(b)





(a) safe RL approach (b) without safe RL approach

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 approach leads to unstable trajectories

[1] W. Cui, J. Li and B. Zhang, "Decentralized Safe Reinforcement Learning for Voltage Control", accepted to Power System Computation Conference (PSCC)

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISORS: BAOSEN ZHANG

SPONSORS: NSF GRANTS ECCS1942326, CNS1931718 and UW CLEAN ENERGY INSTITUTE