Hierarchical Distributed Voltage Regulation in Networked Autonomous Grids

Preprint

Xinyang Zhou,[1] Zhiyuan Liu,[2] Wenbo Wang,[3]
Changhong Zhao,[1] Fei Ding,[1] and Lijun Chen[2]

[1] *National Renewable Energy Laboratory*
[2] *University of Colorado*
[3] *New York University*

Conference Paper
NREL/CP-5D00-73345
August 2019

This report is available at no cost from the National Renewable Energy
Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Hierarchical Distributed Voltage Regulation in Networked Autonomous Grids

Preprint

Xinyang Zhou,[1] Zhiyuan Liu,[2] Wenbo Wang,[3]
Changhong Zhao,[1] Fei Ding,[1] and Lijun Chen[2]

[1] *National Renewable Energy Laboratory*
[2] *University of Colorado*
[3] *New York University*

Hierarchical Distributed Voltage Regulation in Networked Autonomous Grids

Xinyang Zhou*, Zhiyuan Liu*, Wenbo Wang, Changhong Zhao, Fei Ding, Lijun Chen

Abstract— We propose a novel algorithm to solve optimal power flow (OPF) that aims at dispatching controllable distributed energy resources (DERs) for voltage regulation at minimum cost. The proposed algorithm features unprecedented scalability to large distribution networks by utilizing an information structure based on networked autonomous grids (AGs). Specifically, each AG is a subtree of a large distribution network that has a tree topology. The topology and line parameters of each AG are known only to a regional coordinator (RC) that is responsible for communicating with and dispatching the DERs within this AG. The reduced network, where each AG is treated as a node, is managed by a central coordinator (CC), which knows the topology and line parameters of the reduced network only and communicates with all the RCs. We jointly explore this information structure and the structure of the linearized distribution power flow (LinDistFlow) model to derive a hierarchical, distributed implementation of the primal-dual gradient algorithm that solves the OPF. The proposed implementation significantly reduces the computation burden compared to the centrally coordinated implementation of the primal-dual algorithm. Numerical results on a 4,521-node test feeder show that the proposed hierarchical distributed algorithm can achieve an improvement of more than tenfold in the speed of convergence compared to the centrally coordinated primal-dual algorithm.

I. Introduction

The increasing penetration of distributed energy resources (DERs)—such as rooftop photovoltaic, electric vehicles, battery energy storage systems, thermostatically controlled loads, and other controllable loads—has not only provided enormous potential control flexibility that we can explore, but also imposed challenging tasks of optimally coordinating a large number of networked endpoints to satisfy system-wide objectives and constraints such as demand response and voltage regulation. Distributed algorithms are developed to facilitate the scalable control of large networks of dispatchable DERs by distributing the computational burden either coordinated by a (logically) central controller, e.g., [1]–[4], or among neighboring agents without a central controller, e.g., [5]–[9]. In both cases however, the computational loads increase as the system gets larger, making it more difficult to realize fast real-time control.

One way to cope with the computational burden of large systems is to consider the system as a network of autonomous grids (AGs). AGs rely on cellular building blocks that can self-optimize when isolated from neighboring grids and can participate in optimal operations when interconnected to a larger grid [10]. Such structure enables us to divide the potentially immense computation to smaller problems for AGs.

Extensive studies have been done on optimization and control for islanded AGs, e.g., on frequency and voltage regulation within microgrids [11], [12], and on technologies for AGs to follow dispatch power set points from the bulk system operator while respecting operational constraints inside AGs [13], [14]. On the other hand, interactions among AGs in grid-connected mode are crucial for the optimality and stability of a larger network of AGs. References [15]–[17] consider centralized optimal control to balance energy among AGs by simplifying each AG as a node, but they do not model power flow within individual AGs. References [18]–[23] design distributed algorithms to balance power transmission among AGs while respecting operational constraints within AGs without considering power flow dynamics among AGs. Studies have also designed incentive-based distributed algorithms to optimize energy transactions among AGs without modeling the power flow among AGs [20]–[23].

It is crucial to model the power flow both within and among AGs to satisfy network-wide objectives and operational constraints, e.g., voltage regulation; however, very limited literature has done this. Recently, [24] applies a game-theoretic approach to manage a partitioned distribution network based on noncooperative Nash game, but the uniqueness of the equilibrium, the convergence, and the global performance are all difficult to characterize.

This work considers a potentially large distribution network controlled cooperatively by a number of networked AGs. We model the distribution network using the DistFlow model [25], [26], which captures power flows within and among AGs. A regional coordinator (RC) communicates with all the dispatchable nodes within each AG, and a central coordinator (CC) communicates with all the RCs. Each RC knows only the topology and line parameters of the AG that it controls, and the CC knows only the topology and line parameters of the reduced network, which treats each AG as a node and connects all the AGs. Given such information availability, we explore the topological structure of the linearized DistFlow (LinDistFlow) model to derive a hierarchical, distributed implementation of the primal-dual gradient algorithm that solves an optimal power flow (OPF) problem. The OPF problem minimizes the total cost of all the controllable DERs and a cost associated with the total network load subject to voltage regulation

*The first two authors contributed equally.

X. Zhou, C. Zhao, and F. Ding are with the National Renewable Energy Laboratory, Golden, CO, 80401, USA (Emails: {xinyang.zhou, changhong.zhao, fei.ding}@nrel.gov).

Z. Liu and L. Chen are with the College of Engineering and Applied Science, University of Colorado, Boulder, CO 80309, USA (Emails: {zhiyuan.liu, lijun.chen}@colorado.edu).

W. Wang is with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201, USA (Email: ww787@nyu.edu)

constraints. The proposed implementation significantly reduces the computation burden compared to the centrally coordinated implementation of the primal-dual algorithm, which requires a central coordinator for the whole network. The performance of the proposed implementation is verified through numerical simulation of a 4,521-node test feeder. Simulation results show that an improvement of more than tenfold in the speed of convergence can be achieved by the hierarchical distributed method compared to the centrally coordinated implementation. This significant improvement in convergence speed makes real-time grid optimization and control possible. Meanwhile, to our best knowledge, the size of the network in our simulation is the largest in optimization-based control in power system.

The rest of this paper is organized as follows. Section II models the distribution system, formulates the OPF problem, and introduces the primal-dual gradient algorithm for solving the OPF problem. Section III proposes a hierarchical distributed implementation of the primal-dual gradient algorithm. Section IV provides numerical results, and Section V concludes this paper.

II. System Modeling and Problem Formulation

A. Power Flow in Distribution Network

Consider a radial power distribution network denoted by $\mathcal{T} = \{\mathcal{N} \cup \{0\}, \mathcal{E}\}$ with $N + 1$ nodes collected in the set $\mathcal{N} \cup \{0\}$, where $\mathcal{N} := \{1, ..., N\}$ and node 0 is the slack bus, and distribution lines collected in the set $\mathcal{E}$. For each node $i \in \mathcal{N}$, denote by $\mathcal{E}_i \subseteq \mathcal{E}$ the set of lines on the unique path from node 0 to node i, and let p_i and q_i denote the real and reactive power injected, where negative power injection means power consumption and positive power injection means power generation. Let v_i be the magnitude of the complex voltage (phasor) at node i. For each line $(i, j) \in \mathcal{E}$, denote by r_{ij} and x_{ij} its resistance and reactance, and P_{ij} and Q_{ij} the real and reactive power from node i to node j. Let ℓ_{ij} denote the squared magnitude of the complex branch current (phasor) from node i to node j.

We adopt the following DistFlow model [25], [26] for the radial distribution network:

$$P_{ij} = -p_j + \sum_{k:(j,k)\in\mathcal{E}} P_{jk} + r_{ij}\ell_{ij}, \tag{1a}$$

$$Q_{ij} = -q_j + \sum_{k:(j,k)\in\mathcal{E}} Q_{jk} + x_{ij}\ell_{ij}, \tag{1b}$$

$$v_j^2 = v_i^2 - 2(r_{ij}P_{ij} + x_{ij}Q_{ij}) + (r_{ij}^2 + x_{ij}^2)\ell_{ij}, \tag{1c}$$

$$\ell_{ij}v_i^2 = P_{ij}^2 + Q_{ij}^2. \tag{1d}$$

Following [27]–[29], we assume that the active and reactive power loss $r_{ij}\ell_{ij}$ and $x_{ij}\ell_{ij}$, as well as $r_{ij}^2\ell_{ij}$ and $x_{ij}^2\ell_{ij}$, are negligible and can thus be ignored. Indeed, the losses are much smaller than power flows P_{ij} and Q_{ij}, typically on the order of 1%. We further assume that $v_i \approx 1$, $\forall i$ so that we can approximate $v_j^2 - v_i^2 \approx 2(v_j - v_i)$ in Eq. (1c).[1] This approximation introduces a small relative error of at

---

[1]This assumption is not essential because we could work with v_i^2 instead.

most 0.25% under the practically maximum 5% deviation in voltage magnitude.

With these approximations, Eqs. (1) is simplified to the following linear model:

$$\boldsymbol{v} = R\boldsymbol{p} + X\boldsymbol{q} + \tilde{\boldsymbol{v}}, \tag{2}$$

where bold symbols $\boldsymbol{v} = [v_1, \ldots, v_N]^\top$, $\boldsymbol{p} = [p_1, \ldots, p_N]^\top$, $\boldsymbol{q} = [q_1, \ldots, q_N]^\top \in \mathbb{R}^N$ represent vectors, $\tilde{\boldsymbol{v}}$ is a constant vector depending on initial conditions, and the sensitivity matrices R and X, respectively, consist of elements:

$$R_{ij} := \sum_{(h,k)\in\mathcal{E}_i\cap\mathcal{E}_j} r_{hk}, \quad X_{ij} := \sum_{(h,k)\in\mathcal{E}_i\cap\mathcal{E}_j} x_{hk}. \tag{3}$$

Here, the voltage-to-power-injection sensitivity factors R_{ij} (X_{ij}) represent *the resistance (reactance) of the common path of node $\boldsymbol{i}$ and $\boldsymbol{j}$ leading back to node 0.* Keep in mind that this result serves as the basis for designing the hierarchical distributed algorithm to be introduced later. Fig. 1 illustrates $\mathcal{E}_i \cap \mathcal{E}_j$ for two arbitrary nodes i and j in a radial network and their corresponding R_{ij} and X_{ij}.
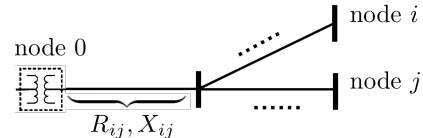

Fig. 1: $\mathcal{E}_i \cap \mathcal{E}_j$ for two arbitrary nodes i, j in the network and the corresponding mutual voltage-to-power-injection sensitivity factors R_{ij}, X_{ij}.

B. OPF Problem

Assume at node $i \in \mathcal{N}$ there is a dispatchable DER (or aggregation of DERs) whose real and reactive power injections are confined as $(p_i, q_i) \in \mathcal{Y}_i$, where $\mathcal{Y}_i$ is a convex and compact set. Let P_0 denote active power injected into the feeder at node 0, which can be approximated by the total active power loads as:

$$P_0 = -P_I - \sum_{i\in\mathcal{N}} p_i, \tag{4}$$

where P_I denotes the uncontrollable total inelastic loads. Note that P_0 is negative if the total active power consumption is larger than the total active power generation. Use $\boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q})$ and $P_0(\boldsymbol{p})$ to represent (2) and (4), respectively, and consider the following OPF problem:

$$\min_{\boldsymbol{p},\boldsymbol{q}} \quad \sum_{i\in\mathcal{N}} C_i(p_i, q_i) + C_0(P_0(\boldsymbol{p})), \tag{5a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{v}} \leqslant \boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q}) \leqslant \overline{\boldsymbol{v}}, \tag{5b}$$

$$(p_i, q_i) \in \mathcal{Y}_i, \forall i \in \mathcal{N}, \tag{5c}$$

where the objective $C_i(p_i, q_i)$ is the cost function for node i, and the coupling term $C_0(P_0(\boldsymbol{p}))$ represents the cost associated with the total network load. For example, $C_0(P_0(\boldsymbol{p})) = \alpha(P_0(\boldsymbol{p}) - \hat{P}_0)^2$ penalizes $P_0(\boldsymbol{p})$'s deviation from a dispatching signal $\hat{P}_0$ with a weight $\alpha > 0$. We make the following assumption for these cost functions.

2

Assumption 1 $C_i(p_i, q_i)$, $\forall i \in \mathcal{N}$ *are continuously differentiable and strongly convex in* (p_i, q_i), *with bounded first-order derivative in* $\mathcal{Y}_i$. *Meanwhile,* $C_0(P_0)$ *is continuously differentiable and convex with bounded first-order derivative.*

C. Primal-Dual Gradient Algorithm

Associate dual variables $\underline{\mu}$ and $\overline{\mu}$ with the left-hand-side and the right-hand-side of (5b), respectively, so that the Lagrangian of (5) is written as:

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{q}; \overline{\mu}, \underline{\mu}) = \sum_{i \in \mathcal{N}} C_i(p_i, q_i) + C_0(P_0(\boldsymbol{p}))$$
$$+ \underline{\mu}^\top(\underline{\boldsymbol{v}} - \boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q})) + \overline{\mu}^\top(\boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q}) - \overline{\boldsymbol{v}}), \quad (6)$$

with (5c) treated as the domain of $(\boldsymbol{p}, \boldsymbol{q})$.

To design an algorithm with provable convergence, we introduce the following regularized Lagrangian with parameter $\phi > 0$:

$$\mathcal{L}_\phi(\boldsymbol{p}, \boldsymbol{q}; \overline{\mu}, \underline{\mu}) = \sum_{i \in \mathcal{N}} C_i(p_i, q_i) + C_0(P_0(\boldsymbol{p}))$$
$$+ \underline{\mu}^\top(\underline{\boldsymbol{v}} - \boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q})) + \overline{\mu}^\top(\boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q}) - \overline{\boldsymbol{v}}) - \frac{\phi}{2}\|\mu\|_2^2, \quad (7)$$

where $\mu := [\underline{\mu}^\top, \overline{\mu}^\top]^\top$.

Theorem 1 *There exists one unique saddle point* $(\boldsymbol{p}^*, \boldsymbol{q}^*; \underline{\mu}^*, \overline{\mu}^*)$ *of* $\mathcal{L}_\phi$.

Proof: The result follows from the fact that $\mathcal{L}_\phi(\boldsymbol{p}, \boldsymbol{q}; \underline{\mu}, \overline{\mu})$ is strongly convex in $\boldsymbol{p}, \boldsymbol{q}$, and strongly concave in $\underline{\mu}, \overline{\mu}$. ∎

Theorem 2 *The difference between the saddle points of* (6) *and* (7) *is bounded, and it is proportional to* ϕ.

We refer the proof of Theorem 2 to [3], [30].

Then, the iterative primal-dual gradient algorithm to find the saddle point of (7) is cast as follows:

$$\boldsymbol{p}(t+1) = \Big[\boldsymbol{p}(t) - \epsilon\Big(\nabla_{\boldsymbol{p}} C(\boldsymbol{p}(t), \boldsymbol{q}(t)) - C_0'(P_0(\boldsymbol{p}(t))) \cdot \mathbf{1}_N$$
$$+ R^\top(\overline{\mu}(t) - \underline{\mu}(t))\Big)\Big]_{\mathcal{Y}}, \quad (8a)$$

$$\boldsymbol{q}(t+1) = \Big[\boldsymbol{q}(t) - \epsilon\Big(\nabla_{\boldsymbol{q}} C(\boldsymbol{p}(t), \boldsymbol{q}(t))$$
$$+ X^\top(\overline{\mu}(t) - \underline{\mu}(t))\Big)\Big]_{\mathcal{Y}}, \quad (8b)$$

$$\underline{\mu}(t+1) = \big[\underline{\mu}(t) + \epsilon(\underline{\boldsymbol{v}} - \boldsymbol{v}(t) - \phi\underline{\mu}(t))\big]_+, \quad (8c)$$

$$\overline{\mu}(t+1) = \big[\overline{\mu}(t) + \epsilon(\boldsymbol{v}(t) - \overline{\boldsymbol{v}} - \phi\overline{\mu}(t))\big]_+, \quad (8d)$$

where $\epsilon > 0$ is some constant stepsize to be determined; $\mathbf{1}_N = [1, \ldots, 1]^\top \in \mathbb{R}^N$, $[\ ]_{\mathcal{Y}}$ is the projection operator onto the feasible set $\mathcal{Y} := \times_{i \in \mathcal{N}} \mathcal{Y}_i$; and $[\ ]_+$ is the projection operator onto the positive orthant.

D. Convergence

We use $\boldsymbol{y} := [\boldsymbol{p}^\top, \boldsymbol{q}^\top]^\top$ to stack all the primal variables and rewrite (8) equivalently as follows:

$$\begin{bmatrix} \boldsymbol{y}(t+1) \\ \mu(t+1) \end{bmatrix} = \left[\begin{bmatrix} \boldsymbol{y}(t) \\ \mu(t) \end{bmatrix} - \epsilon \begin{bmatrix} \nabla_{\boldsymbol{y}} \mathcal{L}_\phi(\boldsymbol{y}(t), \mu(t)) \\ -\nabla_\mu \mathcal{L}_\phi(\boldsymbol{y}(t), \mu(t)) \end{bmatrix}\right]_{\mathcal{Y} \times \mathcal{U}}, \quad (9)$$

where $\mathcal{U}$ is the feasible positive orthant for the dual variables. We further let $\boldsymbol{z} := [\boldsymbol{y}^\top, \mu^\top]^\top$ stack all variables, and $T(\boldsymbol{z})$ denote the operator $\begin{bmatrix} \nabla_{\boldsymbol{y}} \mathcal{L}_\phi(\boldsymbol{y}, \mu) \\ -\nabla_\mu \mathcal{L}_\phi(\boldsymbol{y}, \mu) \end{bmatrix}$.

Lemma 1 $T(\boldsymbol{z})$ *is a strongly monotone operator.*

Proof: See Appendix. ∎

By Lemma 1, there exists some constant $M > 0$ such that for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Y} \times \mathcal{U}$, one has:

$$(T(\boldsymbol{z}) - T(\boldsymbol{z}'))^\top(\boldsymbol{z} - \boldsymbol{z}') \geqslant M\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2. \quad (10)$$

Moreover, based on Assumption 1, the operator $T(\boldsymbol{z})$ is also Lipschitz continuous, i.e., there exists some constant $L > 0$ such that for any feasible $\boldsymbol{z}$ and $\boldsymbol{z}'$, we have:

$$\|T(\boldsymbol{z}) - T(\boldsymbol{z}')\|_2^2 \leqslant L^2\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2. \quad (11)$$

We present the next theorem that ensures the convergence of (9) with a small enough stepsize.

Theorem 3 *If the stepsize* ϵ *satisfies* $0 < \epsilon \leqslant \overline{\epsilon} < 2M/L^2$ *for some* $\overline{\epsilon}$, (9) *converges to the unique saddle point of* (7) *exponentially fast.*

Proof: See Appendix. ∎

E. Motivation for Hierarchical Design

Note that in (8) the update of any p_i (resp. q_i) involves the knowledge of $\sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j - \underline{\mu}_j)$ (resp. $\sum_{j \in \mathcal{N}} X_{ij}(\overline{\mu}_j - \underline{\mu}_j)$). Therefore, at each iteration a coordinator cognizant of the entire network sensitivity matrices R and X is required to first collect updated dual variables from all the nodes, calculate $R^\top(\overline{\mu} - \underline{\mu})$ and $X^\top(\overline{\mu} - \underline{\mu})$, and then send the results back to the corresponding nodes. This becomes computationally more challenging in a larger network containing thousands of or even more controllable endpoints, not to mention the recalculation of the huge R, X matrices in case changes in network topology or regulator taps occur.

This motivates us to design a hierarchical control structure where the large network is partitioned into smaller AGs, each of which is managed locally by its own RC, and there is a CC that manages only a reduced network where each AG is treated as one node. As will be shown in the next section, the hierarchical control design not only distributes the computational burden but also reduces a huge amount of the computation by utilizing the network structure.

III. HIERARCHICAL DISTRIBUTED ALGORITHM

A. Distribution Feeder as Networked Autonomous Grids

Definition 1 *A subtree of a tree* $\mathcal{T}$ *is a tree consisting of a node in* $\mathcal{T}$, ***all of its descendants*** *in* $\mathcal{T}$, *and their connecting lines.*

We group all the nodes of the distribution network $\mathcal{T}$ into (1) K subtrees indexed by $\mathcal{T}_k = \{\mathcal{N}_k, \mathcal{E}_k\}$, $k \in \mathcal{K} = \{1, \ldots, K\}$, and (2) a set $\mathcal{N}_0$ collecting all the other "unclustered" nodes in $\mathcal{N}$. Here, $\mathcal{N}_k$ of size N_k is the set of nodes in subtree $\mathcal{T}_k$ and $\mathcal{E}_k$ contains their connecting lines. Thus, we have $\cup_{k \in \mathcal{K}} \mathcal{N}_k \cup \mathcal{N}_0 = \mathcal{N}$ and $\mathcal{N}_j \cap \mathcal{N}_k = \varnothing, \forall j \neq$

3

Algorithm 1 Hierarchical Distributed Voltage Regulation
___

repeat

[1] node $i \in \mathcal{N}$ updates $(p_i(t+1), q_i(t+1))$ by

$$
\begin{align}
p_i(t+1) &= [p_i(t) - \epsilon(\partial_{p_i} C_i(p_i(t), q_i(t)) - C_0'(P_0(\boldsymbol{p}(t))) + \alpha_i(t))]_{\mathcal{Y}_i}, \tag{12a}\\
q_i(t+1) &= [q_i(t) - \epsilon(\partial_{q_i} C_i(p_i(t), q_i(t)) + \beta_i(t))]_{\mathcal{Y}_i}, \tag{12b}
\end{align}
$$

and $(\underline{\mu}_i(t+1), \overline{\mu}_i(t+1))$ based on local voltage by

$$
\underline{\mu}_i(t+1) = [\underline{\mu}_i(t) + \epsilon(\underline{v}_i - v_i(t) - \phi\underline{\mu}_i(t))]_+, \quad \overline{\mu}_i(t+1) = [\overline{\mu}_i(t) + \epsilon(v_i(t) - \overline{v}_i - \phi\overline{\mu}_i(t))]_+. \tag{13}
$$

[2] RC $k \in \mathcal{K}$ calculates and sends $\sum_{i \in \mathcal{N}_k} \left(\overline{\mu}_i(t+1) - \underline{\mu}_i(t+1)\right)$ to CC; unclustered node $i \in \mathcal{N}_0$ sends $(\overline{\mu}_i(t+1) - \underline{\mu}_i(t+1))$ to CC.

[3] CC computes within the reduced network

$$
\begin{align}
\alpha_k^{\text{out}}(t+1) &= \sum_{h \in \mathcal{K}, h \neq k} R_{n_h^0 n_k^0} \sum_{j \in \mathcal{N}_h} (\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)) + \sum_{j \in \mathcal{N}_0} R_{j n_k^0}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \forall k \in \mathcal{K}, \tag{14a}\\
\beta_k^{\text{out}}(t+1) &= \sum_{h \in \mathcal{K}, h \neq k} X_{n_h^0 n_k^0} \sum_{j \in \mathcal{N}_h} (\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)) + \sum_{j \in \mathcal{N}_0} X_{j n_k^0}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \forall k \in \mathcal{K}, \tag{14b}
\end{align}
$$

and

$$
\begin{align}
\alpha_i(t+1) &= \sum_{k \in \mathcal{K}} R_{i n_k^0} \sum_{j \in \mathcal{N}_k} (\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)) + \sum_{j \in \mathcal{N}_0} R_{ij}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \forall i \in \mathcal{N}_0, \tag{15a}\\
\beta_i(t+1) &= \sum_{k \in \mathcal{K}} X_{i n_k^0} \sum_{j \in \mathcal{N}_k} (\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)) + \sum_{j \in \mathcal{N}_0} X_{ij}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \forall i \in \mathcal{N}_0, \tag{15b}
\end{align}
$$

and sends $(\alpha_k^{\text{out}}(t+1), \beta_k^{\text{out}}(t+1))$ to RC $k \in \mathcal{K}$, and $(\alpha_i(t+1), \beta_i(t+1))$ to unclustered node $i \in \mathcal{N}_0$.

[4] RC $k \in \mathcal{K}$ calculates within AG k:

$$
\alpha_{k,i}^{\text{in}}(t+1) = \sum_{j \in \mathcal{N}_k} R_{ij}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \text{and} \quad \alpha_i(t+1) = \alpha_{k,i}^{\text{in}}(t+1) + \alpha_k^{\text{out}}(t+1), \ \forall i \in \mathcal{N}_k, \tag{16a}
$$

$$
\beta_{k,i}^{\text{in}}(t+1) = \sum_{j \in \mathcal{N}_k} X_{ij}(\overline{\mu}_j(t+1) - \underline{\mu}_j(t+1)), \ \text{and} \quad \beta_i(t+1) = \beta_{k,i}^{\text{in}}(t+1) + \beta_k^{\text{out}}(t+1), \ \forall i \in \mathcal{N}_k, \tag{16b}
$$

and sends $(\alpha_i(t+1), \beta_i(t+1))$ to node $i \in \mathcal{N}_k$.

[5] $\boldsymbol{v}(t+1)$ and $P_0(t+1)$ are updated by the physical system:

$$
\boldsymbol{v}(t+1) = R\boldsymbol{p}(t+1) + X\boldsymbol{q}(t+1) + \tilde{\boldsymbol{v}}, \qquad P_0(t+1) = P_I + \sum_{i \in \mathcal{N}} p_i(t+1). \tag{17}
$$

[6] CC computes/measures $P_0(t+1)$ at the substation and broadcasts $C_0'(P_0(\boldsymbol{p}(t+1)))$.

until stopping criterion is met (e.g., $|P_0(\boldsymbol{p}(t+1)) - P_0(\boldsymbol{p}(t))| < \sigma$ for some small $\sigma > 0$)
___

k. Assume each subtree $\mathcal{T}_k$ is an AG managed by an RC cognizant of the topology of $\mathcal{T}_k$ and communicating with all the controllable nodes within $\mathcal{T}_k$.

Denote the root node of subtree $\mathcal{T}_k$ by n_k^0, and consider a reduced network $\mathcal{T}^r = \{\mathcal{N}^r \cup \{0\}, \mathcal{E}^r\}$ where $\mathcal{N}^r := \cup_{k \in \mathcal{K}} \{n_k^0\} \cup \mathcal{N}_0$ consists of the root nodes of subtrees and all the unclustered nodes, and $\mathcal{E}^r$ consists of their connecting lines. We assume a CC cognizant of the topology of the reduced network $\mathcal{T}^r$ and communicating with all the RCs as well as the unclustered nodes.

Since each indexed subtree is considered as an AG, in this paper, we use the terms "subtree" and "AG" interchangeably.

B. Hierarchical Distributed Algorithm

In this part, we design a hierarchical distributed algorithm for the networked AG structure introduced previously, by exploring the structure of the network matrices R and X. For simplicity, we elaborate the algorithm design for real power injections $\boldsymbol{p}$ only, and that of $\boldsymbol{q}$ follows similarly.
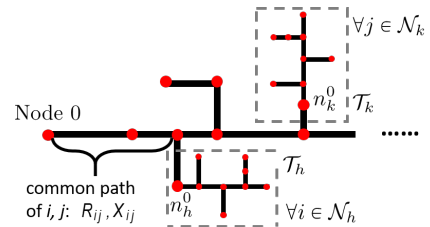

Fig. 2: The unclustered nodes and the root nodes of subtrees together with their connecting lines constitute the reduced network. Two subtrees $\mathcal{T}_h$ and $\mathcal{T}_k$ share the same R_{ij} and X_{ij} for any of their respective nodes i and j.

Rewrite (8a) as:

$$
p_i(t+1) = \Big[p_i(t) - \epsilon\Big(\partial_{p_i} C_i(p_i(t), q_i(t)) - C_0'(P_0(\boldsymbol{p}(t))) + \sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j(t) - \underline{\mu}_j(t)) \Big) \Big]_{\mathcal{Y}_i}, \forall i \in \mathcal{N}. \tag{18}
$$

4

Note that in (18), although $\partial_{p_i} C_i(p_i(t), q_i(t))$ is local information and the scalar $C_0'(P_0(\boldsymbol{p}(t)))$ can be easily broadcast, the last term $\sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j(t) - \underline{\mu}_j(t))$ couples the whole network in principle.

To design a more scalable algorithm, we first introduce the following lemma.

Lemma 2 *Given any two subtrees $\mathcal{T}_h$ and $\mathcal{T}_k$ with their root nodes n_h^0 and n_k^0, we have $R_{ij} = R_{n_h^0 n_k^0}, X_{ij} = X_{n_h^0 n_k^0}, \forall i \in \mathcal{N}_h, \forall j \in \mathcal{N}_k$. Similarly, given any unclustered node $i \in \mathcal{N}_0$ and a subtree $\mathcal{T}_k$ with its root node n_k^0, we have $R_{ij} = R_{in_k^0}, X_{ij} = X_{in_k^0}, \forall j \in \mathcal{N}_k$.*

Proof: We have the two following facts:

1) By (3), R_{ij} (resp. X_{ij}) is the summed resistance (resp. reactance) of the common path of node i and j leading back to node 0.
2) Any node in one subtree and any node in another subtree (or any node in one subtree and one unclustered node) share the same common path back to node 0.

The result is immediate from 1) and 2). ∎

Remark 1 *The results of Lemma 2 can be illustrated in Fig. 2. In Fig. 2, any node i in $\mathcal{T}_h$ and any node j in $\mathcal{T}_k$ share the same $R_{ij} = R_{n_h^0 n_k^0}$ and $X_{ij} = X_{n_h^0 n_k^0}$, which by definition (3) are the summed resistance and reactance of their marked common path connected to node 0.*

Lemma 2 enables us to recalculate the coupling terms in a hierarchical distributed way.
For node $i \in \mathcal{N}_k$: We decompose $\sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j - \underline{\mu}_j)$ as:

$$
\begin{aligned}
&\sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j - \underline{\mu}_j) \\
= &\sum_{j \in \mathcal{N}_k} R_{ij}(\overline{\mu}_j - \underline{\mu}_j) + \sum_{j \in \mathcal{N} \setminus \mathcal{N}_k} R_{ij}(\overline{\mu}_j - \underline{\mu}_j) \\
= &\sum_{j \in \mathcal{N}_k} R_{ij}(\overline{\mu}_j - \underline{\mu}_j) + \sum_{h \in \mathcal{K}, h \neq k} R_{n_h^0 n_k^0} \sum_{j \in \mathcal{N}_h} (\overline{\mu}_j - \underline{\mu}_j) \\
&+ \sum_{j \in \mathcal{N}_0} R_{n_k^0 j}(\overline{\mu}_j - \underline{\mu}_j),
\end{aligned} \tag{19}
$$

where the first part of (19) consists of information within AG k (together with the line parameter from n_k^0 to bus 0, i.e., $R_{n_k^0 n_k^0}$ and $X_{n_k^0 n_k^0}$, which can be informed by CC), the second from all the other AGs, and the third from the unclustered nodes. For convenience, denote:

$$
\begin{aligned}
\alpha_{k,i}^{\text{in}} &= \sum_{j \in \mathcal{N}_k} R_{ij}(\overline{\mu}_j - \underline{\mu}_j), \\
\alpha_k^{\text{out}} &= \sum_{h \in \mathcal{K}, h \neq k} R_{n_h^0 n_k^0} \sum_{j \in \mathcal{N}_h} (\overline{\mu}_j - \underline{\mu}_j) + \sum_{j \in \mathcal{N}_0} R_{n_k^0 j}(\overline{\mu}_j - \underline{\mu}_j).
\end{aligned}
$$

Note that for $i \in \mathcal{N}_k$, $\alpha_{k,i}^{\text{in}}$ is accessible by RC k cognizant of the topology of AG k, and α_k^{out} does not involve the network structure of any other AGs but only that of the reduced network known by CC.
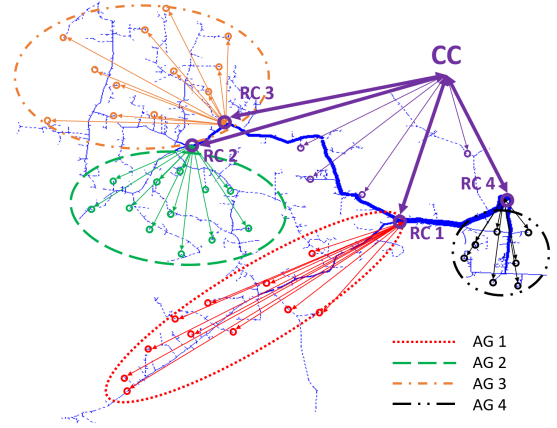

Fig. 3: The 11,000-node test feeder constructed from an IEEE 8,500-node test feeder and a modified EPRI Test Circuits Ckt7. Four AGs are formed for our experiments.

For unclustered node $i \in \mathcal{N}_0$: Similarly, one has:

$$
\begin{aligned}
&\sum_{j \in \mathcal{N}} R_{ij}(\overline{\mu}_j - \underline{\mu}_j) \\
= &\sum_{k \in \mathcal{K}} R_{in_k^0} \sum_{j \in \mathcal{N}_k} (\overline{\mu}_j - \underline{\mu}_j) + \sum_{j \in \mathcal{N}_0} R_{ij}(\overline{\mu}_j - \underline{\mu}_j), \tag{20}
\end{aligned}
$$

whose computation requires only the topology and line parameters of the reduced network known by CC.

Meanwhile, computational loads are also largely reduced because of the following reasons: (1) the term $\sum_{h \in \mathcal{K}, h \neq k} R_{n_h^0 n_k^0} \sum_{j \in \mathcal{N}_h} (\overline{\mu}_j - \underline{\mu}_j)$ requires less computation than the original $\sum_{j \in \mathcal{N}_h, h \neq k} R_{ij}(\overline{\mu}_j - \underline{\mu}_j)$; and, more importantly, (2) α_k^{out} is the same for all the nodes in $\mathcal{N}_k$, which reduces a lot of the repetitive computation that is executed by (8). Equations (19)–(20) motivate us to design the hierarchical distributed algorithm, Algorithm 1. As will be shown in Section IV, the improvement in convergence speed is more than ten times for a 4,521-node feeder, enabling fast real-time grid optimization.

In Algorithm 1, both the CC and RCs are in charge of only a portion of the whole system: the CC manages the reduced network and coordinates RCs and unclustered nodes without knowing any detailed information within any AGs, and the RCs each manage their own AGs without knowing detailed information of other AGs or the reduced network. Also see Fig. 3 for an illustration of clustering and information flows in Algorithm 1.

In addition, because Algorithm 1 is mathematically equivalent to (8), the results in Theorems 1–3 also apply to Algorithm 1.

IV. NUMERICAL RESULTS

In this section, we provide numerical results to show the effectiveness and the efficiency of our design.

A. Network Setup

A three-phase, unbalanced, 11,000-node test feeder is constructed by connecting an IEEE 8,500-node test feeder and a
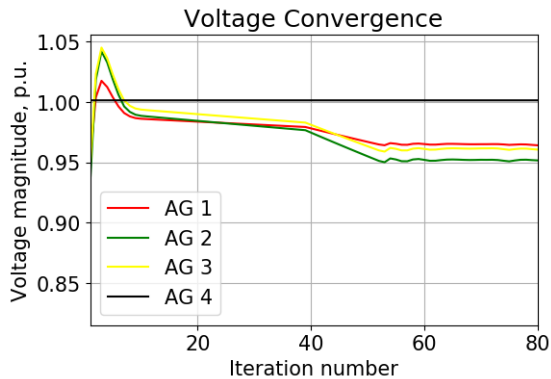
5

Fig. 4: Voltage convergence at selected nodes from four AGs.

Fig. 5: Voltages are controlled within $[0.95, 1.05]$ p.u. by the proposed OPF algorithm.

modified EPRI Ckt7. Fig. 3 shows the single-line diagram of the feeder, where the line width is proportional to the nominal power flow on it. The primary side of the feeder is modeled in detail, whereas the loads on the secondary side (which in this system is aggregation of several loads) are lumped into corresponding distribution transformers, resulting in a 4,521-node network with 1,335 aggregated loads. We group all the nodes into unclustered nodes and four AGs marked in Fig. 3. AG 1 contains 357 nodes with controllable loads, AG 2 contains 222, AG 3 contains 310, and AG 4 contains 154. We fix the loads on all 292 unclustered nodes for simplicity.

The three-phase nonlinear power flow model is simulated in OpenDSS. With default control of the capacitors and regulators in OpenDSS [31], we achieve the voltage profile shown in Fig. 5 with orange dots, where undervoltages are observed. We next disable the control of all the capacitors and regulators to obtain the heavily undervoltage scenario shown with blue dots in Fig. 5, and we implement our voltage regulation algorithm.

B. Hierarchical Distributed Voltage Regulation

For each controllable node i, we consider minimizing the cost of its deviation from its original load level p_i^o, q_i^o, i.e., $C_i(p_i, q_i) = (p_i - p_i^o)^2 + (q_i - q_i^o)^2$. Because we focus on voltage regulation here, we set $C_0(P_0) = \alpha(P_0^2 - \hat{P}_0)^2$ with a small $\alpha = 0.0005$ and $\hat{P}_0$ set to 80% of the initial P_0. $\underline{v}_i$ and $\overline{v}_i$ are uniformly set to 0.95 p.u. and 1.05 p.u., respectively. We implement Algorithm 1 with a constant stepsize $\epsilon = 1 \times 10^{-3}$.

1) Convergence: As shown in Fig. 4, convergence of the algorithm takes 50–60 iterations. In addition, thanks to the hierarchical design that largely reduces the computational burden, it takes about 25 seconds for the control of more than 1,000 nodes to converge on a laptop with an Intel Core i7-7600U CPU @ 2.80GHz 2.90GHz, 8.00GB RAM, running Python 3.6 on Windows 10 Enterprise Version. On the other hand, it takes the centrally coordinated primal-dual algorithm (8) (which is implemented by a centralized coordinator for all the nodes) almost four times longer to converge. By further considering the parallel computation of four clusters, the overall improvement in speed is more than tenfold.
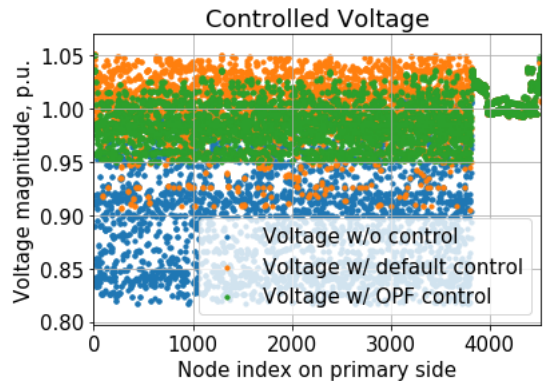
2) Voltage Regulation: We plot the regulated voltages obtained by Algorithm 1 with green dots in Fig. 5. Note that the voltage magnitudes of all the nodes are strictly constrained within the $[0.95, 1.05]$ p.u. bound. In contrast, the default control of regulators and capacitors cannot guarantee that all the voltages are within this bound.

V. CONCLUSION

We proposed a hierarchical distributed implementation of the primal-dual gradient algorithm to solve an OPF problem. The objective of the OPF is to minimize the total cost over all the controllable DERs and a cost associated with the total network load, subject to voltage regulation constraints. The proposed implementation is scalable to large distribution feeders comprising networked AGs. It largely reduces the computational burden compared to the centrally coordinated primal-dual algorithm by utilizing the information structure of the AGs. The performance of our design is analytically characterized and numerically corroborated. The significant improvement in convergence speed shows the great potential of the proposed method for grid optimization and control in real time.

REFERENCES

[1] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *IEEE Trans. on Automatic Control*, vol. 58, no. 11, pp. 2818–2833, 2013.

6

[2] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Trans. on Smart Grid*, vol. 9, no. 2, pp. 942–952, 2018.

[3] X. Zhou, E. Dall'Anese, L. Chen, and A. Simonetto, "An incentive-based online optimization framework for distribution grids," *IEEE Trans. on Automatic Control*, vol. 63, no. 7, pp. 2019–2031, 2018.

[4] X. Zhou, E. Dall'Anese, and L. Chen, "Online stochastic optimization of networked distributed energy resources," *arXiv preprint arXiv:1711.09953*, 2017.

[5] Q. Peng and S. H. Low, "Distributed optimal power flow algorithm for radial networks, I: Balanced single phase case," *IEEE Trans. on Smart Grid*, vol. 9, no. 1, pp. 111–121, 2018.

[6] S. Magnusson, C. Fischione, and N. Li, "Voltage control using limited communication," *arXiv preprint arXiv:1704.00749*, 2017.

[7] P. Šulc, S. Backhaus, and M. Chertkov, "Optimal distributed control of reactive power via the alternating direction method of multipliers," *IEEE Trans. on Energy Conversion*, vol. 29, no. 4, pp. 968–977, 2014.

[8] M. Bazrafshan and N. Gatsis, "Decentralized stochastic optimal power flow in radial networks with distributed generation," *IEEE Trans. on Smart Grid*, vol. 8, no. 2, pp. 787–801, 2017.

[9] C. Wu, G. Hug, and S. Kar, "Smart inverter for voltage regulation: Physical and market implementation," *IEEE Trans. on Power Systems*, 2018.

[10] B. Kroposki, E. Dall'Anese, A. Bernstein, Y. Zhang, and B.-M. Hodge, "Autonomous energy grids," *Proc. of Hawaii International Conference on System Sciences*, 2018.

[11] F. Dörfler, J. W. Simpson-Porco, and F. Bullo, "Breaking the hierarchy: Distributed control and economic optimality in microgrids," *IEEE Trans. on Control of Network Systems*, vol. 3, no. 3, pp. 241–253, 2016.

[12] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Voltage stabilization in microgrids via quadratic droop control," *IEEE Trans. on Automatic Control*, vol. 62, no. 3, pp. 1239–1253, 2017.

[13] E. Dall'Anese, S. S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople, "Optimal regulation of virtual power plants," *IEEE Trans. on Power Systems*, vol. 33, no. 2, pp. 1868–1881, 2018.

[14] A. Singhal and V. Ajjarapu, "A framework to utilize ders' var resources to support the grid in an integrated td system," *arXiv preprint arXiv:1712.07268*, 2017.

[15] Z. Wang, B. Chen, J. Wang, M. M. Begovic, and C. Chen, "Coordinated energy management of networked microgrids in distribution systems," *IEEE Trans. on Smart Grid*, vol. 6, no. 1, pp. 45–53, 2015.

[16] Z. Wang, B. Chen, J. Wang, and C. Chen, "Networked microgrids for self-healing power systems," *IEEE Trans. on Smart Grid*, vol. 7, no. 1, pp. 310–319, 2016.

[17] M. Fathi and H. Bevrani, "Adaptive energy consumption scheduling for connected microgrids under demand uncertainty," *IEEE Trans. on Power Delivery*, vol. 28, no. 3, pp. 1576–1583, 2013.

[18] K. Utkarsh, D. Srinivasan, A. Trivedi, W. Zhang, and T. Reindl, "Distributed model-predictive real-time optimal operation of a network of smart microgrids," *IEEE Trans. on Smart Grid*, 2018.

[19] W. Shi, X. Xie, C.-C. P. Chu, and R. Gadh, "Distributed optimal energy management in microgrids." *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1137–1146, 2015.

[20] H. Wang and J. Huang, "Incentivizing energy trading for interconnected microgrids," *IEEE Trans. on Smart Grid*, vol. 9, no. 4, pp. 2647–2657, 2018.

[21] D. Gregoratti and J. Matamoros, "Distributed energy trading: The multiple-microgrid case," *IEEE Trans. on Industrial Electronics*, vol. 62, no. 4, pp. 2551–2559, 2015.

[22] L. Zhang, Z. Li, and C. Wu, "Randomized auction design for electricity markets between grids and microgrids," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1, pp. 99–110, 2014.

[23] Y. Wang, W. Saad, Z. Han, H. V. Poor, and T. Başar, "A game-theoretic approach to energy trading in the smart grid," *IEEE Trans. on Smart Grid*, vol. 5, no. 3, pp. 1439–1450, 2014.

[24] K. Zhang, W. Shi, H. Zhu, E. Dall'Anese, and T. Basar, "Dynamic power distribution system management with a locally connected communication network," *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[25] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. on Power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[26] ——, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. on Power Delivery*, vol. 4, no. 1, pp. 735–743, 1989.

[27] ——, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[28] X. Zhou, L. Chen, M. Farivar, Z. Liu, and S. H. Low, "Reverse and forward engineering of local voltage control in distribution networks," *arXiv preprint arXiv:1801.02015*, 2018.

[29] Z. Liu, S. You, X. Zhou, G. Ding, and L. Chen, "Signal-anticipating in local voltage control in distribution systems," *arXiv preprint arXiv:1811.09365*, 2018.

[30] J. Koshal, A. Nedić, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[31] R. Arritt and R. Dugan, "The IEEE 8500-node test feeder," *Electric Power Research Institute, Palo Alto, CA, USA*, 2010.

APPENDIX

A. Proof of Lemma 1

Proof: Let $f(\boldsymbol{y}) = \sum_{i \in \mathcal{N}} C_i(p_i, q_i) + C_0(P_0(\boldsymbol{p}, \boldsymbol{q}))$, and $\boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{y}) = \underline{\boldsymbol{\mu}}^\top(\underline{\boldsymbol{v}} - \boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q})) + \overline{\boldsymbol{\mu}}^\top(\boldsymbol{v}(\boldsymbol{p}, \boldsymbol{q}) - \overline{\boldsymbol{v}})$ for simplicity. Then $T(\boldsymbol{z})$ can be equivalently decomposed into the following operators:

$$T(\boldsymbol{z}) = \begin{bmatrix} \nabla_{\boldsymbol{y}} f(\boldsymbol{y}) \\ \nabla_{\boldsymbol{\mu}} \frac{\phi}{2}\|\boldsymbol{\mu}\|_2^2 \end{bmatrix} + \begin{bmatrix} \nabla_{\boldsymbol{y}} \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{y}) \\ -\nabla_{\boldsymbol{\mu}} \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{y}) \end{bmatrix}$$

$$= \begin{bmatrix} \nabla_{\boldsymbol{y}} f(\boldsymbol{y}) \\ \nabla_{\boldsymbol{\mu}} \frac{\phi}{2}\|\boldsymbol{\mu}\|_2^2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -R^\top & R^\top \\ 0 & 0 & -X^\top & X^\top \\ R & X & 0 & 0 \\ -R & -X & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \underline{\boldsymbol{\mu}} \\ \overline{\boldsymbol{\mu}} \end{bmatrix}$$

$$+ \text{ Constant.} \qquad (21)$$

One can verify that the first operator $\begin{bmatrix} \nabla_{\boldsymbol{y}} f(\boldsymbol{y}) \\ \nabla_{\boldsymbol{\mu}} \frac{\phi}{2}\|\boldsymbol{\mu}\|_2^2 \end{bmatrix}$ is strongly monotone since $f(\boldsymbol{y})$ and $\frac{\phi}{2}\|\boldsymbol{\mu}\|_2^2$ are strongly convex in $\boldsymbol{y}$ and $\boldsymbol{\mu}$, respectively. The second (linear) operator is monotone since

$$\begin{bmatrix} 0 & 0 & -R^\top & R^\top \\ 0 & 0 & -X^\top & X^\top \\ R & X & 0 & 0 \\ -R & -X & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -R^\top & R^\top \\ 0 & 0 & -X^\top & X^\top \\ R & X & 0 & 0 \\ -R & -X & 0 & 0 \end{bmatrix}^\top \geq 0.$$

Therefore, $T(\boldsymbol{z})$ is a strongly monotone operator as the result of combining a strongly monotone operator and a monotone operator. ∎

B. Proof of Theorem 3

Proof: Let $\Delta < 1$ be some positive constant. We have

$$\|\boldsymbol{z}(t+1) - \boldsymbol{z}^*\|_2^2$$
$$\leqslant \|\boldsymbol{z}(t) - \epsilon T(\boldsymbol{z}(t)) - \boldsymbol{z}^* + \epsilon T(\boldsymbol{z}^*)\|_2^2$$
$$= \|\boldsymbol{z}(t) - \boldsymbol{z}^*\|_2^2 + \|\epsilon T(\boldsymbol{z}(t)) - \epsilon T(\boldsymbol{z}^*)\|_2^2$$
$$\quad - 2\epsilon(\boldsymbol{z}(t) - \boldsymbol{z}^*)^\top(T(\boldsymbol{z}(t)) - T(\boldsymbol{z}^*))$$
$$\leqslant (1 + \epsilon^2 L^2 - 2\epsilon M)\|\boldsymbol{z}(t) - \boldsymbol{z}^*\|_2^2$$

where the first inequality comes from non-expansiveness of projection operator, and the second from (10) and (11). Notice that (10) and (11) together also guarantee that $M \leqslant L$ so that $1 + \epsilon^2 L^2 - 2\epsilon M$ is always nonnegative. Then based on the condition for Theorem 3 that $0 < \epsilon \leqslant \bar{\epsilon} < 2M/L^2$ one has $\|\boldsymbol{z}(t+1) - \boldsymbol{z}^*\|_2^2 \leqslant \Delta\|\boldsymbol{z}(t) - \boldsymbol{z}^*\|_2^2$ for some constant $0 < \Delta < 1$.

Therefore, (9) converges to the unique saddle point of (7) exponentially fast. ∎