Real-Time Optimal Power Flow

Yujie Tang, *Student Member, IEEE*, Krishnamurthy Dvijotham, and Steven Low, *Fellow, IEEE*

***Abstract*—Future power networks are expected to incorporate a large number of distributed energy resources, which introduce randomness and fluctuations as well as fast control capabilities. But traditional optimal power flow methods are only appropriate for applications that operate on a slow timescale. In this paper, we build on recent work to develop a real-time algorithm for AC optimal power flow, based on quasi-Newton methods. The algorithm uses second-order information to provide suboptimal solutions on a fast timescale, and can be shown to track the optimal power flow solution when the estimated second-order information is sufficiently accurate. We also give a specific implementation based on L-BFGS-B method, and show by simulation that the proposed algorithm has good performance and is computationally efficient.**

***Index Terms*—Optimal power flow, time-varying optimization, quasi-Newton method.**

I. Introduction

OPTIMAL power flow (OPF) is fundamental in power system operations, and there has been extensive research on OPF algorithms. But for most algorithms in the literature, one must wait until the iteration has converged to obtain a solution that can be applied to the network because the intermediate iterates typically do not satisfy power flow equations and are not implementable. This is appropriate for traditional power system applications that typically operate on a slow timescale. However, future power grids will incorporate a large number of distributed energy resources, such as distributed wind and solar generations, electric vehicles, storage devices, smart inverters and other power electronics. They introduce randomness and fluctuations, as well as fast control capabilities to the network. Traditional OPF approaches that work on a slow timescale will be inadequate for this situation, and we need real-time OPF algorithms that can respond quickly to network changes and maintain (sub)optimality.

There are a number of recent works that focus on OPF algorithms that work on a faster timescale. Reference [1] proposed a continuous-time approach based on gradient dynamics for loss minimization; [2] proposed a distributed feedback algorithm for optimal reactive power flow that exploits real-time measurements, based on dual ascent method; [3] developed a fast VAR controller and analyzed its stability; [4] proposed an online OPF algorithm for distribution networks based on projected gradient descent and showed its convergence to the global optimum under certain conditions; [5] designed a composable framework for real-time control of distribution networks based on projected gradient descent, and [6] considered resource agents with bounded disobedience and studied the convergence of projected gradient control under restricted conditions; [7] studied projected gradient descent method on Riemann manifolds in continuous time domain and applied it to the online OPF problem; [8] proposed a real-time feedback controller seeking OPF solutions using SDP relaxation and dual ascent method; [9] employed a linearized power flow model and the double-smoothing technique for the real-time control of distribution networks with time-varying loads and generation limits, and gave theoretical guarantees on the controller's capability of tracking time-varying OPF targets; [10] developed a model-free algorithm based on the extreme-seeking approach; [11]–[13] proposed real-time control schemes that combine frequency control with economic dispatch.

In this paper, we propose a real-time algorithm for time-varying OPF problems based on quasi-Newton methods. For each time step, we measure the values and constraints of the decision variables, use them as the initial point and compute a small correction to track the optimal solution. The computation of the correction is based on a single iteration of a quasi-Newton algorithm, where we estimate the Hessian of the cost function and solve a simple optimization problem with a quadratic objective. Compared to existing literature, the proposed method assumes a single phase nonlinear AC power flow model with arbitrary topology, and utilizes second-order information which is expected to better handle the nonlinearity of the OPF problem. We also prove that the proposed algorithm will have good tracking performance if the second-order information can be estimated with good accuracy. We give an implementation based on the L-BFGS-B method, and show by simulation that the proposed algorithm is computationally efficient and can track the optimal operations with good performance.

II. Model and Formulation

A. Network Model

We consider a power network with a topology represented by a connected undirected graph (N^+, E) where $N^+ := \{0\} \cup N$,

Manuscript received November 1, 2016; revised March 18, 2017 and May 6, 2017; accepted May 7, 2017. Date of publication May 16, 2017; date of current version October 19, 2017. This work was supported in part by the NSF under Grant CCF 1637598, Grant ECCS 1619352, and Grant CNS 1545096, in part by ARPA-E under Grant DE-AR0000699, and in part by DTRA under Grant HDTRA 1-15-1-0003. Paper no. TSG-01523-2016. *(Corresponding author: Yujie Tang.)*

Y. Tang is with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ytang2@caltech.edu).

K. Dvijotham is with Pacific Northwest National Laboratory, Richland, WA 99354 USA (e-mail: dvij@cs.washington.edu).

S. Low is with the Computing and Mathematical Sciences and the Electrical Engineering Departments, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: slow@caltech.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2017.2704922

$N = \{1, 2, \ldots, n\}$ and $E \subseteq N^+ \times N^+$. Bus 0 will be the slack bus, and the phase angle of its voltage will be the reference and taken as zero. In this paper we consider the situation where the network has time-varying loads and generations. The set of time instants is assumed to be a discrete set and is denoted by T.

Let $t \in T$ be an arbitrary time instant. For each bus $i \in N^+$, let $V_i(t)$ be the complex voltage phasor, and $p_i(t), q_i(t)$ be the net real and reactive power injections (generation minus load) at bus i at time t. The complex current phasor through line $(i, j) \in E$ will be denoted by $I_{ij}(t)$. The relation between $(V_i(t))_{i\in N^+}$, $(p_i(t))_{i\in N^+}$, $(q_i(t))_{i\in N^+}$ and $(I_{ij}(t))_{(i,j)\in E}$ is described by physical laws and usually can be written as a set of algebraic equations (when the network is in a steady state) or ordinary differential equations (when dynamics needs to be accounted for). For steady states, the power flow equations

$$\begin{aligned} p_i(t) + \mathrm{j}q_i(t) &= \sum_{j\in N^+} V_i(t)V_j^*(t)Y_{ij}^*, \\ I_{ij}(t) &= -Y_{ij}(V_i(t) - V_j(t)) \end{aligned} \tag{1}$$

are satisfied, where Y_{ij}'s are the entries of the admittance matrix $\mathbf{Y}$ of the network.

For each bus, there are physical constraints on how much power can be injected by the connected devices. We assume that they can be modeled by time-varying constraints

$$(p_i(t), q_i(t)) \in \mathcal{X}_i(t), \quad i \in N^+,$$

where each $\mathcal{X}_i(t)$ is a compact convex subset of $\mathbb{R}^2$ for every $t \in T$. For the slack bus, we assume that $\mathcal{X}_0(t)$ admits a box constraint given by

$$\mathcal{X}_0(t) = \left[\underline{p}_0(t), \overline{p}_0(t)\right] \times \left[\underline{q}_0(t), \overline{q}_0(t)\right].$$

Information about the capabilities of controllable devices, as well as the changes in loads and renewable generations, can be encoded in these time-dependent regions.

The voltages and currents also need to be bounded for operational reasons. We assume that they are given by

$$\begin{aligned} \underline{v}_i \le |V_i(t)|^2 \le \overline{v}_i, \quad i \in N^+, \\ \left|I_{ij}(t)\right|^2 \le \overline{\ell}_{ij}, \quad (i,j) \in E. \end{aligned}$$

For each bus i, we assume that a cost $C_i(p_i, q_i; t)$ will be incurred if some power (p_i, q_i) is injected into the network. The cost functions can be potentially time dependent. We also assume that they are all convex functions of (p_i, q_i) and are twice continuously differentiable.

Table I gives a list of the main notations used throughout the paper.

B. Problem Formulation

Our goal is to minimize the total cost $\sum_i C_i(p_i, q_i; t)$ under the physical and operational constraints for each time t by properly operating the controllable devices in the network. As we have mentioned before, traditional OPF algorithms can be used to find the optimal control strategy on a slow timescale (say every tens of minutes), but will not be suitable for real-time applications on future smart grids. For real-time operations, we need an algorithm that can track the time-varying loads and renewable generations on a faster timescale. We require that the real-time operation at time t should be close to an optimal solution provided by a traditional OPF algorithm with the current feasible region given by $\mathcal{X}_i(t)$, $i \in N^+$; we refer to them as suboptimal strategies.

Although loads and generations change with time, if the time intervals between each real-time updates are sufficiently small, the regions $\mathcal{X}_i(t)$ will not change too much as we proceed from time t to $t + 1$. As a result, in many situations the optimal control at time t is expected to be close to the optimal control at the previous time instant. The objective of the real-time algorithm is then to find a suboptimal strategy for each $t \in T$ for the time-varying OPF

$$\min_{\substack{p_i(t), q_i(t),\\ V_i(t)}} \quad \sum_{i\in N^+} C_i(p_i(t), q_i(t); t) \tag{2a}$$

$$\text{s.t.} \quad p_i(t) + \mathrm{j}q_i(t) = \sum_{j\in N^+} V_i(t)V_j^*(t)Y_{ij}^* \tag{2b}$$

$$(p_i(t), q_i(t)) \in \mathcal{X}_i(t), \quad i \in N^+ \tag{2c}$$

$$\underline{v}_i \le |V_i(t)|^2 \le \overline{v}_i, \quad i \in N^+ \tag{2d}$$

$$\left|Y_{ij}\big(V_i(t) - V_j(t)\big)\right|^2 \le \overline{\ell}_{ij}, \quad (i,j) \in E. \tag{2e}$$

given the current state of the network and the previous real-time operation.

III. A Real-Time OPF Algorithm

A. Algorithm

It can be seen that (2) is a typical OPF problem, where the nonconvex power flow equality constraint (2b) contributes a large portion of the difficulty in finding a solution. We note that the power flow equation is solved automatically by the power network itself, i.e., whenever a particular control is applied, the network itself will implicitly compute the state that satisfies (2b). Mathematically, this is to say that, the variables in (2b) can be partitioned into two sets, one consisting of controllable variables (such as power injections) and the other consisting of dependent variables (such as voltages), and there is a function mapping controllable variables to dependent variables such that they satisfy (2b). This is a consequence of the implicit function theorem. Now we assume that the controllable variables are given by

$$\mathbf{x}(t) := (V_0(t), p_{1:n}(t), q_{1:n}(t))$$

where $p_{1:n}(t)$ denotes $p_1(t), \ldots, p_n(t)$ and $V_0(t)$ is real and positive because $\angle V_0(t) \equiv 0$, and that the dependent variables are given by

$$(V_{1:n}(t), p_0(t), q_0(t)).$$

Then by assuming that the conditions of the implicit function theorem hold, we obtain a function $\mathcal{F} : \mathcal{U} \to \mathbb{C}^n \times \mathbb{R}^2$ where $\mathcal{U}$ is an open subset of $\mathbb{R}_+ \times \mathbb{R}^{2n}$ such that whenever

$$(V_{1:n}(t), p_0(t), q_0(t)) = \mathcal{F}(V_0(t), p_{1:n}(t), q_{1:n}(t))$$

TABLE I
LIST OF SOME NOTATIONS

$N^+ = \{0, \ldots, n\}$	The set of buses of the network
$N = N^+ \backslash \{0\}$	The set of buses excluding the slack bus
$\mathcal{T}$	The set of time instants
$V_i(t)$	The complex voltage phasor at bus i at time t
$p_i(t), q_i(t)$	The real and reactive power injection at bus i at time t
$I_{ij}(t)$	The complex current phasor from bus i to bus j at time t
Y_{ij}	The (i,j)'th entry of the network's admittance matrix
$\mathcal{X}_i(t)$	The feasible region of the real and reactive power injection at bus i at time t
$\underline{v}_i, \overline{v}_i$	The lower and upper bound on the squared voltage magnitude $\lvert V_i(t)\rvert^2$
$\overline{\ell}_{ij}$	The upper bound on the squared current magnitude $\lvert I_{ij}\rvert^2$
$C_i(p_i, q_i; t)$	The cost incurred by injecting (p_i, q_i) at bus i at time t
$\mathbf{x}(t)$	The vector of controllable variables $(V_0(t), p_1(t), \ldots, p_n(t), q_1(t), \ldots, q_n(t))$
$\mathcal{F}$	The power flow function that maps controllable variables to dependent variables satisfying power flow equations
f_t	The sum of the cost function and the penalty functions
$\mathbf{x}^0(t)$	The values of the controllable variables measured at the beginning of period t
$\mathbf{g}_t$	The gradient $\nabla f_t(\mathbf{x}^0(t))$
$\mathbf{B}_t$	A positive definite matrix that serves as an estimate of the Hessian of f_t
$\hat{\mathbf{x}}(t)$	The real-time operation computed by the real-time OPF algorithm; an approximate solution to (5)
$\tilde{\mathbf{x}}(t)$	The exact solution to (5)
$\mathbf{x}^*(t)$	A local optimal solution to (4)
$\lVert\cdot\rVert_{\mathbf{W}}$	The weighted Euclidean norm with respect to the positive definite matrix $\mathbf{W}$, $\lVert\mathbf{x}\rVert_{\mathbf{W}} = (\mathbf{x}^T\mathbf{W}\mathbf{x})^{1/2}$
$\mathcal{P}_S$	The projection operator onto the convex set S

for some $(V_0(t), p_{1:n}(t), q_{1:n}(t)) \in \mathcal{U}$, the variables $(p_{0:n}(t), q_{0:n}(t), V_{0:n}(t))$ will satisfy (2b). We denote the entries of the vector-valued function $\mathcal{F}$ by

$$\begin{aligned} V_i(t) &= V_i(V_0(t), p_{1:n}(t), q_{1:n}(t)), \quad i \in N \\ p_0(t) &= p_0(V_0(t), p_{1:n}(t), q_{1:n}(t)), \\ q_0(t) &= q_0(V_0(t), p_{1:n}(t), q_{1:n}(t)). \end{aligned} \tag{3}$$

The squared magnitude of voltages and currents are then given by

$$\begin{aligned} \lvert V_i(t)\rvert^2 &= \lvert V_i(\mathbf{x}(t))\rvert^2 =: v_i(\mathbf{x}(t)), \\ \lvert I_{ij}(t)\rvert^2 &= \lvert Y_{ij}\big(V_i(\mathbf{x}(t)) - V_j(\mathbf{x}(t))\big)\rvert^2 =: \ell_{ij}(\mathbf{x}(t)). \end{aligned}$$

The idea of partitioning variables into dependent and independent ones was proposed in the reduced gradient method [14] and has been used for solving optimal power flow problems [15]. In practice, it has been empirically observed that, given a "reasonable" set of injections, there is a unique solution that satisfies the operational constraints in most situations. While counterexamples do exist [16], [17], recent work [18]–[20] has identified conditions under which there is a unique power flow solution within a certain domain. This means that we can apply our control to the network and use it to solve the power flow equations for the dependent variables. These dependent variables are then measured to compute the next control. Such an approach will become feasible as utilities around the world are deploying more and more instrumentation that can provide real-time measurements.

Algorithm 1 The Real-Time OPF Algorithm

For each time t,

1) Measure the current controllable and dependent variables, as well as regions $\mathcal{X}_i(t)$ that encode information of loads and generations at time t. Denote the controllable variables at the beginning of period t by $\mathbf{x}^0(t)$.
2) Based on the measurement data, compute $f_t(\mathbf{x}^0(t))$ and
$$\mathbf{g}_t := \nabla f_t\Big(\mathbf{x}^0(t)\Big).$$
3) Solve the following problem (approximately)
$$\begin{aligned} \min_{\mathbf{x}(t)} \ & \mathbf{g}_t^T\Big(\mathbf{x}(t) - \mathbf{x}^0(t)\Big) \\ & + \frac{1}{2}\Big(\mathbf{x}(t) - \mathbf{x}^0(t)\Big)^T\mathbf{B}_t\Big(\mathbf{x}(t) - \mathbf{x}^0(t)\Big) && \text{(5a)} \\ \text{s.t.} \ & (p_i(t), q_i(t)) \in \mathcal{X}_i(t), \quad i \in N && \text{(5b)} \\ & \sqrt{\underline{v}_0} \le V_0(t) \le \sqrt{\overline{v}_0}. && \text{(5c)} \end{aligned}$$
Here the matrix $\mathbf{B}_t$ is a positive definite matrix which serves as an estimate of the Hessian of $f_t(\mathbf{x}(t))$. Denote the solution by $\hat{\mathbf{x}}(t)$.
4) Apply the operation given by $\hat{\mathbf{x}}(t)$ to the network in period t.
5) Update the Hessian estimate to get $\mathbf{B}_{t+1}$.
6) Wait until the next time step and go to Step 1.

By using (3), we avoid the explicit power flow equality constraint. For the nonlinear constraints on voltages $V_i(\mathbf{x}(t))$, $i \in N$, currents $I_{ij}(\mathbf{x}(t))$, and slack bus injections $p_0(\mathbf{x}(t))$ and $q_0(\mathbf{x}(t))$, we employ penalty functions to move them to the objective. Let

$$\begin{aligned} f_t(\mathbf{x}(t)) = & \sum_{i\in N} C_i(p_i(t), q_i(t); t) + C_0(p_0(\mathbf{x}(t)), q_0(\mathbf{x}(t)); t) \\ & + \sum_{i\in N} \mu_i\Big(\phi\big(v_i(\mathbf{x}(t)) - \overline{v}_i\big) + \phi\big(\underline{v}_i - v_i(\mathbf{x}(t))\big)\Big) \\ & + \sum_{(i,j)\in E} \mu_{ij}\phi\big(\ell_{ij}(\mathbf{x}(t)) - \overline{\ell}_{ij}\big) \\ & + \mu_{p_0}\Big(\phi\big(p_0(\mathbf{x}(t)) - \overline{p}_0(t)\big) + \phi\Big(\underline{p}_0(t) - p_0(\mathbf{x}(t))\Big)\Big) \\ & + \mu_{q_0}\Big(\phi\big(q_0(\mathbf{x}(t)) - \overline{q}_0(t)\big) + \phi\Big(\underline{q}_0(t) - q_0(\mathbf{x}(t))\Big)\Big), \end{aligned}$$

and we consider the following problem

$$\begin{aligned} \min_{\mathbf{x}(t)\in\mathcal{U}} \ & f_t(\mathbf{x}(t)) && \text{(4a)} \\ \text{s.t.} \ & (p_i(t), q_i(t)) \in \mathcal{X}_i(t), \quad i \in N && \text{(4b)} \\ & \sqrt{\underline{v}_0} \le V_0(t) \le \sqrt{\overline{v}_0}. && \text{(4c)} \end{aligned}$$

The function $\phi(x)$ is assumed to be continuously differentiable for $x \in \mathbb{R}$ and equal to zero when $x \le 0$. A typical choice is $\phi(x) = \max\{x, 0\}^\kappa$ for some $\kappa > 1$.

The real-time OPF algorithm we propose is given in Algorithm 1. This real-time OPF algorithm is inspired

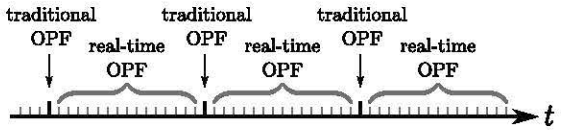


Fig. 1. Illustration of the hierarchical OPF strategy.

by various quasi-Newton methods for numerical optimization [21]. If the loads, generations and the cost functions do not change with time, then the procedure for a single time instant is very similar to a single iteration of a quasi-Newton algorithm that solves the nonlinear program (4). In our situation, the loads, generations and cost functions are changing slowly compared with the time granularity of real-time updates. Then the solution provided by a single iteration of a quasi-Newton algorithm is expected to be a good approximation of the true optimal solution.

We employ the more complicated quasi-Newton methods rather than the projected gradient method because the cost functions $C_i(p_i, q_i; t)$ and the penalty functions can have quite different curvatures (second-order derivatives). On the other hand, we do not restrict the methods of estimating the Hessian as long as $\mathbf{B}_t$ is positive definite, and the projected gradient method can be viewed as a special case of this general algorithm if we let $\mathbf{B}_t = \eta_t \mathbf{I}$ for some $\eta_t > 0$.

The proposed real-time OPF algorithm relies on the assumption that small changes in loads and generations will result in small changes in the optimal operation. However, OPF problems can have multiple local minima. As time changes, these multiple local minima form multiple trajectories, and the global minimum can jump from one trajectory to another. It is also possible that at some time a new trajectory appears or an existing trajectory disappears. In view of these possibilities, we assume that traditional OPF is solved iteratively until convergence in the background on a slower timescale. This "resets" our real-time OPF process periodically with a possibly new initial point. In other words,

- use traditional OPF to solve for the optimal operation on a slow timescale;
- between traditional OPF updates, use the real-time OPF algorithm that can track the time-varying loads and renewable generations on a faster timescale.

Figure 1 gives an illustration of this hierarchical strategy. This paper focuses only on the real-time OPF between "resets".

B. Tracking Performance

In this part we give a preliminary theoretical analysis of the real-time OPF.

Before proceeding, we should first clarify the relationship between $\hat{\mathbf{x}}(t-1)$, the real-time operation at time $t-1$, and $\mathbf{x}^0(t)$, the measurement taken at the beginning of period t. In general, these two quantities can be different due to, for example, the following reasons:

1) For a controllable device, there might be some discrepancy between the control command it receives and its actual power injection. The cause of this discrepancy could be, for example, that the settling time of the device is larger than the interval between real-time updates, or that $\mathcal{X}_i(t)$ is only an approximation of the actual operating region. See [6], [8] for related discussions.
2) Since the feasible set of controllable variables is time-varying, the operation at time $t-1$, which satisfies the constraints specified by $\mathcal{X}_i(t-1)$, may not lie in the feasible set specified by $\mathcal{X}_i(t)$. As time goes from $t-1$ to t, the controllable devices must adjust their power injections so that the hard physical constraints will not be violated.

The details of how the controllable devices implement the received control commands and how they adjust their power injections due to changes in physical conditions are out of the scope of this paper. Here we employ the following simplified model:

$$-(\mathbf{e} + \sigma\,\boldsymbol{\Delta}(t)) \le \hat{\mathbf{x}}(t-1) - \mathbf{x}^0(t) \le \mathbf{e} + \sigma\,\boldsymbol{\Delta}(t), \tag{6}$$

where $\mathbf{e} \in \mathbb{R}^{2n+1}_+$ is a given vector, $\sigma > 0$ is a given constant, and

$$\begin{aligned} \Delta_0(t) &= 0, \\ \Delta_i(t) &= \Delta_{i+n}(t) \\ &= \max_{(p,q)\in\mathcal{X}_i(t-1)} \left\| (p,q) - \mathcal{P}_{\mathcal{X}_i(t)}(p,q) \right\|_\infty, \quad i \in N. \end{aligned} \tag{7}$$

Basically, (6) says that the difference between $\hat{\mathbf{x}}(t-1)$ and $\mathbf{x}^0(t)$ is bounded by two terms. The vector $\mathbf{e}$ gives an upper bound on the discrepancies between the received control commands and the actual power injections. The vector $\boldsymbol{\Delta}(t)$ characterizes how much the feasible regions of injections $\mathcal{X}_i(t)$ change from $t-1$ to t, and $\sigma\,\boldsymbol{\Delta}(t)$ gives an upper bound on how much the controllable devices adjust their power injections due to changes in $\mathcal{X}_i(t)$.

Now, suppose we run the real-time OPF algorithm from time 0 to time T. Let $\mathbf{x}^*(0), \mathbf{x}^*(1), \ldots, \mathbf{x}^*(T)$ be an arbitrary sequence of local optimal solutions to (4), and denote the sequence produced by the real-time OPF algorithm by $\hat{\mathbf{x}}(0), \hat{\mathbf{x}}(1), \ldots, \hat{\mathbf{x}}(T)$. We assume that the traditional OPF resets the real-time OPF at time 0 so that $\hat{\mathbf{x}}(0) = \mathbf{x}^*(0)$. We also assume that $\hat{\mathbf{x}}(t)$ and $\mathbf{x}^0(t)$ are always inside $\mathcal{U}$. The sequence of approximate Hessians will be denoted by $\mathbf{B}_1, \ldots, \mathbf{B}_T$.

We are interested in the following measure of tracking performance

$$\frac{1}{T}\sum_{t=1}^{T} \left\| \hat{\mathbf{x}}(t) - \mathbf{x}^*(t) \right\|_{\mathbf{W}}, \tag{8}$$

where the norm $\|\cdot\|_{\mathbf{W}}$ is defined by $\|\mathbf{x}\|_{\mathbf{W}} := (\mathbf{x}^T\mathbf{W}\mathbf{x})^{1/2}$ and $\mathbf{W}$ is an arbitrary positive definite matrix. The quantity (8) is the average distance between the real-time operations $\hat{\mathbf{x}}(t)$ and the optimal solutions $\mathbf{x}^*(t)$. We introduce $\mathbf{W}$ to allow more freedom in how the entries of the vector are weighted when we calculate the norm. Since the first entry of $\mathbf{x}(t)$ represents the voltage of the slack bus, while other entries are real and reactive power injections, different weights should be employed to produce a sensible norm.

The core step of the real-time OPF algorithm is to solve the problem (5) approximately, and it is expected that the accuracy

of the solution will affect the tracking performance. Denote the exact solution to (5) by $\tilde{\mathbf{x}}(t)$. We assume that the difference between the approximate solution $\hat{\mathbf{x}}(t)$ and the exact solution $\tilde{\mathbf{x}}(t)$ is bounded in the following way

$$\left\|\hat{\mathbf{x}}(t)-\tilde{\mathbf{x}}(t)\right\|_{\mathbf{W}} \leq \eta \tag{9}$$

for some $\eta > 0$ for each t.

Theorem 1: Define

$$e_{\mathbf{W}} = \sqrt{\mathbf{e}^T|\mathbf{W}|\mathbf{e}}, \quad \Delta_{\mathbf{W}}(t) = \sqrt{\mathbf{\Delta}(t)^T|\mathbf{W}|\mathbf{\Delta}(t)},$$

where $|\mathbf{W}|$ denotes the matrix obtained by taking the absolute values of $\mathbf{W}$. Let

$$\lambda_M = \max_{1\leq t\leq T} \inf\{\lambda \in \mathbb{R} : \lambda\mathbf{W} \succeq \mathbf{B}_t\},$$
$$\lambda_m = \min_{1\leq t\leq T} \sup\{\lambda \in \mathbb{R} : \lambda\mathbf{W} \preceq \mathbf{B}_t\}.$$

If for every $t \in \{1,\ldots,T\}$, we have

$$\frac{\|\mathbf{B}_t^{-1}\left(\nabla f_t(\mathbf{x}^*(t)) - \nabla f_t(\mathbf{x}^0(t))\right) - \left(\mathbf{x}^*(t)-\mathbf{x}^0(t)\right)\|_{\mathbf{B}_t}}{\|\mathbf{x}^*(t)-\mathbf{x}^0(t)\|_{\mathbf{B}_t}} \leq \epsilon < \sqrt{\frac{\lambda_m}{\lambda_M}} \tag{10}$$

for some $\epsilon > 0$, then

$$\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{w}} \leq \frac{\epsilon}{\sqrt{\lambda_m/\lambda_M}-\epsilon} \times \frac{1}{T}\sum_{t=1}^{T}\left(\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{w}} + \sigma\Delta_{\mathbf{w}}(t) + e_{\mathbf{w}}\right) + \frac{\eta}{\sqrt{\lambda_m/\lambda_M}-\epsilon}. \tag{11}$$

The proof will be postponed to the Appendix.

The condition (10) roughly means that the approximate Hessian $\mathbf{B}_t$ approximates the true Hessian $\nabla^2 f_t$ along the direction $\mathbf{x}^*(t)-\mathbf{x}^0(t)$ with precision given by ϵ. Indeed, when ϵ is small, we have

$$\mathbf{B}_t^{-1}\left(\nabla f_t\left(\mathbf{x}^*(t)\right) - \nabla f_t\left(\mathbf{x}^0(t)\right)\right) \approx \mathbf{x}^*(t)-\mathbf{x}^0(t),$$

or

$$\nabla f_t\left(\mathbf{x}^*(t)\right) - \nabla f_t\left(\mathbf{x}^0(t)\right) \approx \mathbf{B}_t\left(\mathbf{x}^*(t)-\mathbf{x}^0(t)\right),$$

showing that $\mathbf{B}_t$ gives a good approximation of the curvature information of f_t along $\mathbf{x}^*(t)-\mathbf{x}^0(t)$. Under such conditions, if we temporarily assume that (5) is solved exactly so that $\eta = 0$, then Theorem 1 shows that, the average distance between $\hat{\mathbf{x}}(t)$ and $\mathbf{x}^*(t)$ can be bounded by three terms (multiplied by a factor):

1) The average rate of change of the optimal solutions given by $T^{-1}\sum_t \|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}$;
2) The average of $\sigma\Delta_{\mathbf{W}}(t)$, which characterizes how much the feasible injection regions change from time $t-1$ to time t;
3) The term $e_{\mathbf{W}}$ that describes the discrepancy between the received control commands and the actual power injections for controllable devices.

The first two terms together characterize how fast the OPF problems drift over time, while the third term characterizes the inherent system errors. The multiplicative factor, $\epsilon/(\sqrt{\lambda_m/\lambda_M}-\epsilon)$, decreases as ϵ decreases, which means that the more accurately $\mathbf{B}_t$ approximates the curvature of f_t, the smaller the factor is.

When (5) is solved approximately but (9) is satisfied, there is an additional term on the right-hand side of (11) that characterizes the effect of inaccurate solutions.

Although its conditions cannot be conveniently checked, Theorem 1 can serve as a guide to algorithm design. One implication of Theorem 1 is that, in order for the real-time operations $\hat{\mathbf{x}}(t)$ to track the optimal solutions $\mathbf{x}^*(t)$, the method of producing $\mathbf{B}_t$ should be carefully chosen. Theorem 1 also suggests that approaches based on second-order methods in general have better tracking performance than approaches that only use first-order information, which gives an analytical motivation for the proposed algorithm. In addition, since $\mathbf{B}_t$ is calculated at time $t-1$, the cost functions $C_i(p_i, q_i; t)$ should not change too fast so that $\mathbf{B}_t$ remains a good estimate of the Hessian $\nabla^2 f_t$ at time t.

IV. An Implementation of the Real-Time OPF Algorithm

In Section III we propose a general real-time OPF algorithm inspired by quasi-Newton methods. Implementing it requires solving the following problems:

1) The computation of the gradient $\mathbf{g}_t = \nabla f_t(\mathbf{x}^0(t))$.
2) The method for computing and updating $\mathbf{B}_t$.
3) How to ensure the solution of (5) is in $\mathcal{U}$.

Now we discuss these problems and give a detailed implementation of the real-time OPF based on the L-BFGS-B method.

A. Computation of the Gradient

In order to compute the gradient of f_t, we need to find quantities such as $\partial v_i/\partial x_k$, $\partial \ell_{ij}/\partial x_k$, $\partial p_0/\partial x_k$ and $\partial q_0/\partial x_k$[1] by applying the implicit function theorem on the AC power flow equations. This leads to inverting the Jacobian of the power flow equations. Although the Jacobian is in most cases sparse, its inverse is not and the computation will in general be very time consuming as the number of buses becomes large. Existing literature has proposed different methods for this purpose. Reference [22] employed rectangular representations of the power flow equations and reported improvement in efficiency over traditional methods; [2], [3], and [9] employed particular approximations or linearizations; [4] exploited the radial topology and proposed an iterative approach; [10] used a model-free method based on extreme seeking control.

On the other hand, we note that in practical situations, there will only be a small number of buses or lines whose voltages or currents violate their constraints at each time instant. Since the derivative of $\phi(x)$ is exactly zero when $x \leq 0$, we only need to find $\partial v_i/\partial x_k$ and $\partial \ell_{ij}/\partial x_k$ that correspond to voltages and

[1] In this subsection, we omit the time indices of variables temporarily.

currents that have violated their constraints. They will involve only a small fraction of the inverted Jacobian matrix, which could make the computation much faster.

To be more specific, we use the bus injection model as the AC power flow model and use the exponential form $V_i = |V_i|e^{\mathbf{i}\theta_i}$ to represent the complex voltages.[2] The power flow equations can be written in the form

$$p_0 = F_0^p(V_0, |V_{1:n}|, \theta_{1:n}),\ \ q_0 = F_0^q(V_0, |V_{1:n}|, \theta_{1:n}),$$
$$p_i = F_i^p(V_0, |V_{1:n}|, \theta_{1:n}),\ \ q_i = F_i^q(V_0, |V_{1:n}|, \theta_{1:n}),\ \ i \in N.$$

The partial derivatives of the functions F_0^p, F_0^q, F_i^p and F_i^q can be obtained once we know the values of the involved voltage phasors and their formula can be found, for example, in Newton-Raphson method for solving power flow. By viewing p_0,q_0 and $|V_i|$, θ_i, $i \in N$ as functions of V_0, $p_{1:n}$ and $q_{1:n}$ and taking the derivatives of the above equations, we get

$$\begin{bmatrix} \frac{\partial F_{1:n}^p}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial F_{1:n}^q}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial p_{1:n}} & \frac{\partial |V_{1:n}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} = I_{2n}, \tag{12}$$

$$\begin{bmatrix} \frac{\partial F_{1:n}^p}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial F_{1:n}^q}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial V_0} \\ \frac{\partial \theta_{1:n}}{\partial V_0} \end{bmatrix} = \begin{bmatrix} -\frac{\partial F_{1:n}^p}{\partial V_0} \\ -\frac{\partial F_{1:n}^q}{\partial V_0} \end{bmatrix}, \tag{13}$$

$$\begin{bmatrix} \frac{\partial p_0}{\partial p_{1:n}} & \frac{\partial p_0}{\partial q_{1:n}} \\ \frac{\partial q_0}{\partial p_{1:n}} & \frac{\partial q_0}{\partial q_{1:n}} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_0^p}{\partial |V_{1:n}|} & \frac{\partial F_0^p}{\partial \theta_{1:n}} \\ \frac{\partial F_0^q}{\partial |V_{1:n}|} & \frac{\partial F_0^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial p_{1:n}} & \frac{\partial |V_{1:n}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix}, \tag{14}$$

$$\begin{bmatrix} \frac{\partial p_0}{\partial V_0} \\ \frac{\partial q_0}{\partial V_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_0^p}{\partial V_0} \\ \frac{\partial F_0^q}{\partial V_0} \end{bmatrix} + \begin{bmatrix} \frac{\partial F_0^p}{\partial |V_{1:n}|} & \frac{\partial F_0^p}{\partial \theta_{1:n}} \\ \frac{\partial F_0^q}{\partial |V_{1:n}|} & \frac{\partial F_0^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial V_0} \\ \frac{\partial \theta_{1:n}}{\partial V_0} \end{bmatrix}, \tag{15}$$

where I_{2n} is the $2n \times 2n$ identity matrix, and we use the notation

$$\frac{\partial a_{1:n}}{\partial b} := \left[\frac{\partial a_i}{\partial b}\right]_{i=1,\ldots,n} \in \mathbb{R}^{n\times 1},$$
$$\frac{\partial a}{\partial b_{1:n}} := \left[\frac{\partial a}{\partial b_j}\right]_{j=1,\ldots,n} \in \mathbb{R}^{1\times n},$$
$$\frac{\partial a_{1:n}}{\partial b_{1:n}} := \left[\frac{\partial a_i}{\partial b_j}\right]_{\substack{i=1,\ldots,n \\ j=1,\ldots,n}} \in \mathbb{R}^{n\times n}.$$

First we note from (12) that

$$\begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial p_{1:n}} & \frac{\partial |V_{1:n}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial F_{1:n}^p}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial F_{1:n}^q}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} = I_{2n}.$$

[2]The following method can also be tailored when we use the rectangular representation for voltage phasors as in [22]. Whether one representation is advantageous over the other depends on the specific power network.

Now let

$$\mathcal{J} = \left\{ i \in N : |I_{ij}|^2 \geq \overline{\ell}_{ij} \text{ for some } j \in N^+ \right\}$$
$$\cup \{ i \in N : i \text{ is a neighbor of the slack bus} \},$$
$$\mathcal{I} = \mathcal{J} \cup \left\{ i \in N : |V_i|^2 \geq \overline{v}_i \text{ or } |V_i|^2 \leq \underline{v}_i \right\}.$$

Then we have

$$\begin{bmatrix} \frac{\partial |V_{\mathcal{I}}|}{\partial p_{1:n}} & \frac{\partial |V_{\mathcal{I}}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial F_{1:n}^p}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial F_{1:n}^q}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} = \begin{bmatrix} I_{\mathcal{I}} & 0 \\ 0 & I_{\mathcal{J}} \end{bmatrix}, \tag{16}$$

where $I_{\mathcal{I}}$ is the submatrix formed by the rows of I_n corresponding to $\mathcal{I}$, and similarly for $I_{\mathcal{J}}$. It can be seen that this is a set of $|\mathcal{I}| + |\mathcal{J}|$ linear systems with a common sparse coefficient matrix, and can be solved efficiently when $|\mathcal{I}| + |\mathcal{J}| \ll n$.

After we find $\partial |V_{\mathcal{I}}|/\partial p_{1:n}$, $\partial |V_{\mathcal{I}}|/\partial q_{1:n}$, $\partial \theta_{\mathcal{J}}/\partial p_{1:n}$ and $\partial \theta_{\mathcal{J}}/\partial q_{1:n}$, we use (13) to get

$$\begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial V_0} \\ \frac{\partial \theta_{1:n}}{\partial V_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_{1:n}^p}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial F_{1:n}^q}{\partial |V_{1:n}|} & \frac{\partial F_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial F_{1:n}^p}{\partial V_0} \\ -\frac{\partial F_{1:n}^q}{\partial V_0} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial |V_{1:n}|}{\partial p_{1:n}} & \frac{\partial |V_{1:n}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} -\frac{\partial F_{1:n}^p}{\partial V_0} \\ -\frac{\partial F_{1:n}^q}{\partial V_0} \end{bmatrix},$$

and so

$$\begin{bmatrix} \frac{\partial |V_{\mathcal{I}}|}{\partial V_0} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial V_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial |V_{\mathcal{I}}|}{\partial p_{1:n}} & \frac{\partial |V_{\mathcal{I}}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} -\frac{\partial F_{1:n}^p}{\partial V_0} \\ -\frac{\partial F_{1:n}^q}{\partial V_0} \end{bmatrix}. \tag{17}$$

Then by (14) and (15), and noting that $\partial F_0^p/\partial |V_i| = \partial F_0^p/\partial \theta_i = \partial F_0^q/\partial |V_i| = \partial F_0^q/\partial \theta_i = 0$ if i is not a neighbor of the slack bus, we get

$$\begin{bmatrix} \frac{\partial p_0}{\partial p_{1:n}} & \frac{\partial p_0}{\partial q_{1:n}} \\ \frac{\partial q_0}{\partial p_{1:n}} & \frac{\partial q_0}{\partial q_{1:n}} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_0^p}{\partial |V_{\mathcal{I}}|} & \frac{\partial F_0^p}{\partial \theta_{\mathcal{J}}} \\ \frac{\partial F_0^q}{\partial |V_{\mathcal{I}}|} & \frac{\partial F_0^q}{\partial \theta_{\mathcal{J}}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{\mathcal{I}}|}{\partial p_{1:n}} & \frac{\partial |V_{\mathcal{I}}|}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix}, \tag{18}$$

and

$$\begin{bmatrix} \frac{\partial p_0}{\partial V_0} \\ \frac{\partial q_0}{\partial V_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_0^p}{\partial V_0} \\ \frac{\partial F_0^q}{\partial V_0} \end{bmatrix} + \begin{bmatrix} \frac{\partial F_0^p}{\partial |V_{\mathcal{I}}|} & \frac{\partial F_0^p}{\partial \theta_{\mathcal{J}}} \\ \frac{\partial F_0^q}{\partial |V_{\mathcal{I}}|} & \frac{\partial F_0^q}{\partial \theta_{\mathcal{J}}} \end{bmatrix} \begin{bmatrix} \frac{\partial |V_{\mathcal{I}}|}{\partial V_0} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial V_0} \end{bmatrix}, \tag{19}$$

By the definition of $\mathcal{I}$ and $\mathcal{J}$, we can see that (16), (17), (18) and (19) give all the partial derivatives for calculating $\partial v_i/\partial x_k$ and $\partial \ell_{ij}/\partial x_k$ that correspond to violated constraints as well as $\partial p_0/\partial x_k$ and $\partial q_0/\partial x_k$.

B. The L-BFGS-B Method

The approach of producing the approximate Hessian $\mathbf{B}_t$ is central in the real-time OPF algorithm we proposed. As Theorem 1 points out, the tracking performance is directly related to how well $\mathbf{B}_t$ approximates the true curvature. However, the Hessian of f_t is in general a dense $(2n+1) \times (2n+1)$ matrix, making (5) not very scalable as n increases. Another issue is that (5) involves constraints and we cannot directly use methods for unconstrained optimization.

One way to deal with scalability and constraints is to employ the L-BFGS-B method [23]. Each iteration of the L-BFGS-B method approximately solves a quadratic program of the form

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{g}_k^T(\mathbf{x}-\mathbf{x}_{k-1}) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_{k-1})^T\mathbf{B}_k(\mathbf{x}-\mathbf{x}_{k-1}) \\ \text{s.t.} \ & \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{aligned}$$

and stores the approximate Hessian $\mathbf{B}_k$ and does related calculations by limited-memory BFGS method; the matrix $\mathbf{B}_k$ produced by limited-memory BFGS method is guaranteed to be positive definite. Each L-BFGS-B iteration has very low computational cost as shown in [23] because of the compact representations of limited-memory matrices. If each $\mathcal{X}_i(t)$ can be represented or approximated by a rectangle in $\mathbb{R}^2$ as

$$\mathcal{X}_i(t) = \Big[\underline{p}_i(t), \overline{p}_i(t)\Big] \times \Big[\underline{q}_i(t), \overline{q}_i(t)\Big],$$

then we can almost directly apply this method to our real-time OPF.

We will not describe the L-BFGS-B iteration for real-time OPF in detail here, but only give its outline:

For each time t,

1) Measure the current controllable and dependent variables, as well as bounds $\underline{p}_i(t)$, $\overline{p}_i(t)$, $\underline{q}_i(t)$, $\overline{q}_i(t)$ that encode information of loads and generations at time t. Denote the controllable variables at the beginning of period t by $\mathbf{x}^0(t)$.
2) Based on the measurement data, compute $f_t(\mathbf{x}^0(t))$ and
$$\mathbf{g}_t := \nabla f_t\Big(\mathbf{x}^0(t)\Big).$$
3) Estimate the free variables of the box constraints (5b)-(5c). Let $\mathcal{X}(t)$ denote the feasible region given by (5b)-(5c). We introduce the piecewise linear path
$$\mathbf{y}[s] := \mathcal{P}_{\mathcal{X}(t)}\Big(\mathbf{x}^0(t) - s\,\mathbf{g}_t\Big), \quad s \ge 0,$$
and consider
$$\min_{s\ge 0} \quad m_t(\mathbf{y}[s])$$
where
$$\begin{aligned} m_t(\mathbf{x}) := \ & \mathbf{g}_t^T\Big(\mathbf{x}-\mathbf{x}^0(t)\Big) \\ & + \frac{1}{2}\Big(\mathbf{x}-\mathbf{x}^0(t)\Big)^T\mathbf{B}_t\Big(\mathbf{x}-\mathbf{x}^0(t)\Big). \end{aligned}$$
Here $m_t(\mathbf{y}[s])$ is a univariate piecewise quadratic function in s, and its minimum can be found by enumerating over each linear piece of $\mathbf{y}[s]$. Let the smallest local minimizer be s^c, and let $\mathbf{x}^c = \mathbf{y}[s^c]$ (called the generalized Cauchy point). The free variables will be estimated as those entries of $\mathbf{x}^c$ that strictly satisfy the box constraint.
4) Find
$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} m_t\big(\mathbf{x}^c + \mathbf{u}\big),$$
where $\mathbf{u}$ is restricted to the subspace spanned by the estimated free variables. This is an unconstrained quadratic problem and can be solved explicitly. Let
$$\mathbf{d} = \mathcal{P}_{\mathcal{X}(t)}\big(\mathbf{x}^c + \hat{\mathbf{u}}\big) - \mathbf{x}^0(t).$$
5) Use backtracking to find $\alpha \in (0, 1]$ such that $\mathbf{x}^0(t) + \alpha\mathbf{d} \in \mathcal{U}$ and f_t has sufficient decrease. Let $\hat{\mathbf{x}}(t) = \mathbf{x}^0(t) + \alpha\mathbf{d}$.
6) Evaluate the gradient $\nabla f_t(\hat{\mathbf{x}}(t))$ and update the correction pairs to obtain $\mathbf{B}_{t+1}$.
7) Wait until the next time step and go back to Step 1.

The details of computing $\mathbf{x}^c$, $\hat{\mathbf{u}}$ and $\mathbf{B}_{t+1}$ by limited-memory techniques can be found in [23] and [24]. It should be noted that, we add a backtracking step to ensure that $\hat{\mathbf{x}}(t) \in \mathcal{U}$.

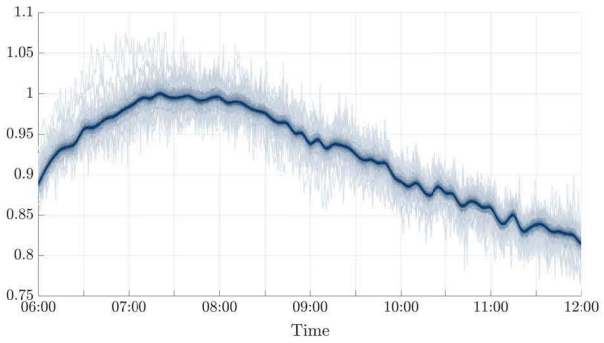


Fig. 2. Curves of $p(t) + \delta p_i(t)$ for all the load buses. Darker line color indicates larger load consumption level in the original IEEE 300 bus test case.

V. Numerical Simulation

We apply the real-time OPF based on L-BFGS-B algorithm to a modified version of the IEEE 300 bus test case with time-varying load profiles. We adopt the network topology, impedance, and generator data of IEEE 300 bus test case which are taken from MatPower [25]. The load profile is based on the total load data of Bonneville Power Administration on 02/08/2016 from 06:00am to 12:00pm [26]. The raw load data is on a 5-minute basis, and we interpolate the data to change the time granularity to 6 seconds and normalize it to have maximum value 1. We denote this base profile by $p(t)$. Then for each load bus i, we first add random noise $\delta p_i(t)$ to the base profile $p(t)$ to simulate fluctuations in the load consumptions; the amplitude of $\delta p_i(t)$ is proportional to the inverse square root of the original load of bus i in the IEEE 300 bus test case. We then multiply $p(t) + \delta p_i(t)$ by the original load of bus i in the IEEE 300 bus test case. The curves of $p(t) + \delta p_i(t)$ for the load buses are shown in Figure 2. We also let each load bus be connected with an inverter-like device that can inject controllable reactive power, whose capacity is 10% of the background real power consumption.

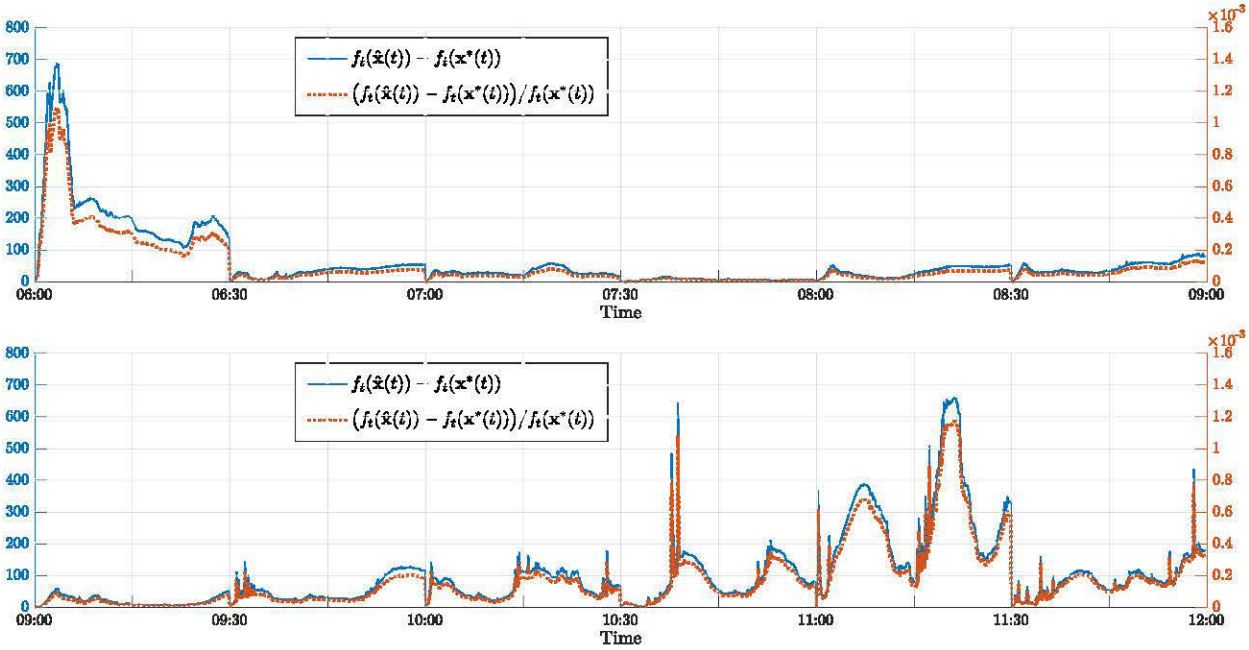


Fig. 3. The absolute and relative gap between the objective values of the real-time operations $\hat{\mathbf{x}}(t)$ and the optimal solutions $\mathbf{x}^*(t)$.

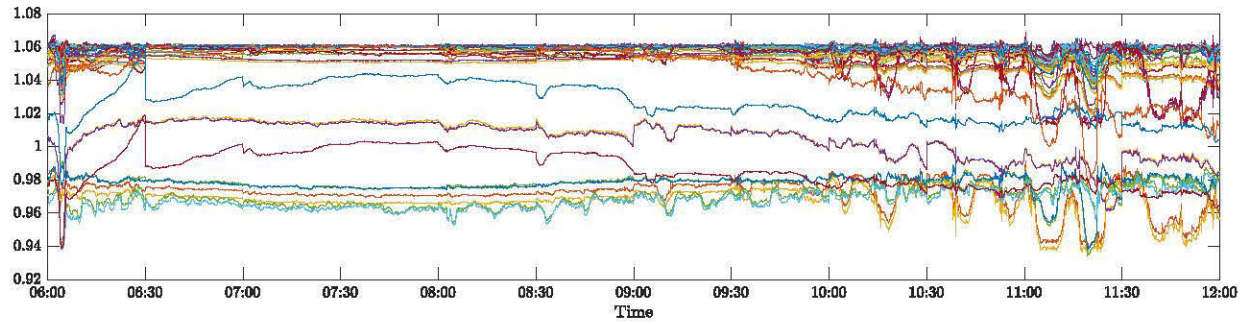


Fig. 4. Voltage profiles of the 43 buses whose voltages have ever violated the constraints $\underline{v}_i \le |V_i(t)|^2 \le \overline{v}_i$ for some t. The vertical axis represents voltage magnitude $|V_i(t)|$ in p.u.

The traditional OPF updates take place every 30 minutes starting from 06:00 am. Between each traditional OPF updates, the real-time OPF operates every 6 seconds. We choose $\phi(x) = \max\{0, x\}^{2.5}$, and the coefficients of the penalty functions are $\mu_i = 5 \times 10^6$ for voltage violations and $\mu_{p_0} = \mu_{q_0} = 10^6$ for constraints on $p_0(t)$ and $q_0(t)$. For the L-BFGS-B method, the number of correction pairs for estimating the Hessian is 12. For simplicity, we let $\mathbf{x}^0(t)$ and $\hat{\mathbf{x}}(t-1)$ be related by $\mathbf{x}^0(t) = \mathcal{P}_{\mathcal{X}(t)}\hat{\mathbf{x}}(t-1)$, which assumes that each controllable device will implement accurately the control command it receives, and changes its power injection to the closest feasible point as $\mathcal{X}_i(t)$ changes. The simulation is run in MATLAB on a Macbook Pro with 2.9GHz Intel i5 CPU and 16GB memory.

Figure 3 shows the absolute and relative gap between the objective values of the real-time operation $\hat{\mathbf{x}}(t)$ and the local optimal solution $\mathbf{x}^*(t)$ to (4), i.e.,

$$f_t\big(\hat{\mathbf{x}}(t)\big) - f_t\big(\mathbf{x}^*(t)\big) \quad \text{and} \quad \frac{f_t\big(\hat{\mathbf{x}}(t)\big) - f_t(\mathbf{x}^*(t))}{f_t(\mathbf{x}^*(t))}.$$

It can be seen that, the relative gap is less than 0.12% for the whole time horizon. In addition, the average relative gap is 0.0133%, while the average absolute gap is 81.38. By further investigating the simulation data, we find that most of the peaks in the gaps are due to penalties from newly violated voltage or slack bus generation constraints, and we can see that the violations can be quickly detected and managed by the algorithm.

Figure 4 shows the voltage profiles of the buses whose voltages have ever violated the constraints $\underline{v}_i \le |V_i(t)|^2 \le \overline{v}_i$ for some t. Here $\underline{v}_i = 0.94^2$ and $\overline{v}_i = 1.06^2$. It can be seen that there are 43 such buses and violations of voltage constraints are not rare, which is inevitable as we employ penalty functions to handle voltage constraints. On the other hand, most of the violations are small. The largest and smallest $|V_i(t)|$ over the whole time horizon are 1.069 and 0.934; moreover,

$$\operatorname*{average}_{i,t:|V_i(t)|>1.06} |V_i(t)| = 1.0607,$$
$$\operatorname*{average}_{i,t:|V_i(t)|<0.94} |V_i(t)| = 0.9384,$$

This suggests that, in practice, we can use slightly tighter or more conservative constraints for the dependent variables, so

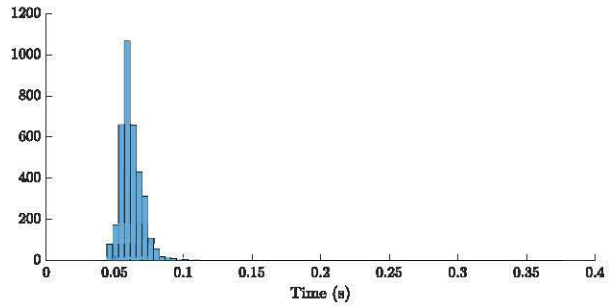


Fig. 5. Histogram of computation times of each real-time update.

that it is unlikely for the original constraints to be violated by the real-time operations $\hat{\mathbf{x}}(t)$.

We also measure the computation time of each real-time OPF update. Figure 5 shows the histogram of the computation times, with the average computation time being 0.062 sec and the maximum being 0.376 sec. We can see that the proposed implementation of the real-time OPF algorithm is quite computationally efficient.

VI. Conclusion and Future Directions

In this paper, we proposed a real-time OPF algorithm based on quasi-Newton methods. This algorithm utilizes real-time measurement data and performs suboptimal updates on a faster timescale than traditional OPF. We studied its tracking performance, and also proposed a specific implementation based on the L-BFGS-B algorithm. Simulations showed that the proposed algorithm can track the optimal operations well and is computationally efficient.

There still remain a number of issues in designing real-time OPF algorithms. Currently in the simulation, the updates are carried out every 6 seconds, and if we use a larger update interval, it is possible for the proposed implementation to lose track of the optimal setpoints for the test case. However, for a large transmission network, 6 second could be too short for handling the communication delays and the dynamics of power devices with large inertia. To extend the time between each updates, we need to improve the algorithm so that it will still work when larger changes in loads and generations occur.

One possible direction is to find more accurate methods of estimating the Hessian. In the proposed implementation, we use the L-BFGS-B method which turns out to work well in the simulation settings, but more accurate estimate of the Hessian may be needed to further improve the tracking performance.

Another possible direction is to introduce dual variables instead of penalty functions. It is possible that by introducing dual variables, one can achieve better convergence and smaller constraint violations, and potentially avoid numerical issues. We are especially interested in combining primal-dual methods with quasi-Newton methods.

Besides improving the tracking performance of the algorithm, we are also interested in developing a distributed algorithm for real-time OPF. As the network scales up, the communication between controllable devices and the control center will become a bottleneck, and distributed algorithms will be much favored. In addition, the feasible set of (5) can be decomposed as the Cartesian product of some low dimensional convex sets, each of which is associated with a bus in the network. It is expected that a distributed implementation can be developed by utilizing this particular structure.

Appendix
Proof of Theorem 1

We write the constraints (5b)-(5c) as $\mathbf{x}(t) \in \mathcal{X}(t)$, where $\mathcal{X}(t)$ is a compact convex subset of $\mathbb{R}^{2n+1}$. First we note that, by the definition of λ_M and λ_m, we have

$$\begin{aligned} \|\mathbf{x}\|_{\mathbf{B}_t}^2 &= \mathbf{x}^T\mathbf{B}_t\mathbf{x} \le \lambda_M \mathbf{x}^T\mathbf{W}\mathbf{x} = \lambda_M\|\mathbf{x}\|_{\mathbf{W}}^2, \\ \|\mathbf{x}\|_{\mathbf{B}_t}^2 &= \mathbf{x}^T\mathbf{B}_t\mathbf{x} \ge \lambda_m \mathbf{x}^T\mathbf{W}\mathbf{x} = \lambda_m\|\mathbf{x}\|_{\mathbf{W}}^2, \end{aligned} \tag{20}$$

for any vector $\mathbf{x}$ and any $t \in \{1, \ldots, T\}$.

At the beginning of time t, the initial point is $\mathbf{x}^0(t)$. Let

$$m_t(\mathbf{x}) := \mathbf{g}_t^T\Big(\mathbf{x} - \mathbf{x}^0(t)\Big) + \frac{1}{2}\Big(\mathbf{x} - \mathbf{x}^0(t)\Big)^T\mathbf{B}_t\Big(\mathbf{x} - \mathbf{x}^0(t)\Big).$$

Then

$$\tilde{\mathbf{x}}(t) = \arg\min_{\mathbf{x}\in\mathcal{X}(t)} m_t(\mathbf{x}),$$

which implies that

$$\begin{aligned} &\frac{d}{dh} m_t\big(\tilde{\mathbf{x}}(t) + h\big(\mathbf{x}^*(t) - \tilde{\mathbf{x}}(t)\big)\big)\Big|_{h=0} \ge 0 \\ \Longrightarrow\quad &\Big(\mathbf{B}_t\Big(\tilde{\mathbf{x}}(t) - \mathbf{x}^0(t)\Big) + \mathbf{g}_t\Big)^T\big(\mathbf{x}^*(t) - \tilde{\mathbf{x}}(t)\big) \ge 0, \end{aligned}$$

and by the first-order optimality condition of $\mathbf{x}^*(t)$, we have

$$\begin{aligned} &\frac{d}{dh} f_t\big(\mathbf{x}^*(t) + h\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big)\big)\Big|_{h=0} \ge 0 \\ \Longrightarrow\quad &\nabla f_t\big(\mathbf{x}^*(t)\big)^T\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big) \ge 0 \end{aligned}$$

Therefore

$$\begin{aligned} &\Big(\nabla f_t\big(\mathbf{x}^*(t)\big) - \nabla f_t\Big(\mathbf{x}^0(t)\Big)\Big)^T\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big) \\ &\quad\ge -\mathbf{g}_t^T\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big) \\ &\quad\ge \Big(\tilde{\mathbf{x}}(t) - \mathbf{x}^0(t)\Big)^T\mathbf{B}_t\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big) \\ &\quad= \big\|\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big\|_{\mathbf{B}_t}^2 + \Big(\mathbf{x}^*(t) - \mathbf{x}^0(t)\Big)^T\mathbf{B}_t\big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big), \end{aligned}$$

and so

$$\begin{aligned} &\big\|\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big\|_{\mathbf{B}_t}^2 \\ &\quad\le \big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big)^T\Big(\nabla f_t\big(\mathbf{x}^*(t)\big) - \nabla f_t\Big(\mathbf{x}^0(t)\Big) \\ &\qquad\qquad - \mathbf{B}_t\Big(\mathbf{x}^*(t) - \mathbf{x}^0(t)\Big)\Big) \\ &\quad= \big(\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big)^T\mathbf{B}_t\Big(\mathbf{B}_t^{-1}\Big(\nabla f_t\big(\mathbf{x}^*(t)\big) - \nabla f_t\Big(\mathbf{x}^0(t)\Big)\Big) \\ &\qquad\qquad - \Big(\mathbf{x}^*(t) - \mathbf{x}^0(t)\Big)\Big) \\ &\quad\le \big\|\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big\|_{\mathbf{B}_t} \\ &\qquad\times \Big\|\mathbf{B}_t^{-1}\Big(\nabla f_t\big(\mathbf{x}^*(t)\big) - \nabla f_t\Big(\mathbf{x}^0(t)\Big)\Big) - \Big(\mathbf{x}^*(t) - \mathbf{x}^0(t)\Big)\Big\|_{\mathbf{B}_t} \\ &\quad\le \epsilon\big\|\tilde{\mathbf{x}}(t) - \mathbf{x}^*(t)\big\|_{\mathbf{B}_t}\Big\|\mathbf{x}^*(t) - \mathbf{x}^0(t)\Big\|_{\mathbf{B}_t}, \end{aligned}$$

or

$$\left\|\tilde{\mathbf{x}}(t)-\mathbf{x}^*(t)\right\|_{\mathbf{B}_t}\le\epsilon\left\|\mathbf{x}^*(t)-\mathbf{x}^0(t)\right\|_{\mathbf{B}_t},$$

where we have used the assumption (10). Then we have

$$\begin{aligned}
&\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{B}_t}\\
&\le\|\hat{\mathbf{x}}(t)-\tilde{\mathbf{x}}(t)\|_{\mathbf{B}_t}+\|\tilde{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{B}_t}\\
&\le\epsilon\|\mathbf{x}^*(t)-\mathbf{x}^0(t)\|_{\mathbf{B}_t}+\|\hat{\mathbf{x}}(t)-\tilde{\mathbf{x}}(t)\|_{\mathbf{B}_t}\\
&\le\epsilon\Big(\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{B}_t}+\|\mathbf{x}^*(t-1)-\hat{\mathbf{x}}(t-1)\|_{\mathbf{B}_t}\\
&\quad+\|\hat{\mathbf{x}}(t-1)-\mathbf{x}^0(t)\|_{\mathbf{B}_t}\Big)+\|\hat{\mathbf{x}}(t)-\tilde{\mathbf{x}}(t)\|_{\mathbf{B}_t}\\
&\le\epsilon\sqrt{\lambda_M}(\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}+\|\hat{\mathbf{x}}(t-1)-\mathbf{x}^0(t)\|_{\mathbf{W}})\\
&\quad+\sqrt{\lambda_M}\|\hat{\mathbf{x}}(t)-\tilde{\mathbf{x}}(t)\|_{\mathbf{W}}+\epsilon\sqrt{\lambda_M}\|\mathbf{x}^*(t-1)-\tilde{\mathbf{x}}(t-1)\|_{\mathbf{W}}.
\end{aligned}$$

By (6),

$$\begin{aligned}
&\left\|\hat{\mathbf{x}}(t-1)-\mathbf{x}^0(t)\right\|_{\mathbf{W}}\\
&\qquad\le\max\{\|\mathbf{u}\|_{\mathbf{W}}:\ -(\mathbf{e}+\sigma\boldsymbol{\Delta}(t))\le\mathbf{u}\le\mathbf{e}+\sigma\boldsymbol{\Delta}(t)\}.
\end{aligned}$$

Since

$$\begin{aligned}
&\max\{\|\mathbf{u}\|_{\mathbf{W}}:\ -(\mathbf{e}+\sigma\boldsymbol{\Delta}(t))\le\mathbf{u}\le\mathbf{e}+\sigma\boldsymbol{\Delta}(t)\}\\
&\quad=\max\{\|\mathbf{u}_1+\mathbf{u}_2\|_{\mathbf{W}}:\ -\mathbf{e}\le\mathbf{u}_1\le\mathbf{e},\\
&\qquad\qquad-\sigma\boldsymbol{\Delta}(t)\le\mathbf{u}_2\le\sigma\boldsymbol{\Delta}(t)\}\\
&\quad\le\max\{\|\mathbf{u}\|_{\mathbf{W}}:\ -\mathbf{e}\le\mathbf{u}\le\mathbf{e}\}\\
&\qquad+\max\{\|\mathbf{u}\|_{\mathbf{W}}:\ -\sigma\boldsymbol{\Delta}(t)\le\mathbf{u}\le\sigma\boldsymbol{\Delta}(t)\},
\end{aligned}$$

and

$$\begin{aligned}
&\max\left\{\|\mathbf{u}\|_{\mathbf{W}}^2:-\mathbf{e}\le\mathbf{u}\le\mathbf{e}\right\}\\
&\qquad\le\max\left\{\sum_{ij}|W_{ij}||u_i||u_j|:\ -\mathbf{e}\le\mathbf{u}\le\mathbf{e}\right\}\\
&\qquad=\mathbf{e}^T|\mathbf{W}|\mathbf{e}=e_{\mathbf{W}}^2,
\end{aligned}$$

and similarly

$$\max\left\{\|\mathbf{u}\|_{\mathbf{W}}^2:\ -\sigma\boldsymbol{\Delta}(t)\le\mathbf{u}\le\sigma\boldsymbol{\Delta}(t)\right\}\le(\sigma\Delta_{\mathbf{W}}(t))^2,$$

we have

$$\begin{aligned}
\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{B}_t}\le\epsilon\sqrt{\lambda_M}\big(&\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}+e_{\mathbf{W}}\\
&+\sigma\Delta_{\mathbf{W}}(t)\big)+\sqrt{\lambda_M}\eta\\
&+\epsilon\sqrt{\lambda_M}\|\hat{\mathbf{x}}(t-1)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}.
\end{aligned}$$

By summing over t, and noting that

$$\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{B}_t}\ge\sqrt{\lambda_m}\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{W}}$$

and $\hat{\mathbf{x}}(0)=\mathbf{x}^*(0)$, we have

$$\begin{aligned}
&\sqrt{\lambda_m}\sum_{t=1}^{T}\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{W}}\\
&\quad\le\epsilon\sqrt{\lambda_M}\sum_{t=1}^{T}\big(\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}+e_{\mathbf{W}}+\sigma\Delta_{\mathbf{W}}(t)\big)\\
&\qquad+\sqrt{\lambda_M}T\eta+\epsilon\sqrt{\lambda_M}\sum_{t=1}^{T-1}\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{W}},
\end{aligned}$$

which leads to

$$\begin{aligned}
&\left(\sqrt{\lambda_m/\lambda_M}-\epsilon\right)\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{x}}(t)-\mathbf{x}^*(t)\|_{\mathbf{W}}\\
&\quad\le\frac{\epsilon}{T}\sum_{t=1}^{T}\big(\|\mathbf{x}^*(t)-\mathbf{x}^*(t-1)\|_{\mathbf{W}}+e_{\mathbf{W}}+\sigma\Delta_{\mathbf{W}}(t)\big)+\eta,
\end{aligned}$$

or equivalently, the bound (11).

REFERENCES

[1] X. Ma and N. Elia, "A distributed continuous-time gradient dynamics approach for the active power loss minimizations," in *Proc. 51st Allerton Conf. Commun. Control Comput. (Allerton)*, Monticello, IL, USA, 2013, pp. 100–106.

[2] S. Bolognani, G. Cavraro, and S. Zampieri, "A distributed feedback control approach to the optimal reactive power flow problem," in *Control of Cyber-Physical Systems*. Cham, Switzerland: Springer Int., 2013, pp. 259–277.

[3] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: Optimality and stability analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3794–3803, Sep. 2016.

[4] L. Gan and S. H. Low, "An online gradient algorithm for optimal power flow on radial networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 625–638, Mar. 2016.

[5] A. Bernstein, L. Reyes-Chamorro, J.-Y. Le Boudec, and M. Paolone, "A composable method for real-time control of active distribution networks with explicit power setpoints. Part I: Framework," *Elect. Power Syst. Res.*, vol. 125, pp. 254–264, Aug. 2015.

[6] A. Bernstein, N. J. Bouman, and J.-Y. L. Boudec. (Nov. 2015). *Design of Resource Agents With Guaranteed Tracking Properties for Real-Time Control of Electrical Grids*. [Online]. Available: https://arxiv.org/abs/1511.08628

[7] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Projected gradient descent on Riemannian manifolds with applications to online power system optimization," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, USA, 2016, pp. 225–232.

[8] E. Dall'Anese, S. V. Dhople, and G. B. Giannakis, "Photovoltaic inverter controllers seeking AC optimal power flow solutions," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2809–2823, Jul. 2016.

[9] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Trans. Smart Grid*, to be published, doi: 10.1109/TSG.2016.2571982.

[10] D. B. Arnold, M. Negrete-Pincetic, M. D. Sankur, D. M. Auslander, and D. S. Callaway, "Model-free optimal control of var resources in distribution systems: An extremum seeking approach," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3583–3593, Sep. 2016.

[11] N. Li, L. Chen, C. Zhao, and S. H. Low, "Connecting automatic generation control and economic dispatch from an optimization view," in *Proc. Amer. Control Conf.*, Portland, OR, USA, 2014, pp. 735–740.

[12] X. Zhang, N. Li, and A. Papachristodoulou, "Achieving real-time economic dispatch in power networks via a saddle point design approach," in *Proc. IEEE Power Energy Soc. Gener. Meeting*, Denver, CO, USA, 2015, pp. 1–5.

[13] C. Zhao, E. Mallada, S. Low, and J. Bialek, "A unified framework for frequency control and congestion management," in *Proc. Power Syst. Comput. Conf. (PSCC)*, Genoa, Italy, 2016, pp. 1–7.

[14] J. Abadie and J. Carpentier, "Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints," in *Optimization*. New York, NY, USA: Academic Press, 1969, pp. 37–47.

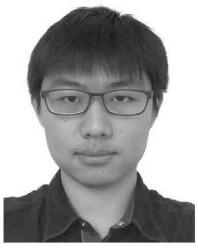
[15] R. Divi and H. K. Kesavan, "A shifted penalty function approach for optimal load-flow," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 9, pp. 3502–3512, Sep. 1982.

[16] A. J. Korsak, "On the question of uniqueness of stable load-flow solutions," *IEEE Trans. Power App. Syst.*, vol. PAS-91, no. 3, pp. 1093–1100, May 1972.

[17] A. Araposthatis, S. Sastry, and P. Varaiya, "Analysis of power-flow equation," *Int. J. Elect. Power Energy Syst.*, vol. 3, no. 3, pp. 115–126, 1981.

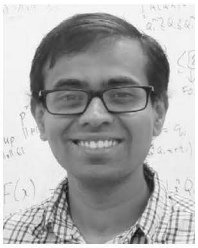
[18] K. Dvijotham, S. Low, and M. Chertkov. (Feb. 2015). *Convexity of Energy-Like Functions: Theoretical Results and Applications to Power System Operations*. [Online]. Available: https://arxiv.org/abs/1501.04052

[19] K. Dvijotham, S. Low, and M. Chertkov. (Jul. 2015). *Solving the Power Flow Equations: A Monotone Operator Approach*. [Online]. Available: https://arxiv.org/abs/1506.08472
[20] C. Wang, A. Bernstein, J.-Y. Le Boudec, and M. Paolone, "Explicit conditions on existence and uniqueness of load-flow solutions in distribution networks," *IEEE Trans. Smart Grid*, to be published, doi: 10.1109/TSG.2016.2572060.
[21] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.
[22] K. Christakou, J.-Y. LeBoudec, M. Paolone, and D.-C. Tomozei, "Efficient computation of sensitivity coefficients of node voltages and line currents in unbalanced radial electrical distribution networks," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 741–750, Jun. 2013.
[23] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.
[24] J. L. Morales and J. Nocedal, "Remark on 'Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization,'" *ACM Trans. Math. Softw.*, vol. 38, no. 1, 2011, Art. no. 7.
[25] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
[26] *BPA: Balancing Authority Load & Total Wind Generation*. Accessed on Oct. 21, 2016. [Online]. Available: https://transmission.bpa.gov/Business/Operations/Wind/



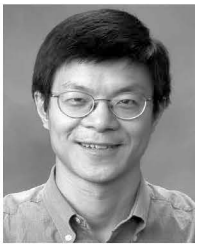
Yujie Tang (S'17) received the bachelor's degree in electronic engineering from Tsinghua University, Beijing, China, in 2013. He is currently pursuing the Ph.D. degree in electrical engineering with the California Institute of Technology, Pasadena, CA, USA.

His research interests include optimization and control of power systems, with a particular focus on real-time online operation of distributed energy resources in smart grids.



Krishnamurthy Dvijotham received the bachelor's degree from the Indian Institute of Technology Bombay, Mumbai, India, in 2008, and the Ph.D. degree in computer science and engineering from the University of Washington, Seattle, Washington, USA, in 2014.

He is a Researcher with the Optimization and Control Group, Pacific Northwest National Laboratory and a Research Assistant Professor with Washington State University, Pullman, WA, USA. His research interests center on algorithmic problems in large scale infrastructure networks and societal systems. His research has won awards at leading conferences in optimization, AI, and machine learning.



Steven Low (F'08) received the B.S. degree from Cornell University, Ithaca, NY, USA, in 1987, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 1992, both in electrical engineering.

He is a Professor of the Department of Computing and Mathematical Sciences and the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA, USA. He was with AT&T Bell Laboratories, Murray Hill, NJ, USA, and the University of Melbourne, Australia. His research on communication networks was accelerating more than 1TB of Internet traffic every second in 2014.

Prof. Low was a co-recipient of the IEEE Best Paper Awards. He was a member of the Networking and Information Technology Technical Advisory Group for the U.S. President's Council of Advisors on Science and Technology in 2006.