Optimal Power Flow: online algorithm, fast dynamics

Steven Low

August 2016
Solar power over land: > 20x world energy demand

network of billions of active distributed energy resources (DERs)

DER: PV, wind tb, EV, storage, smart bldg / appl
System dynamics and controls at different timescales
- require different models
Optimal power flow (OPF)

OPF is solved routinely for

- network control & optimization decisions
- market operations & pricing
- at timescales of mins, hours, days, ...

Non-convex and hard to solve

- Huge literature since 1962
- Common practice: DC power flow (LP)
- Also: Newton-Raphson, interior point, ...

\[
\min c(x) \quad \text{s. t. } \quad F(x) = 0, \quad x \in \bar{x}
\]
Optimal power flow (OPF)

OPF problem underlies numerous applications
• nonlinearity of power flow equations ➞ nonconvexity

Ian Hiskens, Michigan
How to deal with nonconvexity of power flows?

Two ideas

1. exact semidefinite relaxation

Tutorial:
L, Convex relaxation of OPF, 2014
http://netlab.caltech.edu
How to deal with **nonconvexity** of power flows?

Two ideas

1. exact semidefinite relaxation

1. use grid as implicit power flow solver
Key message

Large network of DERs
- Real-time optimization at scale
- Computational challenge: power flow solution

Online optimization [feedback control]
- Network computes power flow solutions in real time at scale for free
- Exploit it for our optimization/control
- Naturally adapts to evolving network conditions

Examples
- Slow timescale: OPF
- Fast timescale: frequency control
Outline

Semidefinite relaxations of OPF
- Power flow models
- Offline algorithms

Online OPF
- Power flow models

Load-side frequency control
- Dynamic models

Tutorial:
L, Convex relaxation of OPF, 2014
http://netlab.caltech.edu

Zhao, Topcu, Li, L, TAC 2014
Online OPF

Gan (FB)  Dvijotham (PNNL)  Tang
Relaxations of OPF

OPF: \[ \min_{x \in X} f(x) \]

relaxation: \[ \min_{\hat{x} \in X^+} f(\hat{x}) \]

But traditional algorithms are all offline …
… unsuitable for real-time optimization of network of distributed energy resources
Relaxations of OPF

OPF: \[ \min_{x \in X} f(x) \]

relaxation: \[ \min_{\hat{x} \in \hat{X}^+} f(\hat{x}) \]

We will compare our online algorithm to relaxation wrt optimality and speed.
OPF

\[
\begin{align*}
\text{min} & \quad c_0(y) + c(x) \\
\text{over} & \quad x, y \\
\text{s. t.} & \quad F(x, y) = 0 \\
& \quad y \leq \tilde{y} \\
& \quad x \in X := x_{\text{min}} \leq x \leq x_{\text{max}}
\end{align*}
\]

- controllable devices
- uncontrollable state
min $c_0(y) + c(x)$
over $x, y$
s. t. $F(x, y) = 0$

power flow equations
OPF

\[
\begin{align*}
\text{min} & \quad c_0(y) + c(x) \\
\text{over} & \quad x, y \\
\text{s. t.} & \quad F(x, y) = 0 & \text{power flow equations} \\
& \quad y \leq \bar{y} & \text{operational constraints} \\
& \quad x \in X := \{x, x, \bar{x}\} & \text{capacity limits}
\end{align*}
\]

Assume: \[\frac{F}{y} \geq 0\] \[y(x)\text{ over } X\]
OPF: eliminate $y$

$$\min_x c_0(y(x)) + c(x)$$

s. t. $y(x) \geq \bar{y}$

$x \in X := \{x, x, \bar{x}\}$
OPF: add barrier

$$\min_x c_0(y(x)) + c(x)$$

s. t.  \(y(x) \geq \bar{y}\)

\[x \in X := \{x, \bar{x}, x\}\]

add barrier function to remove operational constraints

$$\min_{x} L(x, y(x); \quad \)} \quad \text{\(L\): nonconvex}$$

over \(x \in X\)
Online (feedback) perspective

**Network:** power flow solver
\[ y(t) : F(x(t), y(t)) = 0 \]

**DER:** gradient update
\[ x(t+1) = G(x(t), y(t)) \]

control \[ x(t) \]

measurement, communication \[ y(t) \]

cyber network

physical network
Online gradient algorithm

\[
\begin{align*}
\min_{x \in X} & \quad L(x, y(x); m) \\
\text{over} & \quad x \in X
\end{align*}
\]

gradient projection algorithm:

\[
\begin{align*}
x(t+1) &= x(t) - h \frac{L(x, y(x); m)}{L_x(x, y(x); m)} \\
y(t) &= y(x(t))
\end{align*}
\]

- Explicitly exploits network as power flow solver
- Naturally tracks changing network conditions
Online gradient algorithm

\[
\begin{array}{c}
\text{min} & L(x, y(x)) \\
\text{over} & x \in X
\end{array}
\]

gradient projection algorithm:

\[
x(t+1) = x(t) - h \frac{L}{x(t)} \\
y(t) = y(x(t))
\]

Results
1. Optimality
2. Tracking performance
1. Local optimality

Under appropriate assumptions

- $x(t)$ converges to set of local optima
- if #local optima is finite, $x(t)$ converges
1. Global optimality

Assume: $p_0(x)$ convex over $X$

$v_k(x)$ concave over $X$

$$A := \{x \in X : v(x) \leq av + (1-a)v\}$$

**Theorem**

If co{local optima} are in $A$ then

- $x(t)$ converges to the set of global optima
- $x(t)$ itself converges a global optimum if #local optima is finite
1. Global optimality

Assume: \( p_0(x) \) convex over \( X \)
\[ v_k(x) \text{ concave over } X \]

\[ A := \{ x \in X : v(x) \leq a\bar{v} + (1-a)v \} \]

**Theorem**

- Can choose \( a \) s.t.
  \[ A \rightarrow \text{original feasible set} \]
- If SOCP is exact over \( X \), then assumption holds
1. Suboptimality gap

Informally, a local minimum is almost as good as any strictly interior feasible point

\[
\begin{align*}
L(x^*) & \quad L(\hat{x}) & \quad 0
\end{align*}
\]
2. Tracking performance

\[
\begin{align*}
\min_x & \quad c_0(y(x)) + c(x) \\
\text{s. t.} & \quad y(x) \leq \bar{y} \\
& \quad x \in X
\end{align*}
\]

\[
\begin{align*}
\min_x & \quad c_0(y(x), t) + c(x, t) \\
\text{s. t.} & \quad y(x, t) \leq \bar{y} \\
& \quad x \in X
\end{align*}
\]
2. Tracking performance

\[ R(x, x^*) := \sum_{t=1}^{T} c_0(y(x), t) + c(x, t) \]

dynamic regret

\[ c_0(y(x^*), t) + c(x^*, t) \]

optimal cost

\[ \text{cost of Alg} \]

\[ \text{rate of drifting subopt} \]
2. Tracking performance

Theorem

\[
R(x, x^*) := \sum_{t=1}^{T} \left( c_0(y(x), t) + c(x, t) \right) - \sum_{t=1}^{T} \left( c_0(y(x^*), t) + c(x^*, t) \right)
\]

cost of Alg

dynamic regret

\[
R(x, x^*) = O \sqrt{T} \left( 1 + \sum_{t=1}^{T} \left\| x_{t+1}^* - x_t \right\| \right) + \sum_{t=1}^{T} t
\]

rate of drifting

subopt of local min

optimal cost
2. Tracking performance

\[ R(x, x^*) := \sum_{t=1}^{T} c_0(y(x), t) + c(x, t) \]

dynamic regret

\[ R(x, x^*) := \sum_{t=1}^{T} c_0(y(x*), t) + c(x*, t) \]

optimal cost

**Theorem**

- If rate of drifting is \( o\left(\sqrt{T}\right) \) then per-step \( R(x, x^*) \) is asymptotically bounded by \( \bar{\delta} \) (local min)

- Can made \( \bar{\delta} \) arbitrarily small at cost of computation
## Simulations

<table>
<thead>
<tr>
<th># bus</th>
<th>CVX</th>
<th>IPM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>obj</td>
<td>time(s)</td>
<td>obj</td>
<td>time(s)</td>
</tr>
<tr>
<td>42</td>
<td>10.4585</td>
<td>6.5267</td>
<td>10.4585</td>
<td>0.2679</td>
</tr>
<tr>
<td>56</td>
<td>34.8989</td>
<td>7.1077</td>
<td>34.8989</td>
<td>0.3924</td>
</tr>
<tr>
<td>111</td>
<td>0.0751</td>
<td>11.3793</td>
<td>0.0751</td>
<td>0.8529</td>
</tr>
<tr>
<td>190</td>
<td>0.1394</td>
<td>20.2745</td>
<td>0.1394</td>
<td>1.9968</td>
</tr>
<tr>
<td>290</td>
<td>0.2817</td>
<td>23.8817</td>
<td>0.2817</td>
<td>4.3564</td>
</tr>
<tr>
<td>390</td>
<td>0.4292</td>
<td>29.8620</td>
<td>0.4292</td>
<td>2.9405</td>
</tr>
<tr>
<td>490</td>
<td>0.5526</td>
<td>36.3591</td>
<td>0.5526</td>
<td>3.0072</td>
</tr>
<tr>
<td>590</td>
<td>0.7035</td>
<td>43.6932</td>
<td>0.7035</td>
<td>4.4655</td>
</tr>
<tr>
<td>690</td>
<td>0.8546</td>
<td>51.9830</td>
<td>0.8546</td>
<td>3.2247</td>
</tr>
<tr>
<td>790</td>
<td>0.9975</td>
<td>62.3654</td>
<td>0.9975</td>
<td>2.6228</td>
</tr>
<tr>
<td>890</td>
<td>1.1685</td>
<td>67.7256</td>
<td>1.1685</td>
<td>2.0507</td>
</tr>
<tr>
<td>990</td>
<td>1.3930</td>
<td>74.8522</td>
<td>1.3930</td>
<td>2.7747</td>
</tr>
<tr>
<td>1091</td>
<td>1.5869</td>
<td>83.2236</td>
<td>1.5869</td>
<td>1.0869</td>
</tr>
<tr>
<td>1190</td>
<td>1.8123</td>
<td>92.4484</td>
<td>1.8123</td>
<td>1.2121</td>
</tr>
<tr>
<td>1290</td>
<td>2.0134</td>
<td>101.0380</td>
<td>2.0134</td>
<td>1.3525</td>
</tr>
<tr>
<td>1390</td>
<td>2.2007</td>
<td>111.0839</td>
<td>2.2007</td>
<td>1.4883</td>
</tr>
<tr>
<td>1490</td>
<td>2.4523</td>
<td>122.1819</td>
<td>2.4523</td>
<td>1.6372</td>
</tr>
<tr>
<td>1590</td>
<td>2.6477</td>
<td>157.8238</td>
<td>2.6477</td>
<td>1.8021</td>
</tr>
<tr>
<td>1690</td>
<td>2.8441</td>
<td>147.6862</td>
<td>2.8441</td>
<td>1.9166</td>
</tr>
<tr>
<td>1790</td>
<td>3.0495</td>
<td>152.6081</td>
<td>3.0495</td>
<td>2.0603</td>
</tr>
<tr>
<td>1890</td>
<td>3.8555</td>
<td>160.4689</td>
<td>3.8555</td>
<td>2.1963</td>
</tr>
<tr>
<td>1990</td>
<td>4.1424</td>
<td>171.8137</td>
<td>4.1424</td>
<td>2.3586</td>
</tr>
</tbody>
</table>
frequency control

Bialek (Skoltech)  Li (Harvard)  Mallada (JHU)  Topcu (Austin)  Zhao (NREL)
Motivation

- All buses synchronized to same nominal frequency (US: 60 Hz; Europe/China: 50 Hz)
- Supply-demand imbalance ➔ frequency fluctuation
Ubiquitous continuous load-side control can supplement generator-side control

- faster (no/low inertia)
- more reliable (large #)
- better localize disturbances
- reducing generator-side control capacity
How to design load-side frequency control?

How does it interact with generator-side control?
Literature: load-side control

Original idea & early analytical work
- Schwegge et al 1980; Bergin, Hill, Qu, Dorsey, Wang, Varaiya ...

Small scale trials around the world

Early simulation studies

Analytical work – load-side control

Recent analysis – generator-side/microgrid control:
Network model

(generator or load)

\( \hat{d}_i = D_i \)

loads:

damping or uncontrollable

\( i : \) region/control area/balancing authority
Network model

\[ \dot{\theta}_i = \omega_i \]

\[ M_i \dot{\omega}_i = -D_i \omega_i + p_i - \sum_e C_{ie} P_e \]

\[ P_{ij} = b_{ij} \sin(\theta_i - \theta_j) \quad \forall \ i \rightarrow j \]

Generator bus: \( M_i > 0 \)
Load bus: \( M_i = 0 \)
Network model

\[ \dot{\theta}_i = \omega_i \]

\[ M_i \dot{\omega}_i = -D_i \omega_i + \begin{bmatrix} p_i \end{bmatrix} - \sum_e C_{ie} P_e \]

\[ P_{ij} = b_{ij} \sin(\theta_i - \theta_j) \quad \forall \ i \rightarrow j \]

Generator bus: \( p_i \) is real power injection
Load bus: \( p_i \) is controllable load
Generator-side control

\[ \dot{\theta}_i = \omega_i \]

\[ M_i \dot{\omega}_i = -D_i \omega_i + P_i - \sum_{e} C_{ie} P_e \]

\[ P_{ij} = b_{ij} \sin(\theta_i - \theta_j) \quad \forall \, i \rightarrow j \]

generator bus:

primary control \( p^c_i(t) = p^c_i(\theta_i(t)) \)

e.g. freq droop \( p^c_i(\theta_i) = \frac{1}{\tau_{gi}}(a_i + p^c_i) \)

\[ \dot{p}_i = -\frac{1}{\tau_{bi}}(p_i + a_i) \]

\[ \dot{a}_i = -\frac{1}{\tau_{gi}}(a_i + p^c_i) \]
Load-side control

\[ \dot{\theta}_i = \omega_i \]

\[ M_i \dot{\omega}_i = -D_i \omega_i + p_i \sum_{e} C_{ie} P_e \]

\[ P_{ij} = b_{ij} \sin(\theta_i - \theta_j) \quad \forall \ i \to j \]

**Load bus:**

how to design feedback control ?
Suppose the system is in steady state

Then: disturbance in gen/load …
Control goals

- Rebalance power & stabilize frequency
- Restore nominal frequency
- Restore scheduled inter-area flows
- Respect line limits
Load-side controller design

\[ \dot{\theta}_i = \omega_i \]
\[ M_i \dot{\omega}_i = -D_i \omega_i + p_i - \sum_{e} C_{ie} P_e \]
\[ P_{ij} = b_{ij} \sin(\theta_i - \theta_j) \quad \forall \ i \rightarrow j \]

Control goals (while min disutility)

- Rebalance power & stabilize frequency
- Restore nominal frequency
- Restore scheduled inter-area flows
- Respect line limits

Zhao, Topcu, Li, Low
TAC 2014
Mallada, Zhao, Low
Allerton, 2014
Load-side controller design

Design control law whose equilibrium solves:

\[
\min_{d,P} c_i(d_i)
\]

s. t. \( P^m_i d_i = C_{ie} P_e \) node \( i \)

\( C_{ie} P_e = \hat{P}_k \) area \( k \)

\( P_e \quad P_e \quad \overline{P}_e \) line \( e \)

Control goals (while min disutility)

- Rebalance power & stabilize frequency
- Restore nominal frequency
- Restore scheduled inter-area flows
- Respect line limits

freq will emerge as Lagrange multiplier for power imbalance
Load-side controller design

Design control \((G, F)\) s.t. closed-loop system
- is asymptotically stable
- has equilibrium that is optimal

\[
\begin{align*}
\dot{\theta}_i &= \omega_i \\
M_i \ddot{\theta}_i &= -D_i \dot{\omega}_i + p_i - \sum_e C_{ie} P_e \\
P_{ij} &= b_{ij} \sin(\theta_i - \theta_j) \\
\dot{\lambda} &= G(\omega(t), P(t), \lambda(t)) \\
d_i &= F_i(\omega(t), P(t), \lambda(t))
\end{align*}
\]

\[
\begin{align*}
\min_{d, P_{ie}} \quad & c_i(d_i) \\
\text{s. t.} \quad & P_i^m \quad d_i = C_{ie} P_e \quad \text{node } i \\
& C_{ie} P_e = \hat{P}_k \quad \text{area } k \\
& P_e \quad P_e \quad \overline{P}_e \quad \text{line } e
\end{align*}
\]
Load-side controller design

Idea: exploit system dynamic as part of primal-dual algorithm for modified opt

- Distributed algorithm
- Control goals in equilibrium
- Stability analysis

\[
\begin{align*}
\dot{\theta}_i &= \omega_i \\
M_i \ddot{\omega}_i &= -D_i \omega_i + p_i - \sum_e C_{ie} P_e \\
P_{ij} &= b_{ij} \sin(\theta_i - \theta_j) \\
\dot{\lambda} &= G(\omega(t), P(t), \lambda(t)) \\
d_i &= F_i(\omega(t), P(t), \lambda(t)) \\
\min_{d,P} \quad & c_i(d_i) \\
\text{s. t.} \quad & P_i^m \quad d_i = C_{ie} P_e \quad \text{node } i \\
& \quad C_{ie} P_e = \hat{P}_k \quad \text{area } k \\
& \quad P_e \quad \bar{P}_e \quad \text{line } e
\end{align*}
\]
Primary load-side frequency control (linear PF)
• completely decentralized
• **Theorem**: globally stable dynamic, optimal equilibrium

Zhao, Topcu, Li, Low. TAC 2014
Summary: control architecture

**Secondary** load-side frequency control (linear PF)
- communication with neighbors
- **Theorem**: globally stable dynamic, optimal equilibrium

Mallada, Zhao, Low. Allerton 2014
Summary: control architecture

With generator-side control, nonlinear power flow
- Load-side improves both transient & eq
- Theorem: stable dynamic, optimal equilibrium

Zhao, Mallada, Low. CISS 2015
Simulations

Dynamic simulation of IEEE 39-bus system

- Power System Toolbox (RPI)
- Detailed generation model
- Exciter model, power system stabilizer model
- Nonzero resistance lines

Fig. 2: IEEE 39 bus system : New England
Primary control

![Graph showing frequency response over time for primary control systems.](graph.png)
Secondary control

Similar shape but local frequencies differ more, higher control effort, slightly longer settling time
Secondary control

Traditional AGC

Unified Control

AGC blind to line limits

Unified Control enforces line limits
Key message

Large network of DERs
- Real-time optimization at scale
- Computational challenge: power flow solution

Online optimization [feedback control]
- Network computes power flow solutions in real time at scale for free
- Exploit it for our optimization/control
- Naturally adapts to evolving network conditions

Examples
- Slow timescale: OPF
- Fast timescale: frequency control