Autonomous Energy Grid optimization

Steven Low

Caltech

NREL, September 2017
**Risk:** active DERs introduce rapid random fluctuations in supply, demand, power quality increasing risk of blackouts

**Opportunity:** active DERs enables realtime dynamic network-wide feedback control, improving robustness, security, efficiency

Caltech research: distributed control of networked DERs

- Foundational theory, practical algorithms, concrete applications
- Integrate engineering and economics
- Active collaboration with industry
Autonomous energy grid

Computational challenge
- nonlinear models, nonconvex optimization

Scalability challenge
- billions of intelligent DERs

Increased volatility
- in supply, demand, voltage, frequency

Limited sensing and control
- design of/constraint from cyber topology

Incomplete or unreliable data
- local state estimation & system identification

Data-driven modeling and control
- real-time at scale

many other important problems, inc. economic, regulatory, social, ...
Autonomous energy grid

Computational challenge
- nonlinear models, nonconvex optimization

Scalability challenge
- billions of intelligent DERs

Increased volatility
- in supply, demand, voltage, frequency

Limited sensing and control
- design of/constraint from cyber topology

Incomplete or unreliable data
- local state estimation & system identification

Data-driven modeling and control
- real-time at scale

a sample of our work for illustration
Outline

Relaxations of AC OPF
- Dealing with nonconvexity

Distributed AC OPF
- Dealing with scalability

Realtime AC OPF
- Dealing with volatility

Optimal placement
- Dealing with limited sensing/control
Relaxations of AC OPF
dealing with nonconvexity

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

Bose (UIUC)  Chandy  Farivar (Google)  Gan (FB)  Lavaei (UCB)  Li (Harvard)

many others at & outside Caltech …
Optimal power flow (OPF)

Computational challenge
- OPF underlies numerous power system applications but is nonconvex (and NP-hard)

Scalability challenge
- Future smart grid will have billions of intelligent distributed energy resources (DERs)

Our approach
- **Computation:** developed relaxation theory that exploits hidden convexity structure
- **Scalability:** developed distributed algorithms implementable by DERs based on relaxation
Optimal power flow (OPF)

\[
\min_{V \in \mathbb{C}^n} \text{tr}\left( CVV^H \right)
\]

s. t. \[ s_j \leq \text{tr}\left( Y_j^H VV^H \right) \leq \bar{s}_j \]

\[ v_j \leq |V_j|^2 \leq \bar{v}_j \]

\[ C, Y_j \in \mathbb{C}^{n \times n}, \ s_j, \bar{s}_j \in \mathbb{C}, \ v_j, \bar{v}_j \in \mathbb{R} \]

power flow equations: \[ s_j = \text{tr}\left( Y_j^H VV^H \right) \] for node \( j \)

- \( Y_j^H \) describes network topology and impedances
- \( s_j \) is net power injection (generation) at node \( j \)
- “power balance at each node \( j \)” (Kirchhoff’s law)
Optimal power flow (OPF)

\[
\min_{V \in \mathbb{C}^n} \text{tr}(CVV^H)
\]
\[
\text{s.t. } s_j \leq \text{tr}(Y_j^HVV^H) \leq \bar{s}_j
\]
\[
\underline{v}_j \leq |V_j|^2 \leq \bar{v}_j
\]

min generation cost, network loss

generation limits

voltage constraints

nonconvex feasible set

- \(Y_j^H\) not Hermitian (nor positive semidefinite)
- \(C\) is positive semidefinite (and Hermitian)

nonconvex QCQP

[Ian Hiskens]
Equivalent feasible sets

\[
\begin{align*}
\min & \quad \text{tr } CVV^H \\
\text{subject to} & \quad s_j \leq \text{tr} \left( Y_j^H VV^H \right) \leq \bar{s}_j, \quad v_j \leq |V_j|^2 \leq \bar{v}_j
\end{align*}
\]

Equivalent problem:

\[
\begin{align*}
\min & \quad \text{tr } CW \\
\text{subject to} & \quad s_j \leq \text{tr} \left( Y_j^H W \right) \leq \bar{s}_j, \quad v_j \leq W_{jj} \leq \bar{v}_j \\
W & \geq 0, \quad \text{rank } W = 1
\end{align*}
\]
Solution strategy

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

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

If optimal solution \( \hat{x}^* \) satisfies easily checkable conditions, then optimal solution \( x^* \) of OPF can be recovered.
Equivalent relaxations

\[ V \leftrightarrow W \leftrightarrow W^+ \leftrightarrow W_G^+ \]

**Theorem**

- Radial \( G \): SOCP is equivalent to SDP \((V \subseteq W^+ \equiv W_G^+)\)
- Mesh \( G \): SOCP is strictly coarser than SDP

For radial networks: always solve SOCP!
Exact relaxation

For **radial** networks, **sufficient** conditions on
- power injections bounds, or
- voltage upper bounds, or
- phase angle bounds
Exact relaxation

For radial networks, sufficient conditions on
- power injections bounds, or
- voltage upper bounds, or
- phase angle bounds
Exact relaxation

**QCQP** $(C, C_k)$

\[
\begin{align*}
\text{min} & \quad \text{tr}(Cxx^H) \\
\text{over} & \quad x \in \mathbb{C}^n \\
\text{s.t.} & \quad \text{tr}(C_kxx^H) \leq b_k \quad k \in K
\end{align*}
\]

**Graph of QCQP**

$G(C, C_k)$ has edge $(i, j) \iff C_{ij} \neq 0 \quad \text{or} \quad [C_k]_{ij} \neq 0 \quad \text{for some} \ k$

**QCQP over tree**

$G(C, C_k)$ is a tree
Exact relaxation

QCQP \((C, C_k)\)

\[
\begin{align*}
\text{min} & \quad \text{tr}\left( Cxx^H \right) \\
\text{over} & \quad x \in \mathbb{C}^n \\
\text{s.t.} & \quad \text{tr}\left( C_kxx^H \right) \leq b_k \quad k \in K
\end{align*}
\]

Key condition

\[i \sim j: \left( C_{ij}, [C_k]_{ij}, \forall k \right) \text{ lie on half-plane through 0}\]

Theorem

SOCP relaxation is exact for QCQP over tree

Bose et al 2012, 2014
Sojoudi, Lavaei 2013
Implication on OPF

Not both lower & upper bounds on real & reactive powers at both ends of a line can be finite
Example

- Relaxation is exact if $X$ and $Y$ have same Pareto front
- SOCP is faster but coarser than SDP
Potential benefits

<table>
<thead>
<tr>
<th>IEEE test systems</th>
<th>rank((X_0))</th>
<th>SDP cost</th>
<th>MATPOWER cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syst.</td>
<td>(J^\circ)</td>
<td>(\bar{J})</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>5296.7</td>
<td>5296.7</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>576.9</td>
<td>576.9</td>
</tr>
<tr>
<td>118</td>
<td>1</td>
<td>129661</td>
<td>129661</td>
</tr>
<tr>
<td>14A</td>
<td>1</td>
<td>8092.8</td>
<td>9093.8</td>
</tr>
</tbody>
</table>

12.4% lower cost than solution from nonlinear solver MATPOWER

[Louca, Seiler, Bitar 2013]
Potential benefits

Our research

- **Computation**: developed relaxation theory that exploits hidden convexity structure
- **Scalability**: developed distributed algorithms implementable by DERs based on relaxation theory
- **Benefits**: captures values to both utility and users

![baseline graph]

Peak load reduction: 8%
Energy cost reduction: 4%

![optimized graph]
Convex relaxations of OPF

**Distributed OPF**
- Kim, Baldick 1997
- Dall’Anese et al 2012
- Lam et al 2012
- Kraning et al 2013
- Devane, Lestas 2013
- Sun et al 2013
- Li et al 2013
- Peng, Low 2014

**Semidefinite relaxations**
- multiphase unbalanced
  - Dall’Anese et al 2012
  - Gan, Low 2014
- exactness
  - ext refs in tutorial: Low, TCNS 2014
  - http://netlab.caltech.edu
- moment/SoS, based relaxation
  - Phan 2012
  - Gopalakrishnan 2012
  - Louca et al 2013
  - Hijazi et al 2013
  - Andy Sun 2016

**Applications**
- B&B, rank min, QC relaxation,
Challenges

Challenges for practical application

- Relaxation may not be exact
  - Practical application demands a feasible solution
  - No known sufficient condition for exact relaxation for general mesh (transmission) networks
- Semidefinite relaxation (as is) is not scalable
Distributed AC OPF for scalability

Gan (FB)  Peng (Google)
Summary: 3 ideas

1. Solve semidefinite relaxation using branch-flow model (BFM)
   - BFM much more numerically stable
   - assume relaxation is exact (radial nk)

2. Decouple into operations at each bus
   - introduce decoupling variables and consensus constraints
   - message passing between neighboring buses

3. Apply ADMM
   - derive closed-form solution or 6x6 eigenvalue problem for each ADMM subproblem
   - greatly speeds up each ADMM iteration
Summary: simulations

<table>
<thead>
<tr>
<th>network</th>
<th>BIM-SDP</th>
<th></th>
<th></th>
<th>BFM-SDP</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>value</td>
<td>time</td>
<td>ratio</td>
<td>value</td>
<td>time</td>
<td>ratio</td>
</tr>
<tr>
<td>IEEE 13-bus</td>
<td>152.7</td>
<td>1.05</td>
<td>8.2e-9</td>
<td>152.7</td>
<td>0.74</td>
<td>2.8e-10</td>
</tr>
<tr>
<td>IEEE 34-bus</td>
<td>-100.0</td>
<td>2.22</td>
<td>1.0</td>
<td>279.0</td>
<td>1.64</td>
<td>3.3e-11</td>
</tr>
<tr>
<td>IEEE 37-bus</td>
<td>212.3</td>
<td>2.66</td>
<td>1.5e-8</td>
<td>212.2</td>
<td>1.95</td>
<td>1.3e-10</td>
</tr>
<tr>
<td>IEEE 123-bus</td>
<td>-8917</td>
<td>7.21</td>
<td>3.2e-2</td>
<td>229.8</td>
<td>8.86</td>
<td>0.6e-11</td>
</tr>
<tr>
<td>Rossi 2065-bus</td>
<td>-100.0</td>
<td>115.50</td>
<td>1.0</td>
<td>19.15</td>
<td>96.98</td>
<td>4.3e-8</td>
</tr>
</tbody>
</table>

BFM is much more **numerically stable**
SDP relaxations are **exact (wye loads)**

numerically unstable numerically stable

[Gan & Low 2014 PSCC]
Summary: comparison (single phase)

<table>
<thead>
<tr>
<th>Network size N</th>
<th>Total Time S</th>
<th>Avg time ( = S/N )</th>
<th>Centralized (CPU)</th>
<th>Centralized (elapsed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 123 buses</td>
<td>39.5 sec</td>
<td>0.32 sec</td>
<td>1.18 sec</td>
<td>11.4 sec</td>
</tr>
<tr>
<td>Rossi 2,065</td>
<td>1,153</td>
<td>0.56</td>
<td>14.38</td>
<td>157.3</td>
</tr>
<tr>
<td>1,313</td>
<td>471</td>
<td>0.36</td>
<td>8.88</td>
<td>91.2</td>
</tr>
<tr>
<td>792</td>
<td>226</td>
<td>0.29</td>
<td>5.13</td>
<td>50.3</td>
</tr>
<tr>
<td>363</td>
<td>66</td>
<td>0.18</td>
<td>3.08</td>
<td>24.5</td>
</tr>
<tr>
<td>108</td>
<td>16</td>
<td>0.14</td>
<td>0.78</td>
<td>6.5</td>
</tr>
</tbody>
</table>

- Parallel implementation of our distributed algorithm is much faster than solving OPF centrally

footnote: “Centralized” times reported by CVS in Matlab
- Solving SOCP using CVX (not ADMM)
- “CPU” time excludes problem set up before calling convex solver
- “elapsed” time includes setup time in CVX
Summary: simulations

Network (unbalanced)
■ IEEE 13, 34, 37, 123 bus systems

Objective
■ loss minimization

Convergence time (computation only)

<table>
<thead>
<tr>
<th>Network</th>
<th>Diameter</th>
<th>Iterations</th>
<th>Total Time</th>
<th>Avg Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 Bus</td>
<td>6</td>
<td>289</td>
<td>17.11</td>
<td>1.32</td>
</tr>
<tr>
<td>34 Bus</td>
<td>20</td>
<td>547</td>
<td>78.34</td>
<td>2.30</td>
</tr>
<tr>
<td>37 Bus</td>
<td>16</td>
<td>440</td>
<td>75.67</td>
<td>2.05</td>
</tr>
<tr>
<td>123 Bus</td>
<td>30</td>
<td>608</td>
<td>306.3</td>
<td>2.49</td>
</tr>
</tbody>
</table>
Details: 3 ideas

1. Solve semidefinite relaxation using branch-flow model (BFM)
   - BFM much more numerically stable
   - assume relaxation is exact (radial nk)

2. Decouple into operations at each bus
   - introduce decoupling variables and consensus constraints
   - message passing between neighboring buses

3. Apply ADMM
   - derive closed-form solution or 6x6 eigenvalue problem for each ADMM subproblem
   - greatly speeds up each ADMM iteration
BFM and relaxations

DistFlow model (Baran & Wu 1989)

\[ P_{i+1} = P_i - r_i \frac{P_i^2 + Q_i^2}{V_i^2} - P_{Li+1} \]

\[ Q_{i+1} = Q_i - x_i \frac{P_i^2 + Q_i^2}{V_i^2} - Q_{Li+1} \]

\[ V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) + (r_i^2 + x_i^2) \frac{P_i^2 + Q_i^2}{V_i^2} \]

nonconvex!

OPF

\[ \min_{x} f(x) \quad \text{subject to} \quad \text{DistFlow equations} \]

\[ \text{operation constraints} \quad g(x) \leq 0 \]

SOCR relaxation (Farivar & Low 2013)

- Equivalent re-formulation of DistFlow equations (linear + quadratic term)
- SOCP relaxation is often exact, yielding global optimal
- Much more numerically stable than bus injection model
BFM and relaxations

DistFlow model (Baran & Wu 1989)

\[ P_{i+1} = P_i - r_i \frac{P_i^2 + Q_i^2}{V_i^2} - P_{Li+1} \]

\[ Q_{i+1} = Q_i - x_i \frac{P_i^2 + Q_i^2}{V_i^2} - Q_{Li+1} \]

\[ V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) + (r_i^2 + x_i^2) \frac{P_i^2 + Q_i^2}{V_i^2} \]

But DistFlow model is single-phase!

How to generalize to 3-phase unbalanced system?
- Preserve simple analytical structure of 1-phase model
- Preserve superior numerical stability of 1-phase model
Multiphase generalization

DistFlow model for 1-phase

equivalent re-formulation

generalization to 3-phase

SOCP relaxation

SDP relaxation

distributed solution

distributed solution

radial, multiphase, wye + delta

Dall’Anese et al 2013 TSG
Gan & Low 2014 PSCC (above approach)

Zhao et al 2017 IREP

Peng & Low 2017 TSG
Peng & Low 2015 CDC
3phase model

3-phase balanced  (positive sequence)

\[
\begin{bmatrix}
I^a_{jk} \\
I^b_{jk} \\
I^c_{jk}
\end{bmatrix} =
\begin{bmatrix}
y^{aa}_{jk} & 0 & 0 \\
0 & y^{bb}_{jk} & 0 \\
0 & 0 & y^{cc}_{jk}
\end{bmatrix}
\begin{bmatrix}
V^a_j \\
V^b_j \\
V^c_j
\end{bmatrix}
- 
\begin{bmatrix}
V^a_k \\
V^b_k \\
V^c_k
\end{bmatrix}
\]

per-phase analysis

\[I^a_{jk} = y^{aa}_{jk} \left( V^a_j - V^a_k \right)\]

3-phase analysis

\[I_{jk} = y_{jk} \left( V_j - V_k \right)\]

3x3 matrix

3-phase unbalanced  (phase frame)

\[
\begin{bmatrix}
I^a_{jk} \\
I^b_{jk} \\
I^c_{jk}
\end{bmatrix} =
\begin{bmatrix}
y^{aa}_{jk} & y^{ab}_{jk} & y^{ac}_{jk} \\
y^{ba}_{jk} & y^{bb}_{jk} & y^{bc}_{jk} \\
y^{ca}_{jk} & y^{cb}_{jk} & y^{cc}_{jk}
\end{bmatrix}
\begin{bmatrix}
V^a_j \\
V^b_j \\
V^c_j
\end{bmatrix}
- 
\begin{bmatrix}
V^a_k \\
V^b_k \\
V^c_k
\end{bmatrix}
\]

3-phase analysis

\[I_{jk} = y_{jk} \left( V_j - V_k \right)\]
BFM: 3phase (wye)

Auxiliary variables:
\[
\begin{bmatrix}
  v_i & S_{ij} \\
  S_{ij}^H & \ell_{ij}
\end{bmatrix} \succeq 0, \quad \text{rank} \begin{bmatrix} v_i & S_{ij} \\
  S_{ij}^H & \ell_{ij} \end{bmatrix} = 1
\]

Ohm’s law:

Power balance:

\[
v_i = V_i V_i^H \quad \ell_{ij} = I_{ij} I_{ij}^H
\]

\[
S_{ij} = V_i I_{ij}^H
\]

3x3 rank-1 matrices
BFM: 3phase (wye)

Auxiliary variables:
\[
\begin{bmatrix}
v_i & S_{ij} \\
S_{ij}^H & \ell_{ij}
\end{bmatrix} \succeq 0, \quad \text{rank}\begin{bmatrix}
v_i & S_{ij} \\
S_{ij}^H & \ell_{ij}
\end{bmatrix} = 1
\]

6x6 matrix

Ohm’s law:
\[v_j = v_i - (S_{ij}z_{ij}^H + z_{ij}S_{ij}^H) + z_{ij}\ell_{ij}z_{ij}^H\]

3x3 matrices

Power balance:
\[
\sum_{k:k\rightarrow i} \text{diag} (S_{ki} - z_{ki}\ell_{ki}) = \sum_{j:i\rightarrow j} \text{diag} (S_{ij}) + s_{Y,i}
\]

3-vectors (a,b,c)
BFM: 3phase (wye)

Auxiliary variables:
\[
\begin{bmatrix}
  v_i & S_{ij} \\
  S_{ij}^H & \ell_{ij}
\end{bmatrix} \succeq 0, \quad \text{rank} \begin{bmatrix}
  v_i & S_{ij} \\
  S_{ij}^H & \ell_{ij}
\end{bmatrix} = 1
\]

Ohm’s law:
\[
v_j = v_i - (S_{ij} z_{ij}^H + z_{ij} S_{ij}^H) + z_{ij} \ell_{ij} z_{ij}^H
\]

Power balance:
\[
\sum_{k:k \rightarrow i} \text{diag} (S_{ki} - z_{ki} \ell_{ki}) = \sum_{j:i \rightarrow j} \text{diag} (S_{ij}) + s_{Y,i}
\]

equivalent to DistFlow equations if single-phase
OPF (3phase, wye)

\[
\begin{align*}
\min & \quad f(s_Y) \\
\text{over} & \quad (s_Y, v, l, S) \\
\text{s.t.} & \quad v_i \leq v_i \leq \bar{v}_i, \quad s_{Y,i} \in S_{Y,i}, \quad \forall i
\end{align*}
\]

\[
\begin{align*}
v_j &= v_i - (S_{ij} z_{ij}^H + z_{ij} S_{ij}^H) + z_{ij} l_{ij} z_{ij}^H, \quad \forall i \rightarrow j \\
\sum_{k:k \rightarrow i} \text{diag} \left(S_{ki} - z_{ki} l_{ki}\right) &= \sum_{j:i \rightarrow j} \text{diag} \left(S_{ij}\right) + s_{Y,i}, \quad \forall i
\end{align*}
\]

non-convex

branch flow model
SDP relaxation (3phase, wye)

\[
\begin{align*}
\text{min} \quad & f(s_Y) \\
\text{over} \quad & (s_Y, v, \ell, S) \\
\text{s.t.} \quad & v_i \leq v \leq \bar{v}_i, \quad s_{Y,i} \in S_{Y,i}, \quad \forall i \\
\end{align*}
\]

\[
\begin{bmatrix}
v_i & S_{ij} \\
S_{ij}^H & \ell_{ij}
\end{bmatrix} \succeq 0, \quad \text{rank} \left( \begin{bmatrix} v_i & S_{ij} \\ S_{ij}^H & \ell_{ij} \end{bmatrix} \right) = 1, \quad \forall i \to j
\]

branch flow model

6x6: semidefinite constraint

Gan, Low 2014 PSCC
Partition & decouple

\[
\min_{x,y} \sum_{i \in \mathcal{N}} f_i(x_{i0})
\]

s.t. \[
\sum_{j \in \mathcal{N}_i} A_{ij} y_{ji} = 0 \quad i \in \mathcal{N}
\]

\[
x_{i0} \in \mathcal{K}_{i0} \quad i \in \mathcal{N}
\]

\[
x_{i1} \in \mathcal{K}_{i1} \quad i \in \mathcal{N}
\]

\[
x_{i0} = y_{ij} \quad j \in \mathcal{N}_i \quad i \in \mathcal{N}
\]

\[
x_{i1} = y_{ii} \quad i \in \mathcal{N}
\]

\[x_i := (v_i, s_i, l_i A_i, S_i A_i)\]

\[y_{ji} : \text{ decoupling vars}\]

power balance & voltage eqtns

PSD & injection constraints

voltage magnitude constraints

consensus constraints

(co coupling across i)
ADMM

\[
\min_{x,y} f(x) + g(y) \\
\text{s.t. } x \in \mathcal{K}_x, \quad y \in \mathcal{K}_y \\
x = y
\]

\(\lambda\) : Lagrangian multiplier for coupling constraint

augmented Lagrangian:

\[L_\rho(x, y, \lambda) := f(x) + g(y) + \lambda^T(x - y) + \frac{\rho}{2}(x - y)^H \Lambda(x - y)\]

ADMM update at each iteration \(k\)

\[
x^{k+1} = \arg\min_{x \in \mathcal{K}_x} L_\rho(x, y^k, \lambda^k)
\]

\[
y^{k+1} = \arg\min_{y \in \mathcal{K}_y} L_\rho(x^{k+1}, y, \lambda^k)
\]

\[
\lambda^{k+1} = \lambda^k + \rho(x^{k+1} - y^{k+1})
\]

reduce min to
- QP: closed-form soln
- SDP: 6x6 eigenvalue problems

[Peng & L, 2016]
ADMM

Greatly speeds up each ADMM iteration

- much faster than standard iterative solution for each ADMM subproblem

<table>
<thead>
<tr>
<th>per-bus computation time</th>
<th>x-update</th>
<th>z-update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our algorithm</td>
<td>$1.7 \times 10^{-4}$ sec</td>
<td>$5.1 \times 10^{-4}$ sec</td>
</tr>
<tr>
<td>CVX</td>
<td>$2 \times 10^{-1}$ sec</td>
<td>$3 \times 10^{-1}$ sec</td>
</tr>
<tr>
<td>speedup</td>
<td>1,176x</td>
<td>588x</td>
</tr>
</tbody>
</table>

per-bus computation time: time to solve 1 sample ADMM iteration for Rossi circuit with 2,065 buses, divided by 2,065, for both algorithms (single-phase)
Challenges

Challenges for practical application

- **ADMM too slow for high precision solution**
- **Relaxation** (feasible power flow)
  - Wye loads: empirically exact but no proof
  - Delta loads: empirically inexact
- **Offline (distributed) algorithm**
  - Intermediate iterates are not feasible and cannot be applied to network

![Graphs showing residual and objective value evolution](Image)

(a) Primal and dual residual
(b) Objective value
Realtime AC OPF for tracking

Gan (FB)  Tang (Caltech)  Dvijotham (DeepMind)

See also: Dall’Anese et al, Bernstein et al, Hug & Dorfler et al, Callaway et al

Gan & L, JSAC 2016
Tang et al, TSG 2017
OPF

\[ \begin{align*} 
\text{min} \quad & c_0(y) + c(x) \\
\text{over} \quad & x, y \\
\text{s. t.} \quad & \text{operational constraints} \\
& \text{controllable devices} \\
& \text{uncontrollable state} 
\end{align*} \]
OPF

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

power flow equations
OPF

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

Assume: \( \frac{\partial F}{\partial y} \neq 0 \quad \Rightarrow \quad y(x) \quad \text{over} \quad X \)
Static OPF

\[
\begin{align*}
\text{min} & \quad f(x, y(x); \mu) \\
\text{over} & \quad x \in X
\end{align*}
\]

gradient projection algorithm:

\[
\begin{align*}
x(t + 1) &= \left[ x(t) - \eta \frac{\partial f}{\partial x}(t) \right]_x \\
y(t) &= y(x(t))
\end{align*}
\]

active control

law of physics

[Gan & Low, JSAC 2016]
Online (feedback) perspective

### DER: gradient update

\[ x(t+1) = G(x(t), y(t)) \]

#### Control
\[ x(t) \]

#### Measurement, Communication
\[ y(t) \]

### Network: power flow solver

\[ y(t) : F(x(t), y(t)) = 0 \]

- Explicitly exploits network as power flow solver
- Naturally tracks changing network conditions
Drifting OPF

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

static OPF

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

drifting OPF
Drifting OPF

\[
\min f_t(x, y(x); \mu_t)
\]
over \( x \in X_t \)

Quasi-Newton algorithm:

\[
x(t + 1) = \left[ x(t) - \eta(H(t))^{-1} \frac{\partial f_t}{\partial x}(x(t)) \right]_{x_t}
\]

\[
y(t) = y(x(t))
\]

[active control]
[law of physics]

[Tang, Dj & Low, 2017]
Tracking performance

error := \frac{1}{T} \sum_{t=1}^{T} \left\| x^{\text{online}}(t) - x^*(t) \right\|
Tracking performance

error := \frac{1}{T} \sum_{t=1}^{T} \| x^{\text{online}}(t) - x^*(t) \|

Theorem

error \leq \frac{\epsilon}{\sqrt{\lambda_M / \lambda_m} - \epsilon} \cdot \frac{1}{T} \sum_{t=1}^{T} \left( \| x^*(t) - x^*(t-1) \| + \Delta_t \right) + \delta

avg rate of drifting
• of optimal solution
• of feasible injections
Tracking performance

\[
\text{error} := \frac{1}{T} \sum_{t=1}^{T} \left\| x^{\text{online}}(t) - x^*(t) \right\|
\]

**Theorem**

\[
\text{error} \leq \frac{\varepsilon}{\sqrt{\lambda_M / \lambda_m}} \cdot \frac{1}{T} \sum_{t=1}^{T} \left( \left\| x^*(t) - x^*(t-1) \right\| + \Delta_t \right) + \delta
\]

error in Hessian approx
Tracking performance

error := \frac{1}{T} \sum_{t=1}^{T} \left\| x^{\text{online}}(t) - x^*(t) \right\|

\textbf{Theorem}

error \leq \frac{\varepsilon}{\sqrt{\lambda_M / \lambda_m}} \cdot \frac{1}{T} \sum_{t=1}^{T} \left( \left\| x^*(t) - x^*(t-1) \right\| + \Delta_t \right) + \delta

"condition number" of Hessian
Tracking performance

error \( := \) \( \frac{1}{T} \sum_{t=1}^{T} \| x^{\text{online}}(t) - x^*(t) \| \)

Theorem

error \( \leq \) \( \frac{\varepsilon}{\sqrt{\lambda_M / \lambda_m} - \varepsilon} \cdot \frac{1}{T} \sum_{t=1}^{T} \left( \| x^*(t) - x^*(t-1) \| + \Delta_t \right) + \delta \)

“initial distance” from \( x^*(t) \)
Implementation

Implement L-BFGS-B

- More scalable
- Handles (box) constraints $X$

Simulations

- IEEE 300 bus
Tracking performance

IEEE 300 bus
Fig. 3. The absolute and relative gap between the objective values of the real-time operations $\hat{x}(t)$ and the optimal solutions $x^*(t)$.

0.376 sec. We can see that the proposed implementation of the real-time OPF algorithm is quite computationally efficient.

Fig. 4. Histogram of computation times for each real-time update.

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 the updates are carried out every 6 seconds, which could be too short for us to neglect the dynamics for large networks. 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 are allowed.

One possible direction is to find more accurate methods of estimating the Hessian. The L-BFGS-B method turns out to work well as simulations have shown, but we have also found some difficult situations where more accurate estimate of the Hessian is needed.

Another possible direction is to introduce dual variables instead of penalty functions. It has been observed that by introducing dual variables, one can usually 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 number of controllable devices increases, the communication between controllable devices and the control center will become a bottleneck, and distributed algorithms will be much favored.

APPENDIX A PROOF OF THEOREM 1

We write the box constraint (5c)-(5e) as $l(t) \leq x(t) \leq u(t)$.

First we note that, by the definition of $M$ and $m$, we have $kx^2_B(t) = x^T B x = M x^T W x = M kx^2_W$, for any vector $x$ and any $t \in \{1, \ldots, T\}$.

At the beginning of time $t$, the initial point is $x_0(t) = P_t \hat{x}(t-1)$, where $P_t$ is the projection onto the current feasible control region $l(t) \leq x(t) \leq u(t)$, and $\hat{x}(t-1)$ is the previous operation. Let $m_t(x) = g_T x_0(t) + \frac{1}{2} (x - x_0(t))^T B x - x_0(t)^T W x$.

Then the updated operation $\hat{x}(t)$ is the optimal of $\min_{l(t) \leq x \leq u(t)} m_t(x)$. IEEE 300 bus
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
Challenges

Challenges for practical application

- Distributed implementation
- Tracking with lower update speed
- Not all buses have sensors/controllers
Optimal placement
dealing with limited sensing/control

Guo (Caltech)
Summary

Characterization of controllability and observability

- of swing dynamics
- in terms spectrum of graph Laplacian matrix

Implications on optimal placement of controllable DERs and sensors

- set covering problem
Network model

swing dynamics:

\[-M_j \dot{\omega}_j = 1_\mathcal{F}(j) \hat{d}_j + 1_\mathcal{U}(j) d_j - P_j^m + \sum_{e \in \mathcal{E}} C_{je} P_e\]

\[\dot{P}_{ij} = B_{ij} (\omega_i - \omega_j)\]

\[y_j = 1_\mathcal{S}(j) \omega_j\]

weighted Laplacian matrix

\[L = M^{-1/2} CBC^T M^{-1/2}\]
Algebraic coverage

spectral decomposition of $L$

$$L = Q\Lambda Q^T$$

eigenvectors of $L$

$$Q = [\beta_1 \cdots \beta_n]$$

algebraic coverage of bus $j$

$$\text{cov}(j) := \{ s \mid \beta_{sj} \neq 0 \}$$
Theorem
Swing dynamics is controllable if and only if

- $L$ has a simple spectrum holds a.s.
- controllable DERs have full coverage

\[ \bigcup_{j \in U} \text{cov}(j) = \{ \text{all buses} \} \]
Observability

**Theorem**

Swing dynamics is observable if and only if

- $L$ has a simple spectrum holds a.s.
- frequency sensors have full coverage

$$\bigcup_{j \in S} \text{cov}(j) = \{\text{all buses}\}$$
Application

Optimal placement of DER & frequency sensors

- set covering problem
- always install sensors at buses with controllable DERs, and vice versa
Application

IEEE 39-bus New England system

1 pu step disturbance at bus 30

Fig. 2. Line diagram of the IEEE 39-bus New England interconnection test system.
Outline

Relaxations of AC OPF
  - Dealing with nonconvexity

Distributed AC OPF
  - Dealing with scalability

Realtime AC OPF
  - Dealing with volatility

Optimal placement
  - Dealing with limited sensing/control