Log(v) 3LPF: A linearized solution to train reinforcement learning algorithms for distribution systems

Ignacio Losada Carreño, Anna Scaglione
SPADES Y1 workshop. December 2, 2020

Arizona State University
Reinforcement learning on distribution systems, PPO-Clip

Figure 1: Current PyCigar modeling diagram (left) vs proposed architecture (right)
Reinforcement learning on distribution systems, PPO-Clip

- Two neural networks. Weights obtained via SGD
  - Policy function: $\theta_{k+1} = \arg \min_{\theta} g(\hat{R}_t, s_{rl}, a_{rl}, \theta)$
  - Value function: $\phi_{k+1} = \arg \min_{\phi} h(\hat{R}_t, s_{rl}, a_{rl}, \phi)$
- Power Flow (PF) equations are used to compute the rewards
- Rewards are computed for every training iteration, every time step and every action sampled by the algorithm
- **Efficient** and **accurate** PF solvers are necessary

Our target: $\hat{R}_t = f(s_p^1, s_{rl}^2, a_{rl}^3)$

$^1s_p$: Power system state
$^2s_{rl}$: Reinforcement learning state
$^3a_{rl}$: Reinforcement learning action
Distribution systems are unbalanced

- Unbalanced system → 3-phase solvers
- Need for AC modeling vs DC
  - AC PF equations are non-linear
  - Non-linearity is caused by ZIP load models (details later)
- Modeling of line losses through shunt elements (not negligible) needed

PF formulations:

**Current:** \( i = Y_{bus}v \)

**Power:** \( s = D(vv^H Y_{bus}^H) \)

**Figure 2:** Pi-Model representation
Survey of ACPF solvers

- Commercial software PF solvers are **iterative**
  - Newton-Raphson (GridLab-D, PSLF)
  - Gauss-Seidel (OpenDSS, GridLab-D)
  - Forward-Backward Sweep (GridLab-D)

- Previous work, linear approximations
  - Lin3DistFlow [Sankur et al., 2016]. Nominal voltages and no losses. No ZIP models
  - NFA [Fobes et al., 2020]. Real-power only, no ZIP models
  - DCP [Fobes et al., 2020]. DC assumption, ignores reactive power, no ZIP models
  - Lossy Distflow [Schweitzer et al., 2019]. Not valid for complete ZIP models, cannot accommodate modeling of transformers, regulators, losses are parametrized.
  - **LPF** [Li et al., 2017]. Positive-sequence only, lossless, no ZIP models, doesn’t exploit tree structure (use case is transmission systems).
Our contribution

- Modeling capabilities
  - Embedded linear power flow solver
  - ZIP load models, shunt capacitors, regulators, transformers, smart inverters, batteries, and corresponding controls (as of today).
  - Ability to fully control and understand unbalanced 3-phase distribution systems. Implementation of new attack vectors
  - Linear PF and OPF. Applicable to transmission systems.

- Modeling accuracy
  - PF solution validated against OpenDSS
  - Modeling of devices matches that of OpenDSS

- Computational complexity
  - Linear equations allow to solve the system in a single snapshot. In OpenDSS, IEEE-13 11 iterations, IEEE-8500 62 iterations reducing RL training times
  - Ability to exploit graph tree structure
  - Ability to efficiently compute inverse after perturbation
  - Reduced overhead due to api calling external models
Log(v) 3LPF
Kirchhoff: $i_{nm}^{(n)} = i_{nm} + i_{s}^{n}$

Ohm: $i_{nm} = Y_{nm}^{(n)}v_{n} - Y_{nm}^{(m)}v_{m}$

Losses: $i_{s}^{n} = \frac{1}{2} Y_{nm}^{s}v_{n}$

$S_{nm}^{(n)} = v_{n}v_{n}^{H} \left( Y_{nm}^{(n)} + \frac{1}{2} Y_{nm}^{s} \right)^{H} - v_{n}v_{m}^{H} \left( Y_{nm}^{(m)} \right)^{H}$
Log($v$) 3LPF

\[
S^{(n)}_{nm} = v_n v_n^H \left( Y^{(n)}_{nm} + \frac{1}{2} Y^s_{nm} \right)^H - v_n v_m^H \left( Y^{(m)}_{nm} \right)^H
\]

We want to remove the non-linearity $v_n v_n^H$ from the equation that relates power flows to voltage

\[
v_n := [v_n^a, v_n^b, v_n^c]^T, \quad v_n^p = |v_n^p| e^{j\theta_p}
\]

\[
|v_n^p| = e^{\log |v_n^p|}, \quad u_n^p := \log |v_n^p|
\]

\[
v_n^p = e^{u_n^p} e^{j\theta_p}
\]
\[ S_{nm}^{(n)} = v_n v_n^H \left( Y_{nm}^{(n)} + \frac{1}{2} Y_{nm}^s \right)^H - v_n v_m^H \left( Y_{nm}^{(m)} \right)^H \]

We know voltages are around 1 p.u., thus \( \log(1) = 0 \), and we approximate the voltage magnitude in \( v_n^p = e^{u_n^p} e^{j\theta_p} \) using first-order Taylor expansion.

\[ v_n := \begin{pmatrix} e^{u_n^a} e^{j\theta_n^a} \\ e^{u_n^b} e^{j\theta_n^b + \frac{2\pi}{3} - \frac{2\pi}{3}} \\ e^{u_n^c} e^{j\theta_n^c - \frac{2\pi}{3} + \frac{2\pi}{3}} \end{pmatrix} = \Delta_3 \text{diag} \begin{pmatrix} e^{u_n^a} \\ e^{u_n^b} \\ e^{u_n^c} \end{pmatrix} \begin{pmatrix} e^{j\tilde{\theta}_n^a} \\ e^{j\tilde{\theta}_n^b} \\ e^{j\tilde{\theta}_n^c} \end{pmatrix} \]  

\[ v_n \approx \Delta_3 \left( I + \text{diag} \left( u_n \right) \right) e^{j\tilde{\theta}_n} \]  

\[ v_n v_n^H \approx \Delta_3 \left( 11^T + u_n 1^T + 1^T u_n^T + j\tilde{\theta}_n 1^T - j1\tilde{\theta}_n^T \right) \Delta_3^H \]  

\[ v_n v_m^H \approx \Delta_3 \left( 11^T + u_n 1^T + 1^T u_m^T + j\tilde{\theta}_n 1^T - j1\tilde{\theta}_m^T \right) \Delta_3^H \]
Non-linear ACPF: \[ S_{nm}^{(n)} = \mathbf{v}_n \mathbf{v}_n^H \left( \mathbf{Y}_{nm}^{(n)} + \frac{1}{2} \mathbf{Y}_{nm}^{s} \right)^H - \mathbf{v}_n \mathbf{v}_m^H \left( \mathbf{Y}_{nm}^{(m)} \right)^H \]

Reordering and defining the corresponding matrices and vectors

Log(\(\mathbf{v}\)) 3LPF (Linear):
\[ \tilde{s}_{nm} \approx \tilde{\mathbf{Y}}_{\text{bus}} \mathbf{x} \quad \text{where} \quad \mathbf{x} \triangleq \begin{bmatrix} \mathbf{u} \\ \tilde{\theta} \end{bmatrix} \quad (7) \]

\[ \mathbf{x} \approx \tilde{\mathbf{Y}}_{\text{bus}}^{-1} \tilde{s}_{nm} \]

and we may recover the voltage phasors as follows
\[ \mathbf{v} \approx \Delta_3 \text{diag} \left( e^{\mathbf{u}} \right) e^{j\tilde{\theta}_n} \]
ZIP models
ZIP models. Wye-connected loads

**Figure 3:** Wye (left) and delta-connected (right) load

Wye-connected loads: \( S_n^Y = S_n^{Z,Y} + S_n^{I,Y} + S_n^{P,Y} \)

\[
\begin{align*}
Z: & \quad S_n^{Z,Y} = (y_n^Y)^* + 2\text{diag}(y_n^Y)^* u_n \\
I: & \quad S_n^{I,Y} \approx \Delta_3 (i_n^Y)^* + \Delta_3 \text{diag} (i_n^Y)^* u_n + j\Delta_3 \text{diag} (i_n^Y)^* \tilde{\theta}_n \\
P: & \quad S_n^{P,Y} = s_n^Y
\end{align*}
\]
ZIP models. Delta-connected loads

Delta-connected loads: $S_n^\Delta = S_n^{Z,\Delta} + S_n^{I,\Delta} + S_n^{P,\Delta}$

$Z$: $S_n^{Z,\Delta} \approx \left( \text{diag} \left( \Delta_3 \left( \tilde{\gamma}_n^{\Delta} \right)^T 1 \right) + \Delta_3 \left( \tilde{\gamma}_n^{\Delta} \right)^T \right) u_n$

$+ \left( \text{diag} \left( \Delta_3 \left( \tilde{\gamma}_n^{\Delta} \right)^T 1 \right) - \Delta_3 \left( \tilde{\gamma}_n^{\Delta} \right)^T \right) \tilde{\theta}_n$

$I$: $S_n^{I,\Delta} \approx \Delta_3 \Lambda \left( i_n^{\gamma} \right)^* + \Delta_3 \text{diag} \left( \Lambda i_n^{\gamma} \right)^* u_n + j\Delta_3 \text{diag} \left( \Lambda i_n^{\gamma} \right)^* \tilde{\theta}_n$

$P$: $S_n^{P,\Delta} \approx \Lambda s_\ell^\Delta$

**Figure 4:** Wye (left) and delta-connected (right) load
Modeling and control of power delivery elements
Transformers and voltage regulators

Modeled through the **admittance matrix**

\[ S_{nm}^{(n)} = v_n v_n^H \left( Y_{nm}^{(n)} + \frac{1}{2} Y_s^{(n)} \right)^H - v_n v_m^H \left( Y_{nm}^{(m)} \right)^H \]

**Transformers:**

\[ Y_{prim} = ANB \left( Z_{nm}^t \right)^{-1} B^T N^T A^T \]

\[ Y_{prim} = \begin{pmatrix} Y_{nm}^{(n)} & Y_{nm}^{(m)} \\ Y_{mn}^{(n)} & Y_{mn}^{(m)} \end{pmatrix} \]

**Voltage regulators:**

\[ Y_{prim} = \Gamma ANB \left( Z_{nm}^t \right)^{-1} B^T N^T A^T \Gamma^T \]

where \( \Gamma = \text{diag}(\gamma) \) and \( \gamma_i = 1 \pm 0.00625\tau_i \)

\( \tau_i = f(v_{reg}, v_b, v_n) \)

**Attack vectors**

\( v_{reg} \) setpoint, \( v_b \) bandwidth, \( v_n \) measurement

**Figure 5:** WyeG-Delta Transformer
Shunt capacitors and cap controls

Modeled as a wye or delta-connected **constant impedance load**

$$S_n^{Z,Y} = D \left( \nu_n \nu_n^H \Pi (Y_n^Y)^H \Pi^T \right) \quad \text{or} \quad S_n^{Z,\Delta} = D \left( \nu_n \nu_n^H \Pi (Y_n^\Delta)^H \Pi^T \right)$$

**Capacitor banks:**

$$Y_n^Y = \text{diag} \left( y_n^Y \right), \quad Y_n^\Delta = \text{diag} \left( y_n^\Delta \right),$$

$$y_n^Y = \begin{bmatrix} y_{n,a}^Y, y_{n,b}^Y, y_{n,c}^Y \end{bmatrix}^T, \quad y_n^\Delta = \begin{bmatrix} y_{n,a}^\Delta, y_{n,b}^\Delta, y_{n,c}^\Delta \end{bmatrix}^T$$

**Cap controls:**

$$y_n^Y = \sum_{i=1}^{n_c^s} \eta_{c,i} y_{c,i}^Y, \quad y_n^\Delta = \sum_{i=1}^{n_c^s} \eta_{c,i} y_{c,i}^\Delta$$

where \( n_c^s \rightarrow \mathbb{N} \) steps, \( \eta_{c,i} \in \{0, 1\} \) \( \eta_{c,i} = f(\vartheta^4, \vartheta, \vartheta^5) \)

---

4 Control input (current, voltage, kvar, PF, time)
5 Attack vector. Upper and lower limits (operation is outside limits)
MODELED AS A WYE-CONNECTED CONSTANT POWER LOAD

\[ S_{n,Y}^P(t) = s_Y^Y(t) \]

SOLAR RESOURCES AND BATTERIES:

\[ S_{n,Y}^P(t) = s_Y^Y(t) \quad \text{and} \quad S_{n,Y}^P(t) = f(\eta_c,t, \eta_d,t, s_{oc}(t - 1)^6) \]

SMART INVERTERS:

\[ S_{n,Y}^P(t) = f(s_Y^Y(t - 1), v_{n,t})^8 \]

---

\(^6\)State of charge
\(^7\)Voltage measurement
\(^8\)Attack vector. Changes in drop curve settings
Preliminary results
IEEE-123 test case
## OpenDSS vs Log(V) 3LPF

<table>
<thead>
<tr>
<th>RMSE (IEEE-123)</th>
<th>Distflow</th>
<th>Log(V) 3LPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>0.014</td>
<td>0.012</td>
</tr>
<tr>
<td>Phase 2</td>
<td>0.016</td>
<td>0.08</td>
</tr>
<tr>
<td>Phase 3</td>
<td>0.006</td>
<td>0.04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Devices</th>
<th>Buses</th>
<th>Nodes</th>
<th>Losses (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-123</td>
<td>238</td>
<td>132</td>
<td>278</td>
<td>2.63</td>
</tr>
<tr>
<td>IEEE-8500</td>
<td>7282</td>
<td>4876</td>
<td>8531</td>
<td>10.58</td>
</tr>
<tr>
<td>European LV</td>
<td>965</td>
<td>907</td>
<td>2721</td>
<td>0.2703</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Iterations</th>
<th>OpenDSS (s)</th>
<th>Log(V) 3LPF (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-123</td>
<td>3</td>
<td>0.011</td>
<td>0.0001</td>
</tr>
<tr>
<td>IEEE-8500</td>
<td>62</td>
<td>0.244</td>
<td>n/a</td>
</tr>
<tr>
<td>European LV</td>
<td>3</td>
<td>0.087</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Current efforts. Future work

- Exploring techniques to solve the linear system of equations. E.g.
  - Forward-backward sweep
  - Truncated SVD
  - Parallel computation on leaf nodes
- Re-centering around 1 to obtain better approximation
- Sherman-Morrison for matrix inversion after perturbation
- Solving large cases directly from .dss files


Lossy distflow formulation for single and multiphase radial feeders.