Chevauchey Clément - August 2025

SCAN ME

Feeder Study - Method, Controls, and Results

Objectives

- Build a balanced AC feeder in **pandapower** (CIGRE MV template).
- Run **load-flow** (steady state) and **short-circuit** (IEC 60909).
- Export results as **pandas DataFrames** and simple plots.
- Evaluate small corrective actions and select the **smallest feasible fix**
- Limits used: Vmin ≥ 0.95 pu; Vmax ≤ 1.05 pu; line & trafo ≤ 100%.

Results at a glance

Table 1: Results at a glance (baseline vs selected action)

Metric	**Baseline**	**Selected**	**Δ (sel−base)**
$V_{\min}$ (pu)	0.923	0.952	0.029
$V_{\max}$ (pu)	1.030	1.030	0.000
Line max (%)	97.0	81.7	−15.3
Trafo max (%)	101.4	90.1	−11.3

Selected case: `cap_1.0MVAr+rebalance_90pct`
Action: Capacitor 1.0 MVAr @ bus 11; Rebalance ×0.90 (hot LV feeder); OLTC Δtap = 0.
Limits: $V_{\min} \geq 0.95$, $V_{\max} \leq 1.05$, line/trafo ≤ 100%.

Core concepts

Pandapower network model

A pandapower **network** is a single Python object (net) that holds many pandas DataFrames:

- **Input tables**: net.bus, net.line, net.trafo, net.load, net.sgen, …
- **Result tables (created by calculations)**: net.res_* (e.g., net.res_bus, net.res_line, net.res_trafo, net.res_bus_sc).

- **Bus**: connection node (e.g., 20 kV); results give vm_pu (1.0 pu ≈ nominal).
- **Elements**: lines / transformers / loads / generators; you define them → run → read **net.res_***.
- **Slack bus**: balances P & Q so P_in = P_out + losses.
- **Per-unit**: normalized system; keep planning band **0.95 - 1.05 pu**.

Control devices

OLTC (On-Load Tap Changer)

An OLTC is a transformer ratio changer used on energized transformers (e.g., HV/MV or MV/LV). Typical step size is ≈1.25% per tap with a total range around ±8–16%. An AVR holds the controlled bus near a setpoint with a deadband and delay; optional line-drop compensation biases the target to maintain remote-end voltage. OLTCs correct under/over-voltage and slightly reduce current for constant-P loads, but they **do not** remedy thermal overloads driven by real power.

Fixes: undervoltage/overvoltage; slightly reduces current at constant P.
Cannot fix: thermal overload driven by **real power P**; heavy reactive needs far out.

Use OLTC when under/over-voltage and thermal limits are not binding

Shunt Capacitor

A shunt capacitor injects capacitive Q at a bus, reducing upstream reactive current and voltage drop. Placement can be at the substation MV bus (head support) or at the feeder remote end (to lift Vmin). Fixed or step-switched banks (e.g., 1+1+1 MVAr) are common. They address low voltage from reactive drop and can free a few percent of thermal headroom, but they do not solve overloads caused by real power.

Fixes low voltage from reactive drop; frees a few % thermal headroom.
Cannot fix: overload from **real power P**.

Use caps to correct reactive-driven sag at the far end.

OLTC + Caps together

Voltage drop can be approximated as $\Delta V \simeq R \cdot P + X \cdot Q$

OLTC raises the head-end voltage (source side), while capacitors reduce local Q (shrinking the X·Q term) where installed. Together, they lift the entire profile, especially at the remote end - hence the standard practice of AVR-controlled OLTC with switched cap banks.

Workflow (what the script does)

Numbers below map to the script's in-code # comments so readers can jump between report and code.

1. **Headless plotting** - use the Agg backend so figures render on servers/CI.
2. **Limits & policy** - define acceptance limits (e.g., 0.95 - 1.05 pu, ≤95 % loading) and choose selection policy (avoid_rebalance / min_cap / min_thermal).
3. **Output folders** - ensure figures/ and data/ exist.
4. **Robust net saver** - .pkl.gz is the default robust format and JSON/Excel are best-effort
5. **Build baseline** - create CIGRE MV test feeder, run **LF** (pp.runpp).
6. **Baseline SC** - compute **IEC 60909** max fault currents.
7. **Baseline V-profile** - save voltage profile plot with 0.95/1.05 pu rails.
8. **Baseline KPIs/CSV** - export res_bus, res_line, res_bus_sc (if present) + quick KPI prints.
9. **Helpers** - topology scope finder (loads on a trafo's LV feeder), stress-map plot, and before/after voltage overlay plot.
10. **Scenario recorder** - snapshot() stores KPIs plus action metadata (tap/cap/rebalance).
11. **Baseline snapshot** - record the reference case.
12. **OLTC test** - if taps exist, try +1/+2 steps (within limits), snapshot, restore.
13. **Cap sweep @ weakest bus** - try 0/1/2/3 MVAr at the current worst-V bus, snapshot, clean up.
14. **Targeted rebalancing** - scale only loads under the **hottest trafo** (1.00, 0.95, 0.90, 0.85), snapshot, restore.
15. **Small combos** - light **cap + rebalance** grid (e.g., 1 - 2 MVAr × 0.95/0.90), snapshot.
16. **Persist scenarios** - write data/scenario_summary.csv.
17. **Pick "best"** - filter by limits, then sort per policy; export KPIs & manifest.
18. **Apply fix** - deep-copy **before**, apply actions, deep-copy **after**; save plots.
19. **After-state SC** - run IEC 60909 again and export results (if present).
20. **Cleanup/restore** - remove shunt, undo load scaling, reset taps; re-solve.
21. **Finish** - print completion.

Running the study

Environment: pandas 2.3.2; pandapower 3.1.2; Python 3.13; artifacts saved as .pkl.gz to avoid JSON/Excel writer drift.

```
python -m venv .venv && source .venv/bin/activate
pip install pandapower matplotlib pandas
python scripts/run_study.py
```

Outputs (in data/ and figures/) include:

- scenario_summary.csv, run_kpis__*.csv, best_scenario__*.json, run_manifest.txt
- Before/after nets as .pkl.gz (and JSON/Excel where supported)
- Voltage profile (before/after) and stress-map plots (before/after)

Interpreting results (quick checks)

- **Voltages**: net.res_bus.vm_pu - aim for **0.95 - 1.05 pu**.
- **Line/Trafo loading**: net.res_line.loading_percent, net.res_trafo.loading_percent - keep **< 100 %** (planning margins preferably 80 - 90 %).
- **short-circuit (IEC 60909)**: net.res_bus_sc - use for protection sizing/coordination.

Practical cautions

- **Units:** P in MW, Q in MVAr, V in kV; currents often A/kA; many results are in pu or %.
- **Indices vs names:** API functions use **indices**. If you track by name, map name → index first.
- **LF convergence:** islands/no slack/contradictory setpoints → NaNs in res_bus. Ensure a source path and one slack.
- **Version drift:** Plot helpers and JSON/Excel writers vary by pandapower/pandas/Python. The script already **falls back to .pkl.gz** saves when needed.
- **Saving**: the script saves .pkl.gz snapshots for version-robustness and attempts JSON/Excel when supported.

Planning guidance

- If **undervoltage** with moderate loadings → **OLTC up** and/or **small capacitor(s)** at the weak bus.
- If **thermal overload is mainly** from **P** → **reconfiguration / load transfer / reconductoring** (caps help only a little).
- Prefer **switched** capacitor steps (e.g., 1 + 1 + 1 MVAr) to avoid light-load overvoltage and to handle DG export seasons.

Study case

Baseline highlights

- **Vmin** = 0.923 pu (below the 0.95 pu floor)
- **Vmax** = 1.030 pu
- Lines near limit: **Line 1 - 2 = 96.48 %**, **Line 2 - 3 = 96.96 %**
- Transformers: **Trafo 0 - 1 = 101.41 % (25.35 MVA on 25 MVA)**, **Trafo 0 - 12 = 84.70 %**

Implication: Low Vmin with one transformer >100% while a second has headroom suggests an unbalanced feeder loading pattern.

Usual solutions

If a transformer is ≥100% (Like in this case):

- Caps don't fix **P**. They shave **Q** → current drops a bit, but if pf is already decent you only recover a few %.

If the main problem is undervoltage (Vmin < 0.95 pu) and lines/trafo are not hard over:

- **OLTC up +1 (maybe +2)** on the relevant HV/MV trafo. This usually gives a clean +1 - 3% V on the LV side and may slightly **reduce loading** (constant-P loads draw less current at higher V).
- **Check light-load case**: higher taps can cause Vmax > 1.05 pu at night.

If remote-end is still low or reactive flows are heavy:

- **Add shunt capacitor(s)** at or near the **lowest-V bus** (start 1 - 3 MVAr).
- Goal: push Vmin ≥ 0.95 pu **without** pushing any bus > 1.05 pu.
- Prefer **step-switched** caps (e.g., 1 + 1 + 1 MVAr) so you can drop steps at light load.

If one or two lines sit ≥95 - 100% even after OLTC + caps:

- Caps can lower current a bit, but if the bottleneck is real, you need **reconfiguration**, **parallel/reconductor**, or **load transfer**.

Common in practice: OLTC holds the **substation** voltage band (e.g., 1.00 - 1.03 pu). **Switched caps** support the **far end** so the profile stays inside 0.95 - 1.05 pu across seasons. Rebalancing is the structural lever when a trafo/feeder is simply carrying too much.

Script output (With # Comments)

Versions -> pandas: 2.3.2

Environment info for reproducibility (pandas version)

pandapower: 3.1.2

pandapower version used in the study

Voltage profile saved -> figures/voltage_profile.png (see Fig. 1)

Baseline bus-voltage plot was created

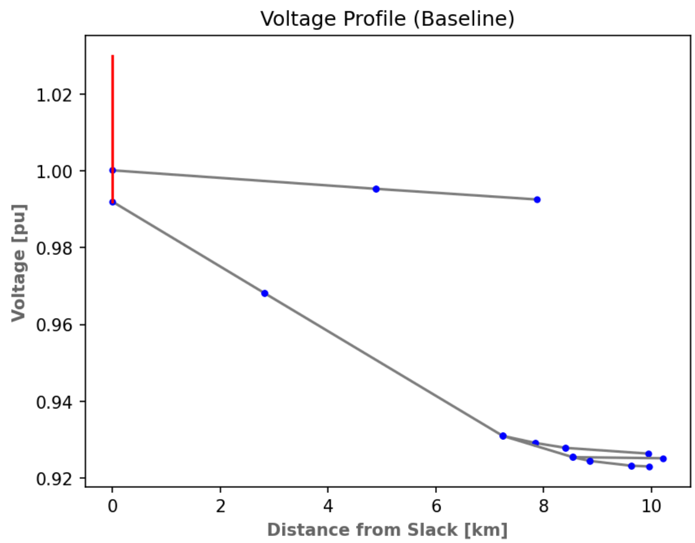

Figure 1 - Baseline voltage profile (figures/voltage_profile.png).

Line loading mean %: 26.68771279701972

Average line loading across the feeder (~27%, healthy overall)

Trafo loading mean %: 93.05476012944422

Average transformer loading (~93%, but see individual trafos below)

Exported res_bus/res_line/res_bus_sc -> data/*.csv

Baseline result tables written to CSV for inspection/sharing

Voltage pu: min=0.923 max=1.030

Baseline voltages: Vmin below 0.95 pu floor; Vmax within 1.05 pu cap

Vmin = 0.923 pu (too low) ~2.7% below a common 0.95 pu floor

Vmax = 1.030 pu (fine).

Top-5 lines by loading %: # Hottest spans first (watch for ≥95-100%)

Line name	Loading percent
1 Line 2-3	96.958807
0 Line 1-2	96.482978
9 Line 3-8	48.143637
2 Line 3-4	37.580668
6 Line 8-9	33.550914

Lines: two spans are near thermal limit
Line 1-2 ⟶ 96.48%
Line 2-3 ⟶ 96.96%

Transformers: one is overloaded
Trafo 0-1: 101.41% (≈ 25.35 MVA on a 25 MVA unit)
Unbalanced feeder loading pattern; power path through 1–2–3 is stressed

Trafo 0-12: 84.70% (≈ 21.17 MVA)

Transformer KPI table (nameplate MVA, loading, headroom, estimated MVA)

name	sn_mva	loading percent	headroom (%)	MVA
0 Trafo 0-1	25.0	101.411473	-1.411473	25.352868
1Trafo 0-12	25.0	84.698048	15.301952	21.174512

0: Overloaded (~101.4%); negative headroom; ≈25.35 MVA on 25 MVA unit
1: Comfortable (~84.7%); ≈21.17 MVA

No transformers with defined OLTC -> skipping OLTC scenario.
Model has no tap data; OLTC trials are auto-skipped

Cap 0.0 MVAr -> Vmin=0.923, Line max=97.0%, Trafo_max=101.4%
Reference point (no capacitor): under-voltage + trafo overload + hot lines
Cap 1.0 MVAr -> Vmin=0.944, Line max=91.4%, Trafo_max=100.0%
+1 MVAr at weakest bus: Vmin improves; currents drop; trafo just at 100%

```
Cap 2.0 MVAr -> Vmin=0.963, Line max=90.4%, Trafo_max=98.7%
# +2 MVAr: Vmin now ≥ 0.95; thermal margins improve further
Cap 3.0 MVAr -> Vmin=0.982, Line max=94.7%, Trafo_max=97.7%
# +3 MVAr: more margin but slightly higher line_max than +2 (network interactions)
-------------------------------------------------------------------
Rebalance x1.00 -> Vmin=0.923, Line max=97.0%, Trafo_max=101.4%
# No change (baseline)
Rebalance x0.95 -> Vmin=0.927, Line max=92.2%, Trafo_max=96.5%
# -5% load on hot feeder proxy: lowers thermal stress; small Vmin lift
Rebalance x0.90 -> Vmin=0.930, Line max=87.4%, Trafo_max=91.6%
# -10%: significant thermal relief; Vmin still < 0.95
Rebalance x0.85 -> Vmin=0.934, Line max=82.7%, Trafo_max=86.7%
# -15%: lots of thermal headroom; voltage still below floor
-------------------------------------------------------------------
Cap 1.0 + Rebalance x0.95 -> Vmin=0.948, Line max=86.5%, Trafo_max=95.0%
# Small cap + small rebalance: close to limits, still Vmin < 0.95
Cap 1.0 + Rebalance x0.90 -> Vmin=0.952, Line max=81.7%, Trafo_max=90.1%
# 1 MVAr + 10% rebalance: all KPIs within limits (Vmin≥0.95, loadings<95)
Cap 2.0 + Rebalance x0.95 -> Vmin=0.967, Line max=85.7%, Trafo_max=93.8%
# Works too, but more MVAr than needed
Cap 2.0 + Rebalance x0.90 -> Vmin=0.970, Line max=81.1%, Trafo_max=88.9%
# Also feasible; higher cap and same rebalance → larger margins
-------------------------------------------------------------------
Scenario summary -> data/scenario_summary.csv
# All scenarios and KPIs sent to CSV
Best candidate (LF): {'case': 'cap_1.0MVAr+rebalance_90pct', 'vmin': 0.9515528091076045, 'vmax':
1.03, 'line_max_%': 81.67415392469702, 'trafo_max_%': 90.12479860758239, 'trafo_Trafo 0-1_%':
90.12479860758239, 'trafo_Trafo 0-12_%': 76.63419674320254, 'dtap': 0, 'q_mvar': 1.0, 'frac': 0.9,
'cap_bus': 11.0, 'rebalance_trafo_idx': 0.0}
# Selector chose smallest fix that meets limits
-------------------------------------------------------------------
Chosen action → cap=1.0 MVAr @ bus 11.0, rebalance×0.9, dtap=0
# Action to apply in the "after" state (no tap steps available)
-------------------------------------------------------------------
Saved pandapower net -> data/net_before__cap-1-0mvar-rebalance-90pct__20250821-210943.pkl.gz
# Before-state network snapshot (portable .pkl.gz)
Saved pandapower net -> data/net_after__cap-1-0mvar-rebalance-90pct__20250821-210943.pkl.gz
# After-state network snapshot (portable .pkl.gz)
Saved: figures/voltage_profile__cap-1-0mvar-rebalance-90pct__20250821-210943.png (see Fig. 2)
# Before/after voltage comparison plot
```

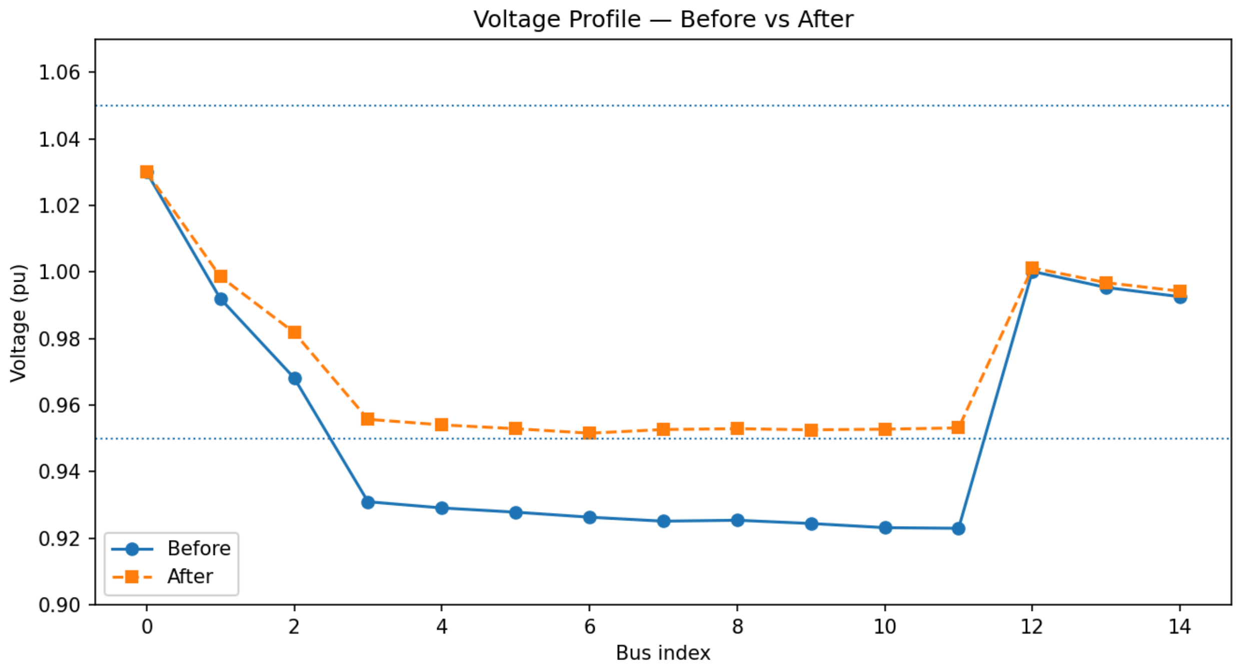

Figure 2 - Voltage Profile - Before vs After (figures/voltage_profile__cap_….png).

Saved: figures/network_stress_before__cap-1-0mvar-rebalance-90pct__20250821-210943.png

"Stress map" by loading (before) (see Fig. 3)

Saved: figures/network_stress_after__cap-1-0mvar-rebalance-90pct__20250821-210943.png

"Stress map" by loading (after) (see Fig. 4)

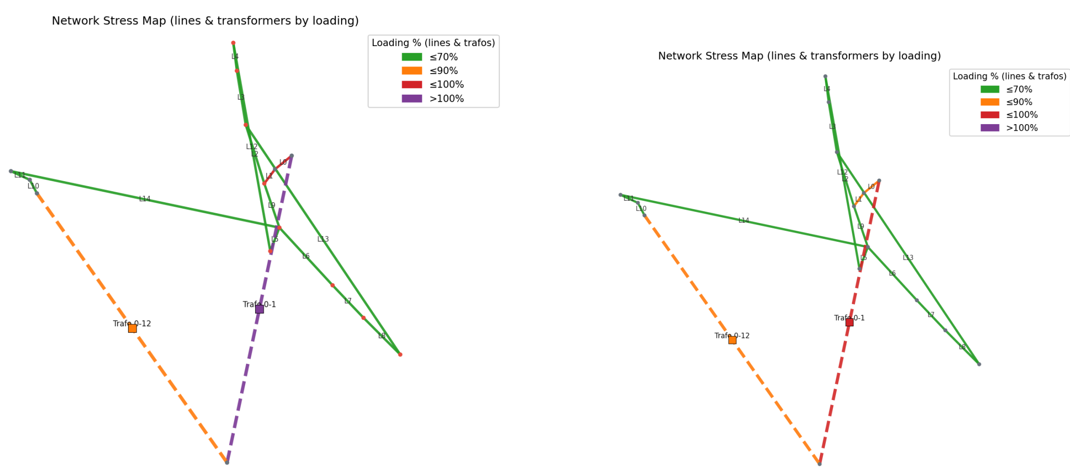

Figure 3-4 - Network stress map (after) (figures/network_stress_after__...png).

Applied: {'q_mvar': 1.0, 'cap_bus_used': 11, 'frac': 0.9, 'tap_steps': 0}
Runtime confirmation of what was actually applied
Done.
Study finished (state restored after saving artifacts)

Results (baseline)

- **Vmin = 0.923 pu**, **Vmax = 1.030 pu** → undervoltage issue.
- **Lines**: two spans near limit: **Line 1 - 2 ≈ 96.5%**, **Line 2 - 3 ≈ 97.0%**.
- **Transformers**: **Trafo 0 - 1 = 101.41% (≈ 25.35 MVA / 25 MVA)**; **Trafo 0 - 12 = 84.70%**.
  ⇒ Overload concentrated on **0 - 1**, with headroom on **0 - 12**.

template in pandapower 3.1.2 has no tap defined ⇒ OLTC test skipped.

(See data/res_line.csv, data/res_trafo.csv, data/scenario_summary.csv)

Scenario scan (selected highlights)

- **Cap 1.0 MVAr only**: Vmin **0.944 pu**, line max **91.4%**, Trafo max **100.0%**.
- **Rebalance ×0.90 only**: Vmin **0.930 pu**, line max **87.4%**, Trafo max **91.6%**.
- **Cap 1.0 MVAr + Rebalance ×0.90**: **Vmin 0.952 pu**, **line max 81.7%, Trafo 0 - 1 90.1%**.

(Full list in **Table S3**: data/scenario_summary.csv.)

Selected action (smallest feasible fix)

- **1.0 MVAr shunt** at **bus 11** (capacitive) **+ 10% load transfer** off Trafo 0 - 1's LV feeder.
- Meets planning limits (**Vmin ≥ 0.95 pu, Vmax ≤ 1.05 pu, and all loadings ≤ 100%**).
- Artifacts:
  - **Before net**: data/net_before__cap-1-0mvar-rebalance-90pct__<timestamp>.pkl.gz
  - **After net**: data/net_after__cap-1-0mvar-rebalance-90pct__<timestamp>.pkl.gz
  - **Plots**: figures/voltage_profile__...png, figures/network_stress_before__...png, figures/network_stress_after__...png

We select a 1.0 MVAr shunt at bus 11 plus a 10% load transfer off the hottest LV feeder. This raises Vmin to ≈0.952 pu and brings line and transformer loadings below 100%, meeting the planning limits (Vmin ≥ 0.95 pu, Vmax ≤ 1.05 pu, loadings ≤ 100%).

Check: 0.9516 ≥ 0.95; 1.030 ≤ 1.05; line 81.7% ≤ 100%; trafo 90.1% ≤ 100% - pass.

Why this works

A small **cap** lowers **Q** where it matters (bus 11), a **10% rebalance** trims **P** through the overloaded path. Together they lift Vmin and relieve thermal limits with minimal intervention.

Caveats & next steps

- If the real feeder does have an OLTC, consider AVR + switched caps (1+1+1 MVAr) and re-run to confirm seasonal margins.
- Validate field locations for load transfer; if structural P-driven overload persists, consider reconductoring or feeder reconfiguration.

Conclusion

Baseline power flow on the CIGRE MV feeder shows
- Undervoltage (Vmin 0.923 pu)
- Two spans near thermal limits (~97%)
- Transformer 0 - 1 overloaded (101.4%).

Small corrective action:
- 1.0 MVAr shunt at the weakest bus plus
- 10% load transfer off the hot LV feeder

Effect:
- Raises Vmin to ∼0.952 pu
- Brings line/transformer loadings below 100%,

This meets the typical planning limits with the smallest intervention tested.

Note that OLTC was not evaluated because the template lacks tap data; in practice, AVR + switched caps is the standard operating strategy.

ANNEX

Create the network and run LF

```
import pandapower as pp, pandapower.networks as pn
net = pn.create_cigre_network_mv(with_der=False)
pp.runpp(net) # fills res_* tables
```

Run short-circuit (IEC 60909)

```
import pandapower.shortcircuit as sc
sc.calc_sc(net, case="max") # results in net.res_bus_sc (version-dependent columns)
```

Add a 2 MVAr shunt at the weakest bus

```
bmin = int(net.res_bus.vm_pu.idxmin())
sh = pp.create_shunt(net, bus=bmin, q_mvar=-2.0, p_mw=0.0, name="cap_2MVAr")
pp.runpp(net)
# ...inspect results...
net.shunt.drop(sh, inplace=True); pp.runpp(net) # clean up
```

Try OLTC steps (if present)

```
t = net.trafo.index[0]
old = int(net.trafo.at[t, "tap_pos"])
for delta in (1, 2):
    net.trafo.at[t, "tap_pos"] = old + delta
    pp.runpp(net)
net.trafo.at[t, "tap_pos"] = old; pp.runpp(net)
```

Note

shunt q_mvar is **negative** in pandapower (capacitive)