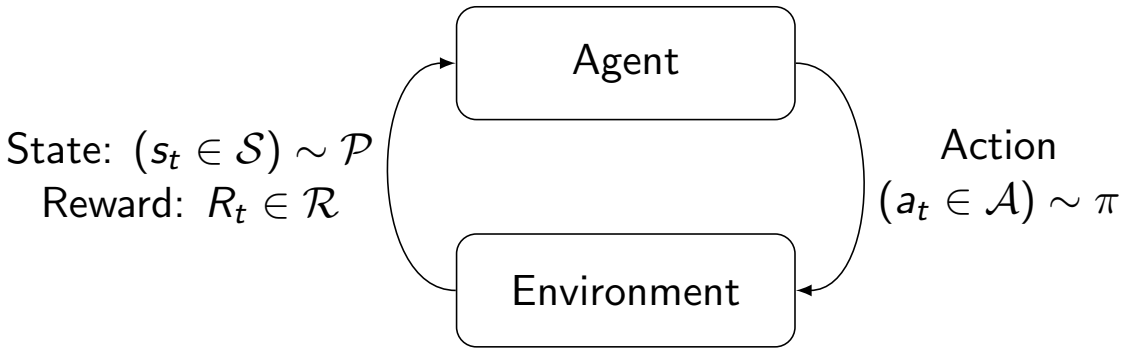
Deep Reinforcement Learning for DER Cyber-Attack Mitigation

CIGAR Workshop 17^{th} March 2021

BERKELEY LAB

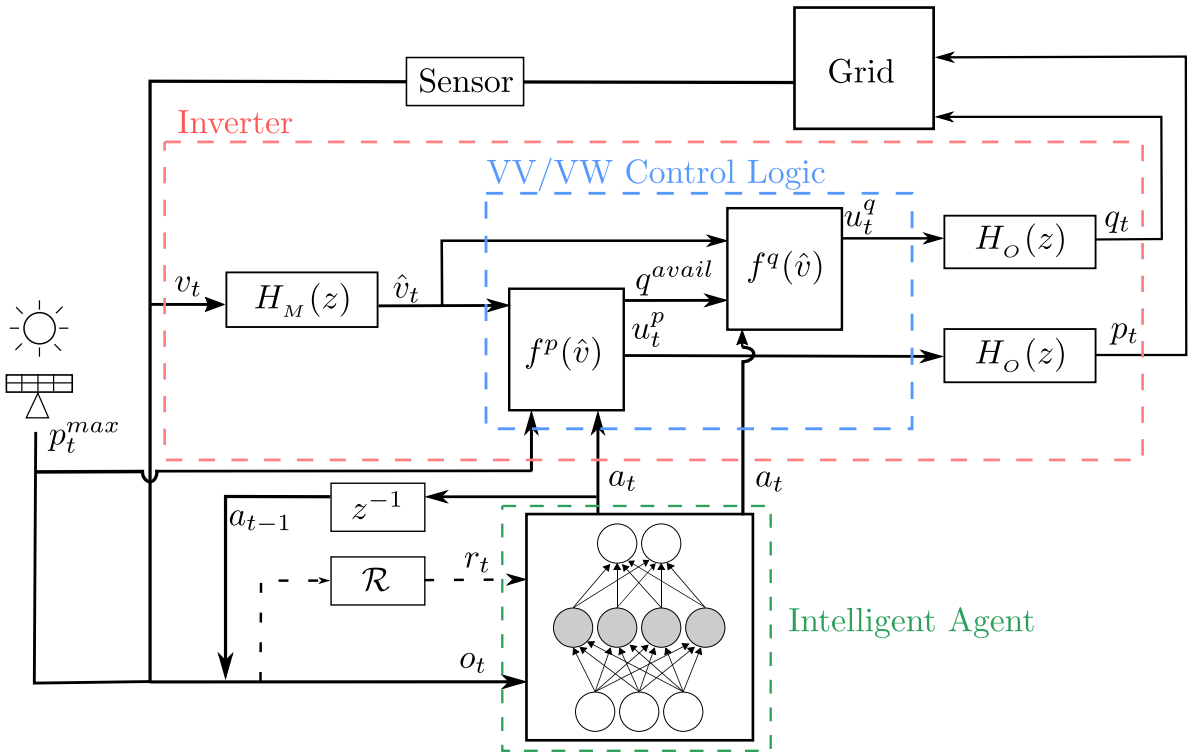
Deep Reinforcement Learning



- Adapt a model-free based approach where DRL models optimal controller using a neural network
- Learns optimal state, s_t, to action, a_t, mapping by repeatedly interacting with environment and receiving a reward R_t
- Weights of these neural networks are learned end-to-end via gradient-based optimization

BERKELEY LAB

Modeling DER Action Space

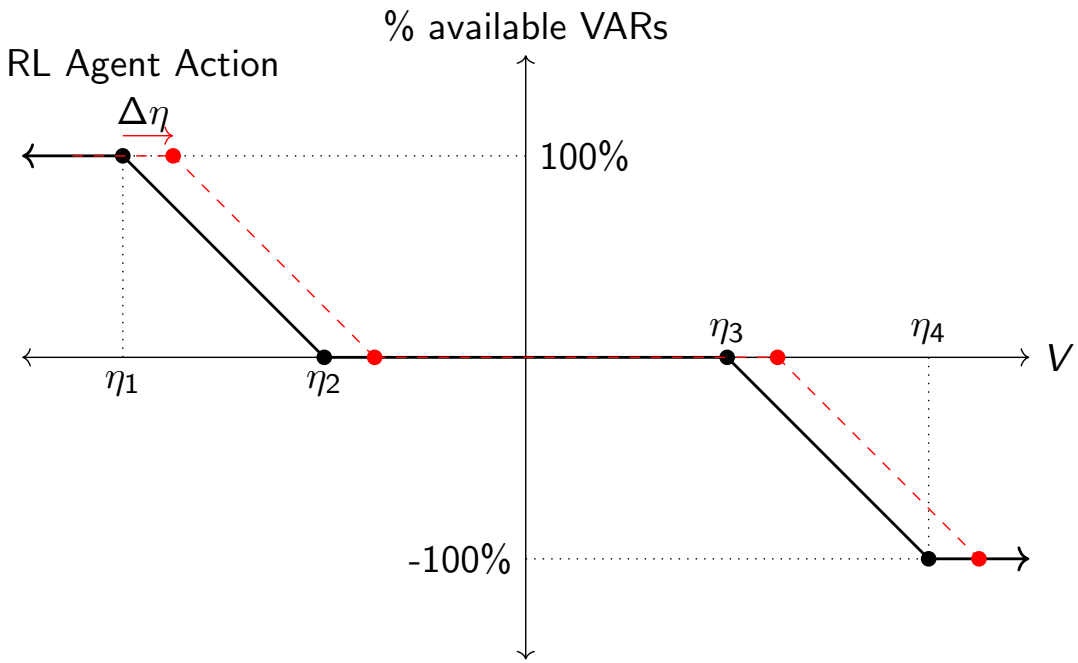


- Voltage measurements are low-pass filtered before active power and reactive power set point calculation
- These set-points are themselves low-pass filtered to ramp rate limit active and reactive power injections

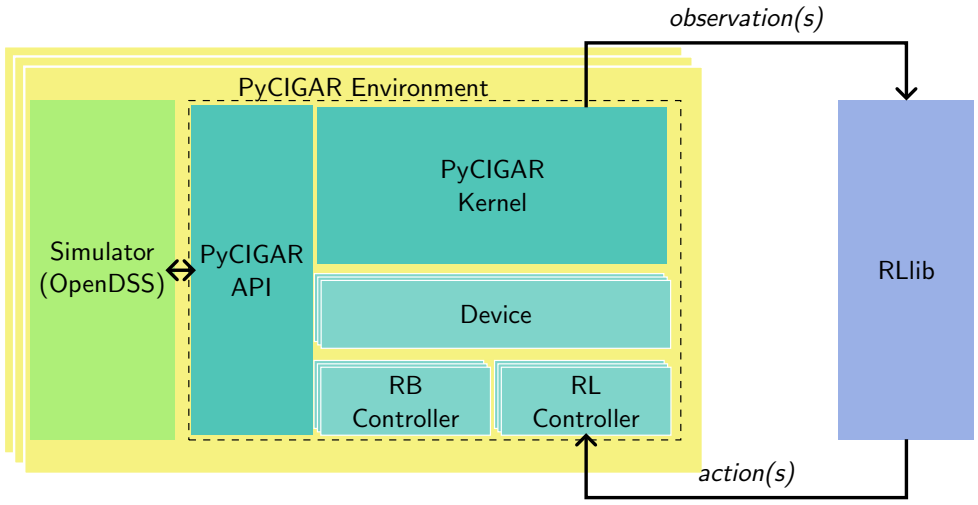
BERKELEY LAB

Modeling DRL Action

- DRL action is the deviation, i.e. $a_t = \Delta\eta$, from default VV/VW parameterization
- Translating curve was found to be preferred action during training
  - Agent learns to indirectly control reactive power injection/consumption



Training

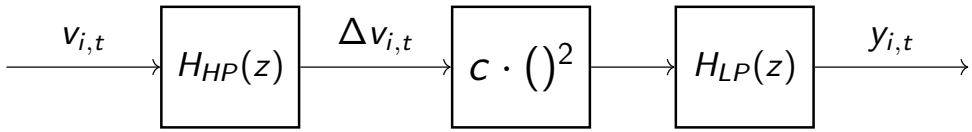


- For training we consider a single RL agent whose observation input vector is the mean of all DER observation input vectors
- This agent then outputs an action that is applied across all inverters in the system
- Once trained, this policy is deployed and acts only on local measurements

BERKELEY LAB

Observation Vector

- We use a simple filter to estimate the energy of the oscillation



The complete observation vector is then given by

- $y_{i,t}$: the estimation of voltage oscillation energy at node i
- $u_{i,t}$: the estimation of voltage unbalance energy at node i
- $v_{i,t}^{a,b,c}$: measurement of the phase voltages at bus i
- $q_{i,t}^{\text{avail, nom}}$: the available reactive power capacity without active power curtailment.
- $a_{i,t-1}^{\text{one-hot}}$: one-hot encoding of the previous action taken by the agent.

BERKELEY LAB

Reward Function

At a timestep t, the reward function, $R_t(a_t, o_t)$, to be maximized is:

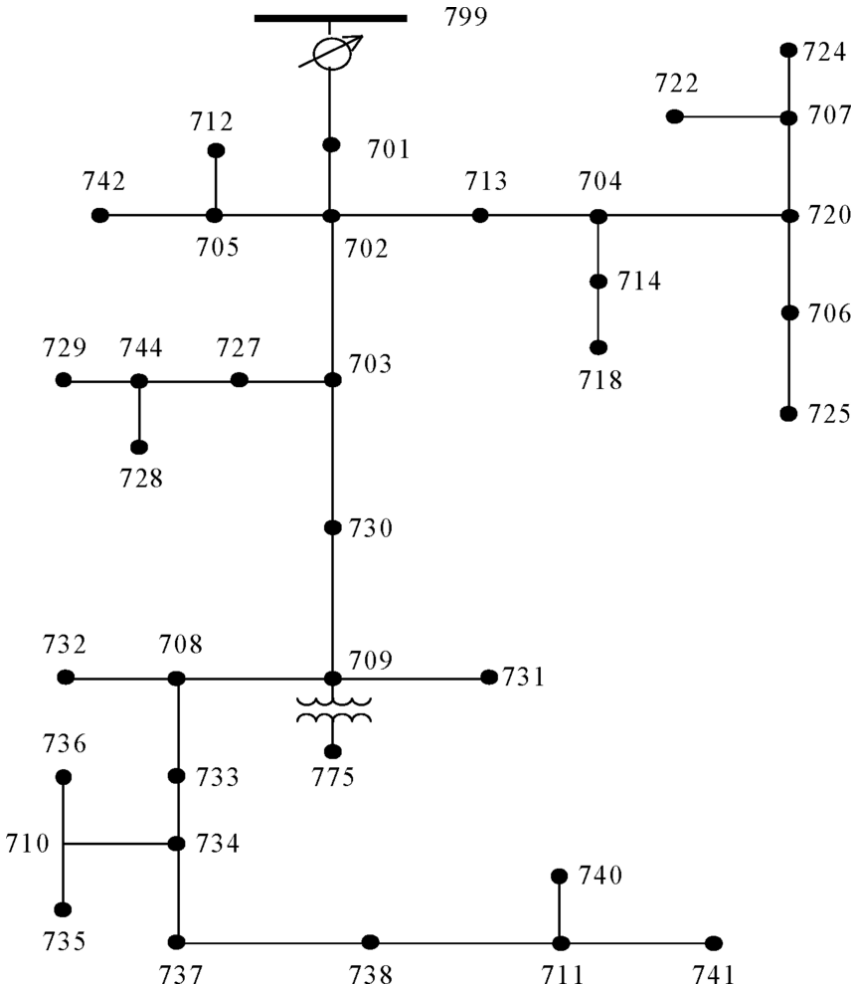
$$R_t = -\left(\frac{1}{|\mathcal{U}|} \sum_{i=1}^{|\mathcal{U}|} \sigma_y y_{i,t} + \sigma_u ||\boldsymbol{u}_t||_\infty + \sigma_a \mathbf{1}_{a_t \neq a_{t-1}} \right.$$
$$\left. + \sigma_0 \|a_t\|_2 + \frac{1}{|\mathcal{U}|} \sum_{i=1}^{|\mathcal{U}|} \sigma_p \left(1 - \frac{p_{i,t}}{p_{i,t}^{\max}} \right)^2 \right).$$

This reward seeks to encourage the agent to

- Minimize system voltage oscillations
- Minimize the worst case voltage unbalance
- Minimize number of VV/VW re-configurations
- Encourage the VV/VW parameterizations to remain close to their default values
- Minimize active power curtailment

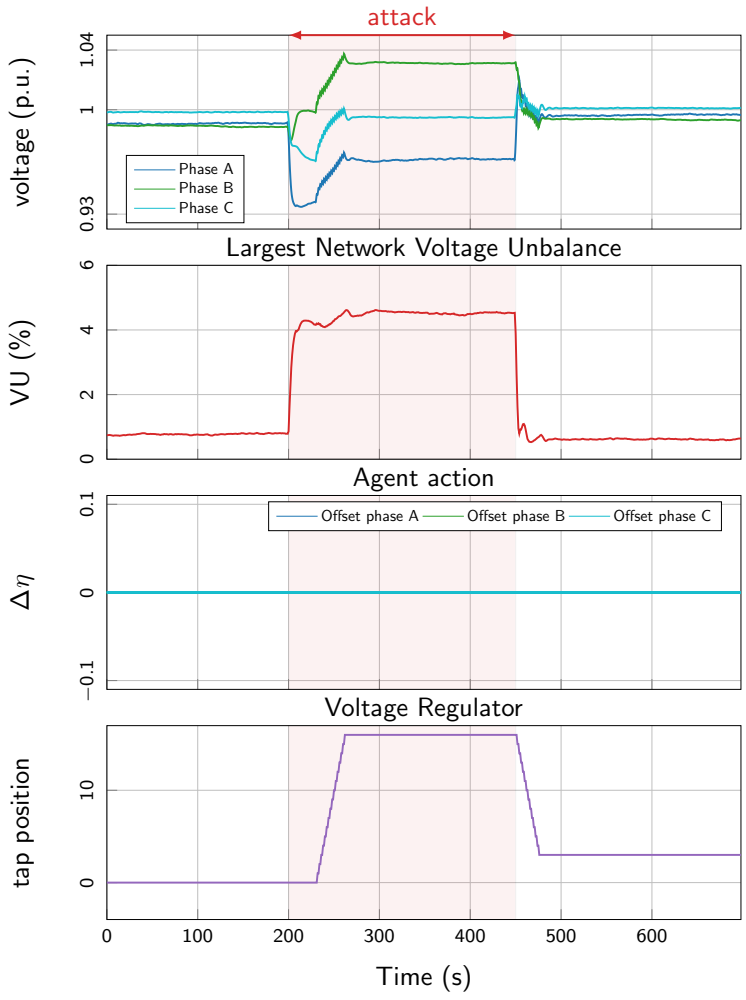
BERKELEY LAB

IEEE 37 Feeder

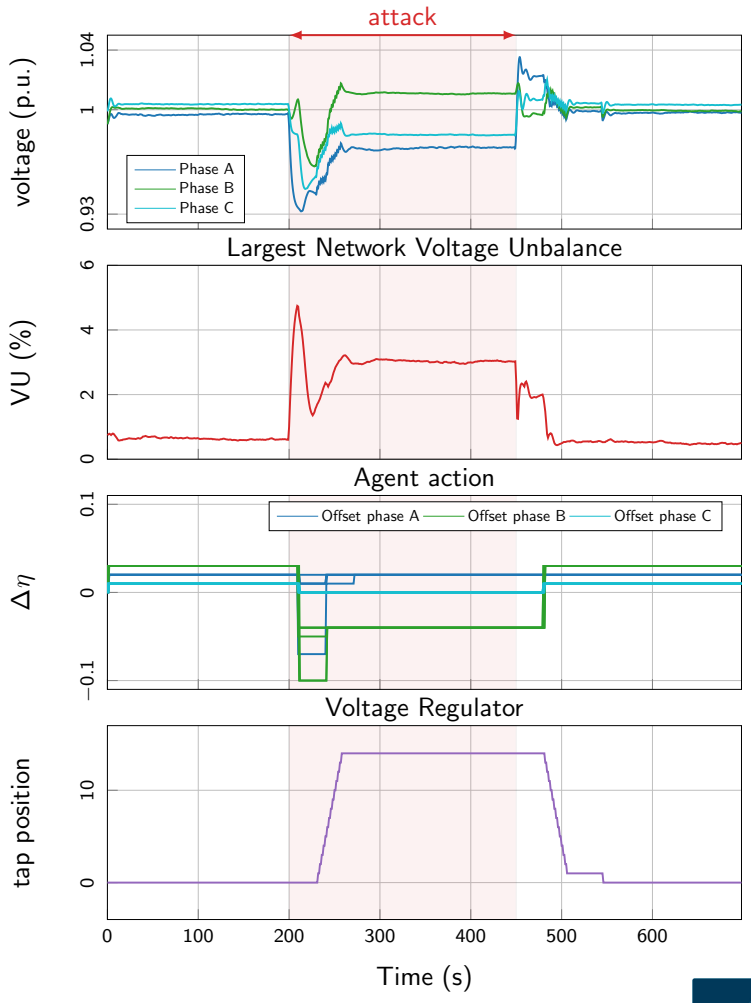


BERKELEY LAB

IEEE 37 Node Imbalance Attack - 40% - Use Regulator



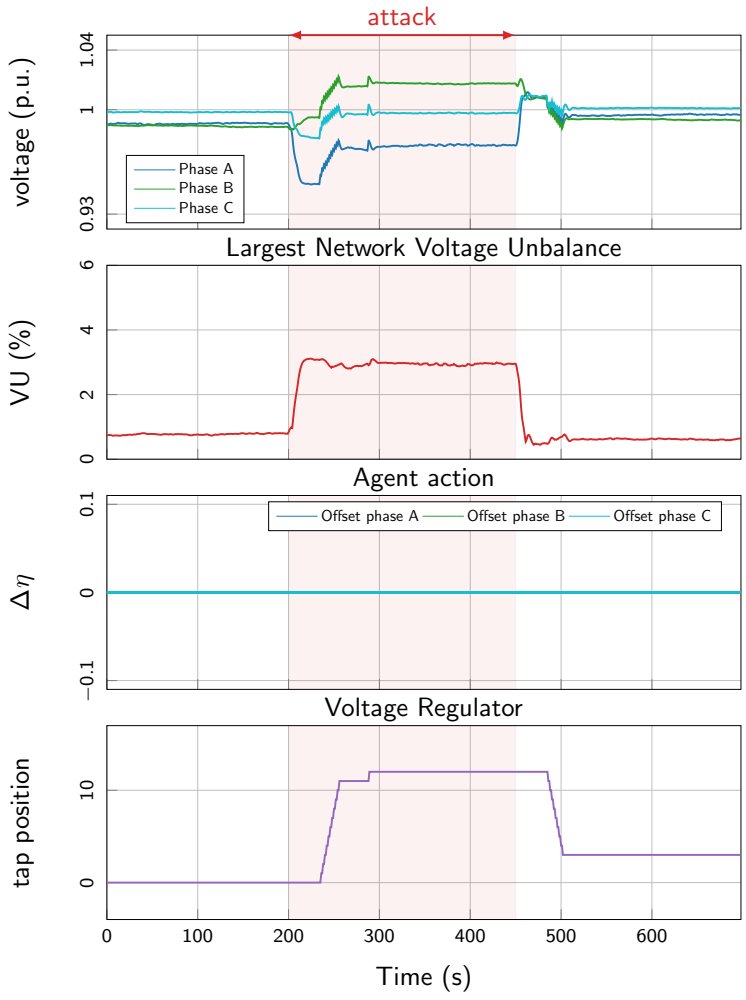
(a) No defense



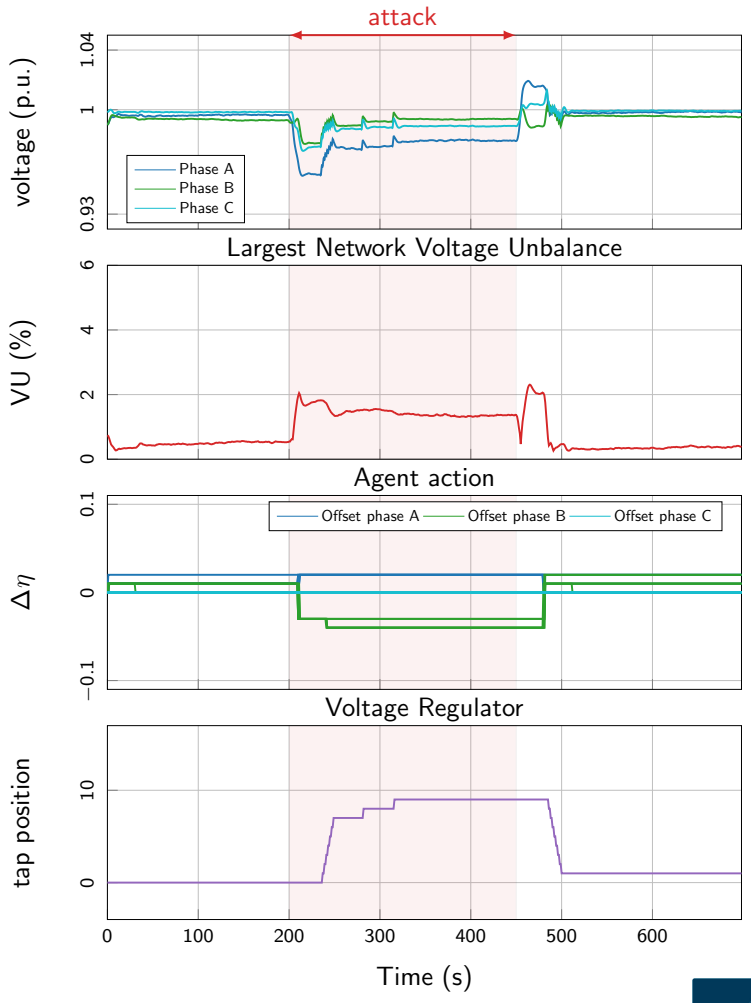
(b) Defense

BERKELEY LAB

IEEE 37 Node Heterogeneous Imbalance Attack - 30%



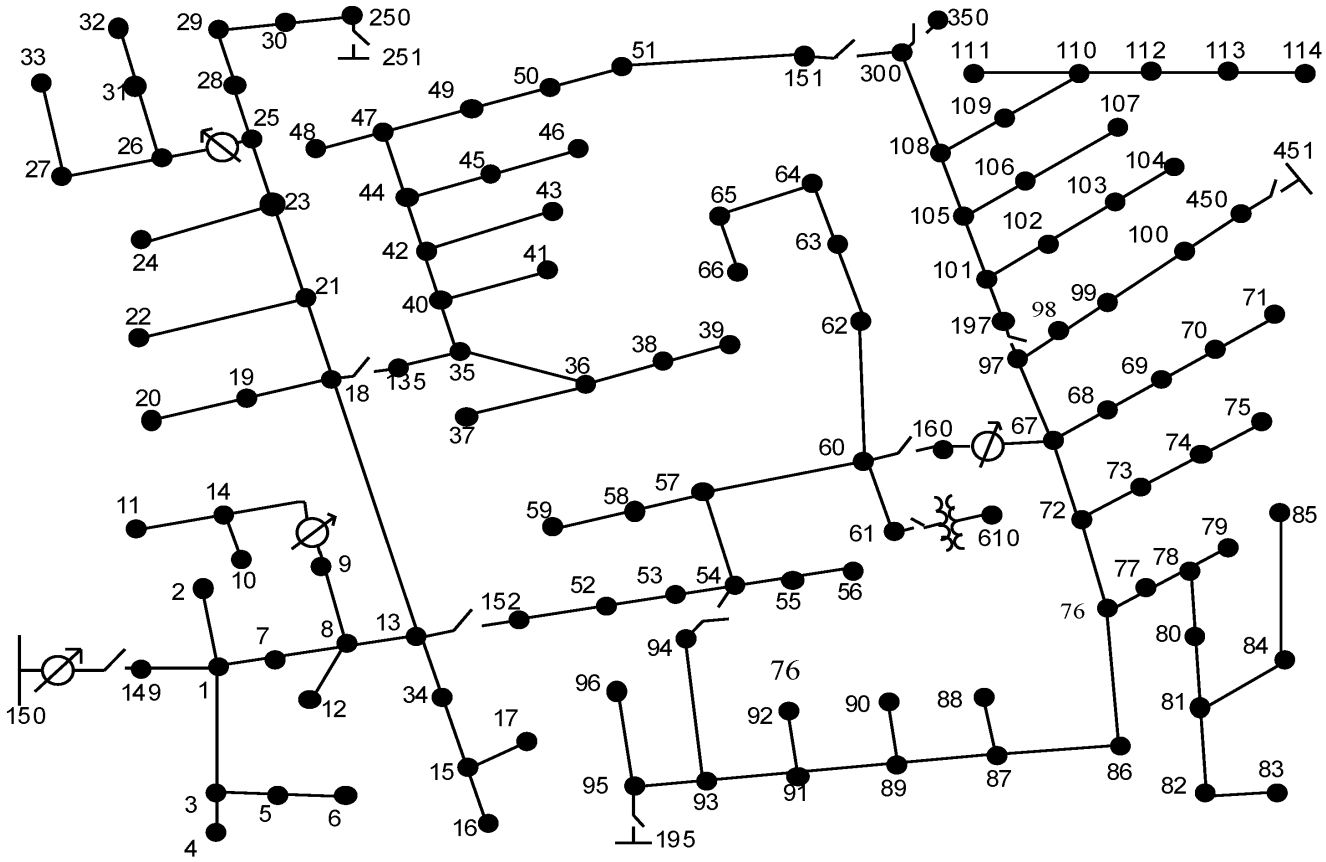
(c) No defense



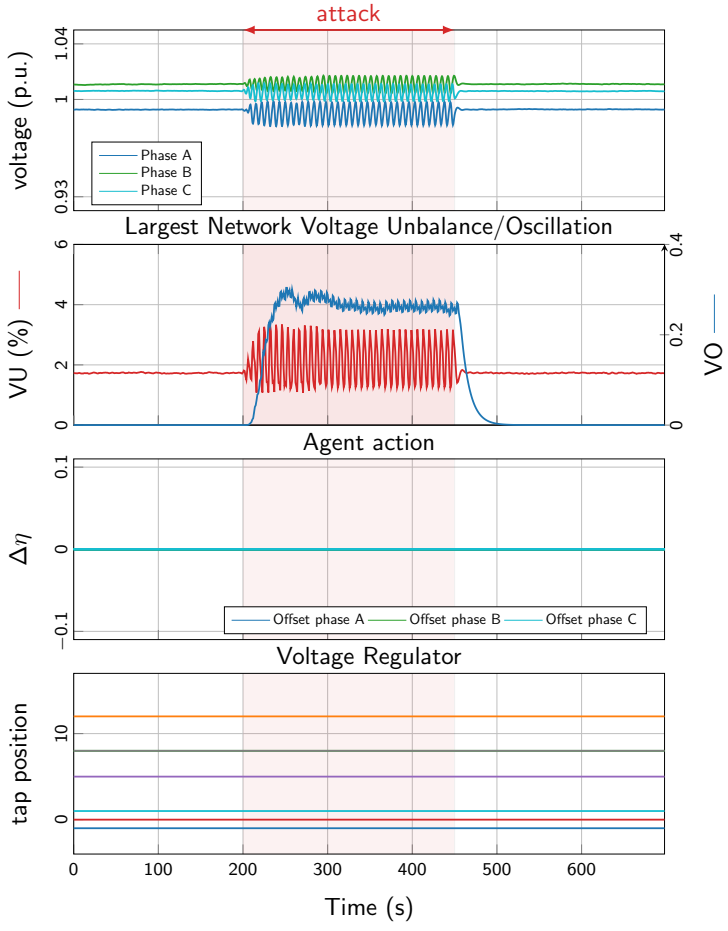
(d) Defense

BERKELEY LAB

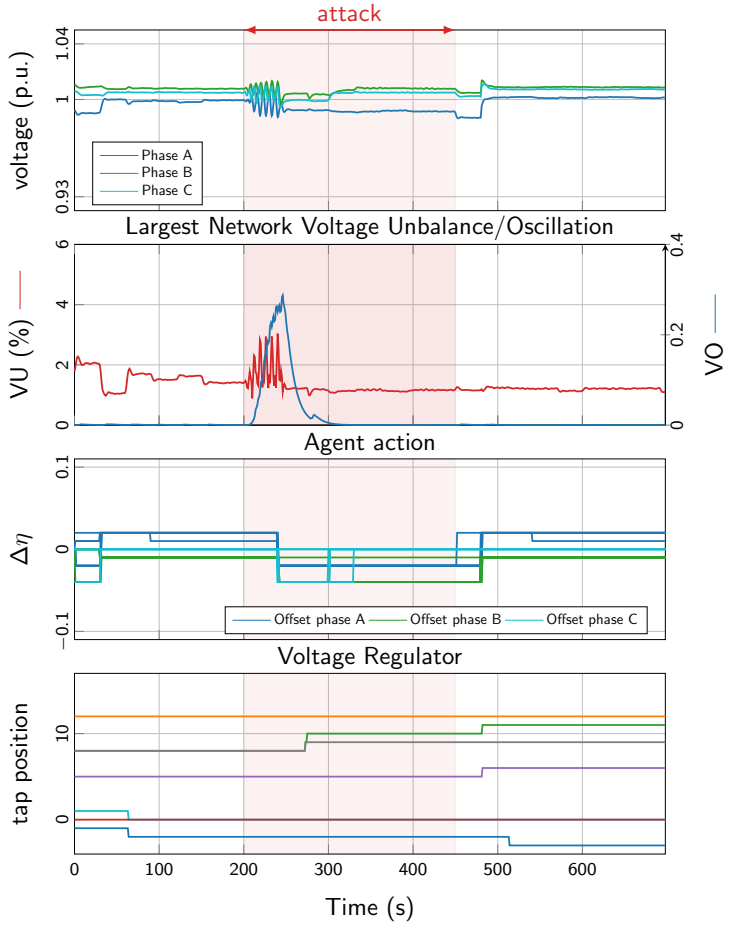
IEEE 123 Feeder



IEEE 123 Node Oscillation Attack with Imbalance - 40%



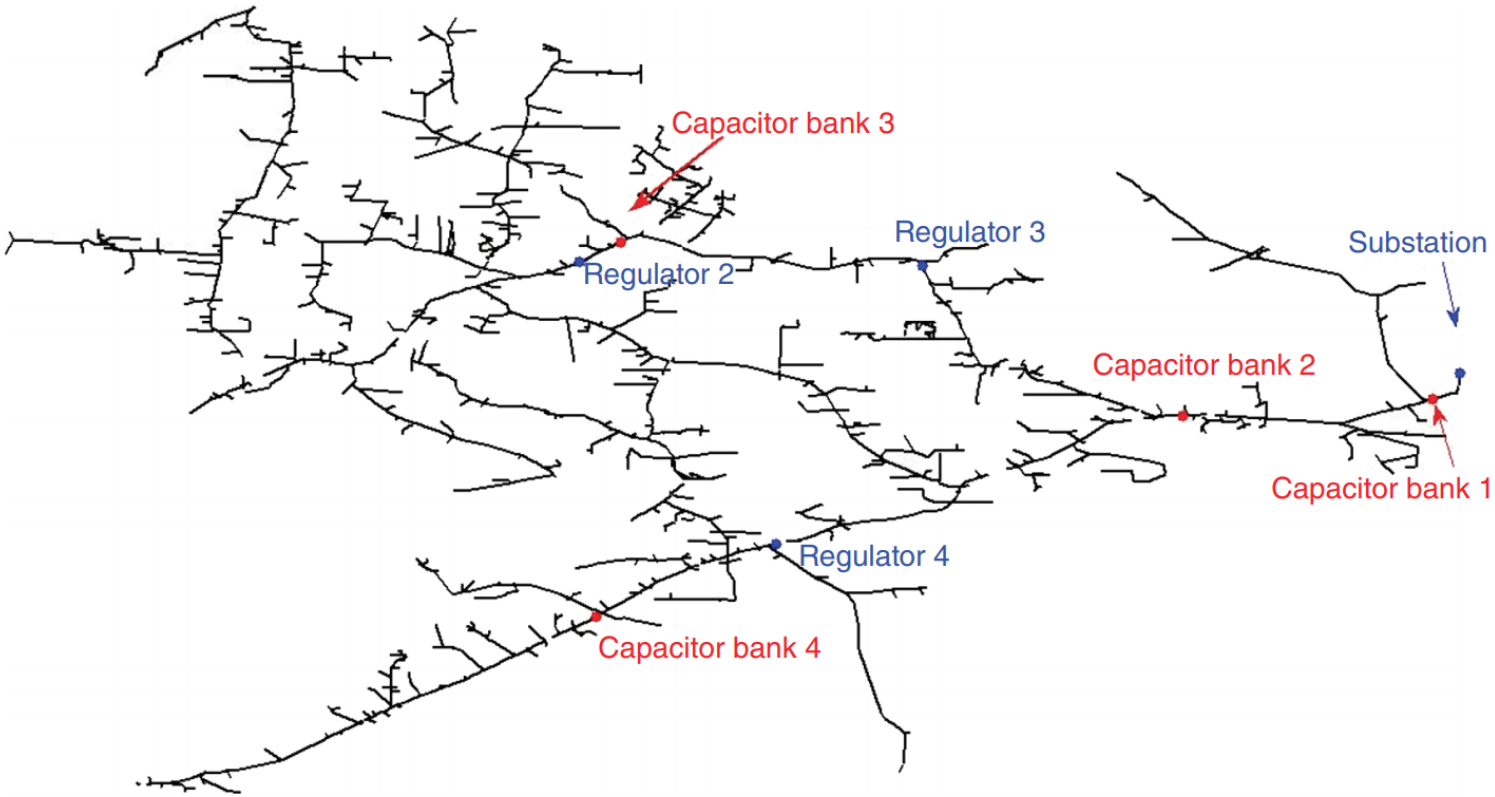
(e) No defense



(f) Defense

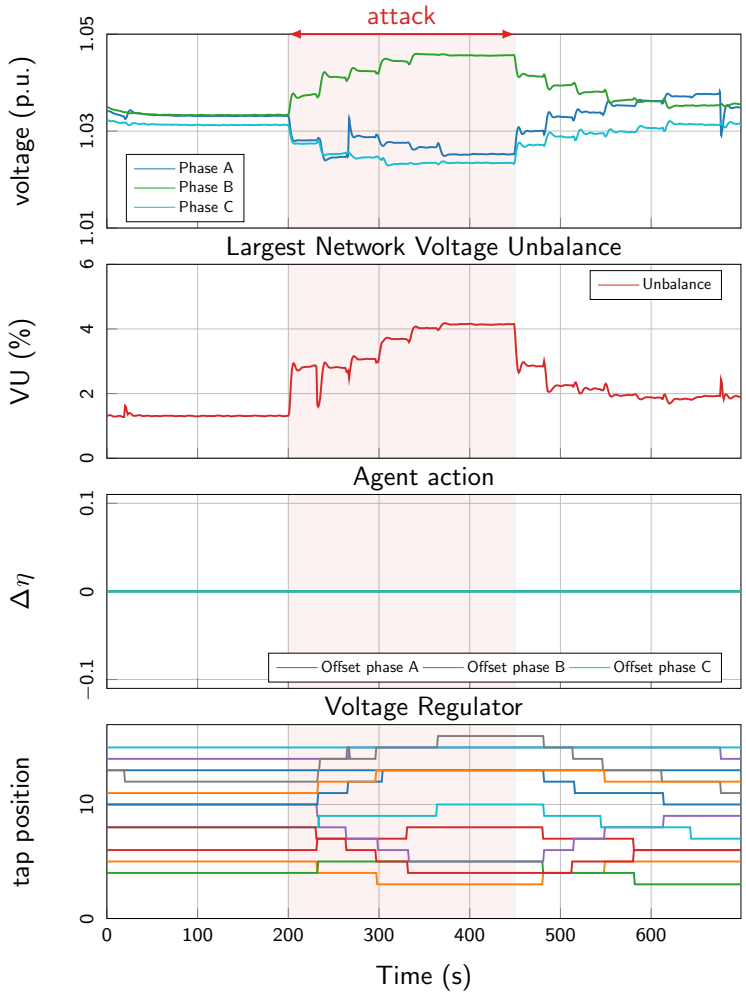
BERKELEY LAB

IEEE 8500 Feeder

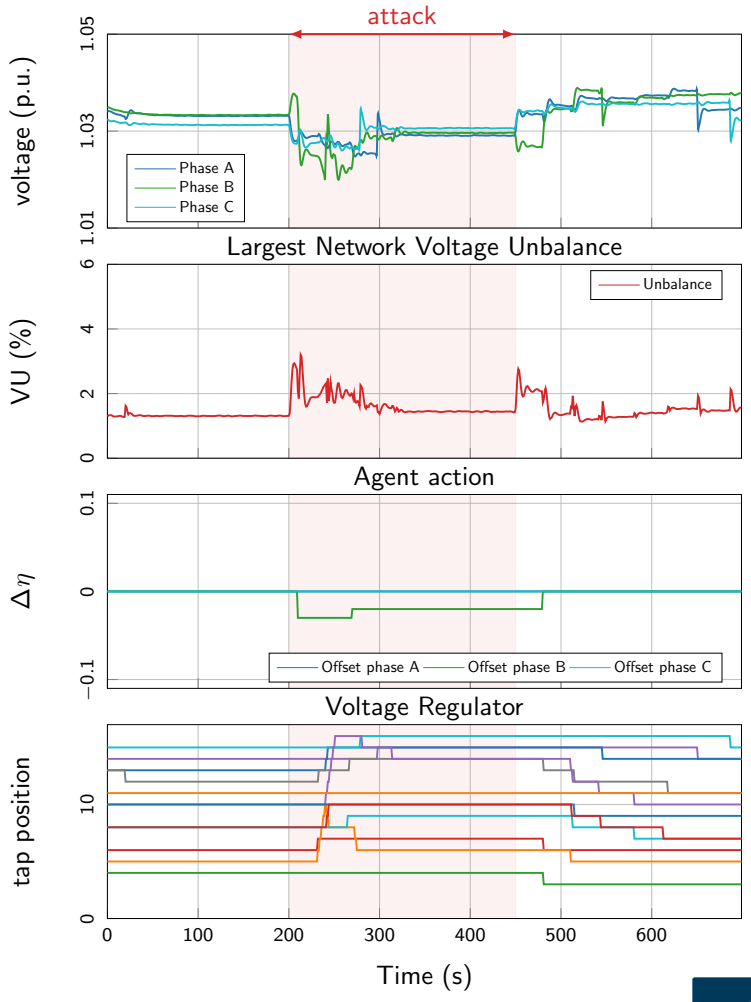


BERKELEY LAB

IEEE 8500 Node Imbalance Attack - 20%



(g) No defense



(h) Defense

BERKELEY LAB