Cybersecurity via Inverter Grid Automatic Reconfiguration (CIGAR) Year 3 Report

Authors
Peisert, Sean
Arnold, Daniel
Roberts, Ciaran
et al.

Publication Date
2023-12-13

Peer reviewed

Cybersecurity via **I**nverter **G**rid **A**utomatic **R**econfiguration (**CIGAR**)
Year 3 Report

PIs: Sean Peisert and Daniel Arnold

Team Members: Ciaran Roberts, Sy-Toan Ngo, Michael Sankur,
Anna Scaglione, Ignacio Losada Carreno, Shammya Saha
Anton Kocheturov, Dmitriy Fradkin
David Pinney, Ryan Mahoney, Lisa Slaughter

Submitted: March 31 2021

1 Executive Summary

This report outlines work undertaken during the third year of the **C**ybersecurity via **I**nverter **G**rid **A**utomatic **R**econfiguration (CIGAR) project, lead by Lawrence Berkeley National Lab (LBNL) and supported by Arizona State University (ASU), Siemens Corporate Technologies, and the National Rural Electric Cooperative Association (NRECA).

Project CIGAR focuses on using the tools from artificial intelligence, specifically Reinforcement Learning (RL), to design algorithms to mitigate the effect of cyberattacks on the autonomous control systems in solar photovoltaic power inverters. Emerging standards such as IEEE 1547 and California Rule 21 encourage the incorporation of various autonomous control functions into solar inverters that will modulate the power injection/consumption of these units in response to locally sensed grid conditions. A thorough overview of the types of smart inverter control functions that Distributed Energy Resources (DER), such as solar photovoltaic and battery systems, could employ is provided in [1]. As the parameters of these control systems can be remotely configured [2], the possibility exists that a hostile entity could gain access to the mechanisms that can wirelessly pass new parameters to these control functions in a portion of DER inverters in a given circuit.

The ramifications of these kinds of attacks are not limited to inconvenience to the asset owner (e.g. the inability to sell power back to the grid or provide power to a residence) or damage to the asset itself. The interaction of these devices with the electric distribution grid can produce very harmful behavior in the form of large oscillations in voltages and power flows [3,4]. Additionally, recent work by the CIGAR team has shown that attacks on smart inverter control systems, coupled with normal voltage regulator behavior, could create large voltage imbalances [5]. In either case, exploitation of the ability to manipulate smart inverter control functions could lead to devices tripping off of the network, or damage to residential/commercial/industrial and utility assets.

In the face of cyberattacks on a subset of DER smart inverters in a given circuit, the CIGAR project is using artificial intelligence to design control systems that manage DER smart inverters which *have not been compromised as part of the cyberattack*. Our work has shown that, given enough non-hacked units in a system, it is possible to adjust the control parameters in non-compromised units to mitigate oscillations and large voltage imbalances caused by hacked units.

This report discusses work undertaken during year 3 of the CIGAR project (which is a 3 year effort). Year 1 of the project focused on the development of the underlying simulation engine needed to train Reinforcement Learning (RL)-based control architectures. Year 2 of the CIGAR project focused on using this training environment to create trained neural networks capable of mitigating cyberattacks on smart inverter autonomous control functions. Year 3 of the CIGAR project focused on training of the RL controller for robustness in choice of network architecture and weather conditions as well as the integration of the CIGAR solution (which we refer to as "PyCIGAR") into the NRECA Open Modeling Framework (OMF). Integration of the RL controller into the OMF allows NRECA member utilities to conduct power flow simulations to understand the effect of threats to DER smart inverter functions in their networks and how their networks could be defended (via the CIGAR solution). As such, this report describes the use of the CIGAR technology to mitigate the effect of two types classes of cyber attacks on sets of compromised DER: 1) attacks designed to create oscillations in system voltages, and 2) attacks designed to create large voltage imbalances in three phase systems. Additionally, this report describes "CyberInverters" module in the OMF, which houses the PyCIGAR software, and discusses the OMF user interface which is used to conduct cyber attack experiments.

This report consists of 5 sections. Section 2 provides an overview of smart inverter voltage regulation functions which are simultaneously the primary threat vectors and the means by which the RL controllers restabilize the distribution grid. Section 3 discusses the reinforcement learning algorithm architecture, the PyCIGAR software package, and presents results showing the behavior of the RL controller in mitigating simulated cyberattacks in OpenDSS. Section 4 shows the API developed by NRECA and LBL that will facilitate user interaction with the PyCIGAR software (once this is integrated into the OMF in Year 3). A discussion of supplemental software built by Arizona State University for generation of synthetic solar generation data is presented in the Appendix.

2 Preliminaries

This section discusses the CIGAR control paradigm, smart inverter Volt-VAR and Volt-Watt control functionality, and presents a stability analysis showing how manipulations of smart inverter autonomous control functions could lead to system voltage instabilities.

2.1 Control Paradigm

The CIGAR project seeks to adjust settings in non-compromised DER solar inverters in the event that a portion of the solar photovoltaic inverters in a given network have been compromised as part of a cyberattack. Given the presence of firmware update systems designed to issue new control setpoints to smart inverters, it is logical to assume that only a portion of the solar photovoltaic inverters in a given distribution grid could be compromised [2]. In this situation, the reinforcement learning algorithms designed in the CIGAR project would issue new setpoints to the remaining set of inverters in the grid in an attempt to mitigate the effect of the cyberattack.

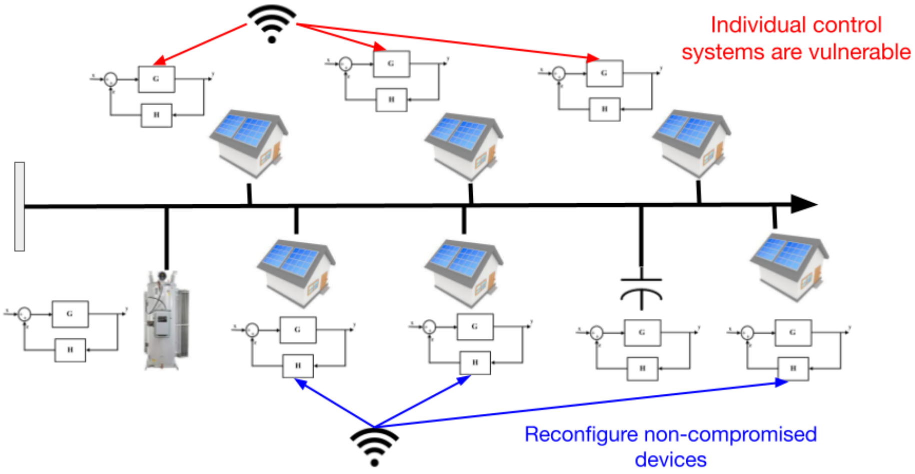

Individual control systems are vulnerable
G
H
Reconfigure non-compromised devices

Figure 1: Depiction of CIGAR control paradigm

The cyber attack and control paradigm is illustrated in Fig. 1 which depicts a distribution grid populated with legacy voltage control systems (voltage regulators and capacitor banks) and several homes equipped with rooftop solar units. The feedback control block diagrams associated with these devices are meant to illustrate that the behavior of each device is governed by feedback control mechanisms whose inputs are local grid states (e.g., voltage, current, etc.). The cyberattack on DER smart inverter control functions (represented by the block diagram feedback loops) is depicted by the red arrows, indicating that the attacker can only adjust a subset of the smart inverters in the system.

The CIGAR control paradigm is illustrated by the blue arrows, which represent the adjustment of the remaining non-compromised set of smart inverters (by the CIGAR reinforcement learning algorithm) to attempt to mitigate the cyberattack. One of the main benefits of the use of reinforcement learning is the ability of the approach to take into account the presence of other dynamics in the system, in the case of Fig. 1 these are the voltage regulator and the capacitor bank, during the decision-making process. Thus, the CIGAR solution is capable of exploiting the presence of other dynamics in the system to determine setpoints for the non-compromised set of DER to best mitigate the cyberattack.

2.2 Linear Power Flow Model Derivation

This section lays the groundwork for a stability analysis that links changes in the parameters of smart inverter autonomous control functions to system voltage instabilities. This section presents a linearized AC power flow model which is subsequently employed in the upcoming stability analysis.

Let the graph $\mathcal{G} = (\mathcal{N} \cup \{0\}, \mathcal{L})$ represent a balanced radial distribution feeder, where $\mathcal{N}$ is the set of nodes (excluding the substation) and $\mathcal{L}$ is the set of line segments, where $|\mathcal{N}| = |\mathcal{L}| = n$. For a given bus $i \in \mathcal{N}$, let $\mathcal{L}_i$ (where $\mathcal{L}_i \subseteq \mathcal{L}$) denote the collection of line segments from node 0 (e.g. the substation) to node i. The *DistFlow* equations [6] capture the relationship between power flowing in line segment $(i, j) \in \mathcal{L}$ and the voltage magnitude drop between nodes i and j:

$$P_{ij} = p_j^c - p_j^g + r_{ij}c_{ij} + \sum_{k:(j,k)\in\mathcal{L}} P_{jk} \tag{1a}$$

$$Q_{ij} = q_j^c - q_j^g + x_{ij}c_{ij} + \sum_{k:(j,k)\in\mathcal{L}} Q_{jk} \tag{1b}$$

$$v_j^2 - v_i^2 = -2\big(r_{ij}P_{ij} + x_{ij}Q_{ij}\big) + \big(r_{ij}^2 + x_{ij}^2\big)c_{ij}^2, \tag{1c}$$

where v_i^2 is node i squared voltage magnitude, P_{ij} and Q_{ij} denote the active/reactive power flowing in line segment (i, j), r_{ij} and x_{ij} are line segment (i, j) resistance and reactance, and c_{ij} are losses. For node i, active (reactive) power consumption is denoted by p_i^c (q_i^c) and active (reactive) power generation, due to DER, is denoted by p_i^g (q_i^g).

Consistent with [3, 7], we neglect losses in (1a) - (1c) which is achieved via setting $c_{ij} = 0$ for all $(i, j) \in \mathcal{L}$. Furthermore, as $v_i \approx 1$ we approximate $v_j^2 - v_i^2 \approx 2(v_j - v_i)$. Let $\beta(j)$ denote the set of all nodes descended from j (including j itself). With these changes, the *DistFlow* model becomes:

$$P_{ij} = \sum_{k\in\beta(j)} \big(p_k^c - p_k^g\big) \tag{2a}$$

$$Q_{ij} = \sum_{k\in\beta(j)} \big(q_k^c - q_k^g\big) \tag{2b}$$

$$v_i - v_j = r_{ij}P_{ij} + x_{ij}Q_{ij}. \tag{2c}$$

The now linearized system of (2a) - (2c) can be more compactly represented via substituting (2a) and (2b) into (2c) and making successive substitutions of voltages from upstream nodes yielding node i voltage as a function of feeder head voltage v_0. Defining the following vectors:

$$\mathbf{v} = [v_1, \ldots, v_n]^T, \quad \mathbf{v_0} = v_0 \mathbf{1} \tag{3a}$$

$$\mathbf{p^c} = [p_1^c, \ldots, p_n^c]^T, \quad \mathbf{p^g} = [p_1^g, \ldots, p_n^g]^T \tag{3b}$$

$$\mathbf{q^c} = [q_1^c, \ldots, q_n^c]^T, \quad \mathbf{q^g} = [q_1^g, \ldots, q_n^g]^T, \tag{3c}$$

then the system of (2a) - (2c) can be recast in vector form:

$$\mathbf{v} = \mathbf{v_0} + \mathbf{R}(\mathbf{p^g} - \mathbf{p^c}) + \mathbf{X}(\mathbf{q^g} - \mathbf{q^c}), \tag{4}$$

where $\mathbf{R}$ and $\mathbf{X}$ are completely positive matrices [3] and

$$R_{ij} = \sum_{(h,k) \in \mathcal{L}_i \bigcap \mathcal{L}_j} r_{hk} \tag{5a}$$

$$X_{ij} = \sum_{(h,k) \in \mathcal{L}_i \bigcap \mathcal{L}_j} x_{hk}. \tag{5b}$$

Defining $\mathbf{Z} = [\mathbf{R}, \mathbf{X}]$, $\mathbf{s}^c = [\mathbf{p}^c, \mathbf{q}^c]^T$, and $\mathbf{s}^g = [\mathbf{p}^g, \mathbf{q}^g]^T$, (4) can expressed compactly as:

$$\mathbf{v} = \mathbf{v_0} + \mathbf{Z}(\mathbf{s}^g - \mathbf{s^c}), \tag{6}$$

2.3 Smart Inverter Models

This section discusses the autonomous smart inverter control functions which are the mechanisms manipulated to create undesired grid conditions during a cyber attack.

Smart inverter VV and VW functions compute reactive and active power setpoints, respectively, as functions of deviations of locally sensed voltages from a nominal value (typically 1 p.u.). Let $f_{p,i}(v_i)$ and $f_{q,i}(v_i)$ denote the VW and VV control functions at node i. We make the following assumptions regarding these functions [3, 8, 9]:

Assumption 1. *The functions $f_{p,i}(v_i)$ and $f_{q,i}(v_i)$ are monotonically decreasing and continuously piece-wise differentiable.*

Assumption 2. *Both $f_{p,i}(v_i)$ and $f_{q,i}(v_i)$ have bounded derivatives, i.e. there exists $C_{p,i} < +\infty$ and $C_{q,i} < +\infty$ such that $|f'_{p,i}(v_i)| \leq C_{p,i}$ and $|f'_{q,i}(v_i)| \leq C_{q,i}$ for all v_i.*

Let $\bar{s}_i$ denote the rated apparent power (i.e. the inverter capacity) of the smart inverter at node i. Similarly, let $\bar{p}_i$ denote the maximum available real power capable of being sourced at the present irradiance level. Following the analysis of [10], $\bar{p}_i$ can be expressed as a fraction of the capacity of the i^{th} inverter:

$$\bar{p}_i = \lambda \bar{s}_i, \quad 0 < \lambda \leq 1, \tag{7}$$

where $\lambda = 1$ corresponds to the inverter generating the maximum amount of real power. In situations where the amount of real power generated is less than $\bar{s}_i$, some inverter devices support the use of the excess system capacity for reactive power generation. The maximum amount of reactive power available for injection/consumption, denoted by $\bar{q}_i(v_i)$, is a function of hardware limitations (q_i^{lim}) and the available reactive power [9]:

$$\bar{q}_i(v_i) = \min\left(q_i^{\text{lim}}, \sqrt{\bar{s}_i^2 - f_{p,i}^2(v_i)}\right). \tag{8}$$

Given the definitions of (7) - (8), generic VV and VW control functions are depicted in Figs. 2 - 3, respectively. A derivation of the Lipschitz constants for both the VV and VW control functions can be found in [9]. Note that both piece-wise linear functions are parameterized by the vector $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_5]$.

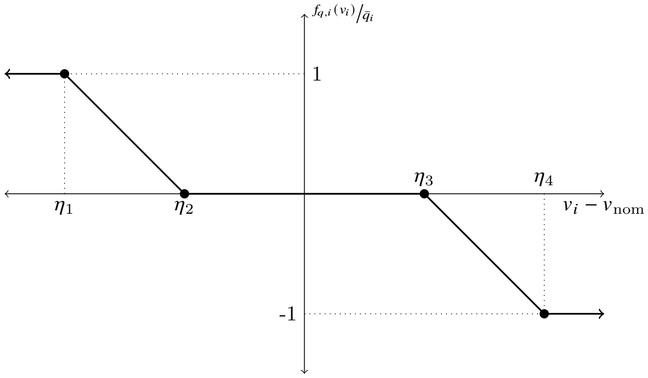


Figure 2: Inverter Volt-VAR curve. Positive values denote VAR injection. v_{nom} is the nominal voltage value.

A block diagram of the smart inverter model is shown in Fig. 4 with VV and VW control logic included in the blue dashed region. Dynamics associated with measuring the grid voltage v are captured by the low pass filter $H_m(s)$ which is used to produce the measured voltage $\hat{v}$ that is input to the VV and VW controllers. The maximum available real power from the solar array, $\bar{p}$, is also input into the VW controller, which along with $\hat{v}$, determines the maximum amount of reactive power available for injection/consumption $\bar{q}$ that is then input to the VV controller. The active and reactive power setpoints produced by the VW and VW controllers are then low pass filtered by $H_O(s)$ to produce the active and reactive power injections that are injected into the grid. These filters serve to limit the rate at which the active and reactive powers injected by PV systems can change and do not represent physical constraints of the smart inverter devices themselves [1].

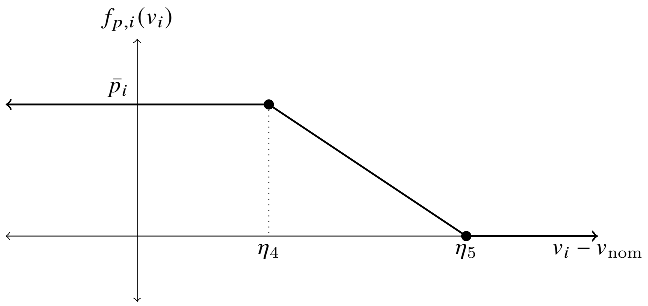


Figure 3: Inverter Volt-Watt curve. Positive values denote watt injection. v_{nom} is the nominal voltage value.

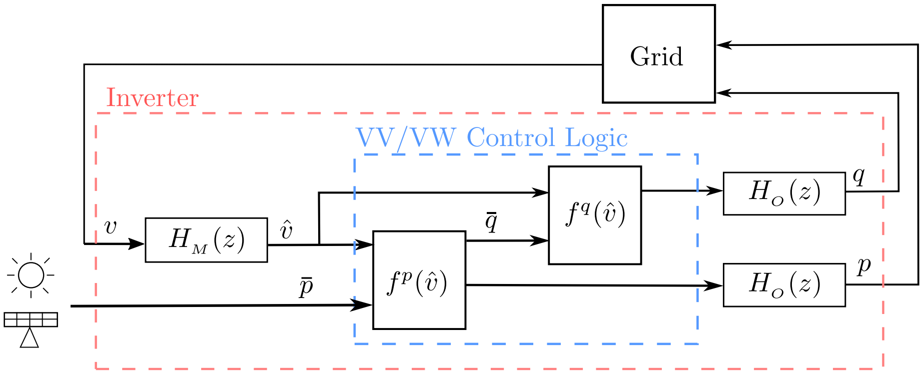


Figure 4: Block diagram of VV and VW control logic of an inverter.

2.4 Stability Analysis

In this section we explore the stability of aggregations of smart inverters modeled by Fig. 4. Without loss of generality, we assume the presence of a VV and VW capable smart inverter at each node in the system. Under the additional assumption that active and reactive power consumption change slowly with respect to inverter control actions, (6) can be recast in the following form:

$$\mathbf{v} = \mathbf{Z}\mathbf{s} + \underbrace{\mathbf{v_0} - \mathbf{Z}\mathbf{s}^{\mathbf{c}}}_{\bar{\mathbf{v}}}, \tag{9}$$

where the superscript has been dropped from $\mathbf{s}$ for convenience and $\bar{\mathbf{v}}$ is treated as constant. Consistent with Fig. 4, the following formulation explicitly models the dynamics associated with measuring the instantaneous voltage (Eq. (9)) *and* computing new active/reactive power injections. Both processes are modeled using first order low pass filters. Letting the vector $\hat{\mathbf{v}}$ denote the collection of *measured* voltages, the inverter dynamics can be expressed as:

$$\mathbf{T}_v \hat{\mathbf{v}} = \mathbf{Z}\mathbf{s} + \bar{\mathbf{v}} - \hat{\mathbf{v}}, \tag{10a}$$

$$\mathbf{T}_s \mathbf{s} = \mathbf{f}(\hat{\mathbf{v}}) - \mathbf{s}, \tag{10b}$$

where $\mathbf{T}_v \in \mathbb{R}^{n \times n}$ and $\mathbf{T}_s \in \mathbb{R}^{2n \times 2n}$ are time constants of measurement and power injection update dynamics (diagonal and positive definite).

In this model, $\mathbf{f}(\hat{\mathbf{v}}) = [\mathbf{f}_p(\hat{\mathbf{v}}), \mathbf{f}_q(\hat{\mathbf{v}})]^T$ is the collection of inverter VV and VW functions at each node in $\mathcal{G}$, where:

$$\mathbf{f}_p(\hat{\mathbf{v}}) = [f_{p,1}(\hat{v}_1), \dots, f_{p,n}(\hat{v}_n)]^T \tag{11a}$$

$$\mathbf{f}_q(\hat{\mathbf{v}}) = [f_{q,1}(\hat{v}_1), \dots, f_{q,n}(\hat{v}_n)]^T, \tag{11b}$$

where, according to Assumptions 1-2, both $f_{p,i}(v_i)$ and $f_{q,i}(v_i)$ are locally Lipschitz with constants $C_{p,i}$ and $C_{q,i}$, respectively. Define the matrices

$$\mathbf{C}_p = \text{diag}([C_{p,1}, \dots, C_{p,n}]) \tag{12a}$$

$$\mathbf{C}_q = \text{diag}([C_{q,1}, \dots, C_{q,n}]) \tag{12b}$$

$$\mathbf{C}_s = \begin{bmatrix} \mathbf{C}_p & \mathbf{C}_q \end{bmatrix}^T. \tag{12c}$$

The following proposition ties the stability of (10a) - (10b) to the system impedances, $\mathbf{Z}$, and $\mathbf{C}_s$.

Proposition 1. *The system of* (10a) - (10b) *is asymptotically stable if:*

$$\|\mathbf{Z}\mathbf{C}_s\|_2 \leq 1 \tag{13}$$

Proof. Noting the equilibrium $(\hat{\mathbf{v}}^*, \mathbf{s}^*)$ of (10a) - (10b) is

$$\mathbf{0} = \mathbf{Z}\mathbf{s}^* + \bar{\mathbf{v}} - \hat{\mathbf{v}}^* \tag{14a}$$

$$\mathbf{0} = \mathbf{f}(\hat{\mathbf{v}}^*) - \mathbf{s}^*, \tag{14b}$$

define the shifted set of coordinates $\Delta\mathbf{v} = \hat{\mathbf{v}} - \hat{\mathbf{v}}^*$, $\Delta\mathbf{s} = \mathbf{s} - \mathbf{s}^*$ which translate the equilibrium to the origin. The dynamics in the new coordinate system are:

$$\mathbf{T}_v \Delta\mathbf{v} = \Big(\mathbf{Z}\Delta\mathbf{s} - \Delta\mathbf{v}\Big), \tag{15a}$$

$$\mathbf{T}_s \Delta\mathbf{s} = \Big(\mathbf{f}(\Delta\mathbf{v} + \hat{\mathbf{v}}^*) - \mathbf{f}(\hat{\mathbf{v}}^*) - \Delta\mathbf{s}\Big). \tag{15b}$$

Define the vector

$$\mathbf{g}(\Delta\mathbf{v}, \Delta\mathbf{s}) = \begin{bmatrix} \mathbf{Z}\Delta\mathbf{s} \\ \mathbf{f}(\Delta\mathbf{v} + \hat{\mathbf{v}}^*) - \mathbf{f}(\hat{\mathbf{v}}^*) \end{bmatrix}. \tag{16}$$

The 2-norm of the vector $\mathbf{g}$ can be expressed as:

$$\|\mathbf{g}\|_2 \leq \left\|\mathbf{Z}\Delta\mathbf{s}\right\|_2 + \left\|\mathbf{f}(\Delta\mathbf{v} + \hat{\mathbf{v}}^*) - \mathbf{f}(\hat{\mathbf{v}}^*)\right\|_2 \tag{17a}$$

$$\leq \left\|\mathbf{Z}\Delta\mathbf{s}\right\|_2 + \left\|\mathbf{C}_s \Delta\mathbf{v}\right\|_2 \tag{17b}$$

$$= \left\| \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{Z} \\ \mathbf{C}_s & \mathbf{0}_{2n \times 2n} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v} \\ \Delta\mathbf{s} \end{bmatrix} \right\|_2 \tag{17c}$$

$$\leq \left\| \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{Z} \\ \mathbf{C}_s & \mathbf{0}_{2n \times 2n} \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} \Delta\mathbf{v} \\ \Delta\mathbf{s} \end{bmatrix} \right\|_2. \tag{17d}$$

Let $\Delta\mathbf{y} = [\Delta\mathbf{v}, \Delta\mathbf{s}]^T$. The dynamics of (15a) - (15b) can be expressed in terms of $\Delta\mathbf{y}$:

$$\mathbf{T}\Delta\mathbf{y} = \Big(-\Delta\mathbf{y} + \mathbf{g}(\Delta\mathbf{y})\Big), \tag{18}$$

where $\mathbf{T} = \text{diag}([\mathbf{T}_v, \mathbf{T}_s])$.

Using the Lyapunov function $V = \frac{1}{2}\Delta\mathbf{y}^T\mathbf{T}\Delta\mathbf{y}$, the derivative of the state trajectories of (18) along V are:

$$\begin{aligned} \dot{V} &= -\Delta\mathbf{y}^T\Delta\mathbf{y} + \Delta\mathbf{y}^T\mathbf{g}(\Delta\mathbf{y}) && \text{(19a)} \\ &\leq \|\Delta\mathbf{y}\|_2^2 + \|\Delta\mathbf{y}\|_2\|\mathbf{g}(\Delta\mathbf{y})\|_2 && \text{(19b)} \\ &\leq \left(-1 - \left\|\begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{Z} \\ \mathbf{C}_s & \mathbf{0}_{2n\times 2n}\end{bmatrix}\right\|\right)\|\Delta\mathbf{y}\|_2^2, && \text{(19c)} \end{aligned}$$

where the terms in parenthesis can be equivalently expressed as $\|\mathbf{Z}\mathbf{C}_s\|_2 \leq 1$. □

Proposition 1 shows that the stability of the feedback interconnection of aggregations of VV/VW controllers and the electric distribution grid will become unstable if the magnitude of the largest slopes of the VV/VW curves (represented by $\mathbf{C}_p$ and $\mathbf{C}_q$) become too steep. Simulated cyberattacks in experiments conducted in CIGAR verify this relationship. This analysis provided the means to create simulated attacks which served as the training environment for reinforcement learning, which is detailed in the next section.

3 PyCIGAR - Reinforcement Learning Control

3.1 Introduction

In recent years, Reinforcement Learning (RL) has become an extremely popular method for solving stochastic optimal control problems which, only a decade ago, were considered intractable. The marriage of RL with deep learning has made it possible to utilize neural networks to approximate complicated conditional probability distributions that to represent optimization value functions or control policies. Modern Deep Reinforcement Learning (DRL) techniques have successfully been applied in a variety of domains and show great promise for addressing problems in the electric power sector. In [11], the authors use the Deep Q Learning (DQN), a reinforcement learning algorithm that combines Q-Learning with deep neural networks, to control both generator dynamic braking and load shedding in the event of a contingency to ensure post-fault recovery. The authors in [12] consider the problem of coordinated voltage regulation using capacitors and smart inverters. Exploiting the timescale separation of these devices, they solve a SOCP convex optimization problem to determine the control policies of the smart inverters while using a DQN network to learn an optimal policy for the capacitor bank switching. In [13], a Deep Deterministic Policy Gradient (DDPG) RL approach is used to coordinate DER to directly modulate active and reactive power injections to regulate grid voltages to maintain normal operating ranges.

The work undertaken in the CIGAR project differs from those described in that this project focuses on developing a control policy that remains inactive during normal operation and is activated during sustained abnormal system conditions (e.g. a during a cyber attack). As such, this controller should not impact a converters response to normal disturbance, e.g. line-to-ground faults. This controller is motivated by cyber-physical security, however, it is agnostic to the cause of the abnormal conditions. Consequently, it can also serve to autonomously re-dispatch controller settings in the event that an intended action has resulted in abnormalities, e.g. networking microgrids with locally optimized controllers.

The remainder of this section is organized as follows. Section 3.2 gives a brief introduction to DRL and discusses the architecture of the training environment (named PyCIGAR) built by the project team. Section 3.3 outlines the control architecture for the use of reinforcement learning to adjust smart inverter control parameters. Section 3.4 presents preliminary results showing the trained agent defeating a cyberattack on inverter DER smart inverter functions.

3.2 Deep Reinforcement Learning

3.2.1 Overview

Reinforcement Learning (RL) is a paradigm of machine learning specialized in solving Markov Decision Processes (MDP) where the state transitions are unknown or too complex to model explicitly. The *agent* is a decision-maker who interacts with the MPD according to a specific policy in an attempt to maximize a particular objective. An environment defined as a MDP has:

- A state space $\mathcal{S}$ containing states observable by the agent;
- An action space $\mathcal{A}$ containing all the possible actions the agent can execute;
- A state transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow [0, 1]$ giving the probability distribution over the next state s' when an action a is taken at state s;
- A reward function $\mathcal{R} : \mathcal{A} \times \mathcal{S} \times \mathcal{S} \longrightarrow \mathbb{R}$ giving the reward received by the agent when the environment transitions from state s to state s' with action a;
- A discount factor $\gamma \in [0, 1]$ representing the trade-off between immediate and future rewards.

The interaction of the agent with the environment described a MDP is depicted in Fig. 5.

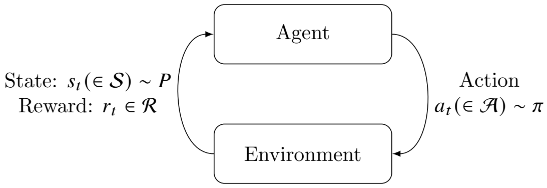

Agent
State: $s_t(\in \mathcal{S}) \sim P$
Reward: $r_t \in \mathcal{R}$
Action
$a_t(\in \mathcal{A}) \sim \pi$
Environment

Figure 5: Agent-Environment interaction in Reinforcement Learning

A RL agent learns optimal actions by interacting with its environment and maximizing its received rewards $r_t \in \mathbb{R}$, dependent on its actions $a_t \in \mathcal{A}$ and the states of the environment $s_t \in \mathcal{S}$. Its objective is to maximize the discounted reward $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t\right]$, with T the terminal time step, by following a deterministic policy $\pi : \mathcal{S} \longrightarrow \mathcal{A}$ or a stochastic one $\pi : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$.

3.2.2 Policy Gradient and PPO

Classic RL relies on feature engineering and is difficult to apply to environments with high dimensional, continuous action / state spaces [14]. Deep RL (DRL) solves these issues by leveraging neural networks and gradient-based optimization to learn the best features and approximate a value function or policy distribution from past experiences. DRL has been successfully applied to robotic control [15], video game [16] [17] and board game [18] [19] playing, and is recently gaining in popularity in a variety of domains [20].

There are many different implementation of DRL. Some approaches use neural networks to explicitly model the value function (or action-value function) to predict the expected reward of a particular state (or action-value pair). Other approaches use neural networks to characterize the policy π governing the behavior of the agent. Given a state s, the policy π will generate the action the agent will follow in interacting with the environment. In fact, some approaches use neural networks to model both the value function and the policy!

A variety of RL architectures have been explored by the CIGAR project team. Thus far, the most reliable have been from a family of approaches known as *Policy Gradient* methods. Policy Gradient methods explicitly model the control policy π of the agent with a neural network, which is repeatedly improved using gradient ascent based samples of rewards from the environment. In its most basic form, the Vanilla Policy Gradient (VPG) method is known to have high variance. One method for reducing this variance is to extend VPG to explicitly model the value function with a neural network (this is an example of an actor-critic method) [21].

Let $\pi_\theta(a|s)$ be a stochastic policy parameterized by θ, that models the probability distribution of action $a \in \mathcal{A}$ given the state $s \in \mathcal{S}$. Let $V^\pi_\phi(s)$ be a value function parameterized by ϕ, estimating the cumulative discounted reward from the current state to the terminal state. The gradient of $J(\theta)$ is:

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\big[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A^\pi_\phi(s_t, a_t)\big] \tag{20}$$

where τ is the trajectory generated by policy π_θ and $A^\pi_\phi(s_t, a_t) = r_t + \gamma V^\pi_\phi(s_{t+1}) - V^\pi_\phi(s_t)\big)$ is the advantage estimation. The policy and value function are updated by gradient ascent:

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta J(\theta) \tag{21}$$

$$\phi_{k+1} = \phi_k + \beta\nabla_\phi(r_t + V^\pi(s_{t+1}) - V^\pi(s_t))) \tag{22}$$

It is known that algorithm of (20) - (22) can be unstable when a large update to the policy agent π_θ occurs. To remedy this, approaches such as Trust Region Policy Optimization (TRPO) [22] are used. TRPO limits the amount by which the policy can be updated at each iteration in training. Proximal Policy Optimization (PPO) [23] simplifies this method and keeps similar performance by using a clipped surrogate objective. PPO is a state-of-the-art method that was successfully used in video games [17] and robotics in simulation [24]. *PPO is presently the employed architecture for the reinforcement learning agent training in the CIGAR project.*

3.2.3 PyCIGAR Architecture

Broadly speaking, there were 2 pieces of software written as part of the CIGAR project: 1) the reinforcement learning training environment, and 2) extensions to the NRECA Open Modeling Framework (OMF) to house the trained agent. Year 2 of the CIGAR project predominantly focuses on the former while laying the groundwork for the latter. Year 3 of CIGAR focused on refining the reinforcement learning environment and integration of trained RL agents into the OMF. The former piece of software has been dubbed "PyCIGAR" to emphasize the python programming language used as the glue to stitch together the different elements of the training environment. PyCIGAR is a Python library for distributed reinforcement learning in a complex environment containing rule-based control devices and RL control devices. To our best knowledge, at this time no such alternative environment has been developed to conduct large scale reinforcement learning on electric power distribution systems. The elements of PyCIGAR are detailed in Fig. 6

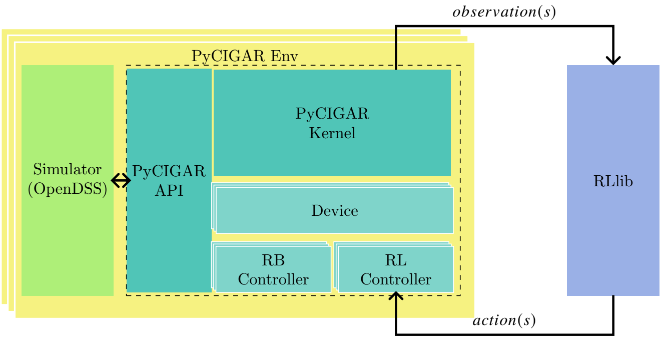


Figure 6: Software elements of PyCIGAR

PyCIGAR is a link between power system simulators and a reinforcement learning library - RLlib [25]. The PyCIGAR library provides a unified API to integrate with different power system simulators without much effort (OpenDSS and GridLAB-D have been tested). Additionally, PyCIGAR uses RLlib to have the capacity of deploying large scale experiments on a server, machine cluster or cloud computer. With PyCIGAR, researchers can easily modify experiment configurations, experiment networks, modify attack scenarios and train agent(s) to mitigate threats on networks.

3.3 CIGAR Control Architecture

Figure 7 shows the smart inverter model discussed in Fig. 4 with the addition of an Artificial Neural Network (ANN) that issues new control setpoints to the inverter Volt-VAR and Volt-Watt control functions.

Figure 7 models the dynamics associated with measuring the grid voltage v as a low pass filter $H_m(s)$ which is used to produce the measured voltage $\hat{v}$. $\hat{v}$ is then input to the VV and VW control functions. The maximum available real power from the solar array, $\bar{p}$, is also input into the VW controller, which along with $\hat{v}$, determines the maximum amount of reactive power available for injection/consumption $\bar{q}$ that is then input to the VV controller. The active and reactive power setpoints produced by the VW and VW controllers are then low

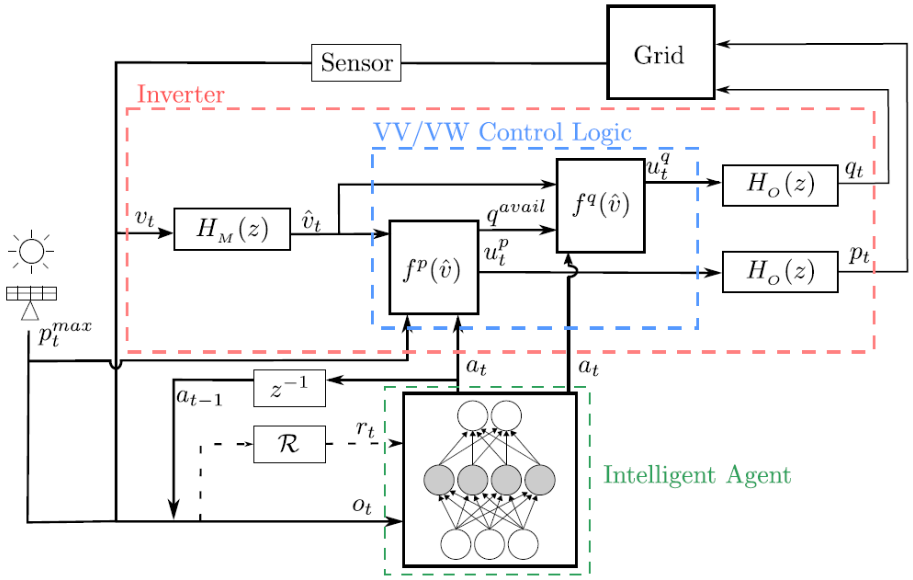


Figure 7: Block diagram of VV and VW control logic of an inverter with neural network for reconfiguring of VV and VW control parameters.

pass filtered by $H_O(s)$ to produce the active and reactive power injections that are injected into the grid. These filters serve to limit the rate at which the active and reactive powers injected by PV systems can change and do not represent physical constraints of the smart inverter devices themselves [1]. The intelligent agent is depicted in the green dashed box which inputs actions at time t, given by a_t, to both the VV and VW controllers. In the context of reinforcement learning, the intelligent agent is a neural network that maps observations of the electric grid and states of the solar inverters into new parameters for solar inverter VV and VW curves. The agent observes active power produced by the solar array, the grid voltage, and past actions input to the VV/VW controllers. Additionally, the agent receives a reward indicating the effectiveness of the last action on mitigating the cyberattack, r_t (see [4, 5] for a detailed description of the reward mechanism).

The action taken by the intelligent agent consist of offsets to default inverter VV and VW control curves as depicted in Fig. 8. As is shown in the figure, the RL agent action, a_t, is implemented as $\Delta\eta$, a translation of the VV or VW curve along the voltage axis. This form of control is an indirect form of active/reactive power injection, as translations of the VV/VW curves along the voltage axis result in new inverter active/reactive power setpoints.

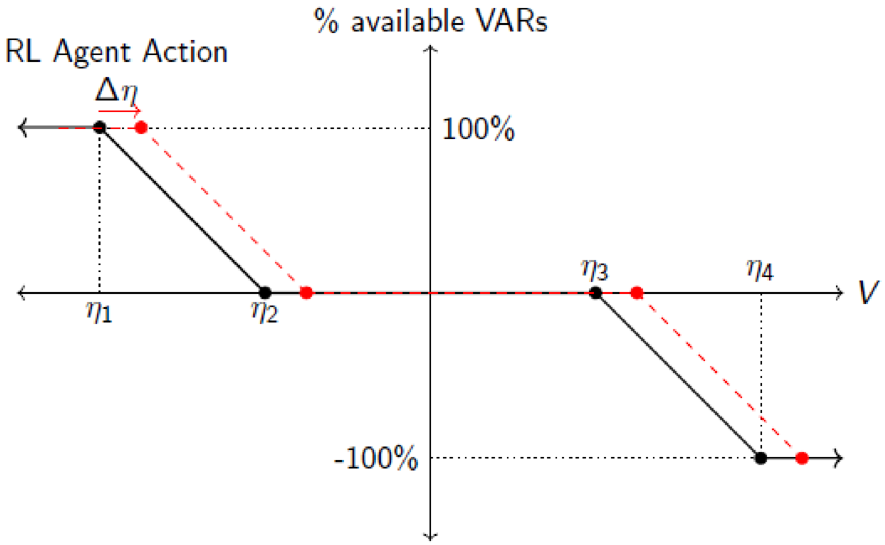


Figure 8: Depiction of control actions affecting inverter control Volt-VAR control setpoints.

3.4 Simulation Results

In this section, we present the results of several experiments where RL controllers trained using the PyCIGAR software framework are used to manage non-compromised inverter VV/VW controllers to mitigate attacks on other smart inverters that create large voltage imbalances as well as large oscillations in the distribution grid.

Figs. 9 - 11 depict the results of experiments conducted on the IEEE 37 node test feeder. The topology of the test feeder is shown in Fig. 9. Figure 10 depicts subplots of timeseries data showing the controller mitigating the effect of a cyber attack designed to create voltage imbalances in the distribution grid. The subplots of the left-hand column show the attack without the presence of RL control, while the set of subplots on the right show the effectiveness of mitigating the imbalance attack. In the particular experiment shown in Fig. 10, the RL agent has learned the dynamics of the voltage regulator and takes these dynamics into account when choosing its next action. This is evident in subplot 3 on the right-hand column where the agent purposefully lowers voltages to trigger the regulator to act, which allows for a better mitigation of the imbalance. Fig. 11 shows another experiment where the RL controller substantially minimizes the amount of voltage imbalance in the system.

Figs. 12 - 13 depict the results of experiments conducted on the IEEE 123 node test feeder. The topology of the test feeder is shown in Fig. 12. Figure 13 depicts subplots of timeseries data showing the controller mitigating the effect of a cyber attack designed to create large oscillations in the distribution grid.

The subplots of the left-hand column show the attack without the presence of RL control, while the set of subplots on the right show the effectiveness of mitigating the oscillation attack. As is shown in the figure, the RL controller substantially minimizes the magnitude of the oscillations in the system during the attack.

Figs. 14 - 15 depict the results of experiments conducted on the IEEE 8500 node test feeder. The topology of the test feeder is shown in Fig. 14. Figure 15 depicts subplots of timeseries data showing the controller mitigating the effect of a cyber attack designed to create voltage imbalances in the distribution grid. The subplots of the left-hand column show the attack without the presence of RL control, while the set of subplots on the right show the effectiveness of mitigating the imbalance attack.

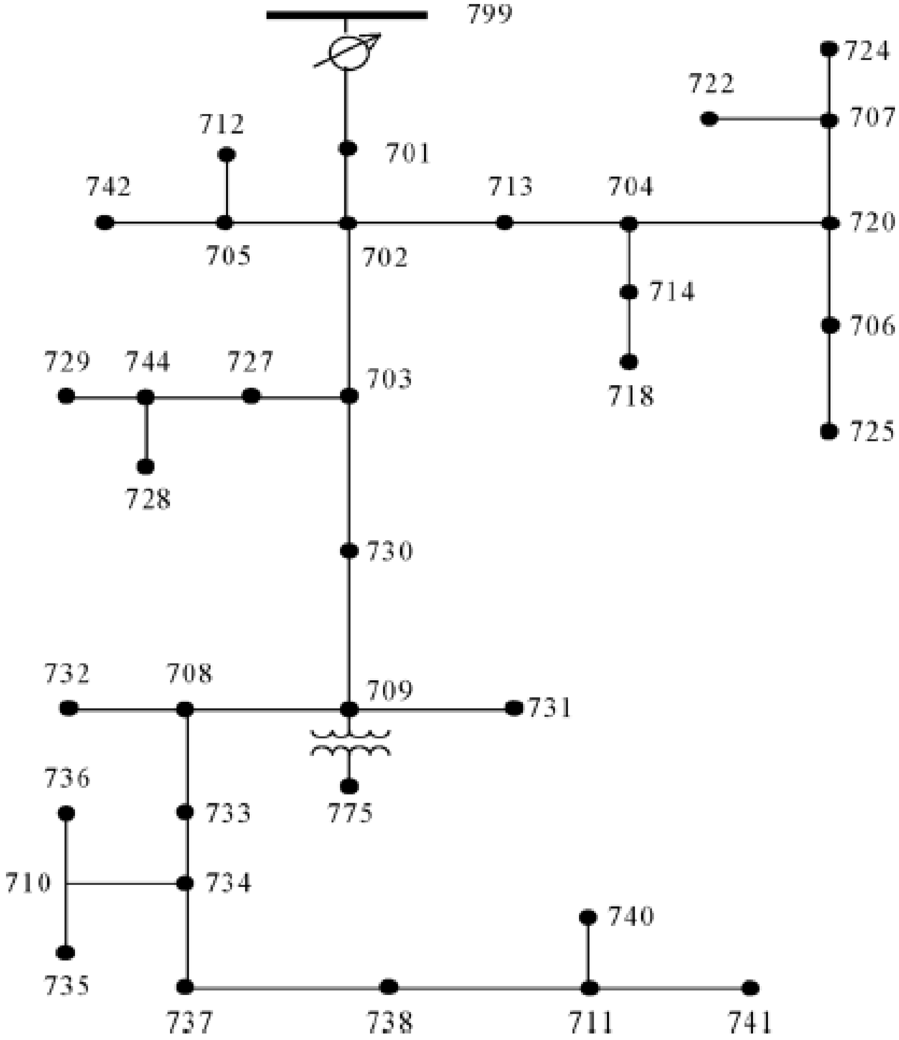


Figure 9: IEEE 37 node test feeder.

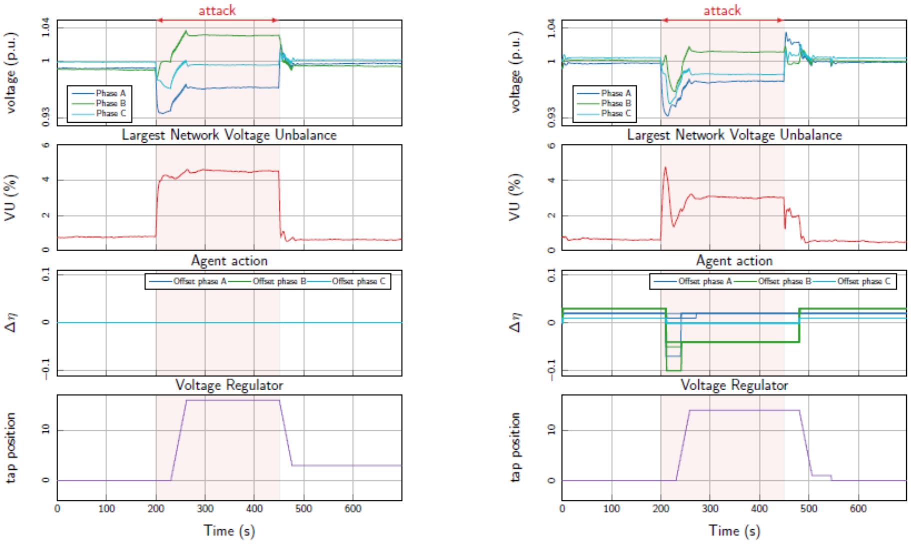


Figure 10: Simulation results showing RL agent learning voltage regulator behavior. The left column of subplots show result of the attack without the presence of reinforcement learning control. The columns on the right show the system behavior when the reinforcement learning control is active.

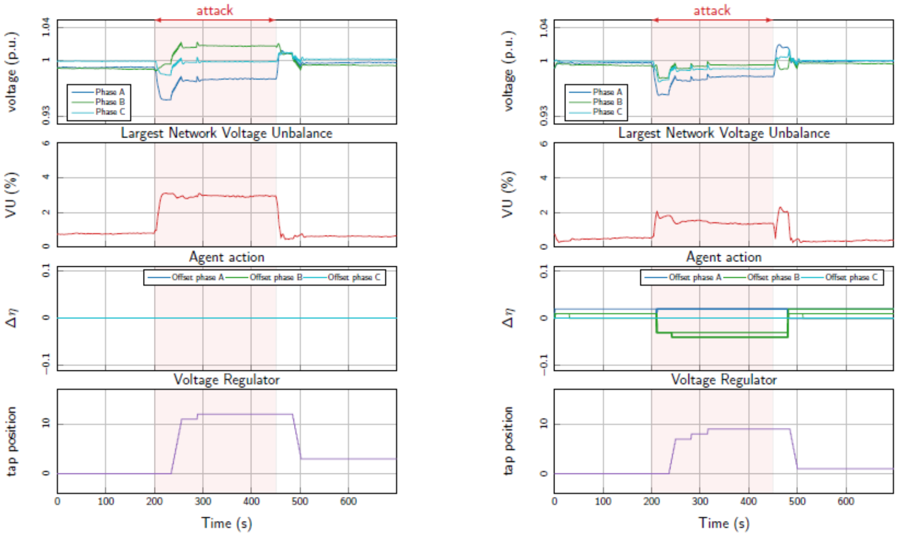


Figure 11: Simulation results showing agent mitigating large voltage imbalances in the system. The left column of subplots show result of the attack without the presence of reinforcement learning control. The columns on the right show the system behavior when the reinforcement learning control is active.

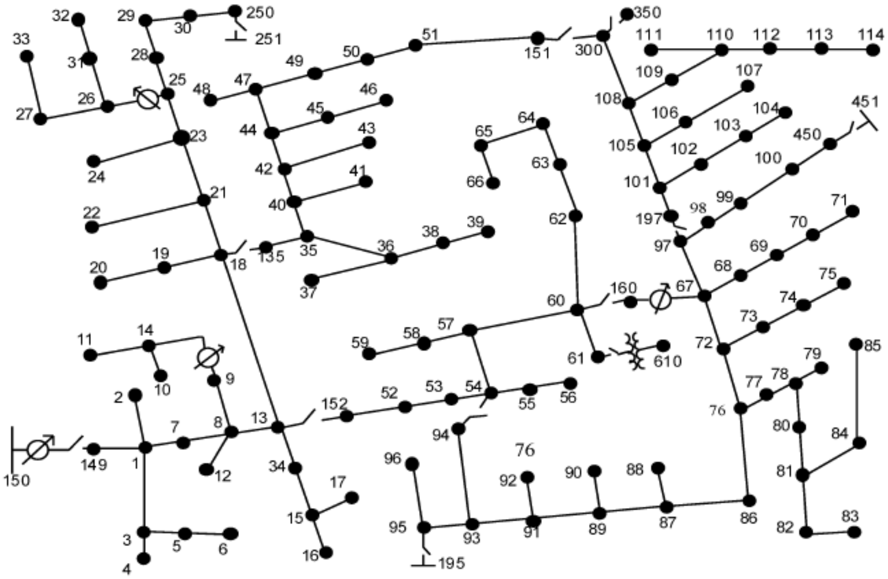


Figure 12: IEEE 123 node test feeder.

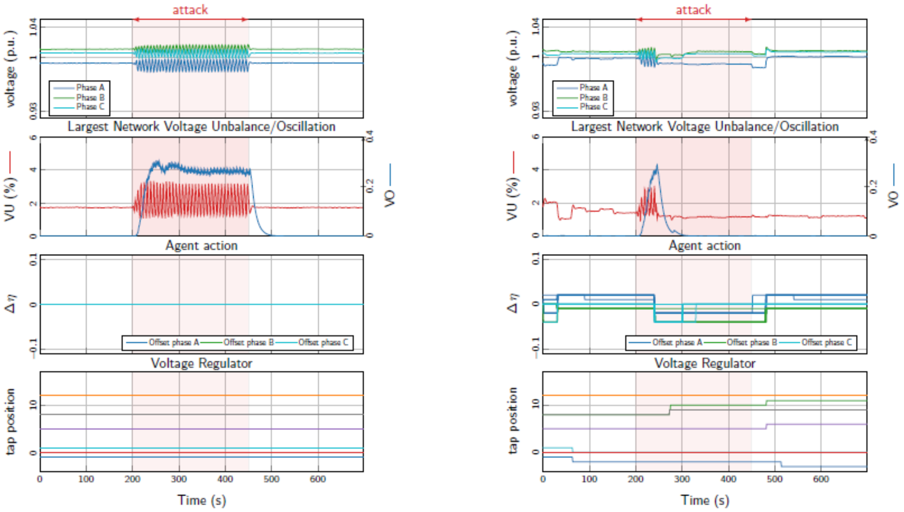


Figure 13: Simulation results showing the agent mitigating large voltage oscillations in the system. The left column of subplots show result of the attack without the presence of reinforcement learning control. The columns on the right show the system behavior when the reinforcement learning control is active.

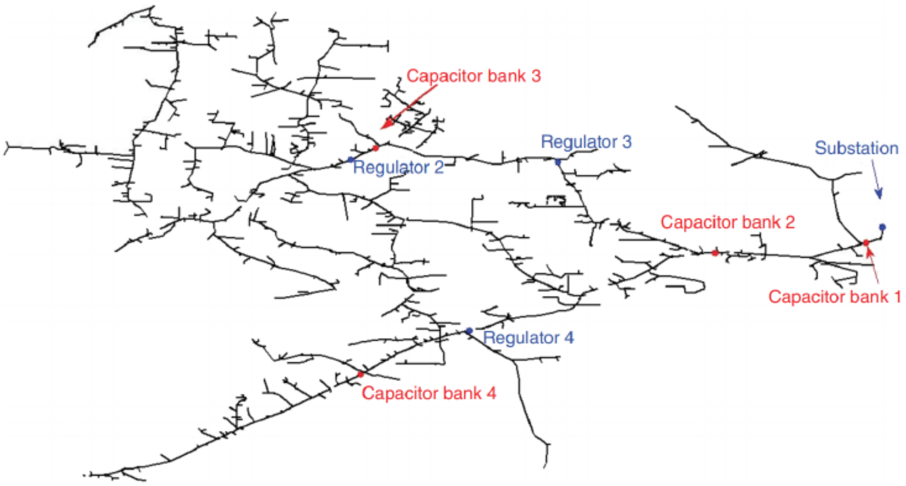


Figure 14: IEEE 8500 node test feeder.

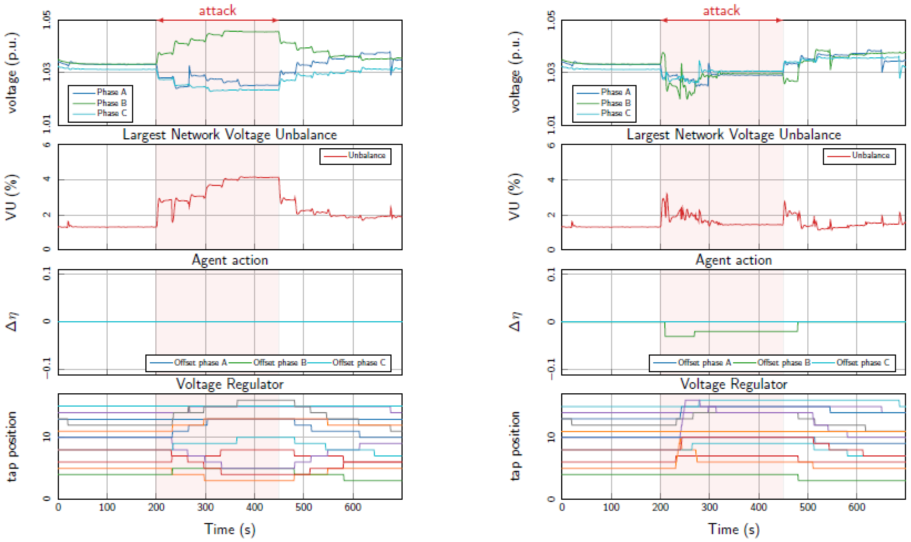


Figure 15: Simulation results showing the agent mitigating large voltage imbalances in the system. The left column of subplots show result of the attack without the presence of reinforcement learning control. The columns on the right show the system behavior when the reinforcement learning control is active.

4 Open Modeling Framework Integration

This section features a description of the "CyberInverters" module integrated into the NRECA Open Modeling Framework as part of the CIGAR project. The remainder of this section contains a presentation NRECA has created to brief security engineers at NRECA member electric utility co-operatives regarding the motivation for the CIGAR project and how the CyberInverters tool can be used for feeder-specific cybersecurity analysis.

On the CyberInverters input screen OMF users will input utility data specific to their networks. This data will subsequently be used by the PyCIGAR software to train and simulate the behavior of a reinforcement learning agent. The features of the model ingestion portion of the API are:

- **Simulation Start Date** - this field denotes the date and time in which the simulation should begin. The field an entry in the format demonstrated in the description.
- **Simulation Length and Length Units** - these fields control the length of time the user would like the simulation to span. The "Simulation Length" field is a simple float, while the "Length Units" field is a dropdown menu containing options to select seconds, minutes, or hours.
- **Feeder** - this button redirects the user to the feeder editor, in which the circuit to be simulated can be created, chosen and edited.
- **Load and PV Output** - this field is a file upload of a .csv that contains load and solar data necessary to run the simulation.
- **Attack Agent Variable** - this field is a dropdown menu containing options for different types of predetermined attack agents, which will simulate an attack on the circuit. The default option is "None" in which the simulation operates with no attack on the circuit.
- **Defense Agent Variable** - this field is a file upload of a HDF5 file representing a defense agent to "protect" the circuit from the attack(s) specified in the attack agent field.
- **Train?** - this field is a simple yes or no dropdown option to either enable or disable the defense agent training algorithm.

Upon running a simulation or training session, some of the features of the output portion of the API are:

- **Power Consumption from Transmission System** - this graph displays the impact of attack/defense on bulk power purchase and system losses across the time period of the simulation.
- **Transmission Voltage** - this graph displays the transmission-level voltage and can give the user a clear representation of stability problems (if any) due to regulation.

- **Substation Power Factor -** this graph shows the impacts of attack/defense agents on power factor at the head of feeder for each step of the simulation.

- **Energy Balance -** this graph provides the user with a sanity check on total energy generation, consumption, and loss.

- **Regulator Tap Changes -** this graph displays tapping actions for each regulator in the circuit across the time of the simulation, which is useful to the user when considering the monetary cost of each regulator.

- **Inverter Outputs -** this output displays a detailed graph for each inverter on the circuit to show the impacts of the attack and defense agents for the specified simulation period. The phase power, both real and imaginary, for each phase is represented.

- **Cap Bank Activation -** this graph displays the tapping actions on each capacitor in the circuit for each time step in the specified simulation period. It is beneficial to the user in analyzing cost by showing the active phases on each capacitor.

- **Triplex Meter Voltages -** this graph shows the minimum, mean, and maximum voltages across all meters in circuit to give the user a representation of voltage abnormality.

- **Solar Data and Other Climate Variables -** these outputs provide a graphical representation of climate variables for each time step in the simulation. Irradiance over time is a very valuable visual tool for the user, as it can be compared to other outputs to see the effects of solar on the circuit. As an example, an inverter attached to a solar panel should see a change in phase power with a change in irradiance, so being able to compare the two graphs can give the user an idea about whether or not the power for that inverter is changing due to an attack/defense or a change in irradiance.

- **Study Details -** this output provides a general overview of the simulation to allow the user to see statistics regarding simulation as well as a map to display the geographic location of the circuit.

- **Raw Input and Output Files -** these links provide user access to any raw input or output file used in the simulation.

Cybersecurity via Inverter-Grid Automatic Reconfiguration (CIGAR) **Final Report**

November 2020
Presented by Ryan Mahoney

Executive Summary

- DER deployments are growing.
- Cyberattacks on DERs and other grid assets are growing.
- We worked with LBNL to create an algorithm for inverter control that could automatically respond to cyberattacks.
- We use reinforcement learning so controls can continuously adapt to changes in an adversary's attack approach.
- Results built into an easy-to-use simulation model on OMF.coop which allows utilities to quantify the effects of a DER cyberattack and evaluate the protective responses available.

A “Smarter” Distribution Grid

Distributed Energy Resources (DERs) have been and continue to be deployed across the larger distribution grid.

Source: California becomes first state to require solar panels on all new homes

Common types of DER:

- Solar (PV)
- Storage (batteries)
- Emergency generators

Common types of DER utilization:

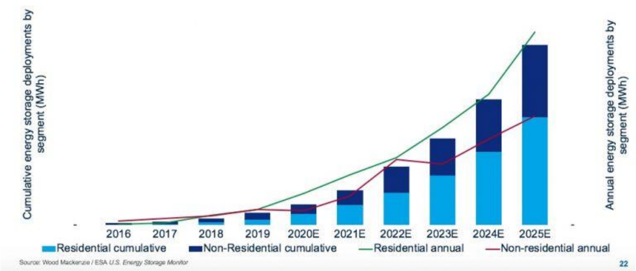
- Islanding
- Microgrids
- “Black start” process assistance
- Sourcing reactive power

Grid Edge and DER Market Update

Storage market capacity forecast, MWh

BTM annual deployments to reach 6.2 GWh in 2025

Distributed storage install base and forecast, MWh 2016-2025E



Source: Wood Mackenzie / ESA U.S. Energy Storage Monitor

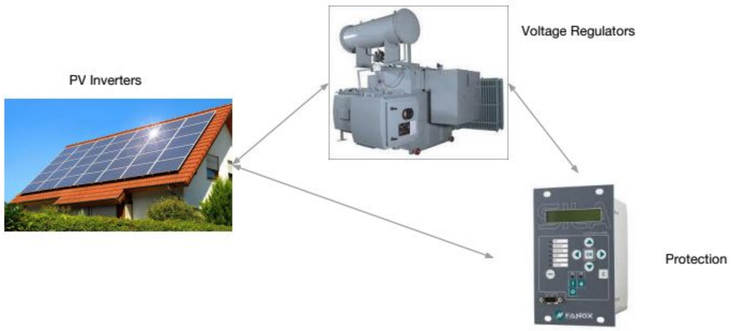
Source: United States Distributed Energy Resources Outlook: DER Installations and Forecasts, 2016-2025E (Wood Mackenzie)

A “Smarter” Distribution Grid (cont.)

With the expanding role of DERs in the field of energy distribution, the development and implementation of “smart” control systems/devices is growing as well. The goal of these systems are to enhance the optimization abilities of the DERs they are associated with.

Examples of DER “smart” devices:

- Inverters
- Protective devices
- Voltage regulators



Example: Enphase and HECO in Oahu

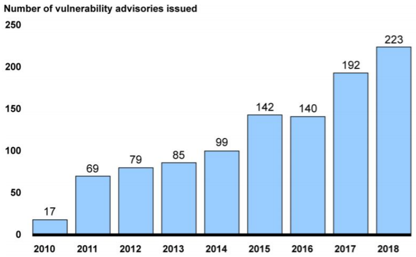
- In 2015, around 800,000 microinverters attached to individual photovoltaic panels in Oahu were remotely reprogrammed by Hawaiian Electric Company and Enphase Energy… in a single day.
- “...Enphase used built-in communications links to upgrade the grid-stabilizing capacity of four-fifths of Hawaii's rooftop solar systems.”
- Possible through the two-way data-over-powerline link that Enphase uses to monitor every one of its microinverters.
- Enphase’s cloud-based systems communicate with each of its panel-level devices every five minutes.
- Existing standards at the time required PV systems to shut down at the first sign of substandard AC voltage and frequency, but the remote update instructed the PV microinverters to ride through irregularities to help stabilize the grid’s AC signal.

Source: 800,000 Microinverters Remotely Retrofitted on Oahu—in One Day

Cybersecurity Concerns

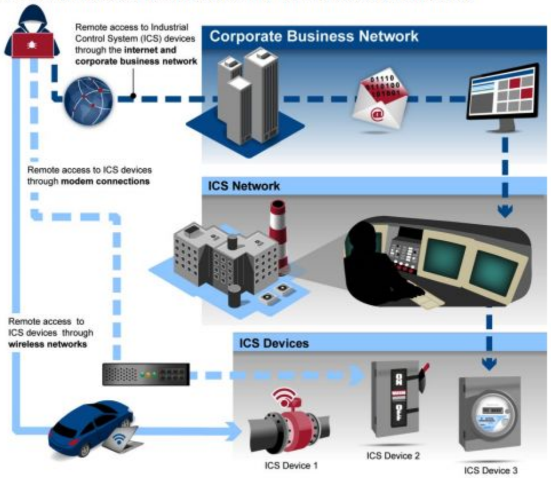
- If these control systems and smart devices are compromised, what could happen and how can we defend against this?
- Complex interaction between different control systems can create more sophisticated attacks.
- Standardization of autonomous DER behavior presents a cyber vulnerability - attacks will be easier to conduct.

Figure 4: Department of Homeland Security Vulnerability Advisories for Industrial Control System Devices, 2010 through 2018



Source: GAO summary of Department of Homeland Security website information. | GAO-19-332

Potential Ways an Attacker Could Compromise Industrial Control System Devices


Remote access to Industrial Control System (ICS) devices through the **internet and corporate business network**
Corporate Business Network
01110
0110100
101001
Remote access to ICS devices through **modem connections**
ICS Network
Remote access to ICS devices through **wireless networks**
ICS Devices
ON
OFF
ICS Device 1
ICS Device 2
ICS Device 3

Source: GAO analysis of Department of Energy and Department of Homeland Security documents. | GAO-19-332

Source: Actions Needed to Address Significant Cybersecurity Risks Facing the Electric Grid

Cybersecurity Concerns

Aurora Generator Test

- In March of 2007, Idaho National Laboratory conducted a demonstration for the U.S. Department of Homeland Security in which they simulated a cyber attack on the control system of power grid equipment.
- The attack opened and closed the circuit breakers of a 2.25 MW diesel generator to be out of phase with the rest of the grid, which led to smoking, shaking and eventually parts of the generator to fall/fly off the machine. (Vijayan)
- This attack simulation took place within 3 minutes, but could have occurred much faster if not for researchers pausing to assess data between each attack iteration.

Source: https://commons.wikimedia.org/w/index.php?curid=34984755

Cybersecurity Concerns

Russian cyber actors in U.S. critical infrastructure sectors

- In March 2018, a joint Technical Alert issued by the FBI and DHS stated that “Russian government cyber actors” have been targeting U.S. critical infrastructure sectors, including energy, nuclear and commercial facilities, since at least March 2016. (Naylor)
- “DHS and FBI produced this alert to educate network defenders to enhance their ability to identify and reduce exposure to malicious activity.” (TA18-074A)
- Electric companies were informed by the U.S. government in the summer of 2017 of a Russian "multistage intrusion campaign," which incorporated common hacking techniques (malware and spear-phishing), had gained access to at least one power plant’s control system. (Naylor)
- "They were not simply looking around that system and reconnoitering it… They were placing the tools that they would have to place in order to turn off the power. That's a serious vulnerability for us, and we're not anywhere near ready to deal with it." - Joel Brenner, head of counterintelligence under the Director of National Intelligence in the Obama administration (Naylor)

Cybersecurity Concerns

December 23, 2015 - cyber attack on the Ukranian power grid

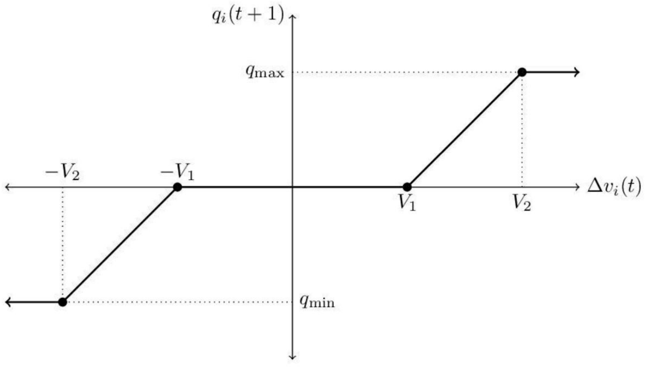
- Kyivoblenergo, a Ukranian regional electricity distribution company, and two other energy distribution companies were hacked, causing outages for around 225,000 customers. (Lee)
- Attack on Kyivoblenergo started at approximately 3:35 p.m. local time. Seven 110 kV and 23 35 kV substations were disconnected for three hours.
- Foreign attacker remotely controlled the supervisory control and data acquisition (SCADA) distribution management system.
- Recovery was done using manual operation of switches, was not able to be recovered using IT or OT systems.

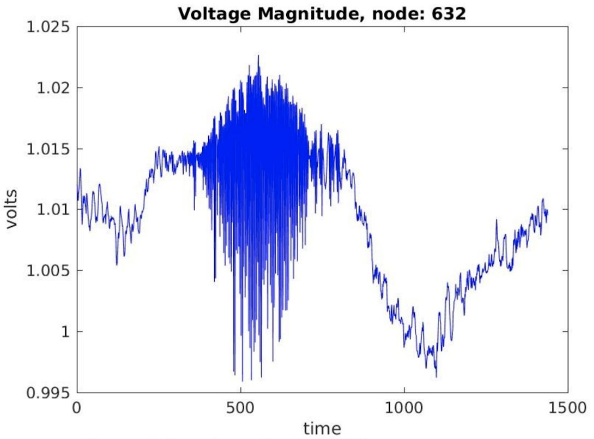
Cybersecurity Concerns

March 5, 2019 denial-of-service attack on U.S. wind and solar assets

- The North American Electric Reliability Corporation (NERC) revealed details about the cyber attack in September 2019.
- Assets revealed to be owned by sPower, one of the largest private owners of operating solar assets in the U.S.
- “Though there was no loss of generation, the March cyberattack impacted the company's visibility into about 500 MW of wind and PV across California, Utah and Wyoming… Attackers exploited a known vulnerability in an unpatched Cisco firewall, causing a series of reboots over 12 hours. (Walton)”
- “The news begs a bigger question about cybersecurity regulations for the energy industry… The manner in which it was carried out was very basic — exposing some essential weaknesses in the way energy companies currently patch and monitor their network devices." -Phil Neray, vice president of security firm CyberX (Walton)

Core Smart Inverter Instability Issue





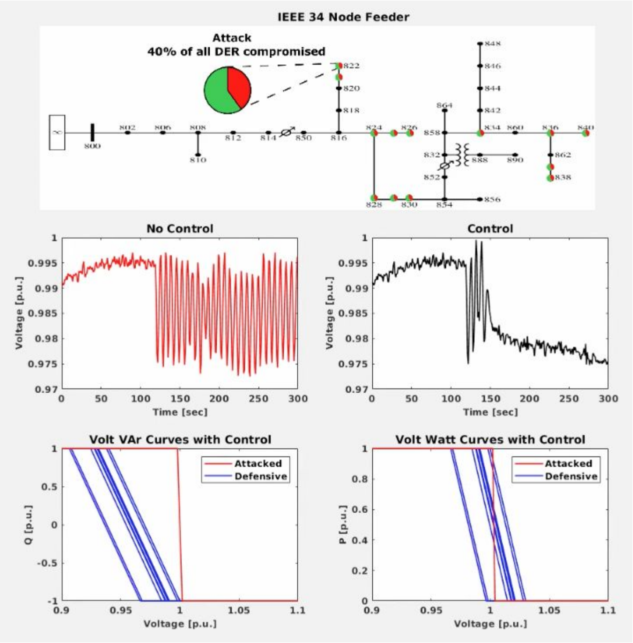
Improperly tuned settings can lead to instabilities

Mitigating Control Instabilities

- An attack can most easily be described as maliciously chosen set-points for the DER’s Volt-VAR (VV) and Volt-Watt (VW) controllers, which will result in improper power injection into the system.
- Adaptive Control - current approach to mitigating instabilities due to a cyberattack, in which the Volt-VAR curve is shifted to the left in order to counteract the shift created by the attack.
  - This is a very basic, naive approach and can only adapt to a single attack that involves voltage oscillation.
  - For systems at a low voltage levels, shifting the VV curve as such could lower the voltage to dangerously low level.
- Reinforcement Learning - train an agent to choose beneficial inverter setpoints.

Mitigating Control Instabilities

- An example of a simulation on a IEEE 34-bus system
  - Both the Volt-VAR and Volt-Watt curves are shifted left.
  - While the adaptive control defense approach manages to mitigate the oscillations in voltage, the resulting voltage [p.u.] drops.



Mitigating Control Instabilities

- "The frequency of attacks are continuing to grow and digitalization and hyper-connectivity are only going to expand the risk… Hackers are getting more and more sophisticated about industrial operations attacks." - Jason Haward-Grau, Chief information security officer, PAS Global (Walton)
- In order to mitigate these types of cyber attacks, we can utilize reinforcement learning (RL) to help formulate more thorough and effective defense strategies.
- Deep reinforcement learning allows for the training of a defense agent, through continuous loops of observing VV/VW curves and manipulating them (by a shift in either direction) and assigning a reward function to find the most effective course of action.
  - This strategy allows for the training of an agent that can handle simultaneous attacks on the system, including attacks causing both voltage oscillation and voltage imbalance.

Mitigating Control Instabilities

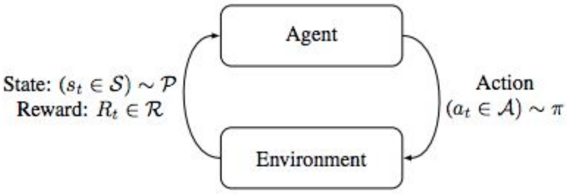
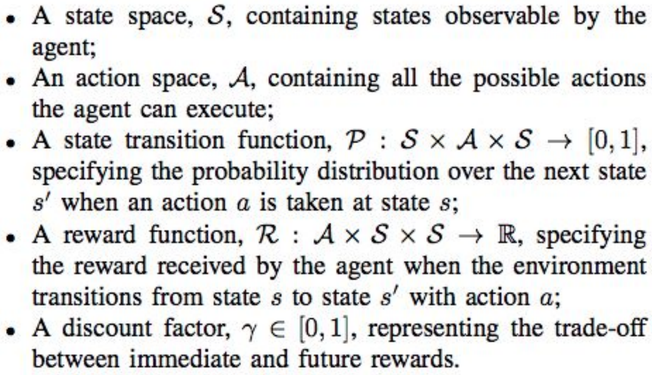

Agent
Environment
State: $(s_t \in \mathcal{S}) \sim \mathcal{P}$
Reward: $R_t \in \mathcal{R}$
Action
$(a_t \in \mathcal{A}) \sim \pi$

Fig. 1: Reinforcement learning loop.



- A state space, $\mathcal{S}$, containing states observable by the agent;
- An action space, $\mathcal{A}$, containing all the possible actions the agent can execute;
- A state transition function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, specifying the probability distribution over the next state s' when an action a is taken at state s;
- A reward function, $\mathcal{R} : \mathcal{A} \times \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$, specifying the reward received by the agent when the environment transitions from state s to state s' with action a;
- A discount factor, $\gamma \in [0, 1]$, representing the trade-off between immediate and future rewards.

CIGAR Project Overview

- Project Dates: 4/1/2018 - 4/1/2021
- Part of CEDS Program
  - Goal: enable distribution grids to adapt to resist/mitigate cyber attacks in real time
- Contributors
  - Lawrence Berkeley National Laboratory (LBNL)
  - Arizona State University (ASU)
  - National Rural Electric Cooperative Association (NRECA)
  - Power Standards Lab (PSL)
  - Siemens Corporation

Data Science and Technology
Lawrence Berkeley National Lab

ASU
Arizona State University

AMERICA'S ELECTRIC COOPERATIVES

PSL
POWER STANDARDS LAB

SIEMENS

CIGAR Project Goals

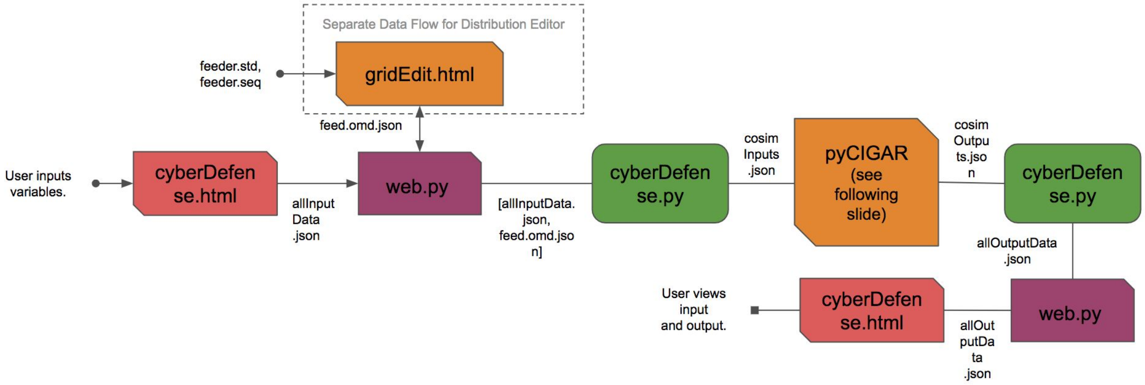
- Study the cybersecurity of systems that control inverters
  - Explore system vulnerabilities through reinforcement learning algorithms
  - Determine approaches to recognize distinguish between cyber attacks through system measurements
- Ensure grid voltage stability due to manipulations of DER and legacy device control settings
  - Make recommendations for control system upgrades and parameters for voltage regulation and protection systems to minimize the effects of cyber attacks
- Internal (NRECA) Goals:
  - Assess risk of inverter-based generation for our utilities
  - Enhance the capability of the OMF
  - Provide a tool for utility engineers and researchers to explore scenarios using dynamic modeling and the test how to mitigate cyber attacks

CIGAR Project Tasks

- Develop feedback control models of DER smart inverter functions, voltage regulation systems, protection systems
- Develop tools to detect instabilities in distribution system voltages
- Create use cases: settings for devices that cause/remedy instabilities in system voltages
- Conduct integrated system (DER, protection, regulation) simulations in GridLAB-D/openDSS
- Prototype reinforcement learning agent to reconfigure control parameters for smart inverters, regulators, protection systems
- Develop reinforcement learning agent module and interface to GridLAB-D/openDSS simulator
- NRECA Contributions
  - Adapt/expand OMF capabilities to support interaction with CIGAR simulation tools
  - Coop Advisory Board

cyberInverters Model Overview

- cyberInverters - OMF model that utilizes **pycigar** simulation tool to show the effects of different “agents” on a distribution grid within a given time series.
- An “agent” or “policy” can either be attacking or defending the grid.



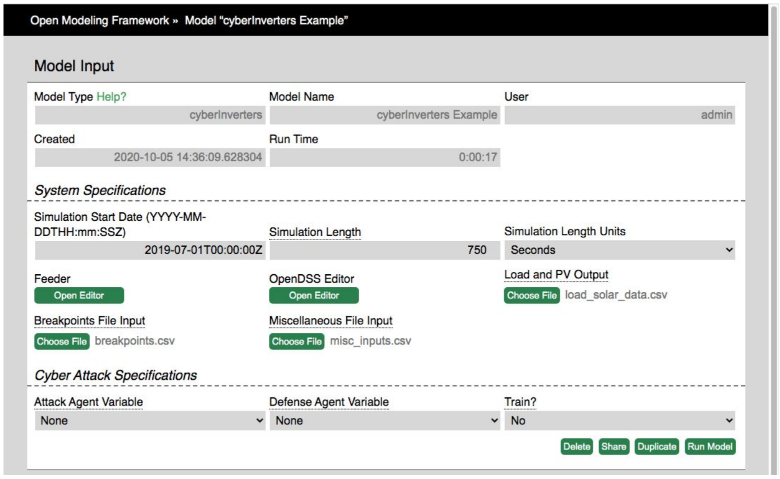
cyberInverters User Stories

1. A distribution planning engineer at an electric utility wants to study the cybersecurity of their system which may have inverters on it at some point. They are able to access a GUI on omf.coop to upload their system data, add/remove/edit inverter based resources like solar or energy storage, simulate a set of cyberattacks, see an analysis of the impacts of those attacks, and see a set of strategies to mitigate those attacks.
2. A researcher from a national lab, university, or electric grid vendor wants to understand the capabilities of the cyberInverters model and/or the underlying APIs for potential use in their work. They can use the utility-focused workflow like in story 1 on omf.coop, they can download and run the source code for all of this from github, and they can contribute back their changes if they really want to.
3. A member of the CIGAR team wants to integrate their code with the OMF and test using OMF features like distribution model editing and visualization. They are able to do this via a Python API and OMF code installable on their local machine.

cyberInverters Model Inputs

System Specifications Inputs

- **Simulation Start Date** - the date and time in which the simulation should begin
- **Simulation Length** and **Length Units** - the length of time the user would like the simulation to span
- **Feeder** - “Open Editor” button redirects user to a visual editor of the feeder on which the simulation is run
- **OpenDSS Editor** - “Open Editor” button redirects user to a text editor of the .dss file which represents the circuit.
- **Load and PV Output** - file upload of a .csv that contains load and solar data necessary to run the simulation
- **Breakpoints File Input** - file upload of a .csv that contains information on the grid’s breakpoints
- **Miscellaneous File Input** - file upload of a .csv that contains general information required for the pycigar tool to run properly



Open Modeling Framework » Model "cyberInverters Example"

Model Input

Model Type Help?	Model Name	User
cyberInverters	cyberInverters Example	admin
Created	Run Time	
2020-10-05 14:36:09.628304	0:00:17	

System Specifications

Simulation Start Date (YYYY-MM-DDTHH:mm:SSZ)	Simulation Length	Simulation Length Units
2019-07-01T00:00:00Z	750	Seconds
Feeder	OpenDSS Editor	Load and PV Output
Open Editor	Open Editor	Choose File load_solar_data.csv
Breakpoints File Input	Miscellaneous File Input	
Choose File breakpoints.csv	Choose File misc_inputs.csv	

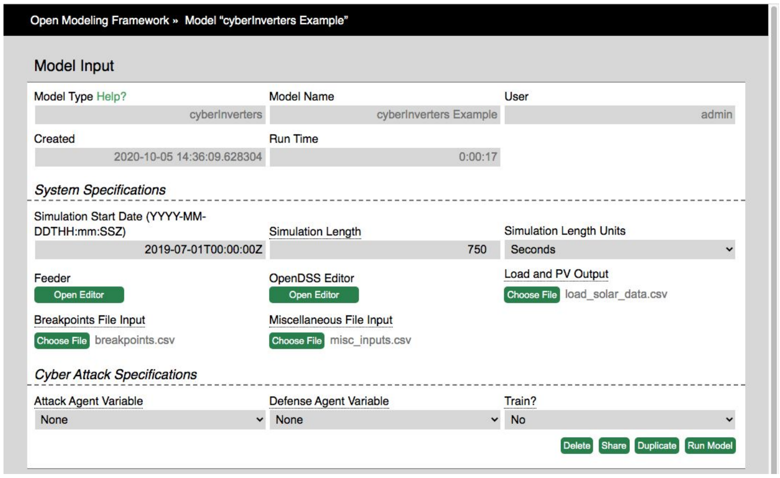
Cyber Attack Specifications

Attack Agent Variable	Defense Agent Variable	Train?
None	None	No

cyberInverters Model Inputs

Cyber Attack Specifications Inputs

- **Attack Agent Variable** - dropdown menu containing options for different types of predetermined attack agents, which will simulate an attack on the circuit
  - The default option is “None” in which the simulation operates with no attack on the circuit.
- **Defense Agent Variable** - dropdown menu containing options for existing defense agents
  - Defense agents are are generated through running a prior training simulation (“Train?” field set to “Yes”).
- **Train?** - “Yes” or “No” dropdown option to either enable or disable the defense agent training algorithm.
  - When “Yes” is selected, a new defense agent will be trained based on the simulation specifications and saved to a folder within the model directory.



Open Modeling Framework » Model "cyberInverters Example"

Model Input

Model Type Help? cyberInverters | Model Name cyberInverters Example | User admin

Created 2020-10-05 14:36:09.628304 | Run Time 0:00:17

System Specifications

Simulation Start Date (YYYY-MM-DDTHH:mm:SSZ) 2019-07-01T00:00:00Z | Simulation Length 750 | Simulation Length Units Seconds

Feeder Open Editor | OpenDSS Editor Open Editor | Load and PV Output Choose File load_solar_data.csv

Breakpoints File Input Choose File breakpoints.csv | Miscellaneous File Input Choose File misc_inputs.csv

Cyber Attack Specifications

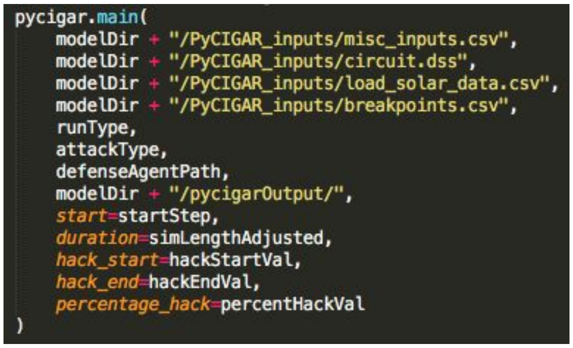
Attack Agent Variable None | Defense Agent Variable None | Train? No

Delete Share Duplicate Run Model

cyberInverters Call to PyCIGAR

Parameters for call to pycigar (in order as they appear):

- **misc_inputs_path** - csv file containing miscellaneous information to run the experiment including the weight of reward
- **dss_path** - openDSS file representing the grid in the experiment
- **load_solar_path** - csv file representing the load and solar profile from experiment start time to end time
- **breakpoints_path** - csv file representing the default VoltVar/VoltWatt break curves of inverters installed in the grid
- **test** - type of test represented by a string [“NO_DEFENSE”, “DEFENSE”, “TRAIN”]
- **type_attack** - type of attack represented by a string [“VOLTAGE_OSCILLATION”, “VOLTAGE_UNBALANCE”]
- **policy** - path to files representing the defense agent for the experiment
  - Only necessary if test=”DEFENSE”
- **output** - path to save the results of of the experiment

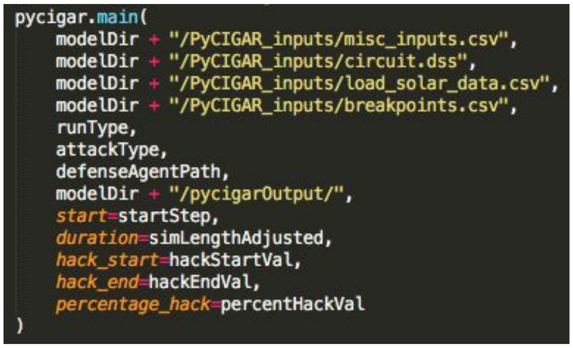


```
pycigar.main(
    modelDir + "/PyCIGAR_inputs/misc_inputs.csv",
    modelDir + "/PyCIGAR_inputs/circuit.dss",
    modelDir + "/PyCIGAR_inputs/load_solar_data.csv",
    modelDir + "/PyCIGAR_inputs/breakpoints.csv",
    runType,
    attackType,
    defenseAgentPath,
    modelDir + "/pycigarOutput/",
    start=startStep,
    duration=simLengthAdjusted,
    hack_start=hackStartVal,
    hack_end=hackEndVal,
    percentage_hack=percentHackVal
)
```

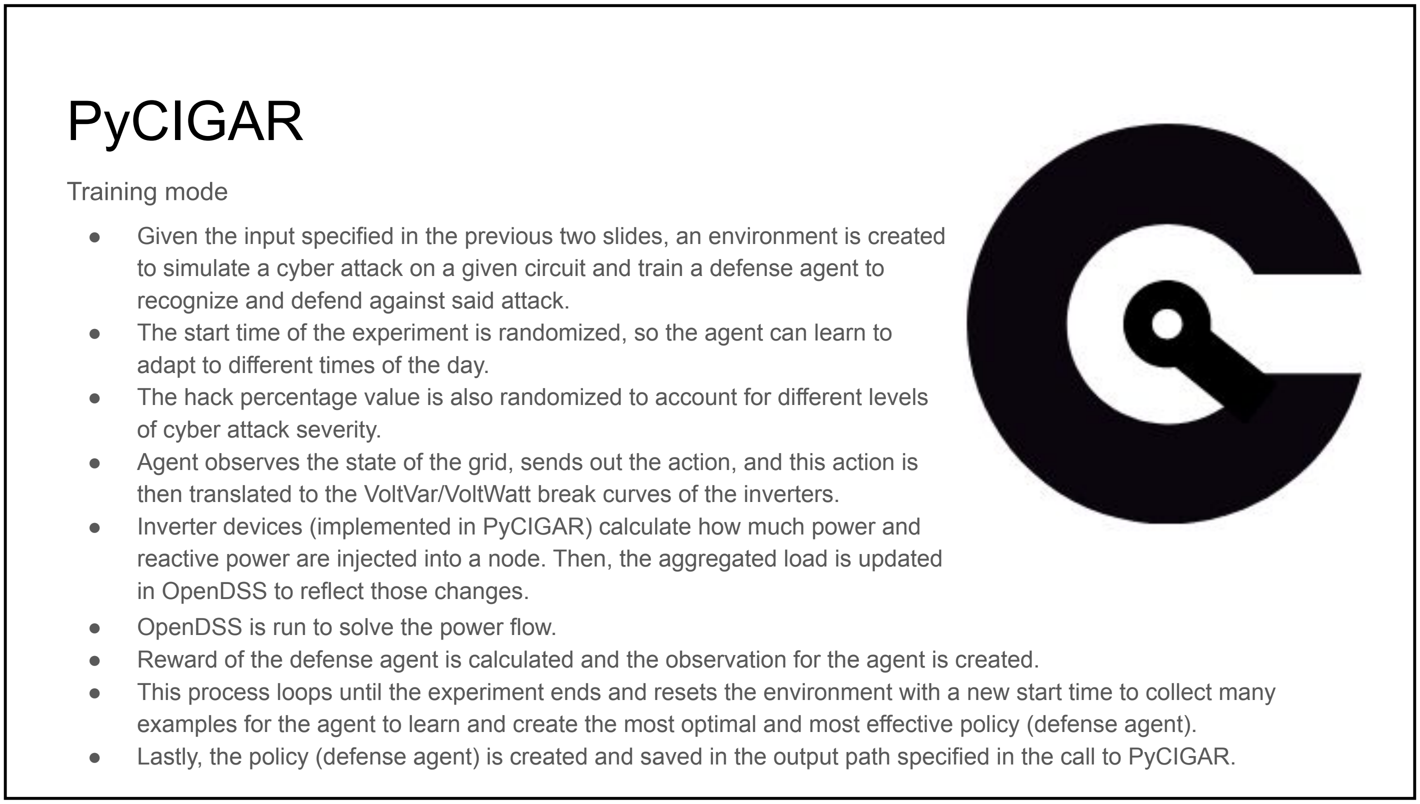
cyberInverters Call to PyCIGAR (cont.)

Parameters for call to pycigar (in order as they appear):

- **start** - integer representing the step in simulation in which the experiment starts
  - if start=100, then the experiment starts with the load and solar profile starting from row 100 in the load_solar_path csv file
- **duration** - integer representing the duration of the experiment in seconds
- **hack_start** - integer representing the number of timesteps (seconds) after start that the hacked inverters are activated
- **hack_end** - integer representing the number of timesteps (seconds) after start that the hacked inverters are deactivated
- **percentage_hack** - provided there is an attack selected, float representing the percentage hack of all the devices
  - Only relevant when an attack is selected
  - All inverters across feeder are “hacked” at the same percentage and at the same time (hack_start)



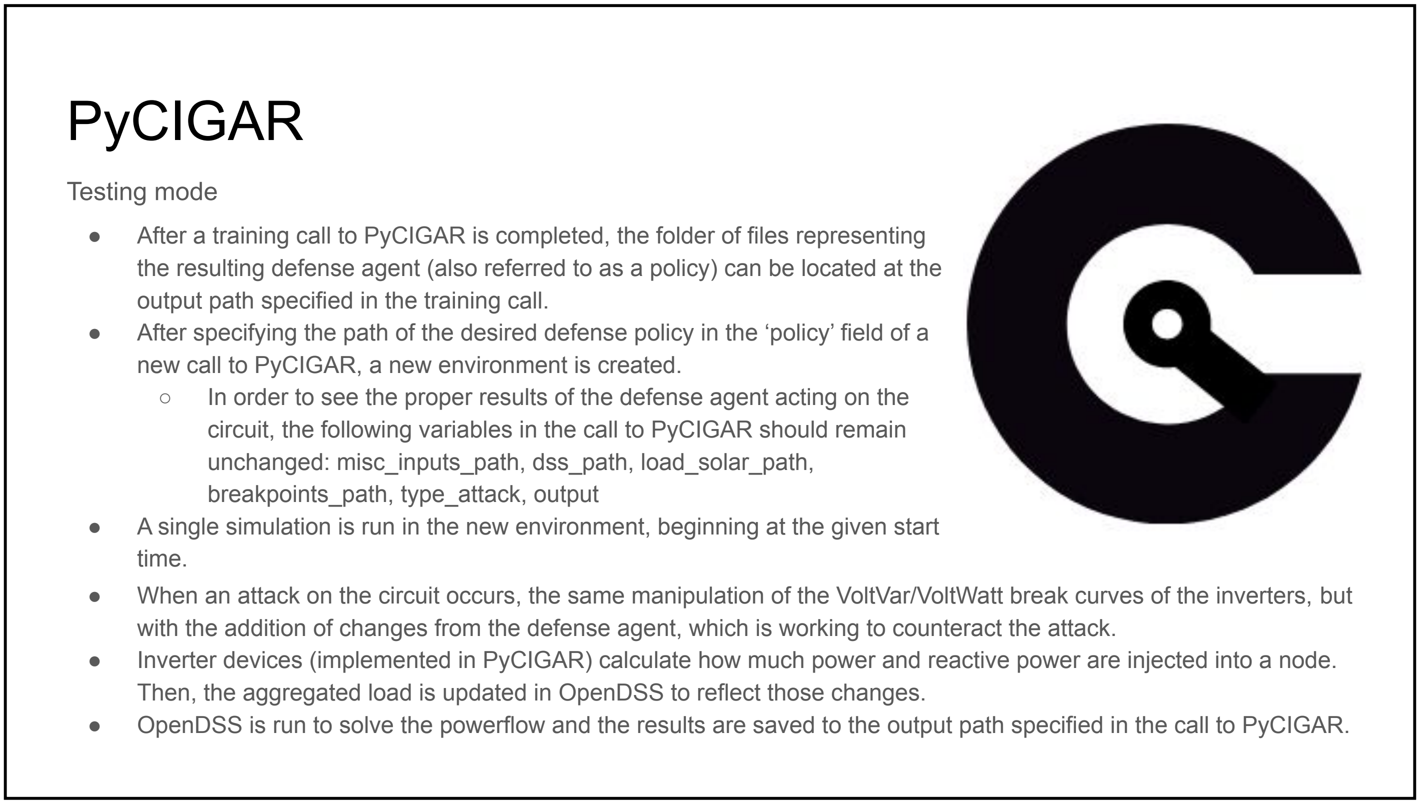
```
pycigar.main(
    modelDir + "/PyCIGAR_inputs/misc_inputs.csv",
    modelDir + "/PyCIGAR_inputs/circuit.dss",
    modelDir + "/PyCIGAR_inputs/load_solar_data.csv",
    modelDir + "/PyCIGAR_inputs/breakpoints.csv",
    runType,
    attackType,
    defenseAgentPath,
    modelDir + "/pycigarOutput/",
    start=startStep,
    duration=simLengthAdjusted,
    hack_start=hackStartVal,
    hack_end=hackEndVal,
    percentage_hack=percentHackVal
)
```



PyCIGAR

Training mode

- Given the input specified in the previous two slides, an environment is created to simulate a cyber attack on a given circuit and train a defense agent to recognize and defend against said attack.
- The start time of the experiment is randomized, so the agent can learn to adapt to different times of the day.
- The hack percentage value is also randomized to account for different levels of cyber attack severity.
- Agent observes the state of the grid, sends out the action, and this action is then translated to the VoltVar/VoltWatt break curves of the inverters.
- Inverter devices (implemented in PyCIGAR) calculate how much power and reactive power are injected into a node. Then, the aggregated load is updated in OpenDSS to reflect those changes.
- OpenDSS is run to solve the power flow.
- Reward of the defense agent is calculated and the observation for the agent is created.
- This process loops until the experiment ends and resets the environment with a new start time to collect many examples for the agent to learn and create the most optimal and most effective policy (defense agent).
- Lastly, the policy (defense agent) is created and saved in the output path specified in the call to PyCIGAR.



PyCIGAR

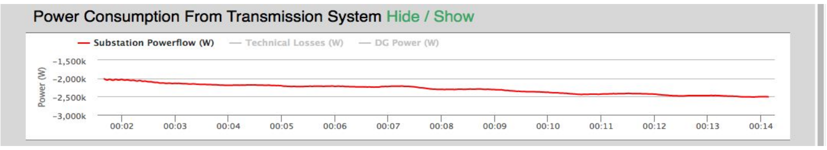
Testing mode

- After a training call to PyCIGAR is completed, the folder of files representing the resulting defense agent (also referred to as a policy) can be located at the output path specified in the training call.
- After specifying the path of the desired defense policy in the ‘policy’ field of a new call to PyCIGAR, a new environment is created.
  - In order to see the proper results of the defense agent acting on the circuit, the following variables in the call to PyCIGAR should remain unchanged: misc_inputs_path, dss_path, load_solar_path, breakpoints_path, type_attack, output
- A single simulation is run in the new environment, beginning at the given start time.
- When an attack on the circuit occurs, the same manipulation of the VoltVar/VoltWatt break curves of the inverters, but with the addition of changes from the defense agent, which is working to counteract the attack.
- Inverter devices (implemented in PyCIGAR) calculate how much power and reactive power are injected into a node. Then, the aggregated load is updated in OpenDSS to reflect those changes.
- OpenDSS is run to solve the powerflow and the results are saved to the output path specified in the call to PyCIGAR.

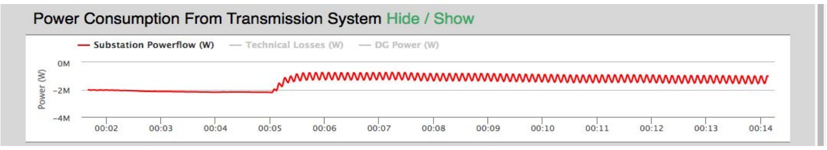
cyberInverters Outputs and Results

- **Power Consumption from Transmission System** - displays the impact of attack/defense on bulk power purchase and system losses across the time period of the simulation

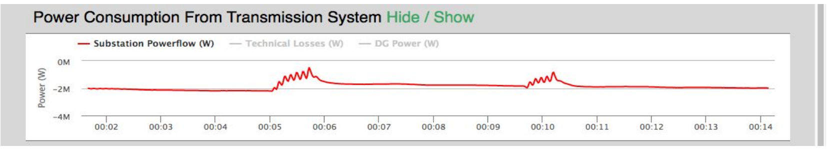
Control (No Attack or Defense) Scenario



Attack Scenario



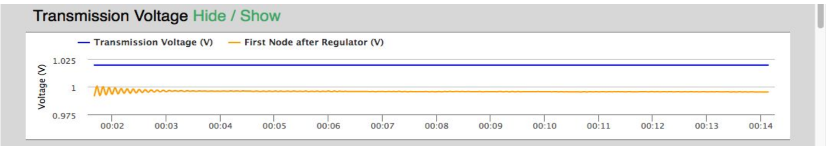
Attack + Defense Scenario



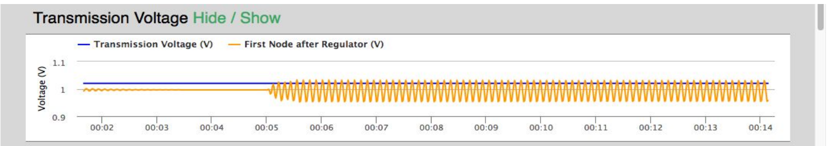
cyberInverters Outputs and Results

- **Transmission Voltage** - displays the transmission-level voltage and can give the user a clear representation of stability problems (if any) due to regulation

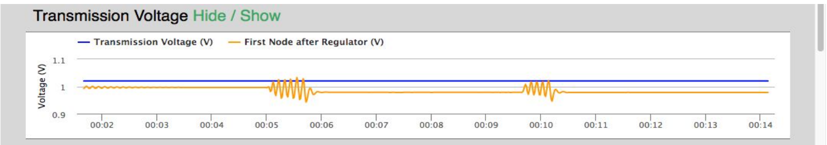
Control (No Attack or Defense) Scenario



Attack Scenario



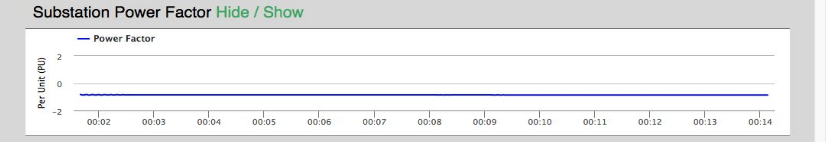
Attack + Defense Scenario



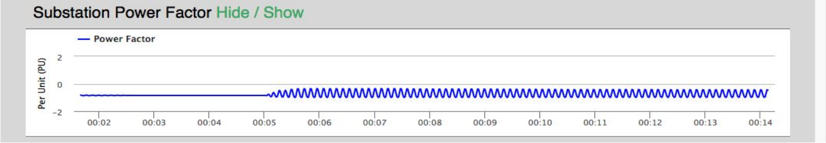
cyberInverters Outputs and Results

- **Substation Power Factor** - shows the impacts of attack/defense agents on power factor at the head of feeder for each step of the simulation

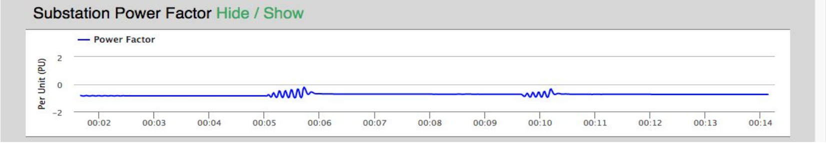
Control (No Attack or Defense) Scenario



Attack Scenario



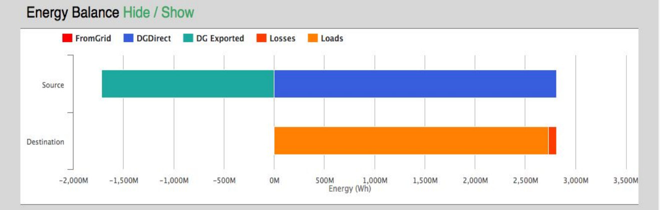
Attack + Defense Scenario



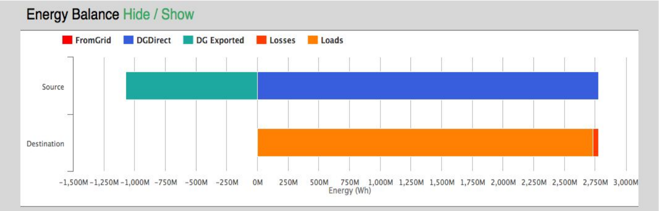
cyberInverters Outputs and Results

- **Energy Balance** - provides the user with a sanity check on total energy generation, consumption, and loss

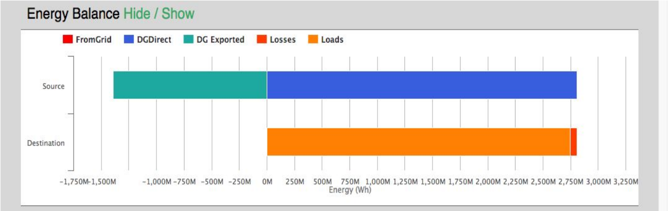
Control (No Attack or Defense) Scenario



Attack Scenario



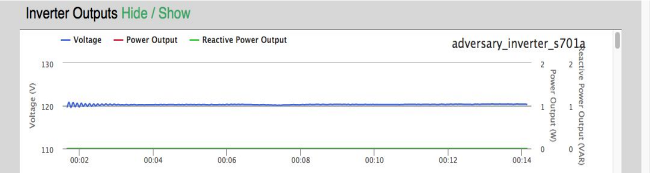
Attack + Defense Scenario



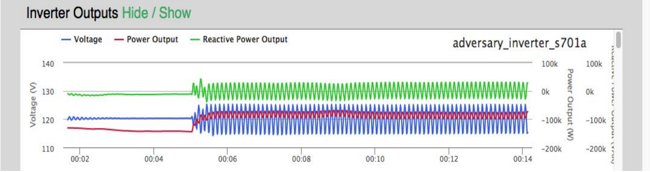
cyberInverters Outputs and Results

- **Inverter Outputs** - detailed graph for each inverter on the circuit to show the impacts of the attack and defense agents for the specified simulation period. The power output, both real and imaginary, for each phase is represented, as well as the voltage readings.

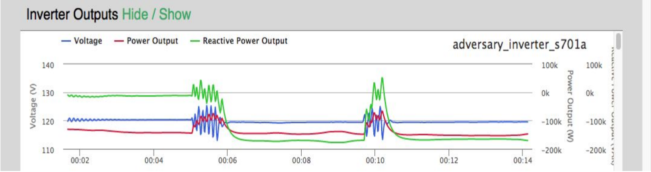
Control (No Attack or Defense) Scenario



Attack Scenario



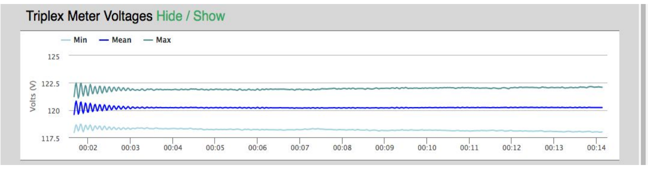
Attack + Defense Scenario



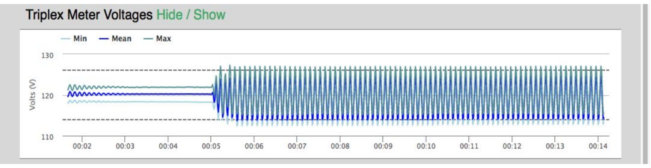
cyberInverters Outputs and Results

- **Triplex Meter Voltages** - this graph shows the minimum, mean, and maximum voltages across all meters in circuit to give the user a representation of voltage abnormality.

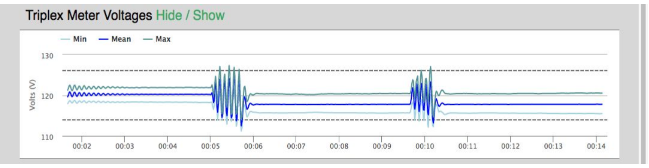
Control (No Attack or Defense) Scenario



Attack Scenario



Attack + Defense Scenario



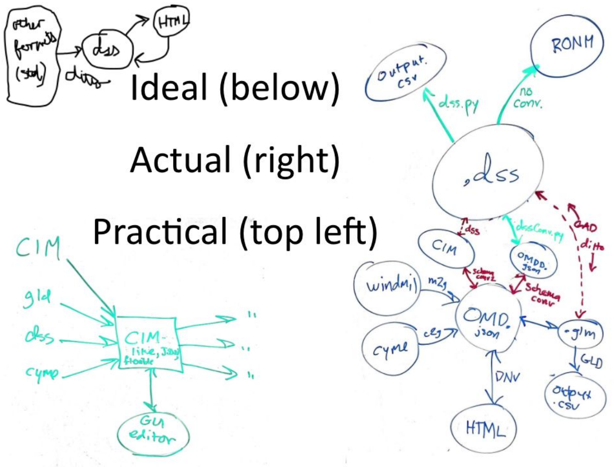
cyberInverters Outputs and Results

Other Outputs:

- **Regulator Tap Changes** - displays tapping actions for each regulator in the circuit across the time of the simulation, which is useful to the user when considering the monetary cost of each regulator
- **Cap Bank Activation** - a graph that displays the tapping actions on each capacitor in the circuit for each time step in the specified simulation period
  - Beneficial to the user in analyzing cost by showing the active phases on each capacitor
- **Solar Data** - a graphical representation of climate variables for each time step in the simulation
  - Irradiance over time is a very valuable visual tool for the user, as it can be compared to other outputs to see the effects of solar on the circuit.
  - As an example, an inverter attached to a solar panel should see a change in phase power with a change in irradiance, so being able to compare the two graphs can give the user an idea about whether or not the power for that inverter is changing due to an attack/defense or a change in irradiance.
- **Study Details** - a general overview of the simulation to allow the user to see statistics regarding simulation as well as a map to display the geographic location of the circuit
- **Raw Input and Output Files** - links providing user access to any raw input or output file used in the simulation

Current and Future Tasks

- Path to deployment: inverter or controller firmware, we're teamed up with Siemens and looking at this.
- Getting openDSS and Gridlab-D to play nicely
  - Better conversion between .dss, .glm and .omd files can allow for multiple input types and a better use of existing OMF tools like distNetViz.
- Gathering more accurate test data
  - Get more real circuits in the model
- Solar data enhancement
- Expanding attack agent list
  - Further investigating [NESCOR scenarios]
- Reaching out to coops
  - Allow coops to test with their data and receive feedback



Ideal (below)

Actual (right)

Practical (top left)

References

- Walton, R. (2019, November 4). *First cyberattack on solar, wind assets revealed widespread grid weaknesses, analysts say*. Utility Dive. https://www.utilitydive.com/news/first-cyber-attack-on-solar-wind-assets-revealed-widespread-grid-weaknesse/566505/
- Naylor, B. (2018, March 23). *Russia Hacked U.S. Power Grid — So What Will The Trump Administration Do About It?*. NPR. https://www.npr.org/2018/03/23/596044821/russia-hacked-u-s-power-grid-so-what-will-the-trump-administration-do-about-it
- Lee, R. (2016, March 18). *Analysis of the Cyber Attack on the Ukrainian Power Grid*. Electricity Information Sharing and Analysis Center. https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf
- Cybersecurity and Infrastructure Security Agency. (2018, March 15). *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors* (TA18-074A). https://us-cert.cisa.gov/ncas/alerts/TA18-074A
- Vijayan, J. (2007, September 28). Simulated attack points to vulnerable U.S. power infrastructure. Computerworld. https://www.computerworld.com/article/2541225/simulated-attack-points-to-vulnerable-u-s--power-infrastructure.html
- Roberts, C., Ngo, S., Milesi, A., Peisert, S., Arnold, D., Saha, S., Scaglione, A., Johnson, N., Kocheturov, A., & Fradkin, D. (2020). Deep Reinforcement Learning for DER Cyber-Attack Mitigation. ArXiv, abs/2009.13088.

CIGAR: Utility Perspective and User Interface

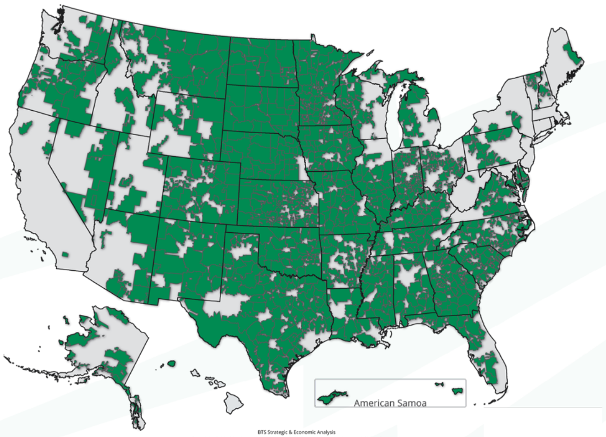
david.pinney@nreca.coop, ryan.mahoney@nreca.coop, lisa.slaughter@nreca.coop

75 Years of Service NRECA America's Electric Cooperatives

March 17, 2021

1

America's Electric Cooperatives



BTS Strategic & Economic Analysis
Copyright © 2016 by the National Rural Electric Cooperative Association. All Rights Reserved.

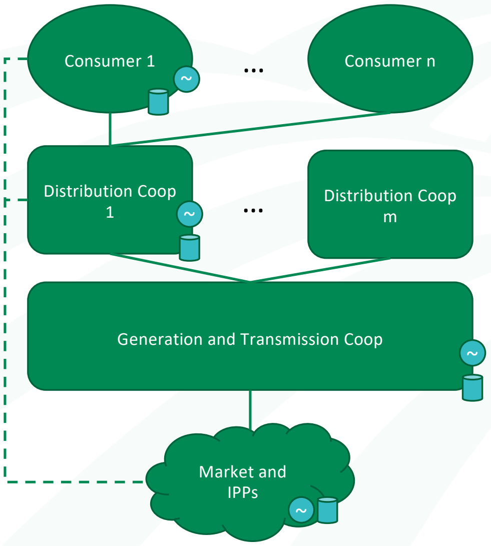
- Serve 42 million people in 47 states through 65 generation & transmission (G&T) co-ops and 840 distribution co-ops
- Own and maintain 42% of the nation's distribution lines
- Average 7.4 consumers per mile of distribution line
- NRECA is a trade association serving the cooperatives through government relations, pension and healthcare services, research, etc.

75 Years of Service NRECA America's Electric Cooperatives

2

Cooperative Power Supply Structure with Inverters

- Hierarchical supply system.
- DERs can go pretty much anywhere.
- Distribution coops typically limited to 5% DER production contractually, storage counting against this limit.
- ITC for solar + storage.
- Storage PPAs gaining popularity.
- Very rare for consumers and distribution coops to access IPPs and the market.
- Behind-the-meter: ideal deployment location?

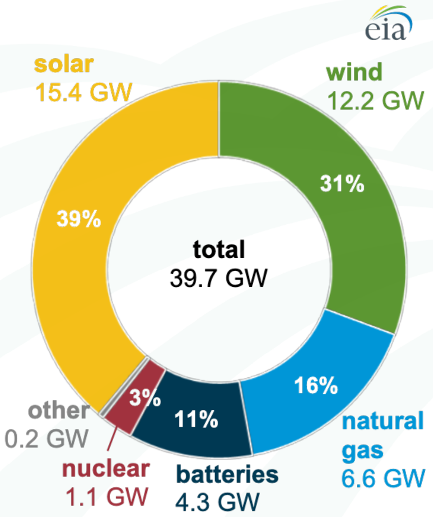


75 Years of Service NRECA America's Electric Cooperatives

3

Motivation for DER Cybersecurity

- Inverter Cyberattacks:
  - Demonstrated in the lab,
  - Executed on the grid,
  - Other nations have access to US infrastructure,
  - Large (800k) firmware updates becoming common
- IEEE1547: From 12 pages to over 100.
- Energy storage deployment and hence inverter deployment is growing +100% annually. EIA 2021 projected additions to right.
- When DERs are deployed we need to know:
  - What inverter settings could destabilize the distribution system for a given circuit.
  - Given a set of hacked devices, what control actions on non-hacked devices could mitigate the destabilizing behavior of the hacked devices.

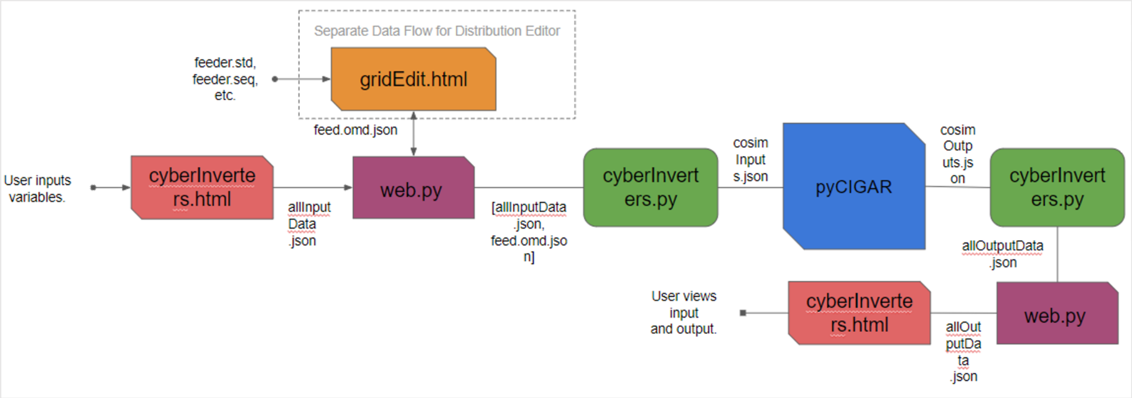


75 Years of Service NRECA America's Electric Cooperatives

4

The PyCIGAR user interface: cyberInverters

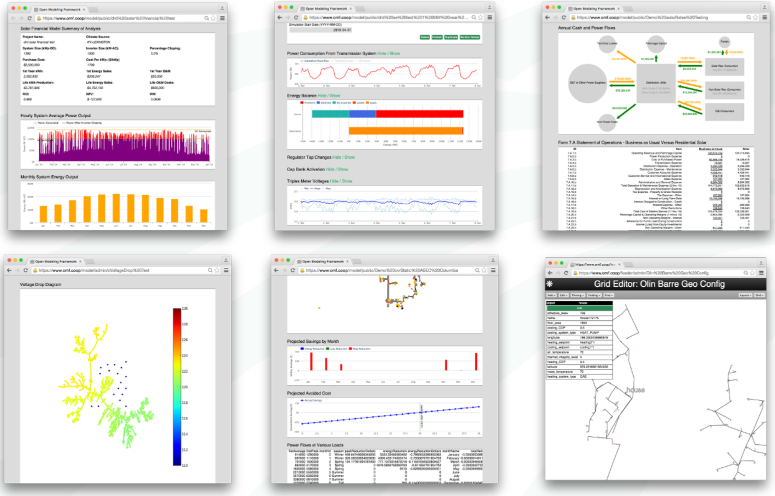
- cyberInverters: OMF.coop model that utilizes pycigar simulation tool to show the effects of different attack/defense agents on a distribution grid over a certain time period.
- Our key user: chief distribution planning engineer.



5

Framework Approach – https://OMF.coop

- Free and open source electric utility modeling software
- Built over the last 6 years by the co-ops and the US Department of Energy
- Python backend and support libraries, light web-based frontend
- Main focus: applications that perform financial and engineering analysis for utilities on emerging technologies (solar, energy storage, networked controls)
- Secondary focus: environment for researchers to develop new models
- Users from 217 organizations (utilities, vendors, universities) as of July 2019

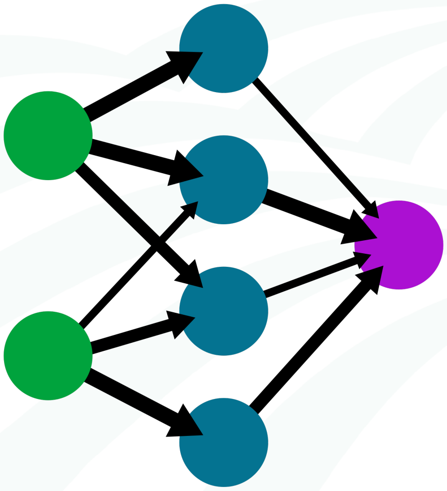


75 Years of Service
NRECA
America's Electric Cooperatives

6

Deep Reinforcement Learning

- Smart *agents* in LBL's pycigar control inverters in the simulation to intelligently counter malicious behavior of hacked devices.
- Agents gain their intelligence through training over millions of scenarios using *deep reinforcement learning*.
- Deep reinforcement learning has been hugely successful in other fields (super-human results in game playing, self-driving cars, facial recognition protein folding, etc.)
- Attacks can also be defined as agents to make training more rigorous (spy-versus-spy).

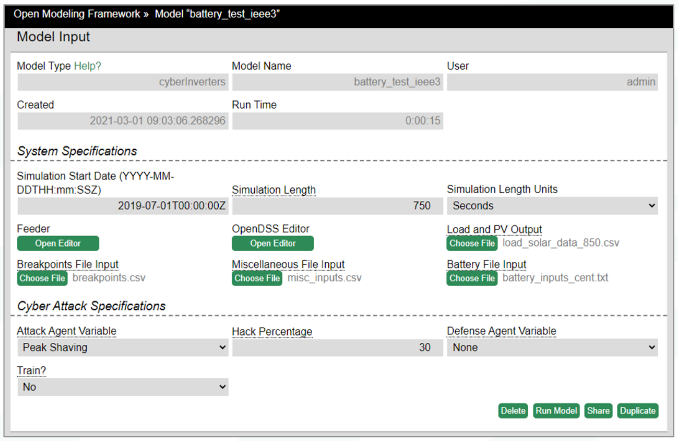


7

cyberInverters Model Inputs

"System Specifications" Section

- **Simulation Start Date** - The date and time at which the simulation begins.
- **Simulation Length** and **Units** - The length of time the simulation will span and the associated units.
- **Feeder** - "Open Editor" button redirects user to a visual editor of the feeder circuit input.
- **OpenDSS Editor** - "Open Editor" button redirects user to a text editor of the .dss circuit definition file.
- **Load and PV Output** - Allows user to upload a .csv containing distributed load and solar generation data at each simulation timestep.
- **Breakpoints File Input** - Allows user to upload a .csv defining each inverter's volt-var curve.
- **Miscellaneous File Input** - Allows user to upload a file containing hyperparameter definitions for the pycigar tool.
- **Battery File Input** - Future functionality. Allows user to upload a .csv defining batteries at various loads.



Open Modeling Framework » Model "battery_test_ieee3"

Model Input

Model Type: cyberInverters
Model Name: battery_test_ieee3
User: admin
Created: 2021-03-01 09:03:06.268296
Run Time: 0:00:15

System Specifications

Simulation Start Date (YYYY-MM-DDTHH:mm:SSZ): 2019-07-01T00:00:00Z
Simulation Length: 750
Simulation Length Units: Seconds
Feeder
OpenDSS Editor
Load and PV Output: load_solar_data_850.csv
Breakpoints File Input: breakpoints.csv
Miscellaneous File Input: misc_inputs.csv
Battery File Input: battery_inputs_cent.txt

Cyber Attack Specifications

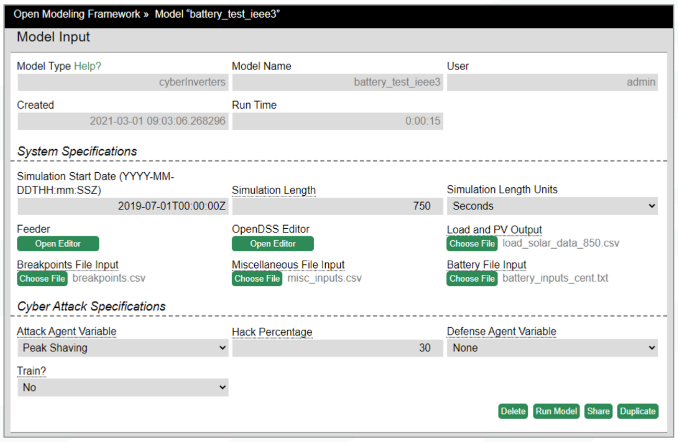
Attack Agent Variable: Peak Shaving
Hack Percentage: 30
Defense Agent Variable: None
Train?: No

8

cyberInverters Model Inputs

"Cyber Attack Specifications" Section:

- **Attack Agent Variable** - Allows selection of different predetermined attack agents that simulate an attack on the circuit.
  - None (Default)
  - Voltage Oscillation
  - Voltage Imbalance
- **Hack Percentage** - Limits attack agent penetration to a random subset of the inverters on the circuit.
- **Defense Agent Variable** - Allows selection of any existing pre-trained defense agents.
- **Train?** - Allows the user to create a defense agent that can be used against a specific attack.
  - Yes - Trains a new defense agent using selected simulation specifications.
  - Saved as a file within the OMF model directory.



Open Modeling Framework » Model "battery_test_ieee3"

Model Input

Model Type Help?: cyberInverters

Model Name: battery_test_ieee3

User: admin

Created: 2021-03-01 09:03:06.268296

Run Time: 0:00:15

System Specifications

Simulation Start Date (YYYY-MM-DDTHH:mm:SSZ): 2019-07-01T00:00:00Z

Simulation Length: 750

Simulation Length Units: Seconds

Feeder: Open Editor

OpenDSS Editor: Open Editor

Load and PV Output: Choose File load_solar_data_850.csv

Breakpoints File Input: Choose File breakpoints.csv

Miscellaneous File Input: Choose File misc_inputs.csv

Battery File Input: Choose File battery_inputs_cent.txt

Cyber Attack Specifications

Attack Agent Variable: Peak Shaving

Hack Percentage: 30

Defense Agent Variable: None

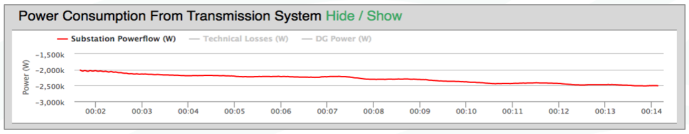
Train?: No

75 Years of Service NRECA America's Electric Cooperatives

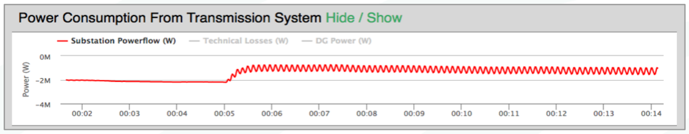
cyberInverters Outputs and Results

Power Consumption from Transmission System - Displays the impact of attack/defense on bulk power purchase and system losses over the simulation duration.

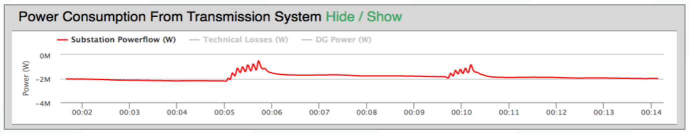
Control (No Attack or Defense) Scenario



Attack Scenario



Attack + Defense Scenario

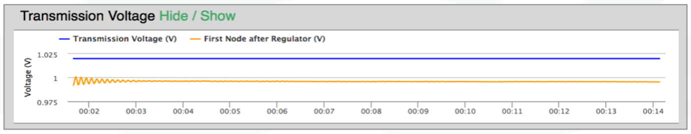


75 Years of Service NRECA America's Electric Cooperatives

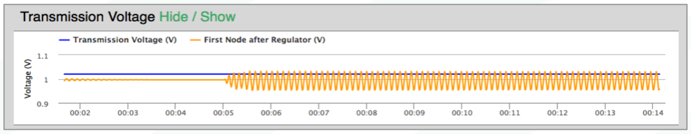
cyberInverters Outputs and Results

Transmission Voltage - Displays the transmission-level voltage and gives the user a clear representation of stability problems (if any) due to voltage regulator actions.

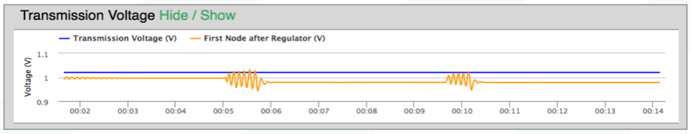
Control (No Attack or Defense) Scenario



Attack Scenario



Attack + Defense Scenario

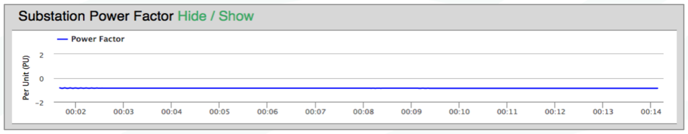


75 Years of Service
NRECA
America's Electric Cooperatives

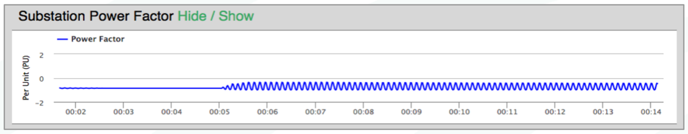
cyberInverters Outputs and Results

Substation Power Factor - Shows the effect of attack/defense agents on power factor at the head of feeder over the simulation duration.

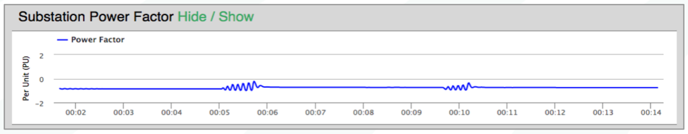
Control (No Attack or Defense) Scenario



Attack Scenario



Attack + Defense Scenario

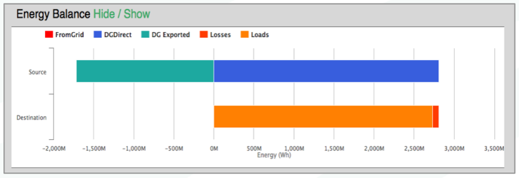


75 Years of Service
NRECA
America's Electric Cooperatives

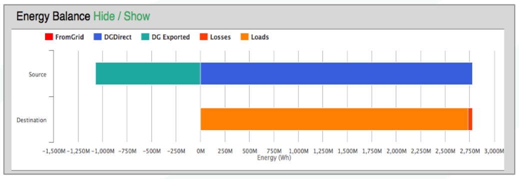
cyberInverters Outputs and Results

Energy Balance - Provides the user with a sanity check on total energy generation, consumption, and loss.

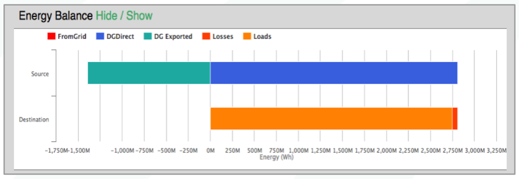
Control (No Attack or Defense) Scenario



Attack Scenario



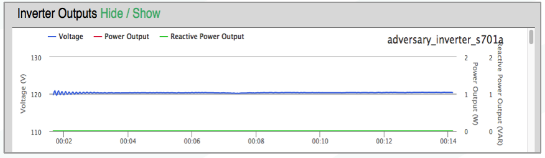
Attack + Defense Scenario



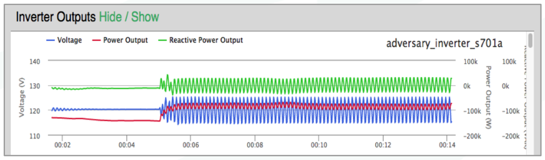
cyberInverters Outputs and Results

Inverter Outputs - Detailed graph for each inverter on the circuit showing the impacts of attack/defense agents over the simulation duration. The voltage readings, real power output, and imaginary power output are represented for each phase.

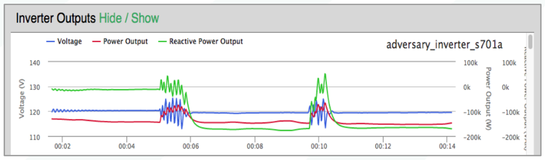
Control (No Attack or Defense) Scenario



Attack Scenario



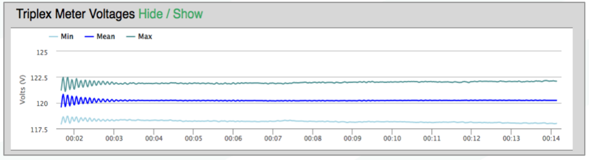
Attack + Defense Scenario



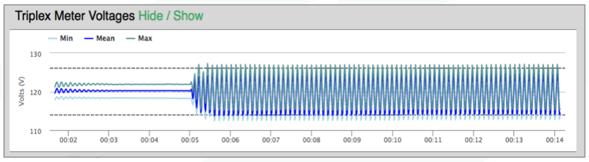
cyberInverters Outputs and Results

Triplex Meter Voltages - Shows the minimum, mean, and maximum voltages across all meters in circuit to give the user a representation of voltage abnormality.

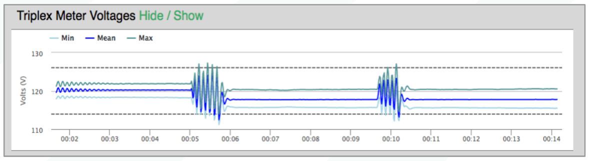
Control (No Attack or Defense) Scenario



Attack Scenario

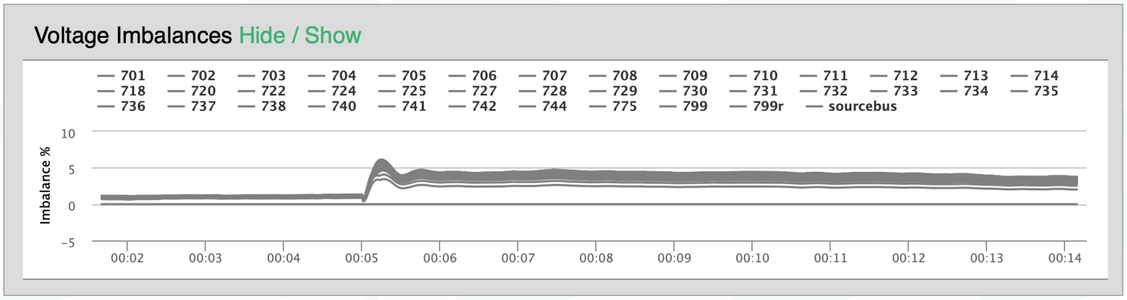


Attack + Defense Scenario

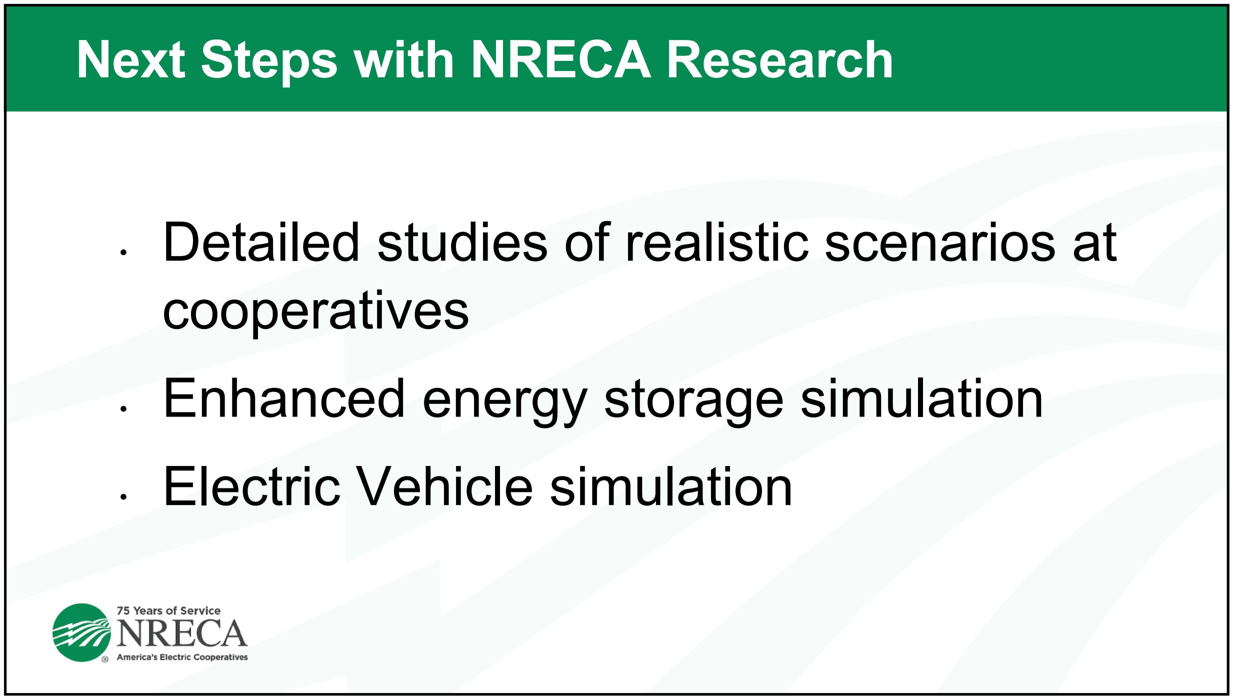


cyberInverters Outputs and Results

Other Outputs - include regulator tap changes, cap bank switching, and voltage imbalance to investigate second order effects.



75 Years of Service
NRECA
America's Electric Cooperatives

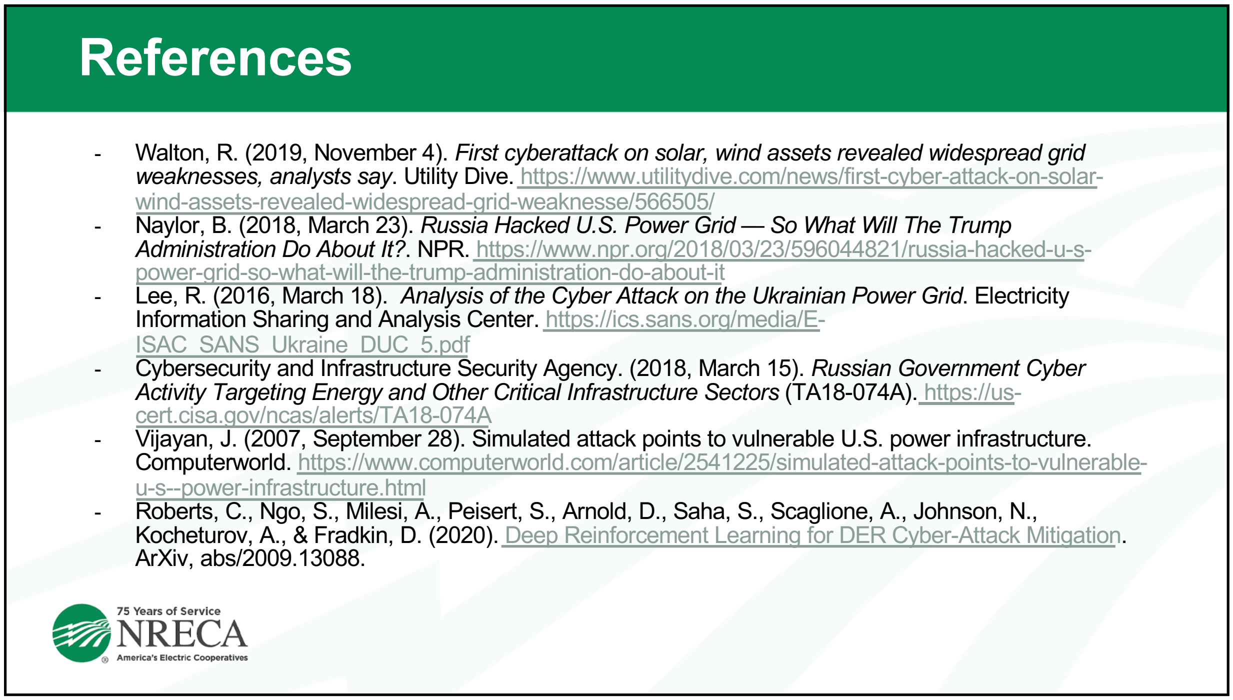


Next Steps with NRECA Research

- Detailed studies of realistic scenarios at cooperatives
- Enhanced energy storage simulation
- Electric Vehicle simulation

75 Years of Service NRECA America's Electric Cooperatives

17



References

- Walton, R. (2019, November 4). *First cyberattack on solar, wind assets revealed widespread grid weaknesses, analysts say*. Utility Dive. https://www.utilitydive.com/news/first-cyber-attack-on-solar-wind-assets-revealed-widespread-grid-weaknesse/566505/
- Naylor, B. (2018, March 23). *Russia Hacked U.S. Power Grid — So What Will The Trump Administration Do About It?*. NPR. https://www.npr.org/2018/03/23/596044821/russia-hacked-u-s-power-grid-so-what-will-the-trump-administration-do-about-it
- Lee, R. (2016, March 18). *Analysis of the Cyber Attack on the Ukrainian Power Grid*. Electricity Information Sharing and Analysis Center. https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf
- Cybersecurity and Infrastructure Security Agency. (2018, March 15). *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors* (TA18-074A). https://us-cert.cisa.gov/ncas/alerts/TA18-074A
- Vijayan, J. (2007, September 28). Simulated attack points to vulnerable U.S. power infrastructure. Computerworld. https://www.computerworld.com/article/2541225/simulated-attack-points-to-vulnerable-u-s--power-infrastructure.html
- Roberts, C., Ngo, S., Milesi, A., Peisert, S., Arnold, D., Saha, S., Scaglione, A., Johnson, N., Kocheturov, A., & Fradkin, D. (2020). Deep Reinforcement Learning for DER Cyber-Attack Mitigation. ArXiv, abs/2009.13088.

75 Years of Service NRECA America's Electric Cooperatives

18

5 Concluding Remarks

The CIGAR project studied the application of reinforcement learning to control non-compromised solar photovoltaic inverters to mitigate the effects of attacks on subsets of solar photovoltaic systems. Compromised devices were assumed to have had their smart inverter Volt-VAR and Volt-Watt curves adjusted to create large voltage oscillations and/or large voltage imbalances in the system. The CIGAR control solution seeks to adjust the remaining set of non-compromised solar inverters through a control policy derived from reinforcement learning to mitigate the oscillation/imbalance while minimizing the amount of active power curtailment and maintaining maximum power quality in the feeder.

The work undertaken in CIGAR demonstrates the effectiveness of the use of reinforcement learning to solve complicated nonlinear dynamic optimization problems for control of distributed energy resources. The reinforcement learning agents were shown to be effective in mitigating cyber attacks on grids of drastically different sizes and topologies. In most studied cases, the CIGAR control strategies operate best when the percentage of smart inverters compromised due to a cyber attack in a given grid is less than 50% of the total installed inverter capacity. For attacks larger than 50% there is simply not enough controllable resource to completely mitigate the effect of the cyberattack. However, in all cases studied, the CIGAR control strategy was shown to ameliorate the severity of the attack. Thus, CIGAR can provide a benefit to the system even when in cases when the attack cannot be completely eliminated.

The PyCIGAR software framework easily allows the incorporation of other controllable Distributed Energy Resources. In a follow-on project, the majority of the CIGAR team is studying the extension of PyCIGAR to incorporate distribution battery storage systems. It is likely that the addition of storage as a controllable resource will enhance the ability of the CIGAR approach to mitigate the effect of larger cyberattacks in distribution grids.

6 Appendix A: Generating High-Resolution Data

The availability of solar resource datasets that can be used to model Distributed Energy Resources (DER) is fundamental to understand the safety of equipment in distribution systems. High-resolution solar data is an essential part in the training of agents for adaptive control algorithms, hence, critical in the development of mitigation schemes used in the event of a cyber-physical attack. Being a weather-dependent natural resource, solar energy variability and uncertainty span different time windows, from seconds to seasons, for which high-resolution datasets are paramount in capturing the underlying phenomena.

Historical solar irradiance datasets are provided by ground-based measurement (e.g. pyranometers) or satellite remote sensing. The frequency of sampling of these sensing devices may vary from a few cycles per second to several seconds or minutes. Often times, these solar products are made available to the public in the form of second or minutely-averaged time series. Nonetheless, the accuracy of measurements from pyranometers is highly dependent on the instrument, acquisition and calibration methods, and the availability of such datasets is limited to sparsely distributed locations. Furthermore, these datasets often contain spurious or missing instances, thus, not well suited for the training of agents that can prevent a cyber-attack on inverter-based energy resources. This indicates the need for complete and reliable nation-wide solar datasets.

6.1 Using the National Solar Radiation Database

In contrast to ground-based measurements, satellite remote sensing can yield high-density measurements which can be used to create high resolution solar datasets in the spatial dimension. The National Solar Radiation Database (NSRDB) is a serially complete state-of-the-art collection of meteorological data developed at the National Renewable Energy Lab (NREL). The NSRDB uses a physics-based solar model (PSM) that leverages satellite-based measurement to generate meteorological variables covering the entire United States along with other international locations. Time series produced by NREL's PSM are provided at every 4-km and 30 minutes from 1998 to 2017. A snapshot of the NSRDB footprint is shown in Figure 16. The dataset includes three of the most common measurements of solar radiation (global horizontal, direct normal and diffuse horizontal irradiance) and other weather variables such as atmospheric pressure, ground-level temperature and wind speeds, or cloud coverage, among others. NREL offers a free Python API to retrieve NSRDB data from any location available, providing flexibility to operators looking to integrate the CIGAR tool.

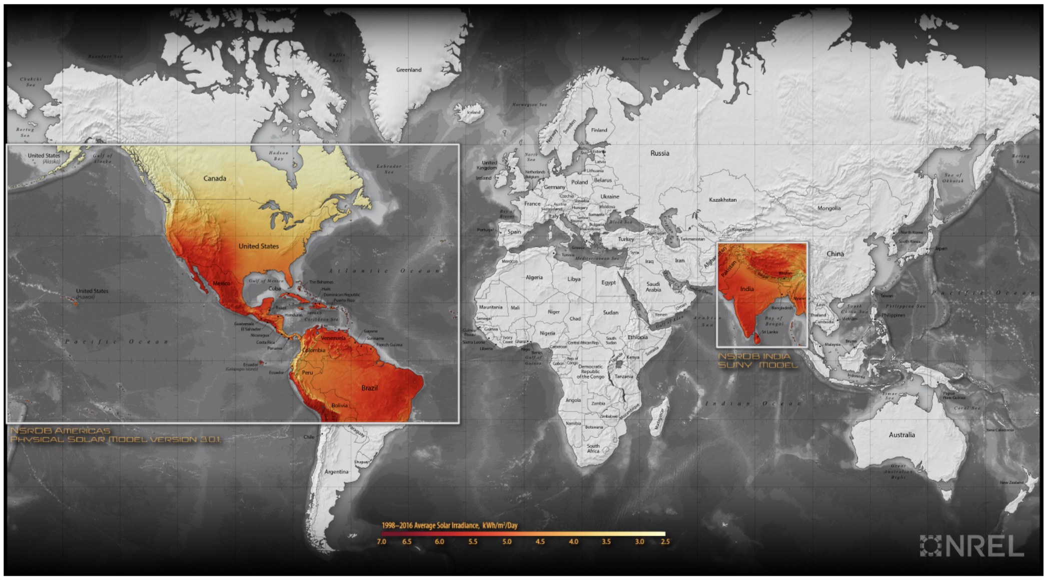


Figure 16: Footprint of the NSRDB

Although the NSRDB complements ground-based solar measurements by making time series available at multiple locations, it does not, however, match the sampling frequency of pyranometers. Consequently, the direct use of the NSRDB may mislead the agent by not capturing sub-hourly and sub-minutely solar variability, resulting in bad control policies. An alternative solution is to interpolate low-resolution data to obtain a more granular dataset. In practice, this method is not suited for the generation of high-resolution data since the variability of solar irradiance in sub-hourly times frames may strongly deviate from the interpolated values. An example is shown in Figure 17

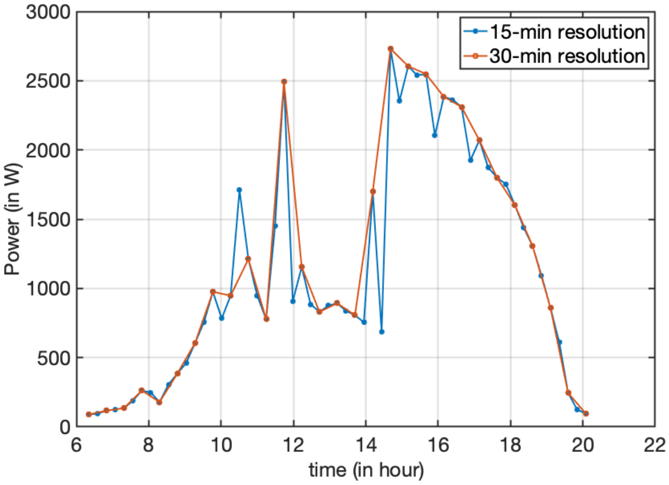


Figure 17: A comparison between a 30-min and 15-min solar profiles

In this task, we solve the issue of augmented resolution by leveraging the two metering approaches. We propose a method to learn the statistics of solar power from ground-based measurements in minutely and sub-minutely time scales under different weather regimes. We then superimpose a random process based on the statistics learned from the ground-based measurements on the solar data generated by the physical model, i.e. the NSRDB. We argue that climatic and topographic effects on solar power are captured by the NSRDB, and that sub-minutely variability of solar power behaves similarly across sites. In other words, the effects of clouds on solar power time frames under 30 minutes will be similar in different areas of the world.

6.2 Modeling PV power generation from an irradiance-based solar model

Solar irradiance received by the photovoltaic panel is used to generate DC current. Our approach to produce power time series from solar irradiance assumes fixed-tilt solar panels since we are mostly concerned about the applications to distribution systems with an abundance of rooftop solar. The angle of incidence θ in a fixed-tilt solar panel is given by

$$\theta = \cos^{-1}\left(\sin\theta_s \cos\left(\gamma - \gamma_s\right)\sin\beta + \cos\theta_s \cos\beta\right) \tag{23}$$

where $\theta_s, \gamma, \gamma_s$ and β are the solar zenith, surface azimuth, solar azimuth and surface tilt angles, respectively. As a result, the irradiance on the plane-of-array I_{POA} can be calculated as follows

$$I_{POA} = I_b + I_{ds} + I_{dg} \tag{24}$$

where I_b, I_{ds}, I_{dg} are the beam, sky diffuse and ground diffuse irradiance, respectively. It should be noted that $I_b = I_n \cos\theta$ where I_n is the normal component of the irradiance. The normal and diffuse components of irradiance are often given as a result of running climate models.

To account for module cover (shading), the calculation of transmitted irradiance yields

$$I_t = I_{POA} - (1 - f) I_b \cos\theta \tag{25}$$

where $1 - f$ denotes an attenuation coefficient and f is given as

$$f = b_0 + b_1\theta + b_2\theta^2 + b_3\theta^3 + b_4\theta^4 + b_5\theta^5 \tag{26}$$

and $b_0, b_1, b_2, b_3, b_4, b_5$ are coefficients of a polynomial fit specific to the glass in the panel. As a result, the DC power output yields

$$\hat{p}^{dc} = I_t p_0^{dc}\Big(1 + \kappa(\tau_c - \tau_r)\Big) \tag{27}$$

where $\hat{p}^{dc}$ is given in watts, p_0^{dc} is the solar panel DC nameplate capacity, $\kappa = -0.5\%/C$ is a temperature coefficient, and τ_c, τ_r are the cell and reference temperatures, respectively. In the following section, we present a solar model that incorporates the stochasticity due to cloud effects on three different regimes, namely sunny, overcast and partly cloudy.

6.3 Cloud regime parametrization

Solar irradiation is attenuated by clouds modeled as a random mask that subtracts a percentage of the light coming from the patch of sky it covers at a certain time. Let the power produced by the PV panel be $w_d[n]$ on day d and time instant $n \in 0, ..., N$. $w_d[n]$ can be expressed as

$$w_d[n] = s_d[n] - \Big(p_d^b[n] + p_d^{dif}[n]\Big) + p_d^e[n] + \eta_d[n] \tag{28}$$

where $s_d[n]$ is the clearsky component, and $p_d^b[n], p_d^{dif}[n]$ are the direct and diffuse components, $p_d^e[n]$ denotes the edge of cloud effect and $\eta_d[n]$ is Gaussian noise. We distinguish between sunny and cloudy days. In the sunny day, we have that $w_n[k] \equiv s_n[k]$. For the cloudy day, the power components are given as follows,

$$\begin{aligned} p_n^b[k] &\approx a_n^b[k] s_n[k], \quad a_n^b[k] = \sum_{\ell \in \mathcal{B}} a_\ell \delta\left[k - r_\ell\right] \\ p_n^d[k] &\approx \sum_q \tilde{h}[q] z_n[k - q] \end{aligned} \tag{29}$$

where $a_n^b[k]$ is the stochastic time series capturing the direct beam sudden power attenuations caused by clouds whose trajectories intersect with that of the sun. The diffuse beam attenuation, instead, is modeled as the convolution of a one-dimensional filter $\tilde{h}[k]$ with a stochastic input $z_n[k]$ that represents the cloud attenuation. To capture the edge of the cloud effect the term $p_n^e[k]$ is introduced when $w_n[k] > s_n[k]$ and is defined as follows

$$p_n^e[k] \approx a_n^e[k] s_n[k], \quad a_n^e[k] = \sum_{\ell \in \mathcal{E}} a_\ell \delta\left[k - k_\ell\right] \tag{30}$$

defined for the edges of some clouds $l \in \mathcal{E}$. As a result, the complete model for the cloudy day is given as

$$w_n[k] = \begin{cases} s_n[k]\left(1 - a_n^b[k]\right) - \sum_q \tilde{h}[q] z_n[k-q] \quad \textbf{when} \quad w_n[k] \leq s_n[k] \\ s_n[k] + a_n^e[k] s_n[k] \quad \textbf{when} \quad w_n[k] > s_n[k] \end{cases} \tag{31}$$

Such a model allows the separation of the components and study a plausible stochastic model for them. In the following subsection, we present three stochastic models for sunny, overcast and partly cloudy weather regimes.

6.4 Stochastic model for sunny, overcast and partly cloudy regimes

We present a model that switches between regimes as shown in Figure 18. The partly cloudy model is slightly more involved due to the presence of the three models, captured by a Hiden Markov Model (HMM). The NSRDB provides deterministically provides the weather regimes by incorporating an extensive cloud classification. This can also be done by analyzing the resulting irradiance values, comparing them with the clearsky components. Therefore, the only underlying stochastic process is that of the HMM.

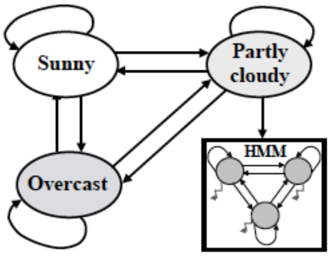


Figure 18: Switching process between stochastic models

In the sunny regime, we assume that the power is given by the clearsky value along with noise. In the overcast case, the clearsky component will be attenuated by the clouds. The sunny day model is given as follows

$$\textbf{Sunny period}: \quad w_d[n] = s[n] + \eta_s[n] \quad \forall n \in 0, \ldots, N \tag{32}$$

where the modelling error is drawn from a Gaussian distribution $\eta_s \sim \mathcal{N}(0, \sigma_c^2)$ and the variance σ_c^2 can be calculated from the error values after fitting the clearsky values. Similarly, we can provide a similar model for the overcast day

$$\textbf{Overcast period}: \quad w_d^{oc}[n] = \alpha_n s[n] + \eta_{oc}[n] \quad \forall n \in 0, \ldots, N \tag{33}$$

where $\alpha_n \forall n \in 0, \ldots, N$ is the attenuation coefficient found by solving a least squares regression problem with respect to the sunny value, and where the residual is $\eta_{oc} \sim \mathcal{N}(0, \sigma_{oc}^2)$.

For the partly cloudy period, we use a Hiden Markov Model (HMM) that can characterize different transitions between states due to the effects of moving clouds on solar irradiance received by the solar panel. The HMM is explained in [26] and is summarized here. A hidden Markov model (HMM) can capture the underlying on-off process that characterizes the sparse parameters in periods with fast moving clouds that cause sharp fluctuations in solar PV power. The observed solar PV power data $w_d[n]$ is modeled as coming from underlying hidden states that are Markovian in nature. Let the state/latent vector, $\mathbf{q}_n$ be a coordinate vector that enumerates all the possible support combinations of sparse parameters for time instant n, i.e. of the vector

$$\mathbf{x}_n = \begin{bmatrix} z_d[k-M+1] & \ldots & z_d[n] & a_d^b[k] & a_d^e[k] \end{bmatrix}^T \tag{34}$$

. The relationship between observations of solar power and latent state is:

$$w_d[n] = \alpha_d s_d[n] - \mathbf{P}\ \mathrm{diag}\left(\mathbf{\Phi}\mathbf{q}_n\right)\mathbf{x}_n \tag{35}$$

$$\mathbb{E}\left(\mathbf{q}_{n+1}|\mathbf{q}_n\right) = \mathbf{A}^T\mathbf{q}_n \tag{36}$$

$$\mathbf{P} = \begin{bmatrix} h[M-1] & \ldots & h[0] & s_d[n] & -s_d[n] \end{bmatrix} \tag{37}$$

Parameter α_d is included to model any constant attenuation over the time period considered unlike the other sparse parameters $\mathbf{x}_n$. Let the total number of states be N_s. Then, $\mathbf{A} \in \mathbb{R}^{N_s \times N_s}$ is the state transition matrix where $\mathbf{A}(i, j)$ is the probability of going from state i to state j. The state vector $\mathbf{q}_n \in \mathbb{R}^{(M+2)}$ is a binary vector taking values from the set of coordinate vectors $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{N_s}\}$ where $\mathbf{e}_i \in \mathcal{R}^{N_s}$ has a 1 at position i and zero elsewhere. Each column in matrix $\mathbf{\Phi} \in \mathbb{R}^{(M+2)\times N_s}$ consists of a possible support of vector $\mathbf{x}_n$.

Certain assumptions are made to decrease the number of states such that there is only 1 non-zero element in the vector $\boldsymbol{x}_n$. Therefore, the number of states are $N_s = M+3$. Noise term can be ignored in the observation (35) since it is small in amplitude relative to the 'noisy' nature of the solar power data that is caused by the fast movement of clouds. All non-zero coefficients in $\mathbf{x}_n$ are hypothesized to come from independent exponential distributions with different parameters since they have different levels of sparsity. While in state i a certain $w_d[n]$ is observed:

$$\textbf{Partly-cloudy period}: \quad w_d[n] = \begin{cases} \alpha_d s_d[n],\ i = 1 \\ \alpha_d s_d[n] - h[i-2] z_d[n-i+2], i = 2,\ldots, M+1, \\ \alpha_d s_d[n] - s_d[n] a_d^b[n],\ i = M+2 \\ \alpha_d s_d[n] + s_d[n] a_d^e[n],\ i = N_s \end{cases} \tag{38}$$

where the following statistical model is postulated for the parameters

$$z_d[n] \sim \exp(\lambda_z), a_d^b[n] \sim \exp(\lambda_{a^b}), a_d^e[n] \sim \exp(\lambda_{a^e}) \tag{39}$$

We showcase our models for a Solarcity site as shown in Figure 19. The dataset is presented as a 15-minute resolution time series. We generate higher resolution, i.e. 1-second, time series with the parameters learned using the stochastic models presented above. Our future work will include learning parameters for different datasets, e.g. PMU data from a site in Riverside, CA, to validate the statistical significance of these values. In the following section, we show some preliminary results for the generation of time series using the NSRDB

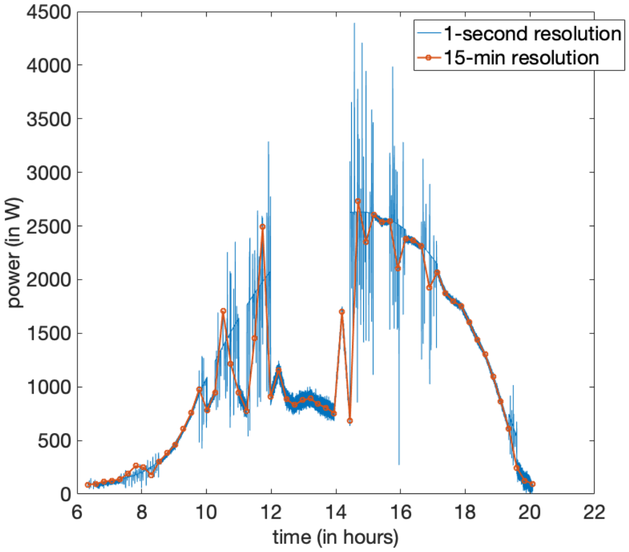


Figure 19: Solarcity 15-minute averaged and generated 1-second power time series

6.5 Showcasing the model for the NSRDB

We showcase our approach by selecting a site from the NSRDB. The NSRDB provides solar irradiance (the global, direct and diffuse components as well as clearsky values) at a 30-min resolution. We generate solar irradiance values with a 1-second resolution and present the results in Figure 20. In future work, we will aim to generate power time series using the NSRDB irradiance and compare them with real datasets as a validation step.

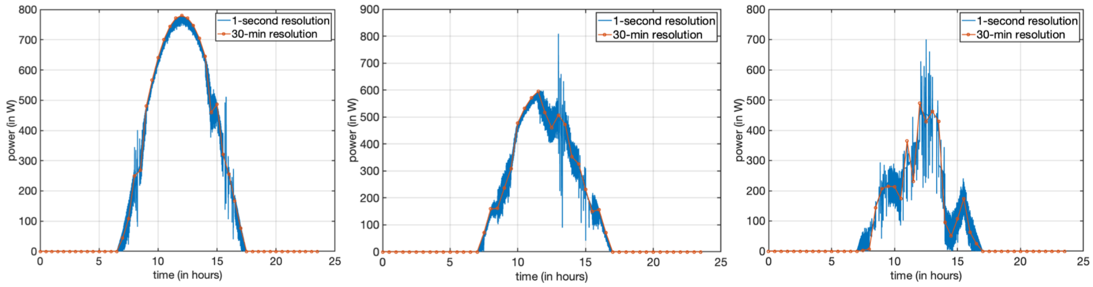


Figure 20: Comparison between NSRDB 30-minute averaged and generated 1-second resolution power. Results are shown for a site in Riverside, for different days. From left to right, January 14, January 20 and February 21 2012

Acknowledgements

This research was supported by the Director, Cybersecurity, Energy Security, and Emergency Response, Cybersecurity for Energy Delivery Systems program, of the U.S. Department of Energy, under contract DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

References

[1] B. Seal, "Common Functions for Smart Inverters, 4th Ed." Electric Power Research Institute, Tech. Rep. 3002008217, 2017.

[2] "Grid modernization strategy - draft report," https://www.hawaiielectriclight.com/Documents/about_us/investing_in_the_future/grid_modernization_strategy_draft.pdf, accessed: 2017-07-15.

[3] M. Farivar, L. Chen, and S. Low, "Equilibrium and dynamics of local voltage control in distribution systems," in *52nd IEEE Conference on Decision and Control*, Dec 2013, pp. 4329–4334.

[4] C. Roberts, S. Ngo, A. Milesi, S. Peisert, S. Saha, A. Scaglione, N. Johnson, A. Kocheturov, D. Fradkin, and D. Arnold, "Deep reinforcement learning for DER cyber-attack mitigation," *arXiv preprint arXiv:2009.13088*, 2020.

[5] C. Roberts, S. Ngo, A. Milesi, S. Peisert, A. Scaglione, and D. Arnold, "Deep reinforcement learning for mitigating cyber-physical der voltage unbalance attacks," *American Control Conference*, accepted.

[6] M. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 735–743, Jan 1989.

[7] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[8] M. Farivar, X. Zho, and L. Chen, "Local voltage control in distribution systems: An incremental control algorithm," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 732–737.

[9] S. S. Saha, D. Arnold, A. Scaglione, E. Schweitzer, C. Roberts, S. Peisert, and N. G. Johnson, "Lyapunov stability of smart inverters using linearized distflow approximation," *IET Renewable Power Generation*, vol. 15, no. 1, pp. 114–126, 2021.

[10] J. H. Braslavsky, L. D. Collins, and J. K. Ward, "Voltage stability in a grid-connected inverter with automatic volt-watt and volt-var functions," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[11] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, 2019.

[12] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, 2019.

[13] C. Li, C. Jin, and R. Sharma, "Coordination of pv smart inverters using deep reinforcement learning for grid voltage regulation," *arXiv preprint arXiv:1910.05907*, 2019.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[15] M. A. OpenAI, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning (2013)," *arXiv preprint arXiv:1312.5602*, vol. 99, 2013.

[17] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[19] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[20] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[22] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[25] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," 2017.

[26] R. Ramakrishna, A. Scaglione, V. Vittal, E. Dall'Anese, and A. Bernstein, "A model for joint probabilistic forecast of solar photovoltaic power and outdoor temperature," *IEEE Transactions on Signal Processing*, vol. 67, no. 24, pp. 6368–6383, 2019.