Restoring Critical Loads in Resilient Distribution Systems Using a Curriculum Learned Controller

Preprint

Xiangyu Zhang, Abinet Tesfaye Eseye, Matthew Reynolds, Bernard Knueven and Wesley Jones

National Renewable Energy Laboratory

Presented at the 2021 IEEE Power & Energy Society General Meeting
July 25-29, 2021

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-2C00-78351
January 2021

Restoring Critical Loads in Resilient Distribution Systems Using a Curriculum Learned Controller

Preprint

Xiangyu Zhang, Abinet Tesfaye Eseye, Matthew Reynolds, Bernard Knueven and Wesley Jones

National Renewable Energy Laboratory

Suggested Citation
Zhang, Xiangyu, Abinet Tesfaye Eseye, Matthew Reynolds, Bernard Knueven and Wesley Jones. 2021. *Restoring Critical Loads in Resilient Distribution Systems Using a Curriculum Learned Controller: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-2C00-78351. https://www.nrel.gov/docs/fy21osti/78351.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-2C00-78351
January 2021

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Office of Electricity Delivery and Energy Reliability.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Restoring Critical Loads in Resilient Distribution Systems Using a Curriculum Learned Controller

Xiangyu Zhang, Abinet Tesfaye Eseye, Matthew Reynolds, Bernard Knueven and Wesley Jones

***Abstract*—In this paper, we propose a curriculum learned reinforcement learning (RL) controller to facilitate distribution system critical load restoration (CLR), leveraging RL's fast online response and its outstanding optimal sequential control capability. Like many grid control problems, CLR is complicated due to the large control action space and renewable uncertainty in a heavily constrained non-linear environment with strong intertemporal dependency. The nature of the problem oftentimes causes the RL policy to converge to a poor-performing local optimum if learned directly. To overcome this, we design a two-stage curriculum in which the RL agent will learn generation control and load restoration decision under different scenarios progressively. Via curriculum learning, the trained RL controller is expected to achieve a better control performance, with critical loads restored as rapidly and reliably as possible. Using the IEEE 13-bus test system, we illustrate the performance of the RL controller trained by the proposed curriculum-based method.**

***Index Terms*—grid resiliency, load restoration, microgrid, reinforcement learning, curriculum learning**

I. Introduction

A resilient distribution system should be able to withstand the impact of extreme events and initiate a swift recovery. In the event of a substation outage and subsequent distribution feeder de-energization, it is essential to rapidly restore critical loads in the system, which is now made possible by the increasing number of distributed energy resources (DERs) installed. Previous studies on this topic include: A restoration process, which first determines post-restoration topology and then restores loads and sets generation outputs, is introduced in [1]. Liu *et al.* [2] leverage both fixed DERs and mobile resources for system restoration using model predictive control (MPC). To cope with generation uncertainty introduced by renewable DERs, in [3], a chance-constrained method is proposed for load restoration to limit risk. However, a bottleneck for these approaches is that they require solutions of resource-intensive optimization problems during online control. This issue of real-time computational complexity is usually circumvented by using a longer control interval and a shorter look-ahead horizon, which inevitably impact the control accuracy.

In the past decade, reinforcement learning (RL), as an alternative to optimization-based approaches, has shown great performance in domains related to optimal sequential decision-making [4]–[6]. Similarly, using RL for critical load restoration (CLR) can be beneficial as CLR is a typical optimal sequential control problem with strong temporal dependency. Additionally, when compared with optimization-based approaches, RL for CLR has following merits: i) RL does not require on-demand computation during real-time control and thus can be faster and fine-granular (w.r.t control interval). ii) Historical renewable generation data can be directly sampled for controller training, avoiding the need for scenario reduction or distribution identification. iii) RL can directly learn from a nonlinear unbalanced multi-phase AC power flow model instead of a simplified one, for a more accurate control performance. iv) An RL controller can also provide ancillary information to facilitate grid operator's decision-making (e.g., value function of RL can provide an expected performance evaluation when the restoration process starts).

Currently, to the best of our knowledge, no other studies have aimed at exploring RL's capability for CLR, with the exception of our earlier work [7]. In [7], from an energy adequacy perspective, we compared RL with MPC for CLR considering imperfect renewable forecasts in a single-bus system. In this work, we extend the work to a network-constrained three-phase unbalanced distribution system, where power flow and voltage violation are considered. This extension, however, introduces more complexities such as nonlinear power flow constraints and a larger action space, inevitably making it harder to learn an RL policy (i.e. the learning can be easily trapped in a poor-performing local optimum). To overcome this, as our major contribution, we propose a novel curriculum learning (CL) approach to guide the RL agent to escape poorly-performing local optima and learn a better control policy in a divide-and-conquer manner.

II. Problem Formulation

A prioritized CLR problem after a distribution system islanded due to the substation outage is investigated. Critical load $i \in \mathcal{L}$ in the system is prioritized by the importance factor ζ^i ($\mathbf{H} = [\zeta^1, \zeta^2, ..., \zeta^N]^\top \in \mathbb{R}^N$ for all loads, and N is the number of loads). Available DERs assets like renewable DERs ($\mathcal{R}$) and dispatchable DERs ($\mathcal{D}$) can be used for restoration. To solve this problem, following assumptions are made:

The authors are with the U.S. National Renewable Energy Laboratory, Golden, CO 80401, USA. xiangyu.zhang@nrel.gov

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the DOE Office of Electricity (OE) Advanced Grid Modeling program. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.

1) Available energy for $\mathcal{D}$ are limited.

2) The length of restoration horizon/substation repair time $\mathcal{T}$ is deterministic and known when restoration starts.

3) Power of critical loads $\mathcal{L}$ ($\mathbf{P} = [p^1, p^2, ..., p^N]^\top \in \mathbb{R}^N$ and $\mathbf{Q} = [q^1, q^2, ..., q^N]^\top \in \mathbb{R}^N$) is time-invariant over $\mathcal{T}$, and can be partially restored with the same power factor.

4) Generation for $\mathcal{R}$ can be predicted accurately up to one hour ahead, beyond which no forecast is available. Considering $\mathcal{T}$ usually spans several hours, the proposed CLR problem is still categorized as stochastic optimal control.

5) Grid topology is assumed to be intact and we defer the inclusion of topology restoration to our future work.

A. Objective

At each control step $t \in \mathcal{T}$, active power set point and power factor angle for DERs (i.e., $\mathbf{P}_t^{\mathcal{G}} \in \mathbb{R}^{|\mathcal{G}|}$ and $\boldsymbol{\alpha}_t^{\mathcal{G}} \in \mathbb{R}^{|\mathcal{G}|}$ for $\mathcal{G} = \mathcal{D} \bigcup \mathcal{R}$, in which $|\cdot|$ calculates the cardinality of a set) and the demand restored for each load (i.e., $\mathbf{P}_t = [p_t^1, p_t^2, ..., p_t^N]^\top \in \mathbb{R}^N$ and $\mathbf{Q}_t = [q_t^1, q_t^2, ..., q_t^N]^\top \in \mathbb{R}^N$) are dynamically determined in order to *maximize* the following objective function:

$$C = \sum_{t \in \mathcal{T}} (C_t^{LR} + C_t^{VV}), \tag{1}$$

where $C_t^{LR} = \mathbf{H}^\top \mathbf{P}_t - \mathbf{H}^\top \text{diag}\{\boldsymbol{\epsilon}\}[\mathbf{P}_{t-1} - \mathbf{P}_t]^+$ represents the load restoration reward, in which $[[x^1, x^2, ..., x^N]]^+ = [(x^1)^+, (x^2)^+, ..., (x^N)^+]^\top$ and $(x^i)^+ = \max(0, x^i)$. Specifically, in C_t^{LR}, the first term encourages load restoration and the second term penalizes shedding previously restored load by $\boldsymbol{\epsilon} = \{\epsilon^1, \epsilon^2, ..., \epsilon^N\}^\top \in \mathbb{R}^N$. Introducing this penalty is to encourage a reliable restoration considering the intermittent renewable generation. At $t = 0$, we assume all loads are not served (i.e., $\mathbf{P}_0 = \mathbf{0}$). Besides load restoration reward, voltage violation is penalized by $C_t^{VV} = -\sum_{n \in \mathcal{N}_b} \lambda[\max(0, V_t^n - \overline{V})^2 + \max(0, \underline{V} - V_t^n)^2]$ in which $\mathcal{N}_b$ is the set of all buses, V_t^n is the voltage magnitude of bus n at time t, $[\underline{V}, \overline{V}]$ are the normal voltage boundaries (e.g., ANSI C.84.1 limits) and λ is the unit penalty, typically a large positive number.

B. Constraints

While maximizing the control credit C, the following operation constraints should be satisfied for all $t \in \mathcal{T}$:

1) *Fuel based DERs*: Power, energy and power factor angle limit should be satisfied; specifically, $\forall g \in \mathcal{D}^{fuel}$, there are:

$$\underline{p^g} \le p_t^g \le \overline{p^g}, \quad \sum_{t \in \mathcal{T}} p_t^g \cdot \tau \le E^g, \quad \alpha_t^g \in [\underline{\alpha^g}, \overline{\alpha^g}], \tag{2}$$

in which τ is the control interval (unit: hour), E^g is the known maximum energy limit (e.g., limited by fuel quantity) and α_t^g is the operating power factor angle ($\alpha_t^g = \arctan(q_t^g / p_t^g)$). Ramping rate limits are not considered.

2) *Storage*: Storage output/state of charge (SOC) feasibility, charging/discharging state transition, initial storage and power factor angle, $\forall \theta \in \mathcal{D}^{ES}$, are constrained by:

$$\begin{aligned} -p^{\theta,ch} \le p_t^\theta \le p^{\theta,dis}, \quad S_{t+1}^\theta = S_t^\theta - \eta_t \cdot p_t^\theta \cdot \tau \\ \underline{S^\theta} \le S_t^\theta \le \overline{S^\theta}, \quad S_0^\theta = s_0, \quad \alpha_t^\theta \in [\underline{\alpha^\theta}, \overline{\alpha^\theta}], \end{aligned} \tag{3}$$

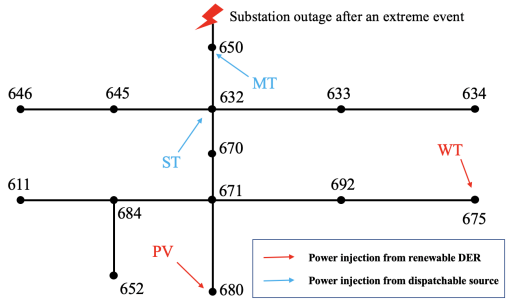


Fig. 1. Illustration of the IEEE 13-bus test system with added DERs.

in which η_t is the energy storage efficiency and $\eta_t = 1/\eta^{dis}$ when battery is discharging ($p_t^\theta > 0$) and $\eta_t = \eta^{ch}$ when it is charging ($p_t^\theta < 0$). S_t^θ and s_0 are the current and initial SOC.

3) *Renewable DERs*: Renewable generation are limited by available natural resources, and the power factor angle should satisfy limits of the inverter; specifically, $\forall r \in \mathcal{R}$, there are:

$$0 \le p_t^r \le \overline{p_t^r}, \quad \alpha_t^r \in [\underline{\alpha^r}, \overline{\alpha_t^r}], \tag{4}$$

in which $\overline{p_t^r}$ is the time-variant available natural resources, and gaps between p_t^r and $\overline{p_t^r}$ represents renewable curtailment.

4) *Loads*: Load pick-up decision should satisfy:

$$\mathbf{0} \le \mathbf{P}_t \le \mathbf{P}, \quad \mathbf{0} \le \mathbf{Q}_t \le \mathbf{Q}, \quad p_t^i / q_t^i = p^i / q^i. \tag{5}$$

5) *Network constraints*: Power flow relationship among all electrical values should be satisfied (e.g., constraints instantiated using an AC power flow model); details are omitted here in the interest of space.

Concretely, in this paper, we consider a modified IEEE 13-bus test system with four DERs: a micro-turbine $\mathcal{D}^{fuel} = \{\mu\}$, an energy storage $\mathcal{D}^{ES} = \{\theta\}$, a wind turbine and a photo-voltaic (PV) generation $\mathcal{R} = \{\omega, \rho\}$, see Fig. 1. These notations for DERs will be used for the rest of the paper.

III. The Curriculum Learning Framework

In this section, we will transform the above-mentioned optimal control problem into a Markov Decision Process (MDP) and then show how to use RL in a CL framework to effectively train a CLR controller.

A. Markov Decision Process (MDP) Formulation

Three key MDP elements (*state*, *action* and *reward*) corresponding to the optimal control problem are defined below.

State reflects the system status of the current step and is used by an RL agent for decision-making. In this study, the MDP state and the state space ($\mathbf{s}_t \in \mathcal{S}$) are defined as:

$$\mathbf{s}_t := [\mathbf{P}_t^\rho, \mathbf{P}_t^\omega, \widetilde{\mathbf{P}}_t, S_t^\theta, \mathring{E}_t^\mu, t] \in \mathcal{S} \subset \mathbb{R}^{2/\tau + N + 3}, \tag{6}$$

in which $\mathbf{P}_t^\rho \in \mathbb{R}^{1/\tau}$ and $\mathbf{P}_t^\omega \in \mathbb{R}^{1/\tau}$ are the PV and wind generation forecast for the next hour (recall Assumption 4 in Section II); $\widetilde{\mathbf{P}}_t := \text{diag}\{\mathbf{P}\}^{-1}\mathbf{P}_t \in \mathbb{R}^N$ shows the fractional load restoration level. S_t^θ and $\mathring{E}_t^\mu$ are the current SOC for the storage and remaining fuel for the micro-turbine, revealing the remaining load supporting capability. Current step index t is also included to inform the progress.

Action represents the decision on control variables the RL agent needs to make at $t \in \mathcal{T}$. Here, action and the action space ($\mathbf{a}_t \in \mathcal{A}$) are defined as:

$$\mathbf{a}_t := [\mathbf{P}_t, p_t^\theta, \alpha_t^\theta, \alpha_t^\omega, \alpha_t^\rho] \in \mathcal{A} \subset \mathbb{R}^{N+4}, \tag{7}$$

in which $\mathbf{P}_t$ is the load pick-up decision; p_t^θ is the storage power output and α_t^i for $i \in \{\theta, \omega, \rho\}$ are the generator/inverter power factor angles for the energy storage, wind turbine and PV system, respectively. Note the micro-turbine is used for power balance and thus its control (p_t^μ and α_t^μ) is not included in (7). In addition, renewable generation (p_t^ρ and p_t^ω) are also not included in $\mathbf{a}_t$ because these renewable energy resources are encouraged to be used entirely by default, but the agent can still decide to curtail renewables (in case of voltage issue) implicitly as shown in Section IV-B.

Rewards are returned to the RL learning agent at each step, given $\mathbf{s}_t$ and $\mathbf{a}_t$, to provide an immediate evaluation of the control. Corresponding to the optimal control problem, the reward is straightforwardly defined as $r_t = C_t^{LR} + C_t^{VV}$.

B. Reinforcement Learning (RL)

In general, training an RL agent is to learn a control policy $\mathbf{a}_t = \pi_{\boldsymbol{\psi}}(\mathbf{s}_t)$ parameterized by a vector $\boldsymbol{\psi}$ that determines a control behavior which will maximize the expected cumulative reward $J(\boldsymbol{\psi}) = \mathbb{E}_{\pi_{\boldsymbol{\psi}}}(\sum_{t\in\mathcal{T}} r_t) = \mathbb{E}_{\pi_{\boldsymbol{\psi}}}(C)$. Policy gradient methods, a category of RL algorithms, use gradient ascent to update policy at each learning iteration by $\boldsymbol{\psi}_{k+1} = \boldsymbol{\psi}_k + \kappa\widehat{\nabla}_{\boldsymbol{\psi}} J(\boldsymbol{\psi})$, in which κ is the step size and $\widehat{\nabla}_{\boldsymbol{\psi}} J(\boldsymbol{\psi})$ is the gradient estimated using experience collected in each learning iteration. Typically, in deep RL, $\pi_{\boldsymbol{\psi}}(\mathbf{s}_t)$ is instantiated using a neural network (called policy network). In this study, our goal is to learn a policy $\pi_{\boldsymbol{\psi}}(\mathbf{s}_t)$ (i.e. train a policy network that maximize $J(\boldsymbol{\psi}) = \mathbb{E}_{\pi_{\boldsymbol{\psi}}}(C)$) and use it to solve the optimal CLR problem proposed in Section II.

C. Curriculum Learning (CL) and Knowledge Transferring

Due to the complexity of the grid control problem and large continuous search spaces (i.e., $\mathcal{S}$ and $\mathcal{A}$), in practice, optimal policy searching oftentimes end up at poor-performing local optima. To ameliorate this, CL, which is expected to find a better local/global optimum of a non-convex training environment [8], is utilized. Specifically, instead of training an RL controller for a difficult problem directly, CL is phased by a curriculum with problems of gradually increased difficulty. By learning to solve these problems and accumulate knowledge progressively, an RL controller can eventually solve the original hard problem with better performance, when compared with a directly trained RL controller.

Originally, as shown in (7), an RL agent must learn entirely from experience the control strategy for both *generation dispatch* and *load restoration* in a non-linear and stochastic environment, which is considered hard. Therefore, we introduce a curriculum with two stages: *1) in Stage I*, an RL controller only learns DERs dispatch ($\mathbf{a}_t^I = [p_t^\theta, \alpha_t^\theta, p_t^\mu, \alpha_t^\mu, \alpha_t^\omega, \alpha_t^\rho] \in \mathcal{A}^I$) and loads are restored using rule-based greedy restoration (i.e., given available generation, higher priority load always restored first). *2) in Stage II*, warm-started with the DERs control knowledge learned in Stage I, the RL controller now learns to solve the original CLR problem by improving control strategy for generation and load restoration simultaneously ($\mathbf{a}_t^{II} = \mathbf{a}_t \in \mathcal{A}$). With such a curriculum, the introduction of Stage I breaks down the learning task and provides a stepping stone for the RL agent to learn the harder problem.

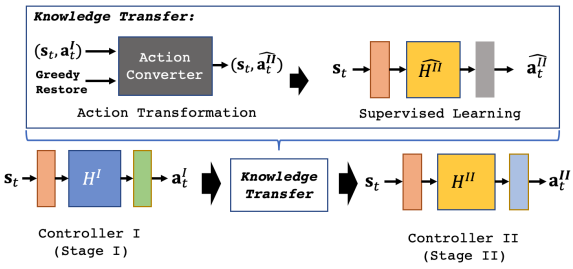


Fig. 2. A two-stage curriculum learning procedure.

In contrast to existing CL practices, in our case, policy networks for the two stages are heterogeneous (different output dimension due to $\mathcal{A}^I \neq \mathcal{A}$). To achieve knowledge transfer between them, we propose two special steps: 1) After Stage I, using the trained Controller I, an adequate number of control trajectories are simulated, from which state-action pairs reflect the controller's behavior are obtained (i.e. $(\mathbf{s}_t, \mathbf{a}_t^I) \in \mathcal{B}$). Then $\mathbf{a}_t^I$ are transformed to the format of $\mathbf{a}_t^{II}$ to get $(\mathbf{s}_t, \widehat{\mathbf{a}_t^{II}}) \in \widehat{\mathcal{B}}$, where $\mathbf{P}_t$ in $\widehat{\mathbf{a}_t^{II}}$ reflects greedy restoration behavior. 2) Using $\widehat{\mathcal{B}}$ as training data set, a neural network is trained via supervised learning. Such network contains the knowledge of Controller I and is then used for warm-starting Controller II given the shared network structure (see Fig. 2).

IV. Case Study

A. Experiment Settings

In the modified IEEE 13-bus system, we created $N = 15$ critical loads distributed in the three-phase system with total loads of $228.9 + j124.8$, $208.7 + j120.4$ and $290.3 + j140.8$ (unit: kW and kvar) for each phase respectively (Specific power for each load and their corresponding bus number are omitted here due to limited space). Importance factors $\mathbf{H}^\top = [1.0, 1.0, 0.9, 0.85, 0.8, 0.8, 0.75, 0.7, 0.65, 0.5, 0.45, 0.4, 0.3, 0.3, 0.2]$ are used. The restoration duration is assumed to be 6 hours with 5-min control intervals ($\tau = 1/12$ and $|\mathcal{T}| = 72$). For any load i, there is $\epsilon^i = 100$, and since $\epsilon^i > |\mathcal{T}|$, it means shedding any previously restored load will be penalized. To avoid voltage violation, $\lambda = 10^6$ is used. DERs' maximum capacities are $\overline{p^\omega} = 400$, $\overline{p^\rho} = 300$, $\overline{p^\mu} = 400$ and $p^{\theta,ch} = -p^{\theta,dis} = 250$ (unit: kW). The power factor angle should follow $\alpha^i \in [0, \pi/4], \forall i \in \{\omega, \rho, \theta, \mu\}$, assuming they can generate, not consume, reactive power. Fuel reserve for micro-turbine can provide 1200 kWh energy; storage's initial SOC is sampled from a truncated Gaussian distribution $s_0 \sim \mathcal{TN}(1000, 250)$ and there is $[\underline{S^\theta}, \overline{S^\theta}] = [160, 1250]$kWh.

B. Learning Environment Design

In this study, OpenDSS [9] is the simulator encapsulated by an RL learning OpenAI Gym [10] environment, providing

power flow solution at each step. Considering the system is islanded, there should be $\sum_{g\in\mathcal{D}\bigcup\mathcal{R}} p_t^g \approx \mathbf{1}^\top \mathbf{P}_t, \forall t \in \mathcal{T}$ (not strict equal due to losses). However, both generation output and load restoration are determined by the RL agent as shown in (7) (we focus on Stage II environment in this section), and the corresponding power balance constraint cannot be guaranteed as enforcing constraints on neural network outputs is in general hard. As a result, we enforce power balancing in the learning environment, using logic shown in Algorithm 1, before power flow computation in each time step. Reactive power is also balanced in the similar manner. In practice, we add a *Vsource* to the same bus with micro-turbine (as a requirement for OpenDSS to simulate islanded system), and it serves as a slack bus but can only compensate small amount of system losses. Regarding renewable DERs, wind and PV generation profile from two months (July and August) are collected, July data is used for RL controller training and August data is for performance evaluating.

Algorithm 1 Active power balancing in simulation

Input: $\mathbf{P}_t$, p_t^ρ, p_t^ω, p_t^θ, $\underline{p^\mu}$, $\overline{p_t^\mu}$

Get feasible generation range:
$\underline{G_t} = p_t^\rho + p_t^\omega + p_t^{\theta,g} + \underline{p^\mu}$, $\overline{G_t} = p_t^\rho + p_t^\omega + p_t^{\theta,g} + \overline{p_t^\mu}$
Define $p_t^{\theta,g} = max(0, p_t^\theta)$, $p_t^{\theta,l} = max(0, -p_t^\theta)$
if $\mathbf{1}^\top \mathbf{P}_t + p_t^{\theta,l} \geq \overline{G_t}$ (Insufficient gen) **then**
  if $p_t^{\theta,l} \geq \overline{G_t}$ (e.g., $p_t^\rho = 0$, $p_t^\omega = 0$ and $\mathring{E}_t^\mu$ is low) **then**
    $p_t^\mu = p_t^{\theta,l}, \mathbf{P}_t^* = \mathbf{0}$
  else
    Start from lower ζ^i, sequentially making $p_t^i = 0$ from $\mathbf{P}_t$ to get $\mathbf{P}_t^*$, until $\mathbf{1}^\top \mathbf{P}_t^* + p_t^{\theta,l} \leq \overline{G_t}$
    $p_t^\mu = \mathbf{1}^\top \mathbf{P}_t^* + p_t^{\theta,l} - \underline{G_t}$
  end if
  $p_t^{\omega,*} = p_t^\omega, p_t^{\rho,*} = p_t^\rho, p_t^{\theta,*} = p_t^\theta$
else if $\mathbf{1}^\top \mathbf{P}_t + p_t^{\theta,l} \geq \underline{G_t}$ (sufficient gen) **then**
  $\mathbf{P}_t^* = \mathbf{P}_t$, $p_t^\mu = \mathbf{1}^\top \mathbf{P}_t^* + p_t^{\theta,l} - \underline{G_t}$
  $p_t^{\omega,*} = p_t^\omega, p_t^{\rho,*} = p_t^\rho, p_t^{\theta,*} = p_t^\theta$
else if $p_t^{\theta,g} \leq \mathbf{1}^\top \mathbf{P}_t + p_t^{\theta,l} < \underline{G_t}$ (excessive gen) **then**
  Curtail renewable generation by
  $\chi^* = \text{argmin}_{\chi\in[0,1)} |\chi p_t^\rho + \chi p_t^\omega + p_t^{\theta,g} - \mathbf{1}^\top \mathbf{P}_t|$, and let
  $p_t^\mu = 0, p_t^{\theta,*} = p_t^\theta, p_t^{\rho,*} = \chi^* p_t^\rho, p_t^{\omega,*} = \chi^* p_t^\omega$
else
  $p_t^\mu = 0, p_t^{\theta,*} = \mathbf{1}^\top \mathbf{P}_t, p_t^{\rho,*} = 0, p_t^\omega = 0$
end if
return $\mathbf{P}_t^*$, $p_t^{\rho,*}$, $p_t^{\omega,*}$, $p_t^{\theta,*}$, p_t^μ

C. Curriculum Learning Necessity

Controller training is conducted on the NREL high-performance computing (HPC) system. We choose the evolution strategy RL (ES-RL) algorithm [11] for Stage I and then use proximal policy optimization (PPO) [12] for Stage II, with a similar rationale as the two-stage policy search proposed in [5]. The policy network used in both stages has hidden layers of [256, 256, 128, 128, 64, 64, 38]. The learning progress is shown in Fig. 3, it can be seen that by using CL (orange and green curves), the average episodic reward of the converged control policy is higher than that of the direct learning (blue curve), which get trapped in a local optimum, even with an appropriate learning rate. The drop of the reward between CL learning stages is due to the required exploratory actions after warm-start, but by the end of the Stage II learning, a higher average reward is achieved, showing a load restoration strategy better than the greedy restoration is obtained.

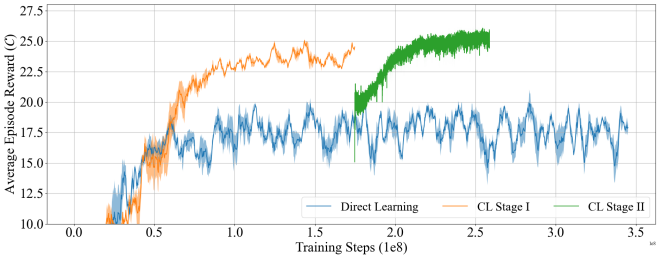


Fig. 3. Learning curves showing average episodic reward C (scaled by 10^{-3}) v.s. training steps. Direct learning and CL Stage I training utilize ES-RL on 10 HPC nodes for two and one hour(s), respectively (direct learning is trained for a longer period to show its convergence). CL Stage II training uses PPO on one HPC node for 20 hours. Each curve is aggregated from three trials.

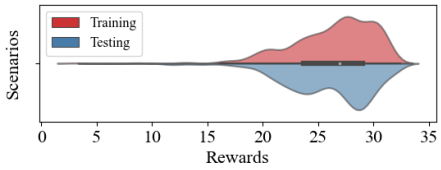


Fig. 4. Performance comparison on both training and testing scenarios (480 uniformly sampled scenarios for each group/month) using Controller II.

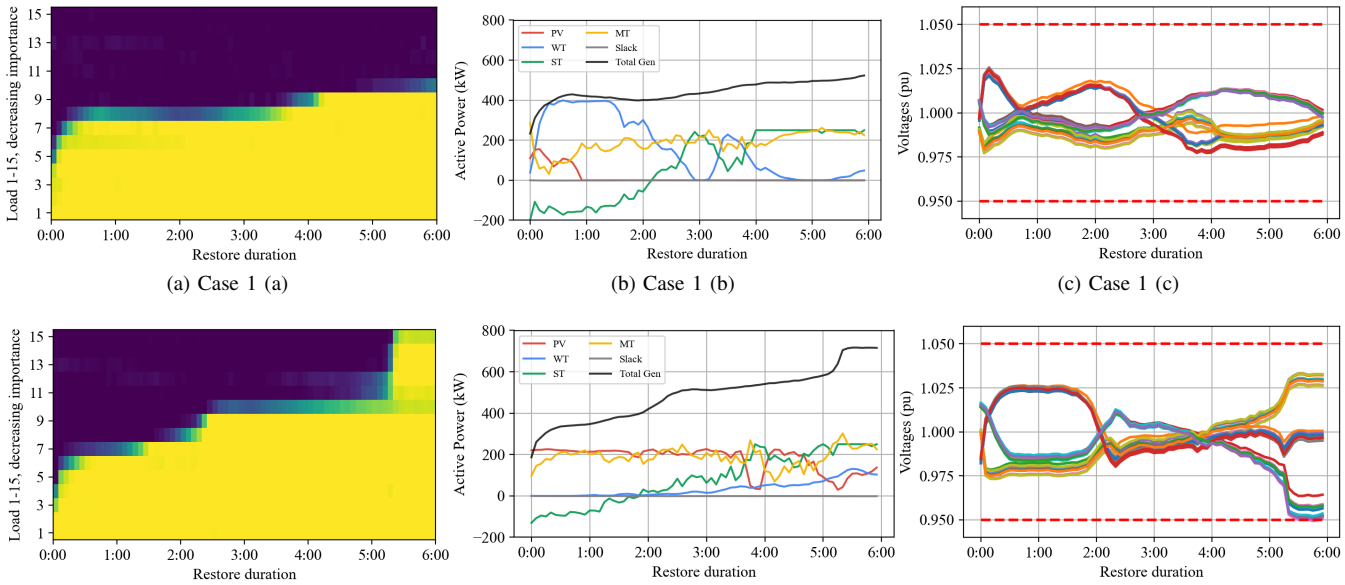
D. Controller Efficacy

To examine the trained controller, we first compare its performance under training scenarios and unseen test scenarios. As shown in Fig. 4, the similarity between reward distributions for both scenario groups indicates RL controller's performance on unseen scenarios does not deteriorate, mainly because the unseen scenarios have similar distribution of renewable generation with the training scenarios as they are from two adjacent months. In practice, this means an RL controller can be trained using recent historical renewable generation data and use it in the near future if CLR event occurs.

To further study the behavior of the learned controller, two specific test scenarios are studied in depth as shown in Fig. 5.

Case 1: Though renewable generation is abundant in the first two hours, the controller doesn't rashly restore as many loads as possible, but choose to charge the battery to hedge against the renewable uncertainty. As a result, when p_t^ω drastically decreases later, almost no load is shed (though small penalty incurs when restoring Load 8 is aborted considering diminishing p_t^ω), providing a reliable and close to monotonic load restoration. Understandably, due to the limited resources in this scenario, not all critical loads are restored.

Case 2: We discovered that when Load 10 is still partially restored, lower priority loads are already restored (see Case 2(a) in Fig. 5). Such control behavior is due to the concern



(a) Case 1 (a) (b) Case 1 (b) (c) Case 1 (c)

(d) Case 2 (a) (e) Case 2 (b) (f) Case 2 (c)

Fig. 5. CLR case study for two testing scenarios (unseen during RL controller training). First column sub-figures show the load pick-up process: brighter color means higher percentage of a load is restored (Yellow: 100%, black: 0%). Second column sub-figures illustrate generation profiles during restoration. Last column sub-figures shows voltage profiles for all buses.

of voltage violation: as shown in Case 2(c) in Fig. 5, several buses in Phase C (the most heavily loaded phase) have voltage close to the lower bound already. Fully restore Load 10, the least important load on Phase C, will cause voltage violation in six buses, as we tested in retrospect.

In both examples, the remaining energy in $\mathcal{D}$ are minimum at the final step and no renewable curtailment is observed. This shows the trained RL controller has successfully learned to fully take advantage of the renewable DERs but not being influenced by their intermittency. Finally, we compare two RL controllers trained via CL and direct learning in testing scenarios: average rewards (C) are 25.90 and 17.66 respectively. This demonstrates CL can facilitate the RL policy search in a complicated environment and eventually learn a better-performing controller for the CLR problem.

V. Conclusion

In this paper, we show that compared with direct RL, CL enables RL to learn a better policy for the CLR problem. The trained controller, upon examination, demonstrates proper behavior for optimal system restoration even for unseen scenarios with renewable uncertainty. These results are expected to show RL's capability for solving grid control problems using state-of-the-art algorithms, techniques and computing platforms, allowing future research to study the practical feasibility of using an RL controller for fast online optimal control. In future work, we will thoroughly compare the proposed RL controller and an up-to-date stochastic optimization-based controller.

References

[1] Ying Wang, Yin Xu, Jinghan He, Chen-Ching Liu, Kevin P Schneider, Mingguo Hong, and Dan T Ton. Coordinating multiple sources for service restoration to enhance resilience of distribution systems. *IEEE Transactions on Smart Grid*, 10(5):5781–5793, 2019.

[2] Weijia Liu, Fei Ding, Kumar Utkarsh, and Shuva Paul. Post-Disturbance Dynamic Distribution System Restoration with DGs and Mobile Resources: Preprint. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020. Available: https://www.nrel.gov/docs/fy20osti/75372.pdf.

[3] Zhiwen Wang, Chen Shen, Yin Xu, Feng Liu, Xiangyu Wu, and Chen-Ching Liu. Risk-limiting load restoration for resilience enhancement with intermittent energy resources. *IEEE Transactions on Smart Grid*, 10(3):2507–2522, 2019.

[4] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[5] Xiangyu Zhang, Rohit Chintala, Andrey Bernstein, Peter Graf, and Xin Jin. Grid-interactive multi-zone building control using reinforcement learning with global-local policy search. *arXiv preprint arXiv:2010.06718*, 2020.

[6] Hanchen Xu, Hongbo Sun, Daniel Nikovski, Shoichi Kitamura, Kazuyuki Mori, and Hiroyuki Hashimoto. Deep reinforcement learning for joint bidding and pricing of load serving entity. *IEEE Transactions on Smart Grid*, 10(6):6366–6375, 2019.

[7] Xiangyu Zhang, Abinet Tesfaye Eseye, Bernard Knueven, and Wesley Jones. Restoring Distribution System Under Renewable Uncertainty Using Reinforcement Learning: Preprint. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020. Available: https://www.nrel.gov/docs/fy21osti/77116.pdf.

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[9] Roger C Dugan and D Montenegro. The open distribution system simulator (OpenDSS): Reference guide. *Electric Power Research Institute (EPRI)*, 2018.

[10] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[11] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.