PowerGym: A Reinforcement Learning Environment for Volt-Var Control in Power Distribution Systems

Ting-Han Fan[1,3] TINGHANF@PRINCETON.EDU
Xian Yeow Lee[2,3] XYLEE@IASTATE.EDU
Yubo Wang[3] YUBO.WANG@SIEMENS.COM
[1]*Princeton University,* [2]*Iowa State University,* [3]*Siemens Technology*

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

Reinforcement learning for power distribution systems has so far been studied using customized environments due to the proprietary nature of the power industry. To encourage researchers to benchmark reinforcement learning algorithms, we introduce PowerGym, an open-source reinforcement learning environment for Volt-Var control in power distribution systems. Following OpenAI Gym APIs, PowerGym targets minimizing power losses and voltage violations under physical networked constraints. PowerGym provides four distribution systems (13Bus, 34Bus, 123Bus, and 8500Node) based on IEEE benchmark systems and design variants for various control difficulties. To foster generalization, PowerGym offers a detailed customization guide for users working with their distribution systems. As a demonstration, we examine state-of-the-art reinforcement learning algorithms in PowerGym and validate the environment by studying controller behaviors. The repository is available at https://github.com/siemens/powergym.

Keywords: Reinforcement learning for physical systems, Benchmark environments, Power distribution systems, Volt-Var control

1. Introduction

Volt-Var control refers to the control of voltage (Volt) and reactive power (Var) in power distribution systems to achieve healthy operation of the systems. By optimally dispatching voltage regulators, switchable capacitors, and controllable batteries, Volt-Var control helps to flatten voltage profiles and reduce power losses across the power distribution systems. It is hence rated as the most desired function for power distribution systems (Borozan et al., 2001).

The core of the Volt-Var control problem is an optimization for voltage profiles and power losses governed by networked constraints. Represent a power distribution system as a tree graph $(\mathcal{N}, \xi)$, where $\mathcal{N}$ is the set of nodes or *buses* and ξ is the set of edges or lines and transformers. Denote node i as j's parent. The physical networked constraints are given in Eq. (1) (Farivar et al., 2013). p, q are active and reactive power consumed at nodes or edges, v, ℓ denote bus voltage magnitude and squared current magnitude, R, X are resistance and reactance. Capital letters stand for given parameters otherwise variables. The constraints in Eq. (1) have quadratic equalities, making any optimization upon it nonconvex. Researchers have either tightly relaxed the constraints with strict nodal injection assumptions (Gan et al., 2014) or used linearization that assumes the distribution systems operates at a fixed operating point (Yang et al., 2016). Both methods require tremendous efforts in trimming and conversion of a model that is readily available in commercial circuit simula-

tion software to specific optimization formulations. Together with many integer decision variables in controllable devices not shown above, the Volt-Var control problem becomes extremely hard to scale to a system with thousands of buses, a typical size for distribution systems.

$$\begin{aligned} p_j &= p_{ij} - R_{ij}\ell_{ij} - \sum_{(j,k)\in\xi} p_{jk} \\ q_j &= q_{ij} - X_{ij}\ell_{ij} - \sum_{(j,k)\in\xi} q_{jk} \\ v_j^2 &= v_i^2 - 2(R_{ij}p_{ij} + X_{ij}q_{ij}) + (R_{ij}^2 + X_{ij}^2)\ell_{ij} \\ \ell_{ij} &= (p_{ij}^2 + q_{ij}^2)/v_i^2, \end{aligned} \tag{1}$$

With recent breakthroughs in deep reinforcement learning (RL), power system researchers have attempted to use RL for power system operations. One such example is learning to operate a transmission systems operation in the L2RPN competition (Marot et al., 2021). Though transmission systems are fundamentally different from distribution systems in both network topology (looped vs. radial) and typical problem types (dynamic vs. quasi-static), RL has shown promising results (Yoon et al., 2020) in operating transmission systems. While there are existing works on RL for distribution systems, researchers have used their own custom training environments. One reason is due to the regulatory and conservative nature of the power engineering industry: being safety-critical, the real-life distribution system topologies and control settings are proprietary. To encourage power systems researchers to make fair comparisons on the developed RL algorithms without having the concern of proprietary information leakage, we have developed PowerGym, a Gym-like environment (Brockman et al., 2016) for optimizing Volt-Var control using IEEE benchmark test systems (PES, 2010; Dugan and Arritt, 2010). It further serves as a base for power systems engineers to implement RL algorithms on their proprietary systems with minor customization.

PowerGym supports Gym-like functionalities such as reset, step, random action sampling, and visualization; hence it is readily applicable with off-the-shelf RL algorithms. On top of the Gym design, PowerGym provides a wide range of environment variations of the IEEE benchmark systems. These variations affect the environment's physical constraints and ultimately the difficulty of the control problem, allowing users to choose an environment that is easier to control but more abstracted or harder to control yet more realistic. Details can be found in Fan et al. (2021).

Our contributions are as follows. We introduce PowerGym as a unified benchmark to help power system researchers evaluate their controls and RL algorithms. To the best of our knowledge, there are no existing state-of-the-art or similar benchmarks. This is the first publicly accessible environment focusing on Volt-Var control in power distribution systems. Second, our implementation philosophy strongly encourages the usability and extendibility of PowerGym. We provide variations of IEEE benchmark systems to enable different control difficulties, along with a detailed customization guide. Additionally, we showcase the applicability of PowerGym on two popular RL algorithms, PPO (Schulman et al., 2017) and SAC (Haarnoja et al., 2018) for validation purposes. We also explain the effect of the horizon and the controllers' physical behaviors via a case study.

2. Related Work

The application of RL to control and manage various aspects of power systems is a well-studied topic in literature (Zhang et al., 2020). There has been renewed interest in this topic due to algo-

rithmic advancements, allowing RL to go beyond tabular settings and scale to large state and action spaces using neural networks as expressive function approximators. Examples of such work include home/building energy control (Sun et al., 2020; Pigott et al., 2021), power systems stability control (Ernst et al., 2004), microgrid control (Henri et al., 2020), and load frequency control for renewable energy (Yan and Xu, 2019). In the context of Volt-Var control, various studies leverage RL to optimize various aspects of the problem, such as emphasizing the constraint satisfaction (Wang et al., 2019, 2020) or the sample efficiency and scalability (Zhang et al., 2021). Nevertheless, most of these results are based on non-standardized implementations of various systems with the environment tuned to the specifics of the problem. This has led to the difficulty of evaluating and comparing results and remains a crucial challenge in these areas, as highlighted by Gao and Yu (2021).

In other domains of deep RL, researchers recognized the importance of having high-quality benchmark environments to facilitate the research into RL application. This has led to the development of environments such as OpenAI Gym for training RL agents to play a variety of games and for robotic control (Brockman et al., 2016), Safety Gym for safe RL exploration (Ray et al., 2019) and ns3-gym for training RL in networks research problems (Gawłowicz and Zubow, 2018). More closely related benchmarks include the Grid2Op (Donnot, 2020) and Gym-ANM (Henry and Ernst, 2021). Grid2Op is the platform used for the Learning To Run Power Network (L2PRN) challenge, controlling unpredictable power generation from renewable energy sources through power systems' topology change (Marot et al., 2020). Gym-ANM is an RL environment that models Active Network Management (ANM) for renewable energy. Nevertheless, a standardized benchmark for Volt-Var control environments with the flexibility of instantiating systems of various sizes and difficulty is still lacking. To this end, we hope that PowerGym serves to fill the gap in the community as a unified benchmark environment for RL research in Volt-Var control.

3. Reinforcement Learning Preliminaries

A reinforcement learning (RL) environment is often modeled by the Markov Decision Process (MDP) with two common MDPs: infinite-horizon discounted MDP $M_d = \langle \mathcal{S}, \mathcal{A}, T, r, \gamma \rangle$ and finite-horizon episodic MDP $M_e = \langle \mathcal{S}, \mathcal{A}, \{T_i\}, \{r_i\}, H \rangle$. $\mathcal{S}$, $\mathcal{A}$ are the state and action spaces. $T/\{T_i\}$, $r/\{r_i\}$ are the stationary/non-stationary state transition function and reward function. $\gamma \in (0, 1)$, $H \in \mathbb{N}$ are the discount factor and the horizon. We assume a stationary state transition in M_e: $T_i = T$ for $i \in [0...H-1]$. The goal of RL is to find a policy to maximize the cumulative rewards:

$$
\begin{aligned}
R_{M_d}(\pi) &= \mathbb{E}\Big[\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i, s_{i+1}) \Big| a_i \sim \pi(\cdot|s_i) \Big] \\
R_{M_e}(\{\pi_i\}) &= \mathbb{E}\Big[\sum_{i=0}^{H-1} r_i(s_i, a_i, s_{i+1}) \Big| a_i \sim \pi_i(\cdot|s_i) \Big]
\end{aligned}
\tag{2}
$$

The optimal policy of M_d is stationary while that of M_e is non-stationary (Agarwal et al., 2019)[Chapter 1]; hence in Eq. (2), the policy is denoted as π in M_d and as $\{\pi_i\}$ in M_e. To reduce the model complexity, most RL experiments are formulated into M_d if the stationarity holds. However, M_e is inevitable when the reward is non-stationary. Depending on the applications, we implement both stationary and non-stationary rewards, which will be discussed in the next section.

4. A Volt-Var Control Environment

4.1. Power Distribution Systems and Objectives

Power distribution systems are networks for delivering electric power from the power transmission system to end consumers. Due to the distribution loss, voltage drops along the power delivery line, possibly causing voltage violations and power losses. Thus, Volt-Var optimization is required. In power distribution systems, the Volt-Var optimization problem is to control devices (e.g., regulators, capacitors, and batteries, represented as x.) under constraints. x affects the voltage, resistance, reactance, and power in the physical networked constraints, so Eq. (1) is a constraint of Eq. (3).

$$\begin{array}{ll} \min_x & f_{\text{volt}}(x) + f_{\text{ctrl}}(x) + f_{\text{power}}(x) \\ \text{s.t.} & \text{Eq. (1) and device constraints.} \end{array} \tag{3}$$

The Volt-Var optimization's objective is a combination of three losses: f_{volt} for voltage violations, f_{ctrl} for control errors, and f_{power} for power losses. The device constraints ensures the devices operates within its physical limits. While Eq. (3) only accounts for a single time step, in practice we solve it at every time step. Solving a sequence of Volt-Var optimization, Eq. (3), becomes a Volt-Var control problem. In short, we call a problem *Volt-Var optimization* if solving a single Eq. (3) and *Volt-Var control* if solving a sequence of Eq. (3) connected by device constraints over time. We use a Python version of OpenDSS to solve for the physical networked constraints of Eq. (3). OpenDSS is an open-source power flow solver developed by EPRI. It takes p, q from Eq. (1) as known variables and solves the nonlinear equations of voltages and currents using fixed-point iterations.

Shifting the focus to elements in power distribution systems, we define each element to be multi-phase following the fact that power is usually delivered in multi-phases. As shown in Figure 1, a (multi-phase) node, or a bus, can be a pure connection point or include node objects like loads, capacitors, or batteries. A (multi-phase) edge is formed by a line, a transformer, or a regulator. Loads model the power consumption from the consumers. Capacitors provide reactive power and batteries are energy (active power) storage. Lines imitate the connection from one (multi-phase) node to another subject to Ohm's law. Transformers and regulators are used for voltage adjustment from one node to another. We highlight that these elements included in PowerGym are designed to mirror the most common elements found in OpenDSS, with the intention of facilitating an easy adoption and common understanding among the power systems community.

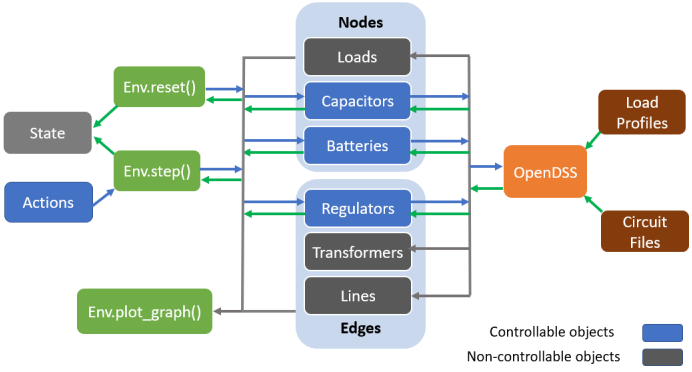

Figure 1: A depiction of PowerGym components and data flow. OpenDSS reads the circuit files and load profiles (discuss in environment registration) to initialize the simulation. The environment functions interact with OpenDSS to get the state transitions.

4.2. Volt-Var Control as An RL Problem

In this subsection, we describe the Volt-Var control problem in the language of RL. As an example, we consider a finite-horizon MDP with $H = 24$ steps as the horizon representing a daily control

Variable	Type	Range
Bus voltage	cont.	$[0.8, 1.2]$
Capacitor status	disc.	$\{0, 1\}$
Regulator taps	disc.	$\{0, ..., N_{\text{reg\_act}} - 1\}$
State-of-charge (soc)	cont.	$[0, 1]$
Discharge power	cont.	$[-1, 1]$

Table 1: Observation Space. Here, both discrete and continuous batteries give the normalized discharge power. $N_{\text{reg\_act}}$ is the number of taps of a regulator (default value of 33 taps, following the convention of OpenDSS.

Variable	Type	Range
Capacitor status	disc.	$\{0, 1\}$
Regulator taps	disc.	$\{0, ..., N_{\text{reg\_act}} - 1\}$
Discharge power	disc.	$\{0, ..., N_{\text{bat\_act}} - 1\}$
Discharge power	cont.	$[-1, 1]$

Table 2: Action Space. The discrete battery uses the discretized discharge power while the continuous battery uses the normalized one. $N_{\text{bat\_act}}$ is the number of a discrete battery's discharge power.

with the control frequency being one action per hour. Nevertheless, the horizon remains a customizable variable in our implementation is further discussed in subsection 5.2 and Fan et al. (2021). The following paragraphs give the details about the observation space, the action space, the state transition, and the reward function of PowerGym.

4.2.1. OBSERVATION AND ACTION SPACES

The observation and action spaces, as summarized in Tables 1 and 2, are products of discrete and continuous variables. The discrete variables are from the physical constraints of the controllers; for example, a capacitor either turns on or off, a regulator operates on a finite number of modes (tap number), and a discrete battery only has a finite number of discharge powers. The continuous variables are normalized into some bounded ranges; for example, the (per-unit) voltage is represented into the unit of the base voltage on a bus, and hence usually bounded in [0.8, 1.2]. The battery's state-of-charge (soc) defined as (charge / max charge) is in [0.0, 1.0]. The continuous battery's normalized discharge power (discharge power / max discharge power) is in [-1.0, 1.0], where negative values denote charging and the positive values mean discharging.

Depending on the device constraints and specific problem formulation, a battery control can be either discrete or continuous. This affects the action space (Table 2) since the action representations are different. However, since we can always post-process the observation after receiving it, we unify the representation of discrete and continuous batteries by mapping the discrete battery's discharge power to the normalized form in the observation space (Table 1). We left the final representation (discrete or continuous) of the battery as part of the design of the problem formulation, which enables greater flexibility for researchers to customize the environment. This will result in an action space that is either multi-discrete or a product of multi-discrete and continuous spaces. Either way, the problem becomes non-trivial for a tabular policy or a policy that encodes the actions as one-hot vectors (e.g., DDQN (Van Hasselt et al., 2016)) because the size of the discrete part of action space scales exponentially in the number of controllers. Further, the possibility of mixing discrete and continuous actions presents an additional challenge to design the architecture of the policy. All things considered, the representations of the observation and action spaces for PowerGym discussed above is designed to reflect a realistic system as close as possible while being flexible enough for customization. Technically, it comes with some non-trivial aspects (mixed actions space) to foster research into developing scalable RL controllers for power distributions systems. Nonetheless, the range of observation and action space can also be masked based on the researcher's use case.

4.2.2. STATE TRANSITION

We now describe the state transition function $s' = T(s, a)$ in PowerGym. With the descriptions in Table 1 and 2, s' is represented as

$$s' = [\text{Vols}(s,a),\ \text{cap}(a),\ \text{reg}(a),\ \text{soc}(s,a),\ \text{dis}(s,a)]. \tag{4}$$

$\text{Vols}(s,a)$ is the next set of voltages and depends on action a and the stochasticity of loads, which we model using the load profiles (discussed in environment design). $\text{cap}(a)$ and $\text{reg}(a)$ are the next statuses of capacitors and regulators, while $\text{soc}(s,a)$ and $\text{dis}(s,a)$ are the next soc's and discharge powers of batteries. Both of $\text{soc}(s,a)$ and $\text{dis}(s,a)$ depend on the current state s because a battery's soc cannot go beyond full charge (soc $= 1$) or depleted (soc $= 0$). To enforce this, we project any *attempted* discharge power to the allowed range based on s, making $\text{dis}(s,a)$ a function of (s,a).

4.2.3. REWARD FUNCTION

We implement the objective of a Volt-Var problem, Eq. (3), into a reward function as follows:

$$r(s,s',i) = -f_{\text{volt}}(s') - f_{\text{ctrl}}(s,s',i) - f_{\text{power}}(s') \tag{5}$$

$$f_{\text{power}}(s') = \text{w}_{\text{power}} \frac{\text{PowerLoss}(s')}{\text{TotalPower}(s')} \tag{6}$$

s is a concatenation of all observations in the current step, s' denotes the next observation, and $i \in [0, ..., H-1]$ is the episode step. The dependency on step i implies the reward could be non-stationary. The power loss term, expanded in Eq. (6), is a ratio of the overall power loss to the total power. The voltage violation and control error terms are expressed in Eq. (7) and (8). Eq. (5) is expressed as $r(s,s',i)$, not $r(s,a,i)$, because the action a is a part of the next state s'. Mathematically, $r(s,a,i)$ and $r(s,s',i)$ are equivalent because $s' = T(s,a)$ is a function of (s,a) under the state transition function T. The voltage violation, Eq. (7), is the sum of worst-case voltage violations among all phases across all buses. The upper/lower violation thresholds ($\overline{V}$/$\underline{V}$) are set as $\pm 5\%$ of the per-unit voltage V due to the US voltage regulation standard (ANSI, 2011).

$$f_{\text{volt}}(s') = \sum_{n \in \mathcal{N}} \Big(\max_{p \in \text{Phases}(n)} V_{n,p}(s') - \overline{V} \Big)_+ + \Big(\underline{V} - \min_{p \in \text{Phases}(n)} V_{n,p}(s') \Big)_+, \tag{7}$$

where $(\cdot)_+$ is a shorthand for $\max(\cdot, 0)$. Thereby, the upper violation $(\max_p V_{n,p} - \overline{V})_+$ is positive when $\max_p V_{n,p} > \overline{V}$ and zero otherwise. The control error, Eq. (8), is a sum of capacitors' and regulators' switching penalties (1st row) and batteries' discharge penalty and soc penalty (2nd row). These penalties discourage the policy from making frequent changes and slow the wear out of the control devices. Note the discharge error $\frac{\text{P}_b(s')_+}{\overline{\text{P}_b}}$, with $\overline{\text{P}_b}$ being the max power, has a $(\cdot)_+$ function as the battery degradation is primarily caused by the battery discharging power $\text{P}_b > 0$. Besides, the soc penalty has an indicator of the last time step $\mathbb{I}_{i=H}$ to encourage a battery b to return to its initial state-of-charge soc_{0b}. Hence, the reward is stationary if $\text{w}_{\text{soc}} = 0$ and non-stationary otherwise.

$$\begin{aligned} f_{\text{ctrl}}(s,s',i) = &\sum_{c \in \text{caps}} \text{w}_{\text{cap}} |\text{Status}_c(s) - \text{Status}_c(s')| + \sum_{r \in \text{regs}} \text{w}_{\text{reg}} |\text{Tap}_r(s) - \text{Tap}_r(s')| \\ &+ \sum_{b \in \text{bats}} \text{w}_{\text{dis}} \frac{\text{P}_b(s')_+}{\overline{\text{P}_b}} + \text{w}_{\text{soc}} \mathbb{I}_{i=H} |\text{soc}_b(s') - \text{soc}_{0b}|, \end{aligned} \tag{8}$$

where $c,\ r,\ b$ represent a capacitor, a regulator, and a battery. Status_c, Tap_r, P_b, soc_b are status of c, tap number of r, discharge power of b, and soc of b. In summary, the composite reward function defined above serves as one possible template for Volt-var control that optimizes the system from multiples aspects. Our results (shown below) have demonstrated that using SoTA RL algorithms, a controller can be trained to yield decent performance. Nevertheless, this reward function remains a modular part of the PowerGym environments and can be easily modified.

5. Design of PowerGym

This section is a brief overview of the designs of PowerGym. To showcase the generality of the design, we mainly discuss the environment registration and customization. Further details on default environments, gym-like usages (e.g., reset, step), and load profiles can be found in Fan et al. (2021).

5.1. Environment Instantiation

Similar to the OpenAI Gym, PowerGym provides the function `make_env()`:

```
make_env(env_name, worker_idx=None)
```

to instantiate an environment, where `env_name` is the name of the registered environment. The argument `worker_idx` (if not `None`) is used for parallel execution, which we detail in the subsection of load profiles. The function `make_env()` reads the following information. First, PowerGym reads DSS-based circuit files into the environment class and uses OpenDSS in the backend to compile the circuit files, as shown in Figure 1. To define the hyper-parameters that affects the RL training under the same system, PowerGym requires information such as the horizon, number of actions of a regulator/battery and customizable weights of the power loss, capacitor's switch loss, regulator's switch loss, battery's discharge loss, battery's state-of-charge (soc) loss in the reward function. The next subsection introduces the customization of such information.

5.2. Environment Registration and Customization

Our implementation allows users to customize the environment by registering a new environment name associated with the required information. This can be achieved by appending the information to the dictionaries within PowerGym's register. An example of defining a new 13-bus environment in the dictionary in shown below. The `dss_file` entry is the main circuit file that OpenDSS compiles. Users may edit `dss_file` to change the underlying circuit's objects (nodes and edges shown in Figure 1 and structure (topology of the power system) to customize the environment from a domain point of view. Users can also modify the hyper-parameters related to the formulation of the RL problem. In the example shown, `max_episode_steps` is the horizon (24 steps by default for daily control, `act_num` is the shorthand of the number of actions, where the battery is continuous if `bat_act_num` is infinity and discrete if finite. The remaining parameters are the weights in the reward function shown in Eq. (5).

```
'13Bus': {
  'system_name': '13Bus',
  'dss_file': 'IEEE13Nodeckt_daily.dss',
  'max_episode_steps': 24,
  'reg_act_num': 33,
  'bat_act_num': 33,
  'power_w': 10.0,
  'cap_w': 1.0/33,
  'reg_w': 1.0/33,
```

```
'soc_w': 0.0/33,
'dis_w': 6.0/33 }
```

Besides the information shown above, PowerGym depends on the load profiles (details in Fan et al. (2021)) and additional auxiliary circuit files. These files are also customizable and can be found in the folder of `systems/{system_name}` of the repository. For example, the above shows users can find customizable files in the folder `systems/13Bus`. For more detailed information on customizing these files, we refer readers to the code's repository.

6. Experimental Validations

6.1. Cumulative Rewards in Default Environments

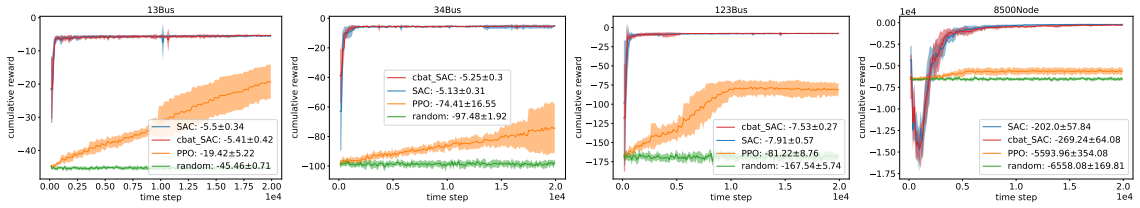

Figure 2: Cumulative rewards on test load profiles for different agents in 40k steps. The labels denote the average and standard deviation of the final rewards.

To demonstrate the applicability of PowerGym, we trained two popular RL algorithms as initial benchmarks on our environments: Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Soft Actor-Critic (SAC) (Haarnoja et al., 2018), with the implementations of PPO based on Fujita et al. (2021) and SAC on Fan and Wang (2021). Since PPO is on-policy while SAC is off-policy, these two algorithms give us a proxy of the expected performance of on-policy versus off-policy algorithms in the environments. For fair comparison, both PPO and SAC have been trained on multi-discrete actions versions of the environment. In addition, SAC has been trained on environments with continuous batteries (cbat) to compare the effect of different battery settings on the difficulty of the environment. The experiments run on a server with one AMD Ryzen Threadripper 3970X CPU and one Nvidia RTX 3090 GPU. The experiments have been designed as follows: The load profiles are randomly partitioned into two halves, one for training and the other for testing. During training, the policy is tested on test load profiles every five episodes; or equivalently every 120 steps as the horizon is 24. Lastly, all experiments are performed across ten random seeds.

In Figure 2, the label "random" denotes an untrained policy that samples actions uniformly from the action space. As expected, SAC converges faster and outperforms PPO across all environments, which aligns with the SAC paper (Haarnoja et al., 2018) that has been demonstrated on the MuJoCo (Todorov et al., 2012). Due to the experiment design (evaluation at every 120 steps), all curves start at step 120 instead of step 0. The first evaluation (step 120) reveals the algorithms' performance based on the first few updates: PPO is similar to random policy while SAC isn't. The fact that PPO's initial performance is close to the random policy validates the clipping nature of its policy gradient. Clipping it's loss function makes PPO updates it's weights conservatively and hence it behaves similar to random policy in the early steps. As for SAC, because its DDPG-style policy gradient (Silver et al., 2014) isn't clipped, SAC suffers more from the initial inaccuracy of Q-function and hence deviates from the random policy in the early steps. Finally, the performances of `SAC` and

cbat_SAC are very similar, implying discrete and continuous batteries share similar behaviors, and SAC successfully adapts to both. To sum up, we demonstrated the applicability of two popular RL algorithms in PowerGym for four different IEEE power distribution systems. In our experiments above, we observed that both PPO and SAC demonstrate learning capabilities when compared to a random policy, with SAC being outperforming PPO and being significantly more sample efficient. We highlight that these results serves to validate the correctness of the PowerGym environment as a benchmark as well as an initial baseline for future research, rather than comparing the applications of existing algorithms.

6.2. Case Study: 123Bus with Different Variations

We take the 123Bus system (Figure 4) as an example to further analyze the behavior of the control policy in PowerGym. Specifically, we focus on the continuous battery scenario because batteries may arbitrarily discharge/charge within an allowable range in practice. For this case study, we consider four variations: vanilla (cbat), scaled loads (cbat_s2.5), with soc penalty (soc), and scaled loads with soc penalty (cbat_soc_s2.5). Scaling the loads of the system reflects a larger power consumption, thus increases the difficulty of controlling the system while a soc penalty introduces a non-stationary reward. As such, we would like to see how the control policy adapts to different scenarios. The first row of Figure 3 visualizes the average switching errors of capacitors and regulators respectively for all four scenarios. Both errors are small in most time steps across all scenarios. Hence, the policies for both capacitors and regulators only make large changes when needed while making small adjustments the rest of the time. Note the behavior of the initial steps and the later steps are different because the RL exploration starts after a 1000 initial random steps.

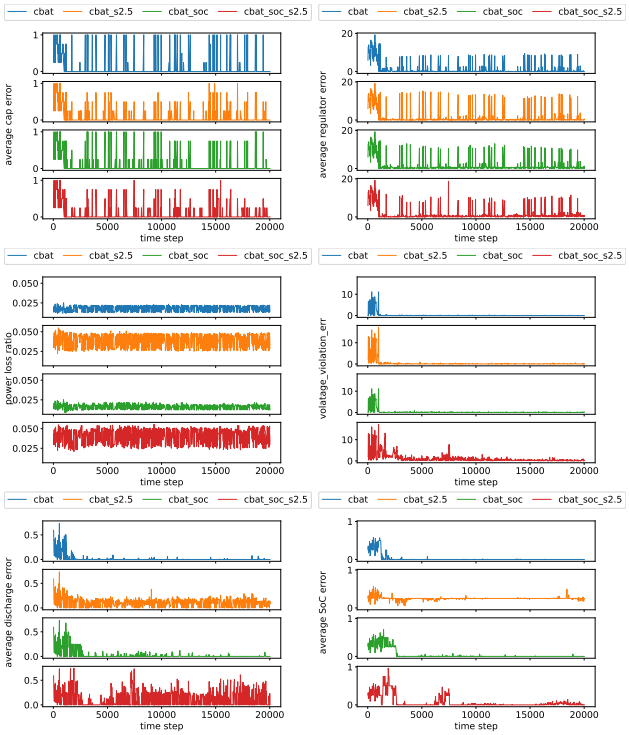

Figure 3: Immediate errors on 4 variations of 123Bus_cbat.

The second row of Figure 3 shows the power loss ratio and voltage violations. Referring to the power loss, we observed that the loss barely improves over time, since it is a challenging objective to optimize. Nevertheless, the power losses on the 2.5-scaled environments are higher than their un-scaled counterparts. This is because large voltage violations cause large voltage differences on the lines, which brings up the power loss on the lines. Additionally, due to the load scaling, the 2.5-scaled environments have the higher voltage violations than the unscaled environments (cbat_s2.5 > cbat and cbat_soc_s2.5 > cbat_soc). Furthermore, the voltage violation of soc_s2.5 is greater than that of s2.5 as the soc penalty makes the policy on batteries more restrictive and non-stationary.

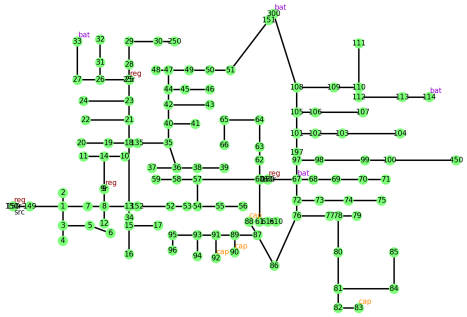

Figure 4: Layout of 123Bus system

Finally, the third row of Figure 3 shows the battery activity in discharge and soc errors. Since the battery is an energy storage device, it is useful when the environment lacks power and has high voltage violations. Hence, the battery barely discharges in the un-scaled environments and maintains mostly zero soc error. As for the scaled environments (`s2.5` & `soc_s2.5`), because `s2.5` discharges frequently, it has smaller voltage violations but higher a soc error. In comparison, `soc_s2.5` discharges less, has a higher voltage violation, but also incurs a smaller soc error. Therefore, there is a trade-off between battery activity and voltage violation in heavily-loaded environments: the more battery activity, the less voltage violation, and the RL controller need to find a delicate balance between the two. In summary, the soc penalty and the load scale affect the difficulty of a PowerGym environment, and this difficulty can be evaluated by power losses, voltage violations, and battery activities. Higher power losses, voltage violations, and battery activities can generally be expected from a harder environment.

6.3. Effects of Varying Horizons

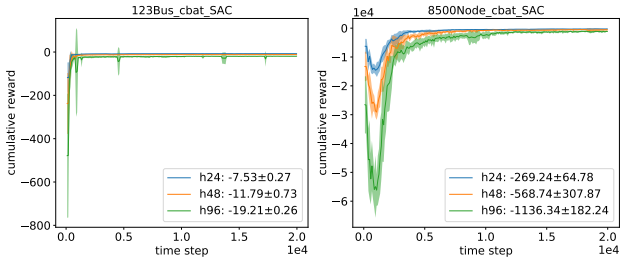

Figure 5: Effects of horizon in 20k steps. h24, h48, h96 denotes horizon = 24, 48, 96 respectively.

Figure 5 shows the cumulative testing reward w.r.t. the horizon for 123Bus and 8500Node systems. We only analyze the continuous battery scenario with the SAC algorithm as this is the setting with the best performance according to Figure 2. As the cumulative reward scales linearly w.r.t. the horizon, h48's cumulative reward is roughly twice of h24's and h96's is four times of h24's. Also, the convergence speeds w.r.t. horizons are similar in 123Bus due to the fact that the 123Bus system is a more stable system and less likely to have voltage violations. On the other hand, 8500Node is less stable, resulting in convergence with longer steps for a longer horizon.

7. Conclusion

We develop a gym-like open-source environment, PowerGym, to facilitate RL research/adaptation for Volt-Var control in power distribution systems. PowerGym encourages power system researchers to make fair comparisons on RL algorithms using the same benchmarking environment. It includes a wide variations (problem size, base voltage violation, load scale, and soc penalty) to study different aspects of the Volt-Var control. PowerGym also acts as a stepping point for researchers/engineers to adopt RL algorithms to power distribution systems in real life and provides a detailed customization guide for researchers/engineers who use PowerGym with their own proprietary power distribution systems. Our RL experiments suggest the correctness of the PowerGym design. The cumulative rewards achieved by our RL agents serve as a baseline for the PowerGym users.

References

Alekh Agarwal, Nan Jiang, and S. Kakade. Reinforcement learning: Theory and algorithms. https://rltheorybook.github.io/, 2019.

ANSI. Electric power systems and equipment – voltage ratings (60 hz) c84.1, 2011. American National Standards Institute (ANSI) Standard: C84.1-2011.

Vesna Borozan, Mesut E Baran, and Damir Novosel. Integrated volt/var control in distribution systems. In *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 01CH37194)*, volume 3, pages 1485–1490. IEEE, 2001.

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

B. Donnot. Grid2op- A testbed platform to model sequential decision making in power systems. . https://GitHub.com/rte-france/grid2op, 2020.

RC Dugan and RF Arritt. The ieee 8500-node test feeder. *Electric Power Research Institute, Palo Alto, CA, USA*, 2010.

D. Ernst, M. Glavic, and L. Wehenkel. Power systems stability control: reinforcement learning framework. *IEEE Transactions on Power Systems*, 19(1):427–435, 2004. doi: 10.1109/TPWRS.2003.821457.

Ting-Han Fan and Yubo Wang. Soft actor-critic with integer actions. *arXiv preprint arXiv:2109.08512*, 2021.

Ting-Han Fan, Xian Yeow Lee, and Yubo Wang. Powergym: A reinforcement learning environment for volt-var control in power distribution systems. *arXiv preprint arXiv:2109.03970*, 2021.

Masoud Farivar, Lijun Chen, and Steven Low. Equilibrium and dynamics of local voltage control in distribution systems. In *52nd IEEE Conference on Decision and Control*, pages 4329–4334, 2013.

Yasuhiro Fujita, Prabhat Nagarajan, Toshiki Kataoka, and Takahiro Ishikawa. Chainerrl: A deep reinforcement learning library. *Journal of Machine Learning Research*, 22(77):1–14, 2021.

Lingwen Gan, Na Li, Ufuk Topcu, and Steven H Low. Exact convex relaxation of optimal power flow in radial networks. *IEEE Transactions on Automatic Control*, 60(1):72–87, 2014.

Yuanqi Gao and Nanpeng Yu. Deep reinforcement learning in power distribution systems: Overview, challenges, and opportunities. In *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2021. doi: 10.1109/ISGT49243.2021.9372283.

Piotr Gawłowicz and Anatolij Zubow. ns3-gym: Extending openai gym for networking research. *arXiv preprint arXiv:1810.03943*, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Gonzague Henri, Tanguy Levent, Avishai Halev, Reda Alami, and Philippe Cordier. pymgrid: An open-source python microgrid simulator for applied artificial intelligence research. *arXiv preprint arXiv:2011.08004*, 2020.

Robin Henry and Damien Ernst. Gym-anm: Reinforcement learning environments for active network management tasks in electricity distribution systems. *Energy and AI*, 5:100092, 2021. ISSN 2666-5468. doi: https://doi.org/10.1016/j.egyai.2021.100092.

Antoine Marot, Benjamin Donnot, Camilo Romero, Balthazar Donon, Marvin Lerousseau, Luca Veyrin-Forrer, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. *Electric Power Systems Research*, 189:106635, 2020. ISSN 0378-7796. doi: https://doi.org/10.1016/j.epsr.2020.106635. URL https://www.sciencedirect.com/science/article/pii/S0378779620304387.

Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aïdan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. Learning to run a power network challenge: a retrospective analysis. *arXiv preprint arXiv:2103.03104*, 2021.

IEEE PES. Ieee pes test feeders. https://site.ieee.org/pes-testfeeders/resources/, 2010. Accessed: 2021-07-28.

Aisling Pigott, Constance Crozier, Kyri Baker, and Zoltan Nagy. Gridlearn: Multiagent reinforcement learning for grid-aware building energy management. *arXiv preprint arXiv:2110.06396*, 2021.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7, 2019.

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

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

Jianwen Sun, Yan Zheng, Jianye Hao, Zhaopeng Meng, and Yang Liu. Continuous multiagent control using collective behavior entropy for large-scale home energy management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 922–929, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Wei Wang, Nanpeng Yu, Jie Shi, and Yuanqi Gao. Volt-var control in power distribution systems with deep reinforcement learning. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7, 2019. doi: 10.1109/SmartGridComm.2019.8909741.

Wei Wang, Nanpeng Yu, Yuanqi Gao, and Jie Shi. Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems. *IEEE Transactions on Smart Grid*, 11(4):3008–3018, 2020. doi: 10.1109/TSG.2019.2962625.

Ziming Yan and Yan Xu. Data-driven load frequency control for stochastic power systems: A deep reinforcement learning method with continuous action search. *IEEE Transactions on Power Systems*, 34(2):1653–1656, 2019. doi: 10.1109/TPWRS.2018.2881359.

Zhifang Yang, Haiwang Zhong, Qing Xia, Anjan Bose, and Chongqing Kang. Optimal power flow based on successive linear approximation of power flow equations. *IET Generation, Transmission & Distribution*, 10(14):3654–3662, 2016.

Deunsol Yoon, Sunghoon Hong, Byung-Jun Lee, and Kee-Eung Kim. Winning the l2rpn challenge: Power grid management via semi-markov afterstate actor-critic. In *International Conference on Learning Representations*, 2020.

Ying Zhang, Xinan Wang, Jianhui Wang, and Yingchen Zhang. Deep reinforcement learning based volt-var optimization in smart distribution systems. *IEEE Transactions on Smart Grid*, 12(1): 361–371, 2021. doi: 10.1109/TSG.2020.3010130.

Zidong Zhang, Dongxia Zhang, and Robert C. Qiu. Deep reinforcement learning for power system applications: An overview. *CSEE Journal of Power and Energy Systems*, 6(1):213–225, 2020. doi: 10.17775/CSEEJPES.2019.00920.