Machine Learning (2024) 113:2675–2700
https://doi.org/10.1007/s10994-023-06422-w

Efficient learning of power grid voltage control strategies via model-based deep reinforcement learning

Ramij Raja Hossain[1] · Tianzhixi Yin[1] · Yan Du[1] · Renke Huang[1] · Jie Tan[2] · Wenhao Yu[2] · Yuan Liu[1] · Qiuhua Huang[1]

Received: 5 April 2023 / Revised: 18 September 2023 / Accepted: 3 October 2023 /
Published online: 6 November 2023

Abstract

This article proposes a model-based deep reinforcement learning (DRL) method to design emergency control strategies for short-term voltage stability problems in power systems. Recent advances show promising results for model-free DRL-based methods in power systems control problems. But in power systems applications, these model-free methods have certain issues related to training time (clock time) and sample efficiency; both are critical for making state-of-the-art DRL algorithms practically applicable. DRL-agent learns an optimal policy via a trial-and-error method while interacting with the real-world environment. It is also desirable to minimize the direct interaction of the DRL agent with the real-world power grid due to its safety-critical nature. Additionally, the state-of-the-art DRL-based policies are mostly trained using a physics-based grid simulator where dynamic simulation is computationally intensive, lowering the training efficiency. We propose a novel model-based DRL framework where a deep neural network (DNN)-based dynamic surrogate model (SM), instead of a real-world power grid or physics-based simulation, is utilized within the policy learning framework, making the process faster and more sample efficient. However, having stable training in model-based DRL is challenging because of the complex system dynamics of large-scale power systems. We addressed these issues by incorporating imitation learning to have a warm start in policy learning, reward-shaping, and multi-step loss in surrogate model training. Finally, we achieved 97.5% reduction in samples and 87.7% reduction in training time for an application to the IEEE 300-bus test system.

Keywords Model-based deep reinforcement learning (MB-DRL) · Surrogate model (SM) · Imitation learning (IL) · Augmented random search (ARS) · Voltage control

Yan Du, Renke Huang, and Qiuhua Huang were at PNNLwhen conducting this study.

Editors: Yuxi Li, Emma Brunskill, Minmin Chen, Omer Gottesman, Lihong Li, Yao Liu, Zongqing Lu, Niranjani Prasad, Zhiwei (Tony) Qin, Csaba Szepesvari, Matthew E. Taylor

Extended author information available on the last page of the article

1 Introduction

The Intergovernmental Panel on Climate Change (IPCC) of the United Nations (UN) has recently indicated that the global greenhouse gas (GHG) emission needs to be halved by 2030 (United Nations, 2023) to keep global warming limited to the target value of +1.5°, necessitating a major transition in global electricity and transportation sector, which are undoubtedly the prominent sources of GHG (CO_2, N_2O, CH_4). To tackle climate change, nations around the world have started transiting towards renewable generations from conventional fossil-fuel-based counterparts, and at the same time, undergoing major policy shifts for massive adoption of electric vehicles (EV) with a goal of net-zero emission and carbon neutrality by 2050 (United States, Europe) (US Department of Energy, 2021; Fetting, 2020), 2060 (China) (International Energy Agency, 2021), 2070 (India) (Birol & Kant, 2022). The increase of the uncertain and intermittent nature of renewable distributed energy resources (DERs), plus newly added loads due to the plug-in EVs, are responsible for the fast and sudden changes in grid operating conditions. An illustration of such conditions can be presented by the well-known net-load "duck curve" (California ISO, 2013). As a result of these rapid transformations of the electric power landscapes, the adoption of new strategies in grid operation and control is becoming significantly important. Generally, system-wide control in power systems can be classified into pre-contingency preventive control and post-contingency emergency control. Predominantly, preventive control design is (a) based on steady-state N-1 contingency studies, (b) includes methods like generation re-dispatch and demand response, and (c) operates at a slower time-scale (in a minute/hour time-scale), whereas post-contingency emergency control design (a) is required to stabilize the system following severe disturbances, (b) operates in faster time-scale (in second time-scale) and (c) includes methods like load shedding, generator tripping, and controlled islanding. Despite past research, fast and efficient designs of emergency control methodologies are still largely open problems owing to the changing operating conditions of power systems, and this paper specifically focuses on the above-mentioned problem with the goal of designing emergency load-shedding control actions to mitigate short-term voltage stability issues (Hatziargyriou et al., 2021) following severe system disturbances.

As discussed, the recent operational challenges due to the proliferation of distributed energy resources (DERs), EVs, and the dynamic nature of loads are making power systems vulnerable to system disturbances. Unless tackled appropriately, severe disturbances can destabilize the system and create large-scale blackouts (Australian Energy Market Operator, 2016). At this juncture, recent studies (Cao et al., 2020; Chen et al., 2022; Glavic, 2019) show that DRL methods can provide more adaptive yet faster solutions compared to the traditional rule-based and model-based methods in power system stability and emergency control applications. DRL-based methods in power systems typically learn optimal policies in a model-free manner (Duan et al., 2019; Huang et al., 2019; Kamruzzaman et al., 2021; Yan & Xu, 2018, 2020), which is an appealing virtue considering the high complexity of power systems. However, owing to direct interaction with the environment during policy learning, model-free methods are not sample efficient (Luo et al., 2022), thus unsuitable for direct application in real-world power grids, where trial-and-error effects are highly costly. In the literature, DRL-based control policies are predominantly trained in a simulated environment instead of directly interacting with the real-world power grid. For first-principle-based dynamic modeling of bulk power systems, large-scale differential algebraic equation

(DAE) models have to be solved, which is computationally intensive. Considering the large amount of explorations required for DRL training, this leads to extensive training and tuning time to obtain a good DRL-based control policy for large-scale power systems. Therefore, the current state-of-the-art model-free DRL algorithms in power systems suffer the major challenges related to (a) sample complexity, and (b) training time.

Model-based DRL (MB-DRL) has been shown to be more sample efficient compared to model-free methods (Wang et al., 2019). Unlike model-free approaches, one type of model-based approaches learns a surrogate (or transition) model of the system dynamics and obtain an optimal policy through interaction with the learned surrogate model. While there are some recent works in applying them in power system steady-state and quasi-steady-state applications (Cao et al., 2022; Kamel et al., 2021; Shuai & He, 2020), there are some grand challenges in applying MB-DRL in power system dynamic stability control such as deriving a sufficiently accurate dynamic model, tackling accumulation of modeling error, and state-action distribution drift commonly seen in large state-action space. To the best of our knowledge, *there is no systematic study of MB-DRL methods in terms of feasibility and applicability for bulk power system dynamic stability control applications*. In this paper, we developed a novel MB-DRL framework for bulk power system voltage stability control that significantly accelerates the training process and improves sample efficiency compared to the state-of-the-art model-free methods. We overcame the challenges mentioned above in training DRL policy with a learned model by introducing (1) multi-step loss in model learning, (2) adaptive model update, (3) robust reward structure, and (4) imitation learning. Our method resulted in 97.5% and 87.7% reductions in sample complexity and training time, respectively. We believe that this is a major boost in making DRL practical for real-world grid dynamic control applications.

1.1 Related works

Learning a policy from scratch without prior knowledge of the underlying process is challenging. Model-based RL (MB-RL), or MB-DRL, is a suitable alternative to facilitate this process. Model-based planning and learning have been discussed extensively in RL literature (Atkeson & Santamaria, 1997; Schaal et al., 1997; Schneider, 1997; Sutton & Barto, 2018). A comprehensive survey on MB-RL methods can be found in the recent review (Luo et al., 2022). The application area of standard approaches on MB-RL mostly focuses on games, robotic control, and autonomous driving. We mainly focus on past works applying MB-RL methods in power systems, and some other related works using surrogate models; but before discussing MB-RL methods, for completeness, we present a brief review of model-free RL/DRL works in power systems.

1.1.1 Model-free RL/DRL in power systems

A significant number of previous efforts utilizing model-free RL methods for power systems applications can be found in a recent review (Chen et al., 2022). The domain of applications broadly comprises frequency regulation, voltage control, and energy management. Leaving aside DRL applications utilizing steady-state (or power flow)-based analysis of power systems, we primarily review relevant studies that discuss model-free DRL applications in emergency control. To this end, there are two types of problems, (a) voltage control and (b) frequency control. In emergency voltage control problems, a deep Q network (DQN)-based load shedding strategy is proposed in Huang et al. (2019). The deep deterministic policy gradient (DDPG) and proximal policy gradient (PPO) methods are utilized in Zhang et al. (2018) and Jiang et al. (2019), respectively, for emergency voltage control. Load-shedding for voltage control is also designed by combining dueling double DQN and behavior cloning (BC) methods (Li et al., 2022). In earlier works, we developed parallel augmented random search (PARS) (Huang et al., 2021), and deep meta reinforcement learning (DMRL)-based approaches (Huang et al., 2022) to address the issues of faster convergence (training), scalability, and adaptation to new scenarios. The safe learning aspects of the PARS algorithm are studied in Vu et al. (2021). The issues of network topology changes are tackled using graph convolutional network (GCN)-based double DQN algorithm in Hossain et al. (2021). Cao et al. (2019) proposed a judgment model for transient stability (stability prediction model), and RL-based (Q-learning) decision-making for reactive compensation. In frequency control, DDPG with multi-agent set-up is used in Yan and Xu (2020), while Xie and Sun (2021) used a novel distributional soft actor-critic (SAC) method. Multi-Q-network-based emergency plans and DDPG-based optimal policy learning are explored in Chen et al. (2020) for emergency frequency control. Lin et al. (2022) solves an out-of-step (OOS) generator tripping problem with a model-free DRL method. For ease of understanding, we summarized these works in Table 1, and it is important to note that these works followed learned optimal policies interacting directly with the grid simulator without using any learned surrogate model (SM) for power systems dynamics. Besides, there are other model-free RL/DRL that can be found in a comprehensive review (Perera & Kamalaruban, 2021). Among them Mahmoud et al. (2021) and Sun et al. (2019) consider voltage and frequency regulation problems, respectively; but, these works do not consider the transient stability problem (Perera & Kamalaruban, 2021).

Table 1 Summary of the RL/DRL methods in emergency control of power systems

References	Application	Algorithm	Type	SM
Huang et al. (2019)	Emergency voltage control	DQN	Model-free	×
Li et al. (2022)	Emergency voltage control	Dueling-DDQN + BC	Model-free	×
Huang et al. (2021)	Emergency voltage control	ARS	Model-free	×
Huang et al. (2022)	Emergency voltage control	ARS + Meta-learning	Model-free	×
Jiang et al. (2019)	Emergency voltage control	PPO	Model-free	×
Li et al. (2021)	Emergency voltage control	DDPG and DQN	Model-free	×
Vu et al. (2021)	Emergency voltage control	ARS + Safe learning	Model-free	×
Hossain et al. (2021)	Emergency voltage control	GCN + DDQN	Model-free	×
Zhang et al. (2018)	Emergency voltage control	DDPG	Model-free	×
Yan and Xu (2020)	Emergency frequency control	Multi-agent + DDPG	Model-free	×
Xie and Sun (2021)	Emergency frequency control	Distributional SAC	Model-free	×
Chen et al. (2020)	Emergency frequency control	Multi-Q + DDPG	Model-free	×
Lin et al. (2022)	OOS generator tripping	DQN	Model-free	×

1.1.2 Model-based RL/DRL and other SM related works in power systems

Model-based RL or DRL studies in power systems have been growing recently. Wang et al. (2021) integrated a deep belief network (DBN)-based surrogate model (SM) into a DRL framework to optimally select the retail energy prices for the community agents. Cao et al. (2022) designed a surrogate model to approximate the nonlinear mapping from the bus active and reactive power injections to the voltage magnitude connected to the DRL-based control design for voltage stabilization. A model-based DRL algorithm with Monte-Carlo tree search for optimal scheduling of a residential micro-grid was developed in Shuai and He (2020). Kamel et al. (2021) introduced a hybrid data-driven and model-based RL for stress reduction of power systems branches. The control problem for branch overload relief is designed based on a sensitivity-based formulation of the power flow model without learning a separate surrogate model. The model-augmented actor-critic method with safety constraints for Volt–VAR control (VVC) is presented in Gao and Yu (2022). This paper learns the environment model for VVC operation utilizing a bootstrap ensemble of probabilistic neural networks.

As indicated earlier, power systems dynamic simulations are computationally intensive. Therefore, developing a surrogate model to replicate the dynamics is a time-efficient solution and has recently been leveraged in some other power grid applications. Rocchetta et al. (2018) and Rocchetta and Patelli (2020) proposed a steady-state power flow emulator, whereas our work focuses on emulating the transient voltage dynamics of power systems. A deep belief network-based surrogate model assisted transient stability constrained optimal power flow solution can be found in Su et al. (2021). The idea of transient stability used in Su et al. (2021) is transient angular stability, which is significantly different from the transient voltage instability (short-term voltage instability) problem considered in our paper. Balduin et al. (2019), a DNN-based surrogate model is introduced to replicate the operation of a low-voltage power grid. In Qiu et al. (2020), a deep-learning-based surrogate model is developed to drastically reduce the real-time computation time of transient stability constrained total transfer capability (TTC) operational planning problem.

Table 2 summarizes the works mentioned in Sec. 1.1.2 and places our work compared to the existing model-based RL/DRL and other SM-based approaches for the power grid.

Table 2 Summary of the Model-based RL/DRL and other SM related works in power systems

References	RL/DRL	SM	Emergency control	Application
Wang et al. (2021)	✓	✓	×	Retail energy pricing
Cao et al. (2022)	✓	✓	×	Voltage stabilization
Shuai and He (2020)	✓	✓	×	Optimal scheduling
Kamel et al. (2021)	✓	×	×	Branch stress reduction
Gao and Yu (2022)	✓	✓	×	Volt–VAR control
Rocchetta et al. (2018) and Rocchetta and Patelli (2020)	×	✓	×	Power flow emulator
Su et al. (2021)	×	✓	×	TSC-OPF
Proposed approach	✓	✓	✓	Voltage control

1.2 Limitations of existing works

With the above discussion, we have found the following limitations and research gaps:

- Existing MB-RL/MB-DRL studies are based on steady-state formulation and do not consider any power system emergency control with transient dynamics. The feasibility and applicability of MB-RL/MB-DRL for emergency control problems with bulk power system dynamics have yet to be addressed.
- Model-free methods have shown the potential of RL/DRL-based approaches in power systems, but these methods require direct interaction with a grid simulator, therefore, perform poorly in terms of sample complexity and training efficiency.

1.3 Main contributions

To address the above-mentioned issues,

- We developed a novel model-based DRL algorithm for emergency voltage control, MB-PARS, which (a) learns a DNN-based surrogate model to simulate power system dynamics, and (b) utilizes a learned surrogate model to train a DRL agent. In the training of the DRL agent, we utilized fast, adaptive, and derivative-free DRL algorithm PARS (Huang et al., 2021). Moreover, we bring the idea of imitation learning (Hussein et al., 2017) to provide a warm start to policy learning. The introduction of MB-PARS greatly reduces the DRL training time and sample complexity.
- We incorporated (1) multi-step prediction loss to improve the prediction capability of the surrogate model and (2) an online update of the surrogate model (in the training phase) to tackle state-action distribution drifting, and (3) reward shaping to accommodate prediction error during the DRL training to help stabilize the training of MB-PARS method. To the best of our knowledge, this is the first model-based DRL algorithm for bulk power system control with voltage transient dynamics being fully considered.

With these contributions, the MB-PARS method is applied for determining an optimal load-shedding strategy against the fault induced delayed voltage recovery (FIDVR) problem (Potamianakis & Vournas, 2006) in the IEEE 300 Bus system. Our study shows that policy training time and sample complexity with MB-PARS can be reduced by 87.7% and 97.5%, respectively, compared to its model-free counterpart. In the testing phase, for new operating conditions including unseen power flow cases and new contingencies, the performance of the trained policy with surrogate model is proven to be satisfactory compared to the state-of-the-art model-free DRL method.

1.4 Organization

The rest of the paper is organized as follows: Sect. 2 includes a comprehensive study on model-based reinforcement learning methods, followed by discussion of distribution drift and data aggregation methods in Sect. 3. Section 4 presents the details of our proposed MB-PARS algorithm and introduces the voltage control problem considered in this paper.

Next, we show the test cases, training and testing results in Sect. 5. At last, conclusions are provided in Sect. 6.

2 Model-based DRL

In this section, we provide a brief overview of model-based DRL, its challenges, and different components.

2.1 Overview

RL is a sequential decision-making process, where an agent interacts with an unknown environment (from the agent's point of view) and collects some abstract signal known as a reward. The problem is studied in a (partially observable) Markov Decision Process (MDP) setting defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ (Sutton & Barto, 2018), where, $\mathcal{S} :=$ state space, $\mathcal{A} :=$ action space and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition function giving the next state $s_{t+1} \in \mathcal{S}$ for a given current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. Besides, for each state-action pair, the environment returns a reward $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$.

In model-based RL, the transition dynamics $\mathcal{P}$ can be represented by (a) a stochastic function $P[s_{t+1} \mid s_t, a_t]$, which is modeled for example as a Gaussian process (Deisenroth & Rasmussen, 2011), $P[s_{t+1} \mid s_t, a_t] = \mathcal{N}(s_t \mid \mu_t, \sigma_t)$, where, μ_t and σ_t are prior mean and standard deviation, or (b) a deterministic function, parameterized by ϕ, with dynamics $s_{t+1} = f_\phi(s_t, a_t)$ (Nagabandi et al., 2018; Yang et al., 2020). In both cases, the model representing the transition dynamics must be learned from prior collected trajectory data. Now, the learned model is utilized to estimate the next-state $\hat{s}_{t+1}$, for a particular state-action pair $\{s_t, a_t\}$. This estimated next state $\hat{s}_{t+1}$ paired with $\{s_t, a_t\}$, are used to compute the reward r_t and thereby the cumulative discounted reward over the entire episode starting for $t = 0$ to T. Thus, the accuracy of the cumulative reward computation depends solely on the accuracy of the probabilistic or parameterized model under consideration, and importantly, the cumulative reward is optimized to retrieve the optimal policy. This is the major challenge for model-based RL compared to model-free RL which does not need any model to learn and is assumed to interact directly with the environment (which may not be even possible for safety-critical applications like power systems).

2.2 DNN-based modeling of transition dynamics

In this work, we modeled the power systems transition dynamics using a DNN $f_\phi(s_t, a_t)$, where ϕ represents the parameter (weights and bias vectors) of the DNN. The most common way is to directly map the next state s_{t+1} as a function of current state s_t, and action a_t, implying $s_{t+1} = f_\phi(s_t, a_t)$ with s_t and a_t as inputs and s_{t+1} as output. Please note that, the reward is determined using the predicted next state s_{t+1}, hence reward function is not learned separately. The data to train $f_\phi(\cdot, \cdot)$ can be collected executing a random policy for different initial state $s_0 \sim \rho_d$ on the actual environment for the horizon $t = 0$ to T, resulting in a collection of ground-truth trajectory data of the form, $\tau = \{(s_0, a_0), (s_1, a_1), \dots, (s_T, a_T)\}$. For a given set of state-transition tuple $\mathcal{D} = \{(s_t, a_t, s_{t+1})\}$, $f_\phi(\cdot, \cdot)$ can be trained by minimizing $\mathcal{L}(\phi) = \frac{1}{|\mathcal{D}|} \sum_{(s_t, a_t, s_{t+1}) \in \mathcal{D}} \|s_{t+1} - f_\phi(s_t, a_t)\|_2^2$.

But, the current model structure $s_{t+1} = f_\phi(s_t, a_t)$ is not efficient, as mentioned in Nagabandi et al. (2018), when s_t and s_{t+1} are close in state-space. Therefore, the learned model $f_\phi(s_t, a_t)$ is prone to make mistakes in predicting future states. Moreover, the problem aggravates with (a) complex dynamics (for instance in power systems) and (b) small duration between two consecutive time step. Additionally, in DRL context, we need to propagate the dynamics for long-horizon roll-outs, where inaccurate dynamics cause the accumulation of error over the entire horizon. To circumvent these issues, we utilized (a) the difference between two consecutive states in modeling $f_\phi(\cdot, \cdot)$, where, $s_{t+1} - s_t = f_\phi(s_t, a_t)$, and (b) multi-step loss function (1) in contrast to the single-step loss function $\mathcal{L}_{ss}(\phi) = \frac{1}{|\mathcal{D}|} \sum_{(s_t, a_t, s_{t+1}) \in \mathcal{D}} \|s_{t+1} - s_t - f_\phi(s_t, a_t)\|_2^2$ while learning $f_\phi(s_t, a_t)$. We follow the steps below to formulate the multi-step (M-step) loss (Yang et al., 2020) based training of $f_\phi(\cdot, \cdot)$:

(a) Make the M-step ground-truth transition tuple, $\mathcal{D}_M = \{(s_t, a_t, s_{t+1}, a_{t+1}, \dots, s_{t+M})\}$

(b) Predict the future states starting from s_t, using $\hat{s}_{t+\tau+1} = \hat{s}_{t+\tau} + f_\phi(\hat{s}_{t+\tau}, a_{t+\tau})$, for $\tau = 0$ to M. It is important to note that except the first step the predicted states $\hat{s}_{t+\tau}$, instead of the ground true state $s_{t+\tau}$, is used as the input of the DNN $f_\phi(\cdot, \cdot)$. The usage of the predicted state as the input of the DNN helps to mitigate the error accumulation for long-horizon prediction.

(c) The DNN is trained using mini-batch stochastic gradient descent minimizing the loss given by (1).

$$\mathcal{L}_{ms}(\phi) = \frac{1}{|\mathcal{D}_M| \times M} \sum_{\substack{(s_{t:t+M}, \\ a_{t:t+M-1}) \\ \in \mathcal{D}_M}} \sum_{\tau=0}^{M-1} \left\| s_{t+\tau+1} - \hat{s}_{t+\tau+1} \right\|_2^2 \tag{1}$$

2.3 Policy derivation with the learned dynamics

After learning the model dynamics, the next important task is to learn policy using the learned model. The goal of model-based RL is to learn a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ for the agent, such that it maximizes the expected cumulative reward over a horizon T, for some given initial state distribution ρ_d and following the learned transition dynamics given by $f_\theta(\cdot, \cdot)$. Mathematically, $\pi^* = \text{argmax}_\pi J(\pi)$, where $J(\pi) = \mathbf{E}_{s_0 \sim \rho_d} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1}) \right]$, $a_t = \pi(s_t)$, $s_{t+1} = f(s_t, a_t)$, and γ is the discount factor. In the general DRL set-up, the policy is parameterized by θ, hence the cumulative reward function $J(\pi)$ becomes $J(\pi_\theta) = \mathbf{E}_{s_0 \sim \rho_d} \left[\sum_{t=0}^{T} \gamma^t r(s_t, \pi(\theta, s_t), s_{t+1}) \right]$, an implicit function of θ, and consequently, the problem converts to finding the θ^*, which maximizes $J(\pi_\theta)$, i.e., $\theta^* = \text{argmax}_\theta J(\pi_\theta)$. For this work, we adopted derivative-free PARS, developed in Huang et al. (2021). Most of the state-of-the-art gradient-based and value-based algorithms are hard to manage in large-scale power grid problems. The details can be found in Huang et al. (2021). In short, the reasons are (a) ineffective action-space exploration, (b) difficulties in gradient computation due to non-smoothness in the environment and reward structure, (c) challenges in parallel implementation, and (d) high sensitivity to the hyper-parameters, which makes the training notoriously difficult. Unlike gradient-based and value-based methods, the derivative-free techniques are (a) easy to implement, (b) easy to parallelism, and (c) easy to tune (due to the lesser number of hyper-parameters).

3 Distribution drift in model-based DRL and data aggregation method

This section considers the implications of dynamical surrogate model-based policy learning, and we inferred that:

- The supervised learning of surrogate model $f_\phi(s_t, a_t)$ requires a ground-truth data set, and the creation of such a comprehensive data set representing different aspects of the state-action distribution is challenging for high dimensional complex systems, e.g., power systems. Moreover, the ground-truth data generation for large-scale power systems needs to be carried out by an expensive physics-based grid simulator. Therefore, training the surrogate model with a relatively smaller data set is more customary.
- The optimality of the learned policy is closely reliant on the accuracy of the trained dynamical surrogate model. Incorporation of multi-step loss and learning differences of states (instead of learning the next state directly) can enhance the generalization capability of the learned surrogate model up to a certain extent, but the training phase performance of the learned policy can fall drastically if the learned surrogate model encounters out-of-distribution state-action pairs, which are not similar to the state-action distribution used in the training phase of surrogate model $f_\phi(s_t, a_t)$, and as mentioned earlier, for high dimensional problem setting, such distributional drift is unavoidable and poses significant challenges in the learning of optimal policy.

```
Require: Retrieve ground-truth interaction data-set $\mathcal{D}_{\text{offline}}$.
Require: Initialize empty data buffer $\mathcal{D}_{\text{online}}$ with size limit.
Require: Initialize policy $\pi_\theta$.
Train surrogate model $f_\phi(\cdot,\cdot)$ using $\mathcal{D}_{\text{offline}}$.
for iteration $k = 1, \cdots, H$ do
    Update policy $\pi_\theta \rightarrow \pi_{\theta+}$ using trained $f_\phi(\cdot,\cdot)$.
    Collect and add ground-truth data to $\mathcal{D}_{\text{online}}$ with updated policy $\pi_{\theta+}$.
    if surrogate model update is true then
        Aggregate Data set $\mathcal{D} = \mathcal{D}_{\text{online}} + 25\%$ of $\mathcal{D}_{\text{offline}}$.
        Retrain surrogate model $f_\phi(\cdot,\cdot)$ with $\mathcal{D}$.
    end if
    $\pi_\theta \leftarrow \pi_{\theta+}$
end for
return Trained policy $\pi_\theta$
```

Algorithm 1 Data Aggregation Method

To solve these issues, we need to reduce the drift between the state-action distribution used for training of $f_\phi(s_t, a_t)$ and the state-action distribution observed under current policy $\pi_\theta(\cdot)$, and following Ross et al. (2011) and Moya et al. (2023) we adopted an online data aggregation method during policy learning. This is achieved by (a) collecting new ground-truth transition data using the current policy $\pi_\theta(\cdot)$, and (b) retraining $f_\phi(s_t, a_t)$ online during policy learning, with a mixture of previously stored and newly collected ground-truth interaction data. The detailed procedure can be found in Algorithm 1.

4 MB-PARS framework for voltage control

This section presents the main algorithms and the architecture details of our proposed MB-PARS method for the FIDVR-related voltage stabilization problem.

4.1 Parallel augmented random search

Parallel Augmented Random Search (PARS), developed in our previous work (Huang et al., 2021), is a scaled-up version of ARS algorithm (Mania et al., 2018) to tackle large-scale grid control problems. PARS is a nest parallelism scheme utilizing the inherent parallelism found in ARS and is implemented using the Ray framework (Moritz et al., 2018). As mentioned earlier, the main objective in policy search is to find θ, which maximizes (2).

$$J(\pi_\theta) = \mathbf{E}_{s_0 \sim \rho_d}\left[\sum_{t=0}^{T} \gamma^t r(s_t, \pi(\theta, s_t), s_{t+1})\right] \tag{2}$$

In (2), $s_0 \sim \rho_d$ represents the randomness of the environment and can be encoded by $\xi := s_0 \sim \rho_d$, and this simplifies the expression in (2) into $J(\pi_\theta) = \mathbf{E}_\xi[\mathbf{r}(\pi_\theta, \xi)]$, where, $\mathbf{r}(\pi_\theta, \cdot) = \sum_{t=0}^{T} \gamma^t r(s_t, \pi(\theta, s_t), s_{t+1})$ is the reward achieved by the policy $\pi_\theta = \pi(\theta, \cdot)$ for a single trajectory rollout. Unlike the existing policy gradient algorithms (PPO, SAC, DDPG, TRPO, TD3, A3C), PARS performs direct exploration in parameter θ space rather than the action space. Detailed explanations of this algorithm can be found in Mania et al. (2018) and Huang et al. (2021). In short, PARS is as follows: (1) the algorithm selects random noises $\delta_1, \dots, \delta_N$ to perturb the policy parameter θ (Mania et al., 2018; Plappert et al., 2017), (2) utilizes the perturbed direction, generates rollouts and computes episode rewards $\mathbf{r}(\pi_\theta, \cdot)$, (3) finally, selects the top-performing directions and updates θ, making it align in the best possible direction. The algorithm uses a finite difference approximation to modify θ instead of back-propagation with 5 main hyper-parameters: $\alpha :=$ step size, $N :=$ policy perturbation direction per iteration, $v :=$ standard deviation of exploration noise, $b :=$ number of top-performing directions, and $m :=$ number of rollouts per perturbation direction. We also observed that existing DRL algorithms such as PPO, A2C, SAC have more than 10 hyper-parameters to tune (Raffin et al., 2021).

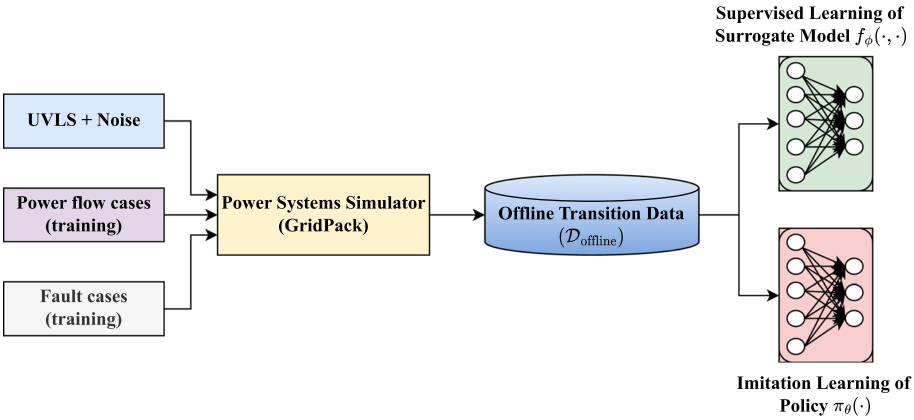

UVLS + Noise
Power flow cases (training)
Fault cases (training)
Power Systems Simulator (GridPack)
Offline Transition Data ($\mathcal{D}_{\text{offline}}$)
Supervised Learning of Surrogate Model $f_\phi(\cdot,\cdot)$
Imitation Learning of Policy $\pi_\theta(\cdot)$

Fig. 1 Initial phase: surrogate model training and imitation learning

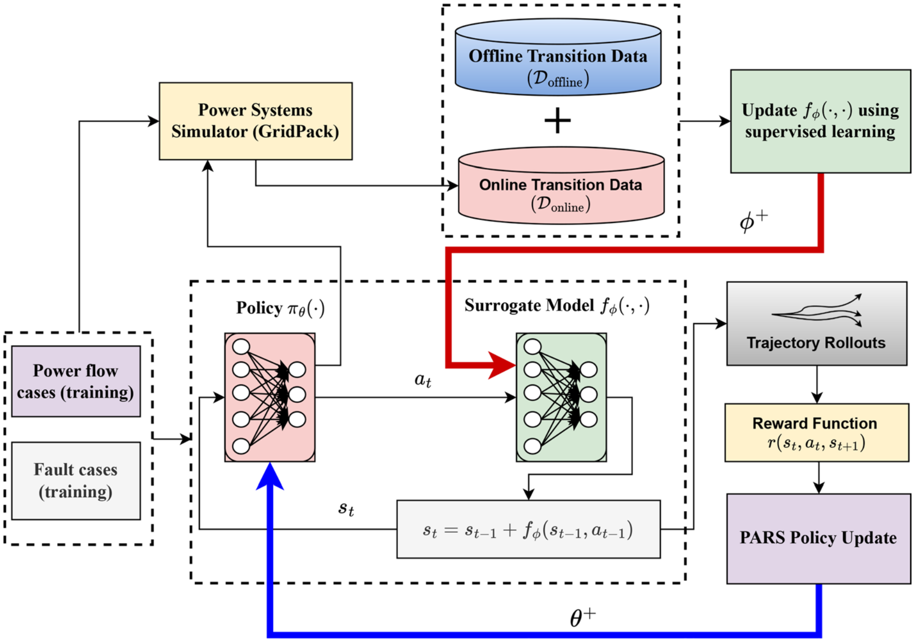

Offline Transition Data ($\mathcal{D}_{\text{offline}}$)
+
Online Transition Data ($\mathcal{D}_{\text{online}}$)
Power Systems Simulator (GridPack)
Update $f_\phi(\cdot,\cdot)$ using supervised learning
ϕ^+
Power flow cases (training)
Fault cases (training)
Policy $\pi_\theta(\cdot)$
Surrogate Model $f_\phi(\cdot,\cdot)$
a_t
s_t
$s_t = s_{t-1} + f_\phi(s_{t-1}, a_{t-1})$
Trajectory Rollouts
Reward Function $r(s_t, a_t, s_{t+1})$
PARS Policy Update
θ^+

Fig. 2 Flowchart of MB-PARS policy update

4.2 MB-PARS algorithm and implementation architecture

We propose a novel MB-PARS algorithm that combines the concept of model-based RL detailed in Sect. 2 and the PARS algorithm to accelerate the control policy learning in large-scale grid control problems.

The key steps of the MB-PARS algorithm are shown in Algorithm 2. The conceptual architecture of Algorithm 2 is shown in Figs. 1 and 2. The learning process can be broadly divided into the following major parts:

4.2.1 Generation of offline training data

We utilized some given rule-based policy, for instance, under voltage load shedding (UVLS) rule, mixed with random noise to generate the system trajectories for different operational scenarios selected for training. In the algorithmic context, we formally define the operational scenarios selected for training as the task set $\mathcal{T}$. The offline data set $\mathcal{D}_{\text{offline}}$ is built by preprocessing the trajectory data as multi-step transition tuples. More details of the data generation are discussed in Sect. 5.2.

4.2.2 Learning of surrogate model

The DNN-based surrogate dynamic model $f_\phi(\cdot, \cdot)$, ϕ representing DNN parameters, is trained on $\mathcal{D}_{\text{offline}}$ minimizing the multi-step loss function given in (1). We utilize the trained surrogate model (faster in computation speed) to generate the rollouts while exploring the parameter space of DRL policy at step-12 of Algorithm 2.

4.2.3 Imitation learning-based policy initialization

The basic idea behind imitation learning-based initialization is to provide a warm start in the MB-PARS policy search. We used a UVLS-based policy to collect training data sets for the surrogate model. Please note that UVLS policy is extensively used in industry for decades (Taylor, 1992), and is a simple rule-based method. We followed the standard behavior cloning (BC) approach to mimic the UVLS policy. BC approaches have been used in other applications, such as locomotion (Nakanishi et al., 2004), autonomous cars (Pomerleau, 1988). Based on the collected data, BC approach learns a policy through supervised learning on demonstration state-action pairs (Nair et al., 2018). This learned policy imitating UVLS policy is referred as the initial policy and utilized to initialize the weights and biases of the target MB-PARS policy. Compared to using initial randomized weights for the MB-PARS policy, which is the common practice, imitation learning greatly reduces the search time at the initial phase of the policy optimization process.

4.2.4 Policy learning

The policy network $\pi(\theta)$ is modeled using a long short-term memory (LSTM) network. LSTM is a type of recurrent neural network capable of learning long-term dependencies. It should be noted that the proposed MB-PARS is utilized for voltage control problems; that is why LSTM is used to learn the temporal correlation of the voltage observations. As mentioned earlier, policy learning is achieved through the derivative-free PARS algorithm. But, before starting the policy training, the weights θ of the policy network are initialized by leveraging imitation learning, as discussed earlier.

4.2.5 Online retraining of surrogate model

To mitigate the issue that arises from the distribution mismatch, as mentioned in Sect. 3, the algorithm generates an online ground-truth data sample using updated policies and adds it to the online data buffer $\mathcal{D}_{\text{online}}$. Then, it retrains the surrogate model at a given frequency (see step-19 to step-21 of Algorithm 2) with a combined set of online and offline data.

```
1:  Create offline data set $\mathcal{D}_{\text{offline}}$.
2:  Instantiate a surrogate DNN model $f_\phi(\cdot,\cdot)$, where $\phi :=$ weights and bias.
3:  Train $f_\phi(\cdot,\cdot)$, and set surrogate model update interval $\mathcal{F}$.
4:  Instantiate the LSTM-based DNN policy $\pi_\theta = \pi(\theta,\cdot)$.
5:  Conduct imitation learning using $\mathcal{D}_{\text{offline}}$. Initialize the policy weights $\theta_0 \in \mathbf{R}^{n\times p}$ with the imitation learning policy weights.
6:  Set hyper-parameters $\alpha, N, v, b, m$, the running mean $\mu_0 = \mathbf{0} \in \mathbf{R}^n$, and standard deviation $\Sigma_0 = \mathbf{I} \in \mathbf{R}^{n\times n}$ of MDP states $(s_t)$, total iteration number $H$, and decay rate $\epsilon$.
7:  for iteration $k = 1, \cdots, H$ do
8:      Randomly pick $N$ directions $\delta_1, \cdots, \delta_N$ for policy perturbation.
9:      Call the current version of surrogate model $f_\phi(\cdot,\cdot)$.
10:     for each $\delta_i|_{i=0}^{N}$ do
11:         Perturb the policy weights $\theta$ in both $\pm$ direction of $\delta_i$:
            $\theta_{ki+} = \theta_{k-1} + v\delta_i$ and $\theta_{ki-} = \theta_{k-1} - v\delta_i$
12:         For each task $p \in \mathcal{T}$, observe the initial state $s_0$, and generate total $2 \times m$ rollouts using the surrogate model $f_\phi(s_{k,t}, a_{k,t})$.
13:         During each rollout, normalize the states $s_{k,t}$ at instant $t$, using $s_{k,t} = \frac{(s_{k,t}-\mu_{k-1})}{\Sigma_{k-1}}$; next obtain the action $a_{k,t} = \pi(\theta_k, s_{k,t})$, and get the new state $s_{k,t+1}$ using state transition rule mentioned in Section $4.3.3$. Update $\mu_k$ and $\Sigma_k$ with $s_{k,t+1}$.
14:         Calculate the average rewards over $m$ rollouts $\{\mathbf{r}_{ki+}, \mathbf{r}_{ki-}\}$, respectively for $\pm$ perturbation.
15:     end for
16:     Select top $b$ directions among $\delta_1, \cdots, \delta_N$ based on max $\{\mathbf{r}_{ki+}, \mathbf{r}_{ki-}\}$ and calculate their standard deviation $\sigma_b$.
17:     $\theta_{k+1} = \theta_k + \frac{\alpha}{b\sigma_b} \sum_{i=1}^{b} (\mathbf{r}_{ki+} - \mathbf{r}_{ki-}) \rightarrow$ Update policy weight.
18:     Evaluate the current policy $\pi_{\theta_{k+1}}$ in the ground-truth environment and generate new multi-step transition tuples for all tasks $\in \mathcal{T}$ and add it to $\mathcal{D}_{\text{online}}$ (having a maximum limit).
19:     if $k/\mathcal{F} = 0$ and $k > 0$ then
20:         Follow step-6 to step-7 of Algorithm-1.
21:     end if
22:     $\pi_\theta \leftarrow \pi_{\theta+}$
23:     Decay $\alpha$ and $v$ with rate $\epsilon$: $\alpha = \epsilon\alpha$, $v = \epsilon v$.
24: end for
25: return $\theta$
```

Algorithm 2 MB-PARS (Proposed Method)

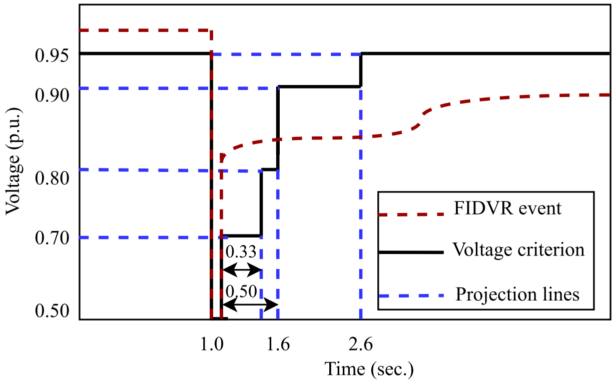


Fig. 3 Transient voltage recovery criterion (PJM, 2021)

4.3 RL/DRL formulation of emergency voltage control problem

This paper considers a short-term voltage instability problem originating from Fault-induced delayed voltage recovery (FIDVR). FIDVR has occurred in various US utilities. Briefly, it is defined as the phenomenon whereby system voltage remains at significantly reduced levels for several seconds after a fault has been cleared. In general, the stalling of air-conditioner (A/C) motors (1-phase induction motors) and prolonged tripping is the root cause of FIDVR. To mitigate FIDVR, it is required to have a well-proof emergency control plan according to the voltage recovery criterion given in PJM transmission planning criteria (PJM, 2021) (shown in Fig. 3). Traditionally, rule-based under-voltage load shedding (UVLS) schemes are used as a part of the control plan (Taylor, 1992). But, in general, UVLS policy does not provide an optimal solution and may cause unnecessary tripping of essential loads. To find the optimal solutions, we need to formulate the problem as a non-convex, nonlinear constrained optimization as detailed in our recent work (Huang et al., 2019). Huang et al. (2019) also provides the details of the MDP formulation of the above-mentioned optimization problem to implement RL-based control schemes. Next, we briefly discuss the MDP structure of the load-shedding-based voltage control problem for completeness.

4.3.1 State

In voltage control problem with load shedding as control action, obviously the main variables of interest are, $V_t := [V_t^1, \dots, V_t^m]^\top$: the voltage measurement of m no. monitored buses at time t, can be denoted as s_t. Consequently, the surrogate model should learn the voltage transitions under applied control actions a_t, using the relation: $s_{t+1} = s_t + f_\phi(s_t, a_t)$. In our previous work (Huang et al., 2019), it is observed that $P_{D,t} := [P_{D,t}^1, \dots, P_{D,t}^n]^\top$ the percentage load remaining at the n no. controlled buses at time t contain relevant information helping in the policy learning process. Hence, we stacked them together to form $\bar{s}_t := [s_t, P_{Dt}]^\top$ the state of the underlying MDP.

4.3.2 Action

The action a_t is to perform load shedding at each of the n no. of controlled buses. We design the load shedding as a continuous control action, and at each time step it can vary from 0 to 20% of the respective bus load. Thus, $a_t = [a_t^1 \dots a_t^n]^\top$, and the action space is $[-0.2, 0]$ (minus means shedding).

4.3.3 State transition

As mentioned earlier, in this paper, the state transition is achieved through the learned surrogate model. Please note that $\bar{s}_t$ contains s_t which follows the relation: $s_{t+1} = s_t + f_\phi(s_t, a_t)$, while $P_{D,t+1} = P_{D,t} - a_t$ can be derived based on the available load $P_{D,t}$ and applied load shedding action a_t. With slight abuse of notation, for ease of understanding, we used s_t to denote the state of the underlying MDP.

4.3.4 Reward

The training of the agent depends primarily on the reward structure. We follow the reward structure used in our previous work (Huang et al., 2019, 2021, 2022) with certain vital modifications. The main objective of this problem is to meet the voltage recovery criteria as shown in Fig. 3 with a minimum amount of load shedding. According to the standard, the voltages should return to at least 0.8, 0.9, and 0.95 p.u. within 0.33, 0.5, and 1.5 s, respectively. Overall the idea is as follows: if the agent fails to recover the voltage above the minimum required voltage level within a certain time duration (usually 4 s) of fault clearance instant T_{pf}, we penalize the agent heavily with a large negative number $-R$, while for successful recovery of the voltages above the minimum required voltage level, the reward is computed as a weighted sum of voltage deviations (according to the specified standard), the amount of load shedding (voltage recovery with minimum possible load interruption) and the penalty for invalid action (action of load shedding even when the load at the bus is already 0). The reward function used in Huang et al. (2021) addresses all the issues mentioned above, but in the context of the current problem where we are utilizing the surrogate model to generate rollouts, we need to tackle some other issues originating from the inaccuracy of the surrogate model dynamics (modeling error). Thus, we introduced two major changes in the reward equation compared to the one used in Huang et al. (2021). The reward r_t is shown in (3)–(4) followed by the discussion on the modifications incorporated in the current work.

$$r_t = \begin{cases} -R, & \text{if } V_t^i < (V_{r_4} - d) \text{ and } t > T_{pf} + 4 \\ -R \times e^{-[\min_i\{V_t^i\} - (V_{r_4} - d)] \times \tau}, & \text{if } (V_{r_4} - d) \le V_t^i < V_{r_4} \text{ and } t > T_{pf} + 4 \\ c_1 \sum_i \Delta V_t^i - c_2 \sum_j \Delta P_t^j - c_3 u_{ilvd}, & \text{otherwise} \end{cases} \tag{3}$$

$$\Delta V_t^i = \begin{cases} \min\{V_t^i - V_{r_1}, 0\} & \text{if } T_{pf} < t < T_{pf} + 0.33 \\ \min\{V_t^i - V_{r_2}, 0\} & \text{if } T_{pf} + 0.33 < t < T_{pf} + 0.5 \\ \min\{V_t^i - V_{r_3}, 0\} & \text{if } T_{pf} + 0.5 < t < T_{pf} + 1.5 \\ \min\{V_t^i - V_{r_4}, 0\} & \text{if } T_{pf} + 1.5 < t \end{cases} \tag{4}$$

Table 3 Power flow scenarios for training and testing

Power flow scenarios	Generation	Load
Scenario 1	100% of total generation	100% of total load
Scenario 2	115% of total generation	115% of total load
Scenario 3	85% of total generation	85% of total load
Scenario 4	92% of total generation	80% of total load in Zone-1

Table 4 Bus indices of fault locations for training and testing

Training	Testing
3, 5, 8, 12, 17, 23	3, 5, 8, 12, 17, 23
	26, 1, 4, 6, 7, 9, 10, 11, 13
	14, 16, 19, 20, 21, 22, 25, 87, 102
	89, 125, 160, 320, 150, 123, 131, 130

where V_t^i is the voltage for bus i at time t, ΔP_t^j is the load shedding amount in p.u. for load bus j at time t, u_{ilvd} is the invalid action penalty, $V_{r_1}, V_{r_2}, V_{r_3}, V_{r_4}$ are the time-based requirement of different voltage levels similar to the standard values given in Fig. 3 (PJM, 2021), and c_1, c_2, c_3 are the weight factors for reward computation for successful recovery of voltage above minimum voltage requirement V_{r_4} after $T_{pf} + 4$ s.

- We introduced a soft penalty if the agent fails to meet the minimum voltage recovery criteria to tackle the model inaccuracy and maintain learning stability. In other words, instead of penalizing the agent heavily if the voltage computed by the surrogate model is close but less than V_{r_4} after $T_{pf} + 4$ (minimum voltage recovery criteria), we penalize the agent according to the soft function $-R \times e^{-[\min_i\{V_t^i\}-(V_{r_4}-d)]\times\tau}$. Here, we used d as a dead band near the minimum required voltage V_{r_4} and τ as a time constant.
- In earlier model-free PARS implementation (Huang et al., 2021), the values of the weights c_1 and c_2 with $(c_1 < c_2)$ penalizes voltage deviations compared to the penalty for taking actions. This caused an implicit jump in the reward function near the minimum voltage requirement, adversely affecting the agent's learning with an imperfect surrogate model-based framework. However, this jump does not cause any adverse issues while learning through the ground-truth physics-based grid simulator. We modified the values c_1 and c_2 and with $(c_1 > c_2)$ to eliminate such implicit jumps to complete the training successfully in the presence of the imperfect surrogate model.

5 Test cases and results

In this section, we first describe the training and testing environment for implementing the proposed method. Then we present the simulation results and comparison with model-free PARS to demonstrate the superiority of the proposed methods in terms of training efficiency and control performance.

5.1 Test case: IEEE 300-bus system

The training and testing of the proposed method are performed with a modified IEEE 300-bus system with loads larger than 50 MW within Zone 1 represented by WECC composite load model (Huang et al., 2019). More details on the IEEE 300-bus test system can be found in our previous work (Huang et al., 2021). In short, the IEEE 300-bus system is a widely adopted open test system for bulk power system and its dynamic simulation is carried out by the open-source platform GridPACK (Huang et al., 2017).

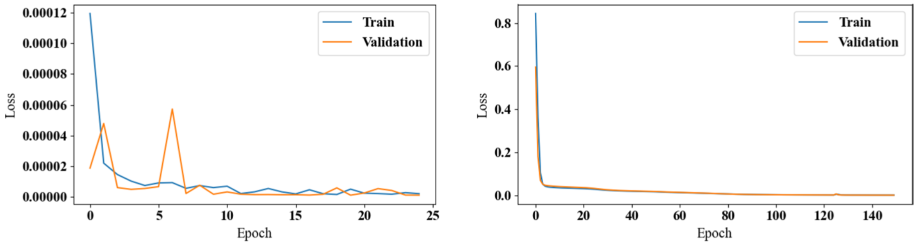
In the training phase, 6 operational scenarios comprising base power flow cases and 6 fault buses are considered. The faults are applied at 1.0 s at the respective buses, and self-cleared after a duration of 0.1 s. While, in the testing phase, we considered 132 operational scenarios comprising 4 power flow scenarios and 33 fault buses. Compared with the training cases, we added 27 more fault buses and 3 more power flows during testing. The reason for applying new test scenarios is to validate the adaptability of the proposed data-driven control policy. The detailed power flow conditions for training and testing are shown in Table 3, while Table 4 shows the details of the fault buses. Load-shedding control actions are considered for all buses with a dynamic composite load model at Zone 1 (34 buses). The load shedding percentage at each control step can vary from 0 to 20%. The agent is designed to provide control decisions (including no action as an option) to the grid in every 0.1 s. The observations included voltage magnitudes at 142 buses within Zone 1 and the remaining fractions of 34 composite loads. We also included the fault information as part of the input values for the policy training, consisting of the fault bus, fault start time, fault duration, and the time distance of the current step to the fault start time. Thus, the dimension of the observation space is 180. The PARS algorithm without a surrogate model is the baseline method we used to compare the proposed model-based approach. The goal is to achieve similar control performance while greatly reducing the training time and sample complexity of the reinforcement learning algorithm.

5.2 Surrogate model training

A fully-connected neural network (FCNN) is considered to represent the surrogate model $f_\phi(\cdot,\cdot)$, which replicates the power system dynamics. The FCNN has 3 hidden layers of size [1000, 500, 200] with ReLU as the nonlinear activation function. The activation function considered for the output layer is sigmoid. For the training of the surrogate model, we created an offline data set utilizing (a) a UVLS policy and adding random noise to the obtained actions and (b) power flow and fault scenarios defined under the training phase. For the offline data generation, (a) select an operational scenario among task set $\mathcal{T}$, (b) generate voltage trajectory data under the action sequences provided by the UVLS policy + random noise using GridPack-based time-domain simulation, and (c) create the M-step transition tuple $\{s_t, a_t, s_{t+1}, a_{t+1}, \dots, s_{t+M}\}$ from the voltage trajectory data and add to $\mathcal{D}_{\text{offline}}$. With this offline data set, we conducted supervised learning to train the surrogate model. Note the training process follows the steps in Sect. 2.2 to create the multi-step loss function defined in (1). We choose a 5-step loss in this experiment, i.e., $M = 5$. The FCNN is trained using mini-batch stochastic gradient descent with optimizer ADAM having a learning rate $0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-7$. The plots of training and validation

Table 5 Hyper-parameters for training IEEE 300-bus system

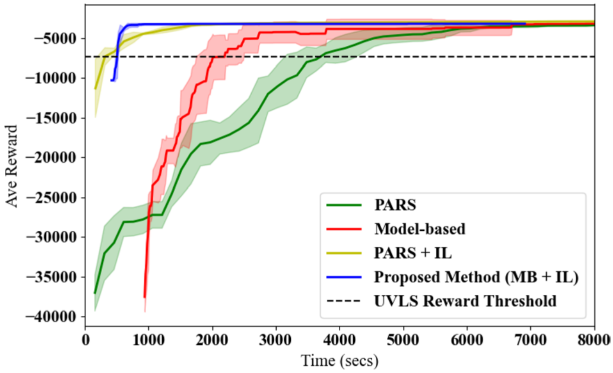
Parameters	Values
Policy network size (hidden layers)	[32, 32]
Number of directions (N)	60
Top directions (b)	30
Step size (α)	1
Step size (α) (with IL)	0.05
SD of exploration noise (υ)	2
SD of exploration noise (υ) (with IL)	0.1
Decay rate (ε)	0.9985
Decay rate (ε) (with IL)	0.9999



(a) MSE Losses of the SM (b) MSE Losses of the IL Policy

Fig. 4 Training plots for the surrogate model (SM) and imitation learning (IL) policy

Fig. 5 Comparison of training average rewards with clock time



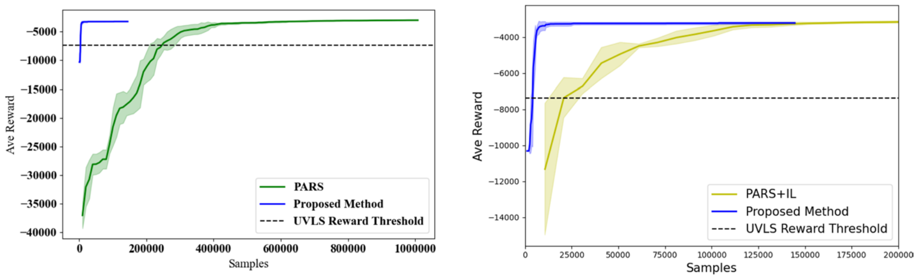
losses of the surrogate model are shown in Fig. 4a, which indicates a good convergence. This completes the initial offline training of the surrogate model. According to Algorithm 2, during online retraining (intermediate training) of the surrogate model, we follow warm start training. The data aggregation of this online update is done following Algorithm 1. Please note that $\mathcal{D}_{\text{online}}$ has a size limit and can store transition data for the last 20 policy iterations, creating transition tuples for 20 trajectory data for each task $\in \mathcal{T}$. The trained surrogate model generates rollouts in the policy learning phase (step-12 of Algorithm 2) and eliminates the need for using a physics-based power systems simulator, e.g., GridPack. For the IEEE 300-bus test system, the surrogate model is run approximately 10 times faster than GridPack, with per step times 0.0038 ± 0.0001 s. and 0.0441 ± 0.0023 s, respectively. It should be noted that the difference is expected to increase for larger-scale power systems.

5.3 Policy training

First, we conducted imitation learning using the UVLS-based offline data (also used for surrogate model learning). Please see Sect. 4.2.3 for more details on imitation learning, which initializes the weights of the policy network, an LSTM network, for this study. Figure 4b shows a good convergence of training and validation mean-squared error (MSE) loss during the training of the imitation learning policy. With the trained surrogate model and imitation learning-based initialized policy network, we started MB-PARS policy training following Algorithm 2. The hyperparameters used in the training of MB-PARS policy are given in Table 5. Figure 5 shows the comparison of average rewards during the RL policy training for different methods, such as (1) model-free baseline PARS (PARS), (2) Surrogate model + PARS (Model-based): PARS algorithm with the surrogate model, (3) model-free baseline PARS + imitation learning (PARS + IL), (4) Surrogate model + PARS + IL (Proposed method). Please note that model-free baseline PARS is trained with the physics-based dynamic power system simulator (GridPACK). Results were obtained by averaging over five different random seeds. The plots in Fig. 5, where the *x*-axis presents the actual clock time, clearly indicate that the proposed method is superior to the other methods in terms of actual training time. Besides, there are some important points to note regarding this result:

- The starting time of the model-based approach is later than the model-free counterpart. This is because of the offline training time for the surrogate model, which is taken into consideration to have a fair comparison regarding algorithm efficiency.
- Different starting reward values in Fig. 5 clearly show that imitation learning provides a better start in policy search than starting from a randomized initial policy.
- Imitation learning reduces the variance in average reward during training and helps the training process to stabilize in the case of the model-based approach, where the training performance is also dependent on the learned surrogate model. Our investigation also finds that the use of imitation learning reduces the chance of training divergence.

On the implementation side, PARS is a nested parallelism scheme implemented using RAY framework (Moritz et al., 2018), and it follows the same architecture for all the variants mentioned here. Our proposed method uses the surrogate model, which accelerates the rollout generation time, thereby helping to improve the overall training time. One policy iteration step includes (a) 12 s (approx.) for the trajectory rollout generation using the trained surrogate model considering multiple rollouts per task $\in \mathcal{T}$ (see Algorithm 2: Step-10 to Step-15), and (b) 4.5 s (approx.) for new ground-truth data generation using the GridPack simulator. New data generation considers only single rollouts per task $\in \mathcal{T}$. The online retraining of the surrogate model is done following a model update interval $\mathcal{F} = 10$ and takes 30 s for each update.



(a) MB + IL vs. PARS

(b) MB + IL vs. PARS + IL

Fig. 6 Comparison of training sample efficiency

5.4 Comparison of training time and sample efficiency

It should be noted that according to the recent literature, model-free baseline PARS (PARS) (Huang et al., 2021) is one of the current state-of-the-art; hence while comparing the sample efficiency and training time, we provide the comparison values of our proposed method with respect to baseline PARS. Our proposed MB-PARS approach in Algorithm 2 (also labeled as ‘MB + IL’) converged around 800 s as shown in Fig. 5, while the state-of-the-art model-free PARS algorithm (Huang et al., 2021) converged around 6500 s. This shows an 87.7% reduction in training time. It is also important to note that rollout generation in model-free PARS is achieved with GridPACK, one of the fastest simulators in the power system. Additionally, compared to PARS + IL, our proposed method started late (considering offline surrogate model training time) but improved quickly and converged faster than the PARS + IL approach. Please note the approaches using imitation learning have a minimal variance in reward plots of Fig. 5. The warm start provided by the IL effectively filters out unnecessary actions, thereby reducing variance in policy searching across different seeds. Next, we compare the sample efficiency. Figure 6a shows the comparison of sample efficiency between the baseline method PARS and our proposed method. Our proposed approach converged around 15,000 samples, while the PARS algorithm converged around

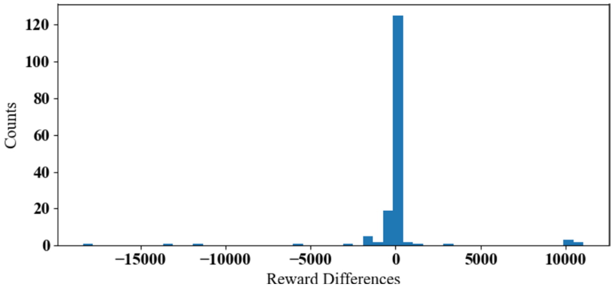
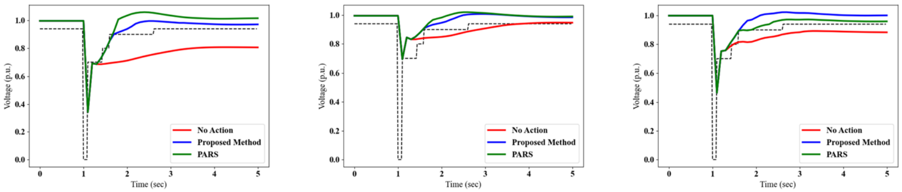


Fig. 7 Reward differences between the proposed method to PARS

Table 6 Comparisons between the proposed method and PARS

Metrics	Proposed method	PARS
Convergence time	800 s	6500 s
Convergence samples	15,000	600,000
Testing reward	– 3859.60	– 3828.91

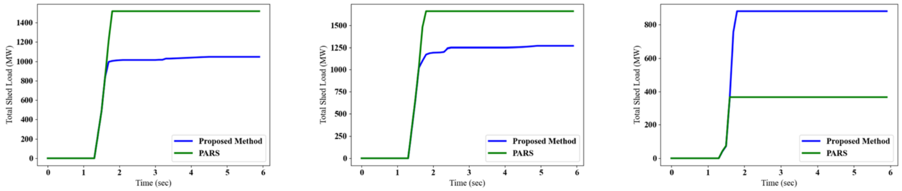


(a) Plot of voltages at Bus 13 for fault at Bus 21

(b) Plot of voltages at Bus 3 for fault at Bus 2

(c) Plot of voltages at Bus 14 for fault at Bus 4

Fig. 8 Plot of voltages for sample scenarios



(a) Total load shed for fault at Bus 21

(b) Total load shed for fault at Bus 2

(c) Total load shed for fault at Bus 4

Fig. 9 Plot of total load shed for sample scenarios

600,000 samples. This shows a 97.5% reduction in samples needed from the power system simulator. This result really highlights the sample efficiency of the proposed method. Fig. 6b shows the comparison of sample efficiency between the PARS + IL and the proposed method. The PARS + IL algorithm converged around 150,000 samples, 11 times more than MB + IL, indicating the proposed method has a 90% reduction in samples nee ded.

5.5 Policy testing

To show the adaptability of the trained policy in unknown scenarios, we tested our trained policy with 33 different fault buses (see Table 4) with 4 different power flow scenarios (see Table 3). Please note that our main contribution is to accelerate the training procedure and improve the sample efficiency of the underlying DRL method. The improvements in training performances are shown in Sect. 5.4. Baseline PARS (Huang et al., 2021) proved efficient in tackling voltage instability-related issues in power systems. Here, we are mainly interested in showing the equivalence in the testing performances for various scenarios. That's why the reward differences between the proposed method and the baseline PARS policy are plotted in Fig. 7 over 132 (= 33 × 4) different cases. Please note that the reward difference plot is zero-centric, confirming the similarity. We also compare the average reward of baseline PARS and our proposed method over the same testing cases (see Table 6). The baseline PARS policy and our method are also close to each other regarding the average reward.

Finally, to better understand the voltage recovery with trained policies, we plotted voltage curves and corresponding load-shedding for fault at bus 21, bus 2, and bus 4 in Figs. 8 and 9. Even though both methods achieved the desired voltage performance for these faults, our proposed method shed less load for faults at buses 21 and 2, while for the fault at bus 4, the proposed method shed more load than PARS. Both recovery rate and load shed amount are part of the reward function, and they jointly influence the overall performance, leading us to compare the reward values in Fig. 7, as discussed earlier.

5.6 Discussions on practical applications

Due to the mission-critical nature of the power grid, the data set for surrogate model training is generated using a simulation platform, e.g., GridPack. Using a simulation environment is common in power systems control room operations for different studies (Brosinsky et al., 2018). In general, the simulation environment utilizes a base model for the real-world power grid and certain variations with respect to that base model. But power systems constantly change; therefore, for certain operational changes, if the underlying real-world grid varies widely, the base model itself needs to be updated; hence, the surrogate model and the policy require retraining in the updated setting.

6 Conclusions

This paper proposes a model-based training approach for the PARS algorithm, aided by imitation learning, to solve the FIDVR problem. The proposed algorithm utilized a surrogate model that learned about the power system dynamics to generate roll-outs in the RL training stage to reduce the training time. We also added imitation learning to this process to provide a warm start to the RL policy, thus reducing the early searching time of the policy training. By testing the proposed approach in the IEEE 300-bus system, we show that the policy trained using the surrogate model with imitation learning achieved similar control performance with the PARS policy trained using GridPACK while needing only 13% of the training time as PARS. This result is significant because long training time is the bottleneck for many RL applications, especially in the power system research field. With the fast-changing nature of the power systems, an efficient training procedure requiring only one-tenth of the training time of the previous method provides a promising avenue for future research and applications.

Author contributions Conceptualization: QH, RH, JT; Methodology: QH, RH, JT, WY, TY, RRH; Formal analysis and investigation: RRH, TY, YD; Writing—original draft preparation: RRH, TY; Writing—review and editing: QH, JT, WY, RH, YD, YL; Funding acquisition: QH, RH; Resources: QH, YL, JT; Supervision: QH, RH.

Funding This work is supported by funding from the U.S. Department of Energy (DOE) Advanced Research Projects Agency-Energy (ARPA-E) OPEN 2018 program. The Pacific Northwest National Laboratory (PNNL) is operated by Battelle for the U.S. Department of Energy (DOE) under Contract DE-AC05-76RL01830.

Data availability Data are available upon request.

Code availability Code is available upon request.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

International Energy Agency (2021). *An energy sector roadmap to carbon neutrality in China*. OECD Publishing.

Atkeson, C. G., & Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In *Proceedings of international conference on robotics and automation* (Vol. 4, pp. 3557–3564).

Australian Energy Market Operator (2017). Black system South Australia 28 September 2016: Final report. https://aemo.com.au/

Balduin, S., Tröschel, M., & Lehnhoff, S. (2019). Towards domain-specific surrogate models for smart grid co-simulation. *Energy Informatics, 2*(1), 1–19.

Birol, F., & Kant, A. (2022). India's clean energy transition is rapidly underway, benefiting the entire world.

Brosinsky, C., Westermann, D., & Krebs, R. (2018). Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers. In *2018 IEEE international energy conference (ENERGYCON)* (pp. 1–6).

California ISO (2013). California ISO-fast facts. https://www.caiso.com/documents/flexibleresourceshelprenewables_fastfacts.pdf

Cao, D., Hu, W., Zhao, J., Zhang, G., Zhang, B., Liu, Z., Chen, Z., & Blaabjerg, F. (2020). Reinforcement learning and its applications in modern power and energy systems: A review. *Journal of Modern Power Systems and Clean Energy, 8*(6), 1029–1042.

Cao, J., Zhang, W., Xiao, Z., & Hua, H. (2019). Reactive power optimization for transient voltage stability in energy internet via deep reinforcement learning approach. *Energies, 12*(8), 1556.

Cao, D., Zhao, J., Hu, W., Ding, F., Yu, N., Huang, Q., & Chen, Z. (2022). Model-free voltage control of active distribution system with PVs using surrogate model-based deep reinforcement learning. *Applied Energy, 306*, 117982.

Chen, C., Cui, M., Li, F., Yin, S., & Wang, X. (2020). Model-free emergency frequency control based on reinforcement learning. *IEEE Transactions on Industrial Informatics, 17*(4), 2336–2346.

Chen, X., Qu, G., Tang, Y., Low, S., & Li, N. (2022). Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid, 13*(4), 2935–2958.

Deisenroth, M., & Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 465–472).

Duan, J., Shi, D., Diao, R., Li, H., Wang, Z., Zhang, B., Bian, D., & Yi, Z. (2019). Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Transactions on Power Systems, 35*(1), 814–817.

Fetting, C. (2020), *The european green deal*. ESDN Report, December (2020)

Gao, Y., & Yu, N. (2022). Model-augmented safe reinforcement learning for Volt–VAR control in power distribution networks. *Applied Energy, 313*, 118762.

Glavic, M. (2019). (Deep) reinforcement learning for electric power system control and related problems: A short review and perspectives. *Annual Reviews in Control, 48*, 22–35.

Hatziargyriou, N., Milanovic, J., Rahmann, C., Ajjarapu, V., Canizares, C., Erlich, I., Hill, D., Hiskens, I., Kamwa, I., Pal, B., Pourbeik, P., Sanchez-Gasca, J., Stankovic, A., Van Cutsem, T., Vittal, V., & Vournas, C. (2021). Definition and classification of power system stability-revisited and extended. *IEEE Transactions on Power Systems, 36*(4), 3271–3281.

Hossain, R. R., Huang, Q., & Huang, R. (2021). Graph convolutional network-based topology embedded deep reinforcement learning for voltage stability control. *IEEE Transactions on Power Systems, 36*, 4848–4851.

Huang, R., Jin, S., Chen, Y., Diao, R., Palmer, B., Huang, Q., & Huang, Z. (2017). Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform. In *2017 IEEE power and energy society general meeting* (pp. 1–5).

Huang, R., Chen, Y., Yin, T., Huang, Q., Tan, J., Yu, W., Li, X., Li, A., & Du, Y. (2022). Learning and fast adaptation for grid emergency control via deep meta reinforcement learning. *IEEE Transactions on Power Systems, 37*, 4168–4178.

Huang, R., Chen, Y., Yin, T., Li, X., Li, A., Tan, J., Yu, W., Liu, Y., & Huang, Q. (2021). Accelerated derivative-free deep reinforcement learning for large-scale grid emergency voltage control. *IEEE Transactions on Power Systems, 37*(1), 14–25.

Huang, Q., Huang, R., Hao, W., Tan, J., Fan, R., & Huang, Z. (2019). Adaptive power system emergency control using deep reinforcement learning. *IEEE Transactions on Smart Grid, 11*(2), 1171–1182.

Huang, Q., Huang, R., Palmer, B. J., Liu, Y., Jin, S., Diao, R., Chen, Y., & Zhang, Y. (2019). A generic modeling and development approach for WECC composite load model. *Electric Power Systems Research, 172*, 1–10.

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys, 50*(2), 1–25.

Jiang, C., Li, Z., Zheng, J., & Wu, Q. (2019). Power system emergency control to improve short-term voltage stability using deep reinforcement learning algorithm. In *2019 IEEE 3rd international electrical and energy conference (CIEEC)* (pp. 1872–1877).

Kamel, M., Dai, R., Wang, Y., Li, F., & Liu, G. (2021). Data-driven and model-based hybrid reinforcement learning to reduce stress on power systems branches. *CSEE Journal of Power and Energy Systems, 7*(3), 433–442.

Kamruzzaman, M., Duan, J., Shi, D., & Benidris, M. (2021). A deep reinforcement learning-based multi-agent framework to enhance power system resilience using shunt resources. *IEEE Transactions on Power Systems, 36*(6), 5525–5536.

Li, J., Chen, S., Wang, X., & Pu, T. (2021). Research on load shedding control strategy in power grid emergency state based on deep reinforcement learning. *CSEE Journal of Power and Energy Systems, 8*, 1175–1182.

Lin, B., Wang, H., Zhang, Y., & Wen, B. (2022). Real-time power system generator tripping control based on deep reinforcement learning. *International Journal of Electrical Power and Energy Systems, 141*, 108127.

Li, X., Wang, X., Zheng, X., Dai, Y., Yu, Z., Zhang, J. J., Bu, G., & Wang, F.-Y. (2022). Supervised assisted deep reinforcement learning for emergency voltage control of power systems. *Neurocomputing, 475*, 69–79.

Luo, F. -M., Xu, T., Lai, H., Chen, X. -H., Zhang, W., & Yu, Y. (2022). A survey on model-based reinforcement learning. arXiv:2206.09328

Mahmoud, M., Abouheaf, M., & Sharaf, A. (2021). Reinforcement learning control approach for autonomous microgrids. *International Journal of Modelling and Simulation, 41*(1), 1–10.

Mania, H., Guy, A., & Recht, B. (2018). Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in neural information processing systems* (Vol. 31).

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., & Stoica, I., (2018). Ray: A distributed framework for emerging AI applications. In *13th USENIX symposium on operating systems design and implementation)* (pp. 561–577).

Moya, C., Lin, G., Zhao, T., & Yue, M. (2023). On approximating the dynamic response of synchronous generators via operator learning: A step towards building deep operator-based power grid simulators. arXiv preprint arXiv:2301.12538

Nagabandi, A., Kahn, G., Fearing, R. S., & Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 7559–7566).

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 6292–6299).

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems, 47*(2–3), 79–91.

Perera, A., & Kamalaruban, P. (2021). Applications of reinforcement learning in energy systems. *Renewable and Sustainable Energy Reviews, 137*, 110618.

PJM (2021). Exelon transmission planning criteria. https://www.pjm.com/-/media/planning/planning-criteria/exelon-planning-criteria.ashx?la=en

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., & Andrychowicz, M. (2017). Parameter space noise for exploration. arXiv preprint arXiv:1706.01905

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems* (Vol. 1, pp. 305–313).

Potamianakis, E. G., & Vournas, C. D. (2006). Short-term voltage instability: Effects on synchronous and induction machines. *IEEE Transactions on Power Systems, 21*(2), 791–798.

Qiu, G., Liu, Y., Zhao, J., Liu, J., Wang, L., Liu, T., & Gao, H. (2020). Analytic deep learning-based surrogate model for operational planning with dynamic TTC constraints. *IEEE Transactions on Power Systems, 36*, 3507–3519.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research, 22*(1), 12348–12355.

Rocchetta, R., & Patelli, E. (2020). A post-contingency power flow emulator for generalized probabilistic risks assessment of power grids. *Reliability Engineering and System Safety, 197*, 106817.

Rocchetta, R., Zio, E., & Patelli, E. (2018). A power-flow emulator approach for resilience assessment of repairable power grids subject to weather-induced failures and data deficiency. *Applied energy, 210*, 339–350.

Ross, S., Gordon, G., & Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 627–635).

Schaal, S., et al. (1997). Learning from demonstration. *Advances in Neural Information Processing Systems, 9*, 1040–1046.

Schneider, J. G. (1997). Exploiting model uncertainty estimates for safe dynamic control learning. In *Advances in neural information processing systems* (pp. 1047–1053).

Shuai, H., & He, H. (2020). Online scheduling of a residential microgrid via Monte-Carlo tree search and a learned model. *IEEE Transactions on Smart Grid, 12*(2), 1073–1087.

Su, T., Liu, Y., Zhao, J., & Liu, J. (2021). Deep belief network enabled surrogate modeling for fast preventive control of power system transient stability. *IEEE Transactions on Industrial Informatics, 18*(1), 315–326.

Sun, J., Zhu, Z., Li, H., Chai, Y., Qi, G., Wang, H., & Hu, Y. H. (2019). An integrated critic-actor neural network for reinforcement learning with application of DERs control in grid frequency regulation. *International Journal of Electrical Power and Energy Systems, 111*, 286–299.

Sutton, R., & Barto, A. (2018). *Reinforcement learning: An introduction*. MIT Press.

Taylor, C. W. (1992). Concepts of undervoltage load shedding for voltage stability. *IEEE Transactions on Power Delivery, 7*(2), 480–488.

United Nations (2023). Intergovernmental Panel on Climate Change longer report. https://www.ipcc.ch/report/ar6/syr/

US Department of Energy (2021). How we're moving to net-zero by 2050. https://www.energy.gov/articles/how-were-moving-net-zero-2050

Vu, T. L., Mukherjee, S., Huang, R., & Huang, Q. (2021). Safe reinforcement learning for grid voltage control. arXiv preprint arXiv:2112.01484

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P., & Ba, J. (2019). Benchmarking model-based reinforcement learning. arXiv preprint arXiv:1907.02057

Wang, X., Liu, Y., Zhao, J., Liu, C., Liu, J., & Yan, J. (2021). Surrogate model enabled deep reinforcement learning for hybrid energy community operation. *Applied Energy, 289*, 116722.

Xie, J., & Sun, W. (2021). Distributional deep reinforcement learning-based emergency frequency control. *IEEE Transactions on Power Systems, 37*, 2720–2730.

Yang, Y., Caluwaerts, K., Iscen, A., Zhang, T., Tan, J., & Sindhwani, V. (2020). Data efficient reinforcement learning for legged robots. In *Proceedings of the conference on robot learning. Proceedings of machine learning research* (Vol. 100, pp. 1–10).

Yan, Z., & Xu, Y. (2018). 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.

Yan, Z., & Xu, Y. (2020). A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system. *IEEE Transactions on Power Systems, 35*(6), 4599–4608.

Zhang, J., Lu, C., Fang, C., Ling, X., & Zhang, Y. (2018). Load shedding scheme with deep reinforcement learning to improve short-term voltage stability. In *2018 IEEE innovative smart grid technologies-Asia (ISGT Asia)* (pp. 13–18).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ramij Raja Hossain[1] · Tianzhixi Yin[1] · Yan Du[1] · Renke Huang[1] · Jie Tan[2] · Wenhao Yu[2] · Yuan Liu[1] · Qiuhua Huang[1]

✉ Tianzhixi Yin
tianzhixi.yin@gmail.com

Ramij Raja Hossain
rhossain@ieee.org

Yan Du
ydu0116@gmail.com

Renke Huang
huangrenke@gmail.com

Jie Tan
jietan@google.com

Wenhao Yu
magicmelon@google.com

Yuan Liu
yuan.liu@pnnl.gov

Qiuhua Huang
qiuhua.huang@ieee.org

[1] Pacific Northwest National Laboratory (PNNL), Richland, WA 99354, USA

[2] Google Brain, Google Inc., Mountain View, CA 94043, USA