Reinforcement learning for robust voltage control in distribution grids under uncertainties

Aleksandr Petrusev, Muhammad Andy Putratama, Rémy Rigo-Mariani, Vincent Debusschere, Patrick Reignier, Nouredine Hadjsaid

Reinforcement Learning for Robust Voltage Control in Distribution Grid under Uncertainties

Aleksandr Petrusev[1,2], Muhammad Andy Putratama[1], Rémy Rigo-Mariani[1], Vincent Debusschere[1], Patrick Reignier[2], Nouredine Hadjsaid[1]

[1]Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab, 38000, Grenoble, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000, Grenoble, France

ARTICLE INFO

Keywords:

Voltage control, Reinforcement Learning, TD3PG, PPO, Flexibility, PV production, Batteries, Distribution Grid, Second-order conic relaxation, Optimal Power Flow

ABSTRACT

Traditional optimization-based voltage controllers for distribution grid applications require consumption/production values from the meters as well as accurate grid data (i.e., line impedances) for modeling purposes. Those algorithms are sensitive to uncertainties, notably in consumption and production forecasts or grid models. This paper focuses on the latter. Indeed, line parameters gradually deviate from their original values over time due to exploitation and weather conditions. Also, those data are oftentimes not fully available at the low voltage side thus creating sudden changes between the datasheet and the actual value. To mitigate the impact of uncertain line parameters, this paper proposes the use of a deep reinforcement learning algorithm for voltage regulation purposes in a distribution grid with PV production by controlling the setpoints of distributed storage units as flexibilities. Two algorithms are considered, namely TD3PG and PPO. A two-stage strategy is also proposed, with offline training on a grid model and further online training on an actual system (with distinct impedance values). The controllers' performances are assessed concerning the algorithms' hyperparameters, and the obtained results are compared with a second-order conic relaxation optimization-based control. The results show the relevance of the RL-based control, in terms of accuracy, robustness to gradual or sudden variations on the line impedances, and significant speed improvement (once trained). Validation runs are performed on a simple 11-bus system before the method's scalability is tested on a 55-bus network.

I. INTRODUCTION

One of the main tasks of distribution system operators (DSO) is to control voltage levels within specified limits, typically in the range [0.95-1.05] normalized per unit (p.u.) under normal conditions. DSOs can use different levers to adapt the voltage. Conventional regulation consists of the control of On-Load-Tap-Changer (OLTC) or capacitor banks [1]. In recent years, new control capabilities emerged due to the increasing penetration of Distributed Energy Resources (DER) and demand-side management. Among those controls are shifts in prosumers consumption, regulation of solar and wind power plants, and charge and discharge of storage systems (fixed or mobile like electric vehicles) [2]. All these options denoted as "flexibilities", can be used for a wide range of grid operations apart from voltage regulation, such as loss minimization, peak load shaving, and so on. Various optimization-based solutions have been proposed in the literature to model and control these flexibilities to maintain the voltage within specific limits, such as [3] and [4]. These optimization approaches require consumption values from the meters as well as grid data (i.e. topology and line impedance). Those data are typically integrated into model predictive control (MPC) strategies that rely on time series forecasts for the bus power (i.e. load/generation).

* Institute of Engineering Univ. Grenoble Alpes

The performance of those controllers depends on the capacity to mitigate uncertainties. The first cause of uncertainty that comes to mind would be the forecasting of consumption and production. However, even with perfect predictions, model-based optimization controllers' performance can be degraded by drifting line parameters from their original values due to aging and weather conditions (continuous change) or by a lack of knowledge of the grid (abrupt change). Indeed, the impedances may not be perfectly known at the low voltage level in cases of old facilities. This directly impacts the accuracy of the system model that is embedded in MPC architectures. Thus, potentially significant errors may occur in the grid state estimation within the controllers, consequently reducing the voltage regulation quality. Moreover, optimization-based algorithms can require significant time to compute the controls, especially when considering uncertainties mitigation as a core feature, e.g., stochastic optimizations [5]. In the literature, uncertainties, when considered, mostly referred to the bus power values (i.e. load and/or generation), while neglecting the lack of knowledge on the grid itself, notably impedances. Also, considering multiple heterogeneous degrees of freedom (i.e., photovoltaic, PV, and energy storage systems, ESS, inverters, OLTC, or reconfigurations) may be too optimistic, if not unrealistic, for real-life case studies. The paper thus tackles the robustness of the control to lines impedances variations.

Artificial intelligence (AI) offers a relevant alternative to deal effectively and rapidly with the different sources of uncertainties, not only on grid impedances but also on local production and consumption forecasts. Neural networks already provided noticeable rapidity in grid voltage prediction, allowing us to consider them in quasi-real-time (<1s) [6] and proved efficient to compute optimal flexibility control of inverters[7] and various DER[8]. These experience-based strategies are not able to perform optimization and map already calculated controls from classical optimization algorithms with input-measured data (i.e., supervised learning). Moreover, such neural networks cannot cope with the physical parameters drifting over time. Reinforcement learning (RL) algorithms can overcome this drawback, as they do not need the expected outputs of the controller, but only an evaluation of the quality of the performed controls. For instance, [9] investigates the performance of various RL algorithms for the energy management of storage units and loads in grids with high penetration of DER. However, the implemented algorithms assume to have a perfect forecast about temperature, load and local generation, and nominal values of all line parameters, which is not completely realistic. Adaptive Q-learning is also tested to control the demand side of home area grids [10]. The proposed control strategy is applied for each household separately but does not consider electrical grid constraints.

Many investigations have been carried out for voltage control using different RL algorithms. A multi-agent algorithm based on the deep deterministic policy gradient method (DDPG) was presented in [11], with a grid regulation based on the control of generation units. Several iterations during the execution were necessary to obtain voltage values. A voltage control with PV inverters, OLTC, and switched capacitors was proposed in [12]. The control is based on Soft Actor-Critic (SAC) algorithm and requires measurements of active/reactive consumption and generation at each node, which can be problematic due to privacy concerns. Two reinforcement learning techniques performing voltage regulation with generation units and relying on Deep Q-Network (DQN) and DDPG showed promising results [13]. Control of PV generation by DDPG was extended to static Var compensator in [14]. Coupling DDPG with a surrogate neural network (to learn the nonlinear mapping between the active/reactive power injections and the voltage magnitude at each node) allowed to control of distributed generation units as well as of tap transformer position in [15]. DDPG

considers the limited knowledge of grid parameters, but an external predictor of load and generation is needed. Finally, a two-stage voltage control strategy (based on DDPG as well) was proposed in [16] to mitigate voltage violations caused by the uncertainties of EVs and load consumption.

The presented work compares two RL algorithms: Twin Delayed Deep Deterministic Policy Gradient (TD3PG) [17] and Proximal Policy Optimization (PPO) [18]. Indeed, they can be more performant than most state-of-the-art algorithms on technical aspects [17]. They are implemented here to handle the problem of voltage control without knowing the exact values of the line parameters for medium voltage (MV) and low voltage (LV) distribution grids and considering limited flexibility levers. Being pre-trained, they require an extremely short time to compute the optimal controls at every execution (<<1s). The main contributions of this work are:

- A comparison for two different types of RL algorithms that show the most promising results comparing to other algorithms [17]: off-policy TD3PG and on-policy PPO [18] using a proposed performance metric and a large range of sensitivity analyses;
- A comparison with a conventional optimization-based approach about robustness to the grid impedance uncertainties of both the online and offline policies.

The main interests of the implemented algorithms are that:

- They do not require an external forecast and implicitly embed a prediction of the bus power to compute the controls;
- No potentially sensitive private data such as bus load and generation are needed as the controls are only computed based on the voltage measurements;
- Two-stage training (combining offline and online training) is proposed to cope with impedance uncertainties (thanks to the online phase) and avoid dangerous voltage values at the distribution grid during online training (thanks to the offline phase).

The rest of the paper is organized as follows: Section II describes the case study and the principle of reinforcement learning algorithms. Section III discusses the implementation of the voltage control and its tuning concerning the main hyperparameters of the algorithms. Section IV presents the results of voltage control for both RL algorithms, investigating their robustness to line impedance deviation, compared to an optimization-based control. Scalability tests on a 55-bus system are also introduced. Finally, the last section provides conclusions and perspectives for future works.

LIST OF THE MAIN ACRONYMS

RL	Reinforcement Learning
SAC	Soft Actor-Critic
PPO	Proximal Policy Optimization
DDPG	Deep Deterministic Policy Gradient
TD3PG	Twin Delay Deep Deterministic Policy Gradient
OPF	Optimal Power Flow
VPI	Voltage Performance Index
SOC	State of Charge

II. CONSIDERED PROBLEM AND PROPOSED FRAMEWORK

A. Case Study

The simulation environment consists of a medium/low voltage grid simulated using Pandapower. The considered grid (shown in Fig. 1) comprises eleven nodes ($n \in N$), six loads, three PV, and four flexibilities, i.e., batteries with inverters ($f \in F$) that regulate active and reactive power flows respectively. The model integrates real electrical consumption profiles for loads (from the "Smart meters in London" project [19]), representative PV generation profiles from the National Renewable Energy Laboratory (NREL) [20], and datasheet-based characteristics for the line impedances. Batteries have 90 % efficiency for charging and discharging, capacities of 500 kWh, and a rated power of 300 kW.

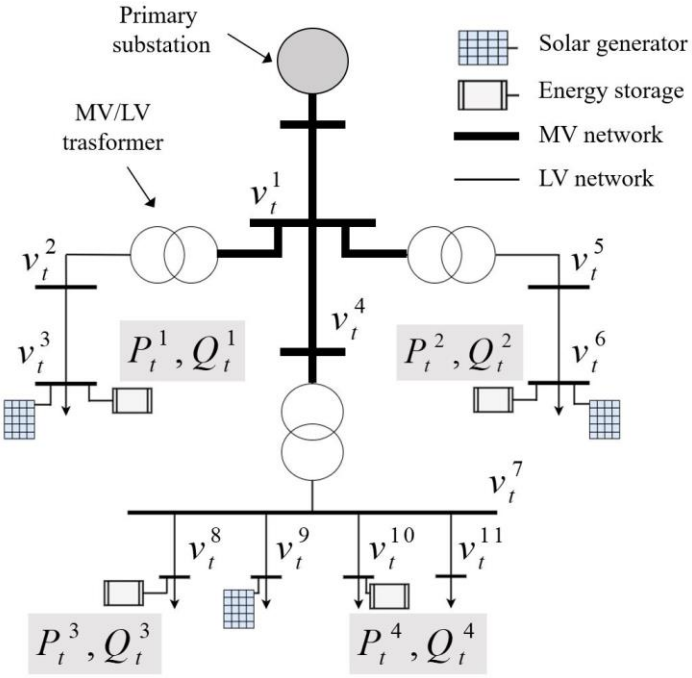

Fig. 1: Considered MV/LV grid, including local storage and production in selected nodes

B. Baseline: Conventional Optimization-Based Voltage Control

A reference model-based voltage optimization control, adapted from [21], is considered a baseline for comparison. This controller is formulated as a multi-objective optimal power flow (OPF) problem with a second-order conic relaxation to account for the grid model. The main objective is to minimize the total voltage deviation over a predefined time horizon T (1). The problem is subject to grid and battery constraints that are not represented here for the sake of clarity but are described in [21]. The losses are integrated into the objective function so that the conic relaxation of the power flow constraint is valid. Thus, as discussed in [21], careful tuning of the objective function loss weight C_{loss} shall be conducted so that priority is given to the penalization of the voltage deviations ΔV_t^n beyond the acceptable limits.

$$\min\left(\sum_{t=1}^{T}\sum_{n=1}^{N}\Delta V_t^n + C_{loss}\sum_{t=1}^{T}\sum_{l=1}^{L}P_{loss\,t}^{\ l}\right) \quad \textbf{(1)}$$

where L is the set of lines and P_{losst}^{l} the losses in line l at time t.

C. Definitions and Algorithms Selection for Reinforcement Learning

Typical supervised learning relies on a training set of examples provided by a knowledgeable external supervisor [22]. Reinforcement learning, on the contrary, learns from its own experience [26]. The main elements of RL training are:

- *Environment*: the considered system represented at time t by state variables s_t;
- *Agent*: the controller that takes the action a_t, depending on s_t, which leads to the next state s_{t+1} once the actions are processed by the environment;
- *Reward* r_t: the controller objective metric calculated from s_{t+1}, to assess the relevance of the actions a_t taken.

Each transition sample [s_t; s_{t+1}; a_t; r_t] is saved into a large table ("replay buffer"). During the learning phase, which is separated from sample acquisition, the algorithm selects random samples from this table thus limiting correlations in the observations and smoothing changes in the data distribution. The final goal of the RL algorithm is to find an optimal policy π which is a mapping between states and actions, $a_t = \pi(s_t)$. The policy is generated so that it maximizes the cumulative reward over a time T, specified by the user.

The two main types of RL algorithms are *i*) on-policy learning (for instance advantage actor-critic, A2C, trust region policy optimization, TRPO, and PPO), which learns about the return from actions taken using its current policy, and *ii)* off-policy learning (for instance DDPG, SAC, and TD3PG), which learns about one policy, π_1, while the reward observations are generated by action sequence from another policy, π_2. TD3PG is chosen as the baseline off-policy algorithm thanks to its technical improvements over DDPG[17]. Also, it can outperform SAC in higher dimensions due to its deterministic policy. As for on-policy algorithms, PPO is chosen as it is simpler to implement than TRPO, due to its soft constraints in the objective function, and empirically seems to perform at least as well as TRPO and A2C [18]. Both chosen algorithms contain two parts: an "Actor" part, which learns an optimal policy, and a "Critic" part, which is used to predict a return (discounted sum of future rewards). Both parts are represented by respective neural networks and their training consists in optimizing their parameters (i.e., weights and biases). The "Critic" part is different for the two algorithms. In TD3PG it computes an action-value function $Q(s_t,a_t)$, which is defined as the expected future reward of starting in state s_t, taking action a_t and following a given policy π, and expressed in (2) [22]:

$$Q_t^{\pi}(s_t, a_t) = \sum_{k=0}^{\infty} \mathrm{E}_{\pi}\left[\gamma^k \times r_{t+k+1} \mid s_t, a_t\right] \quad \textbf{(2)}$$

where E_π is the expectation and γ is the discount factor, that determines the importance of future rewards and is usually between [0.95;0.99].

In PPO, the Critic network computes the state-value function $V_t^{\pi}(s)$, which is defined as the expected future reward of starting in state s_t and following a given policy π, and expressed in (3) [22]:

$$V_t^{\pi}(s_t) = \sum_{k=0}^{\infty} \mathrm{E}_{\pi}\left[\gamma^k \times r_{t+k+1} \mid s_t\right] \quad \textbf{(3)}$$

The second part of the algorithms, the "Actor" part, computes a_t for a given s_t. Its training consists of updates of its network parameters through gradient ascent to get output a_t, which leads to higher expected return, computed based on approximations of the "Critic" network.

Thus, the algorithms concurrently learn a state-value or action-value function and a policy. They use the transitions (obtained by the acting agent) saved in the "replay buffer" to learn the state/action-value functions, which are in turn used to learn the policy. Such interactions of "Actor" and "Critic" networks allow gradually getting closer to the desired goal, i.e., the optimal policy π.

D. Operation Principle

The model of the considered grid is connected to the RL algorithm for actions and observations exchanges, as presented in **Erreur ! Source du renvoi introuvable.**. This figure describes the algorithm operation along the training period with successive iterations between the modeled system (environment/grid simulation) and the agent. Each iteration consists of an exchange of information – i.e. observation and reward value returned by the environment and controls outputs from the RL algorithms. The operational phase, once the training is done, follows a similar flow with a controller that computes the controls at time *t+1* from measurements/states at time *t*. This state s_t is represented by the following vector (4):

$$s_t = \left[\mathbf{v_t^n}, \mathbf{P_t^f}, \mathbf{Q_t^f}, \mathbf{SOC_t^f}, t \right] \quad \textbf{(4)}$$

where *t* denotes the current timestep, $\mathbf{v_t^n}$ the voltage at each node of the grid, $\mathbf{P_t^f}$ the active power of the batteries, $\mathbf{Q_t^f}$ the reactive power of inverters, and $\mathbf{SOC_t^f}$ the state of charge of the batteries. As previously mentioned, the algorithm does not require access to the node consumption/production data but only the voltage measurements. From those observations, the RL algorithm computes the action a_t as the reference setpoints (active/reactive) for the flexibilities at the next timestep *t+1*, as expressed in (5). Thus, the RL algorithm implicitly accounts for built-in load and local generation forecast for the next timestep based on the current measurements.

$$a_t = \left[\mathbf{P_{t+1}^f}, \mathbf{Q_{t+1}^f} \right] \quad \textbf{(5)}$$

Based on the new flexibilities setpoints and PV/load values at timestep *t+1*, the simulation computes the new state variables for the grid, thus providing the observation s_{t+1} to the RL algorithm. From this observation, the algorithm also calculates the corresponding reward r_t. As expressed in (6), the reward tends to penalize the voltage deviations above the normal operations, set between 0.95 p.u. and 1.05 p.u. Additionally, to preserve the storage availability, an exponential reward penalizes the SOC values the more they deviate from 0.5 in absolute value The coefficient ω is used in the exponent to increase the impact of strong SOC deviations, α is a penalty for exceeding voltage limits, and β is a linear coefficient penalty for too low or high SOC.

$$r_t = -\alpha . \sum_{n=1}^{N} \max\left(0, (v_{t+1}^n - 1.05)\right)^2 + \max\left(0, (0.95 - v_{t+1}^n)\right)^2 - \frac{1}{\beta} . \sum_{f=1}^{F} \exp\left(\omega \cdot |SOC_{t+1}^f - 0.5|\right) \quad \textbf{(6)}$$

The objective of the reward function is to penalize the SOC deviations around 50 %, with increasing values when reaching overcharge or over-discharge (i.e. exponential). An alternative could have consisted in

penalizing only the values over the 0% -100% tolerance (e.g. with min/max functions). However, this would have led to a discontinuous, non-derivative reward function. The proposed exponential formulation was chosen instead to simplify the training that relies on gradient descent algorithms

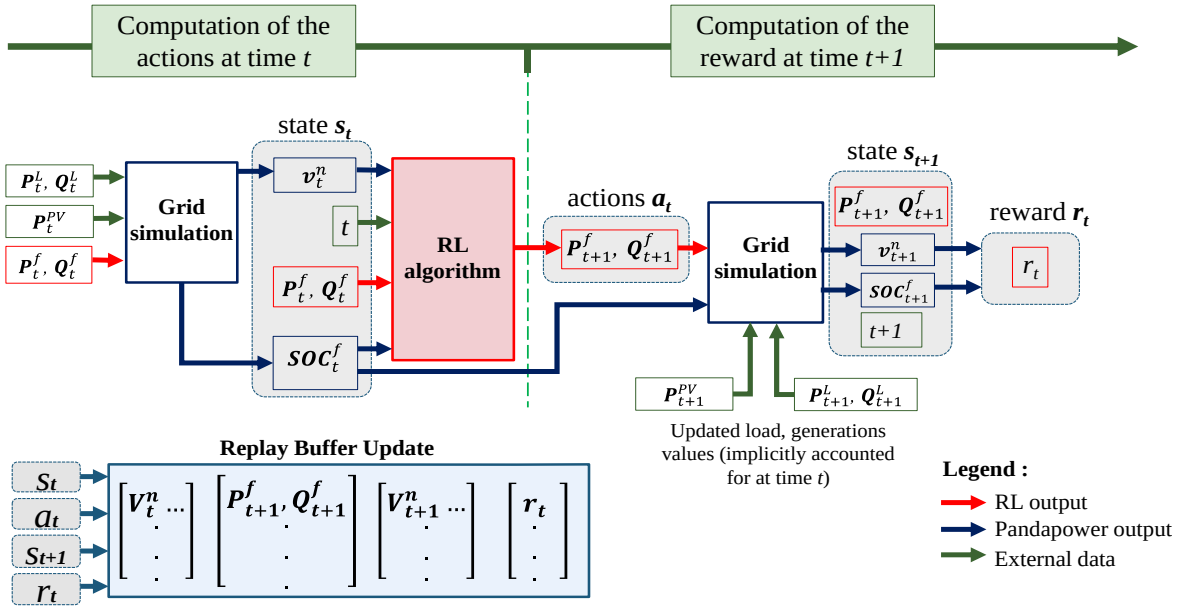

Fig. 2: RL algorithms training and interaction with the study case (grid displayed in Fig. 1)

The objective of the RL algorithm is to maximize the reward (or minimize the negative reward in this case) over time. Empirically chosen coefficients (α, β, and ω) allow the RL-based controller to maintain the voltage within the given limits while avoiding complete depletion or overcharging of batteries (further discussed in Section III). The proposed control scheme also presents the interest of being versatile in terms of control application. It would only require adapting the reward function with any other considered objective for grid management operations

E. Two-Stage Offline and Online Training

A two-stage training scheme is proposed to cope with impedance deviations during operation (online phase) and minimize (thanks to the offline phase) the occurrence of dangerous regimes in the electrical grid during the online training (as presented in Fig. 3). The idea is to first train the algorithm in an offline mode on a target grid model that uses nominal impedance values from the manufacturer. Using data on consumption and local production over a full year, the algorithm can run simulations over multiple 'simulated' years, choosing different actions for the same states. Control performances are assessed on test simulations with consumption and local generation profiles for a second year.

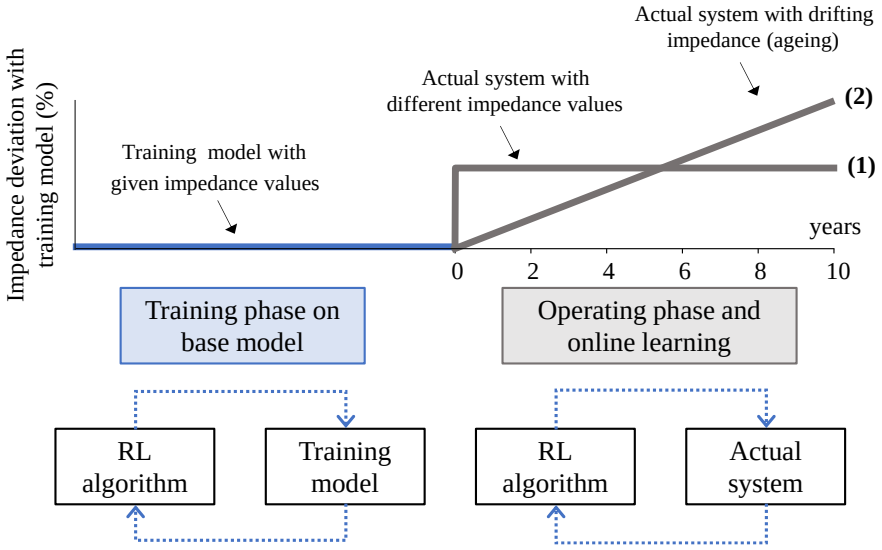

Fig. 3: Offline and Online training for impedance uncertainties mitigation. Step variation of impedances in scenario (1) and gradual variation in scenario (2)

In the second step, the pre-trained offline algorithm is connected to the actualized electrical grid where two scenarios are considered (Fig. 3):

(1) – a significant difference between the expected impedance values (used during offline training) and the real ones (abrupt deviation). This scenario simulates a lack of knowledge, a priori, of the system.

(2) – the impedance values are close to the expected ones, but continually drift from the nominal ones during the online phase due to exploitation (continual deviation). This simulates potential aging effects.

Preliminary offline training of the algorithm prevents actions that could lead to dangerous voltage values in the actual operational phase. By exchanging actions and observations with the grid, the algorithm can gradually learn changing impedances, adjust its policy, and produce more accurate optimization results compared to the pre-trained offline solution (refer to Section IV.C).

III. TUNING THE AI ALGORITHMS

A. Performance Metric

To assess the performance of all investigated control strategies, a voltage performance index (VPI) is defined in (7). It calculates how many times, on average, the algorithm violates voltage constraints for each node in the grid.

$$VPI = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T}\delta_t^n}{|N|\cdot|T|}\cdot 100\% \qquad \begin{cases}\delta_t^n = 0 \; if \; 0.95 \le V_t^n \le 1.05 \\ \delta_t^n = 1 \; otherwise\end{cases} \tag{7}$$

As previously mentioned, one-year profiles of generation and consumption are used for the offline training, and another year's profiles for testing. The VPI over the test year without any control is significant (19.5 %) with both under and overvoltages. Due to optimization time for the benchmark algorithm (around 5 hours for a full-year profile on 15 cores, 96Gb RAM machine), only four representative weeks of the year

(one per season) are used to calculate the yearly VPI. This allowed significant reducing computational times (notably considering the need for multiple runs) while still providing representative results (VPI is 18.7 % over the four representative weeks rather than 19.5 % for the whole year).

B. Sensitivity Analyses

B.1. Training Settings

RL algorithms are very sensitive to hyperparameters. The speed of convergence with the same initial set of random transitions can differ significantly depending on the initialization and algorithm settings. Similarly, the same initialization and hyperparameters cannot guarantee similar convergence or similar (suboptimal) solutions due to the stochastic nature of the training algorithms. However, a replay buffer with a sufficient number of transition samples partially mitigates this stochastic problem. In combination with correctly selected hyperparameters and fixed initialization, such replay buffer allows obtaining close to optimum control strategies for each new training (i.e. for independent runs). A sensitivity analysis over 1500 runs was performed to choose TD3PG and PPO best hyperparameters. Values were selected to provide the highest performance during the offline learning stage (i.e., the lowest VPI) for the considered environment.

The load and generation profiles have a resolution of 30 min, i.e., 48 measurements per day. Thus, for simplicity, the size of each training epoch (i.e., time horizon) was set to 48 steps, where each step corresponds to one interaction between the controller and its environment. PPO uses a stochastic policy, where each action is taken from a generated probability distribution. TD3PG, on the contrary, uses a deterministic policy. For this reason, at the beginning of the training, TD3PG performs a search space exploration by executing a given number of random actions. In our case, 24000 random interactions (500 epochs) showed the best results in terms of compromise between suboptimal solutions and excessive exploration. The total number of epochs for the training was set at 4000 with an update every 8 steps for TD3PG and 30,000 with 20 updates at the end of each epoch for PPO. Activation functions of Tahn and ReLU were chosen for the Actor and Critic network hidden layers respectively. Adam optimizer [25] was used to implement the stochastic gradient descent to train these networks. The main sensitivity analyses for TD3PG and the subsequent hyperparameters selection for PPO and TD3PG are presented in the following subsections.

B.2. Impact of neural network structure

At first, the impact of the model's structures and the number of neurons (for both, "Actor" and "Critic" networks) was assessed. To ensure the same initial conditions for comparison purposes, a common random seed (initial configuration) was chosen for the investigated neural structures. That allowed having the same initialization of all network weights and the same direction of the gradient at the beginning of the training. The results displayed in Fig. 4*a* show the VPI results over the training epochs for the four test weeks.

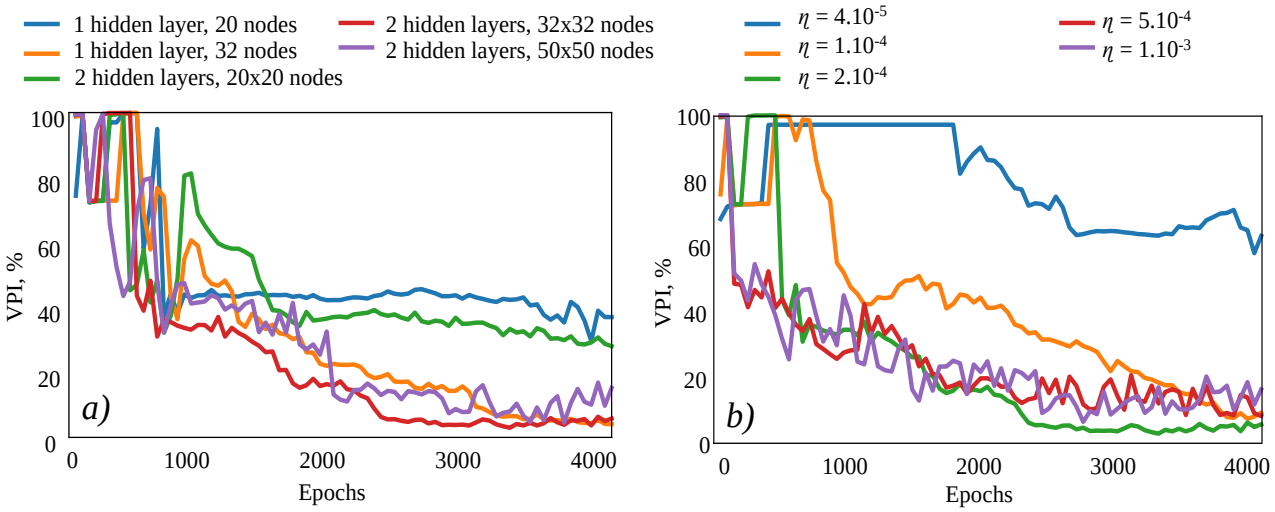

Fig. 4: Learning curves of TD3PG – *a)* for different numbers of nodes and layers in the neural networks – *b)* for different learning rates

The structure with 20 neurons and one hidden layer displays the worst results with a VPI still above 30 % after 4000 epochs. Neural networks with one hidden layer of 32 neurons achieve 4.1 % VPI only at the end of the training. It is better than the structure with two hidden layers of 50 neurons each but still worse than the 32x32 structure, which shows 4.5 % and 2.9 % of VPI respectively. It can be concluded, that one hidden layer is not enough to capture all dependencies, and a structure of 50x50 neurons needs too many samples to converge to a better solution.

B.3. Impact of learning rate

Fig. 4*b* illustrates the results of a sensitivity analysis of the convergence to the learning rate η for both "Actor" and "Critic" neural networks. Large learning rates (of $5e^{-4}$ and $1e^{-3}$) allow faster convergence at the beginning of the training process. However, after 500 epochs smaller learning rate $2e^{-4}$ starts monotonic improvement, and after 2000 epochs reach a VPI of 2.8%. Indeed, large learning rates take too large steps at each update and may miss optimum solutions. On the contrary, with too small learning rate values, the networks can get stuck into less relevant local optimums. Thus, a learning rate of $2e^{-4}$ presents a good trade-off for the considered voltage control tasks.

B.4. Impact of batch size and replay buffer size

The dependency of the training results to batch (i.e., the number of samples used for each update of neural networks) and to the replay buffer sizes is presented in Fig. 5a with training executed over 4000 epochs (no improvement with longer training). Note that, for each couple (replay buffer size, batch size) the best VPI obtained along the training was kept, which was not necessarily the value obtained at the end of the training.

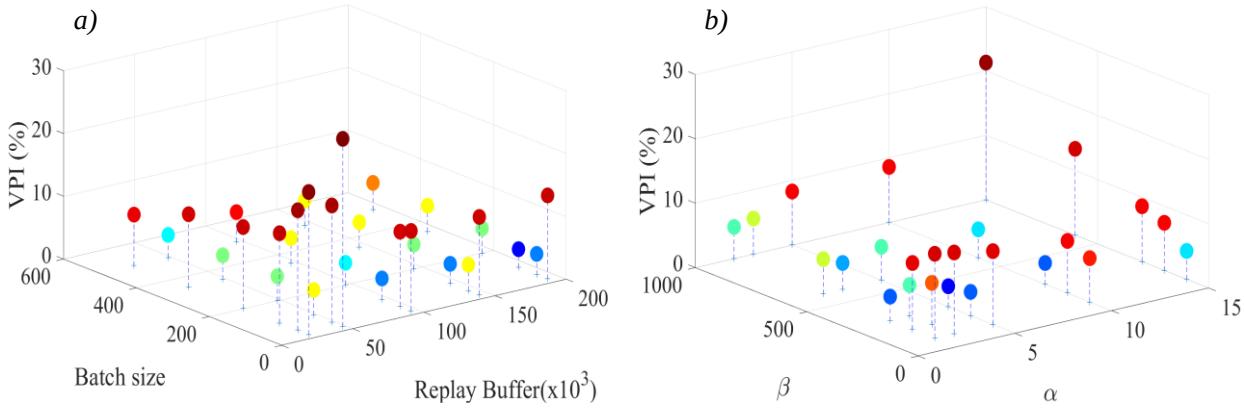

Fig. 5: VPI with TD3PG – *a*) for different batch and replay buffer sizes– *b)* for different parameters of the reward function expressed in (6)

The size of the replay buffer directly affects the training process. Small replay buffers are the most suitable for a changing environment, allowing the use of only the more recent interaction results for the training. This also helps converge faster to the current local optimum. Bigger replay buffers, on the contrary, allow better generalization and smooths changes in the data distribution. An incorrectly sized replay buffer will slow down the learning of the correct value function, thus justifying its analysis [23].

According to [24], small batch sizes tend to converge to flat local optimums that vary only slightly within a small neighborhood of this optimum, whereas large batch sizes converge to sharp local optimums. Additionally, small batch size training finds minimizers further away from the initial weights, compared to large batch size training. As seen in Fig. 5a, for a batch size of 500, the best size of the replay buffer is 48000. However, as the batch size decreases, the best replay buffer size grows. Thereby the batch size of 100 with a replay buffer capable of saving all interactions during 4000 epochs showed the lowest VPI and was selected for the rest of the work.

B.5. Impact of reward coefficients

Finally, the dependency of the training results on the coefficients of the reward function expressed in (6), α, and β is investigated. Not only is the ratio between α and β important, but also their absolute value because the activation function (Tanh) acts differently for different ranges. This assertion is confirmed by results presented in Fig. 5b. Three points, [α=5, β=600], [α=10, β=300], and [α=15, β=200] have the same ratio, but the VPI is equal to 5.2%, 3.3%, and 7.4% respectively. When $1/\beta$ is too low, the algorithm neglects the SOC value. That leads to situations where its batteries are fully charged or discharged, and cannot be used at the right time. Similarly, when α is too low the algorithm focuses mainly on keeping the SOC close to 0.5 while ignoring voltage deviations. In that case, flexibilities are underused.

B.6. Validation over the four test weeks

A set of preliminary tests was performed to validate the representativeness of the four chosen weeks. Four models were extracted from the 1500 runs (two with low VPI and two with high VPI) and executed on the whole test year. Their VPI was calculated and then compared with the corresponding VPI over the four test weeks. The results, displayed in TABLE I, show very close performances of the model in both periods. The conservativeness of the results over the four representative weeks indicates that if the controller is less performant over the whole year it is also less performant over the representative period.

TABLE I COMPARISON OF VPI FOR 4 WEEKS AND 1 YEAR				
Model N°	1	2	3	4
4 weeks	2.9%	3.1%	19,1%	18.6%
1 year	3.6%	3.5%	19.0%	19.1%

B.7. Parameters Values

TABLE II gives the best values considered for all the hyperparameters of the algorithms. Note that the sensitivity analysis considered all those hyperparameters. However, for the sake of clarity and conciseness, only some of the sensitivities of selected parameters were presented in the previous subsection. To build them, all hyperparameters that were not under the test were also fixed according to TABLE II.

TABLE II

Parameter	Value	Parameter	Value
MAIN PARAMETERS OF THE TD3PG ALGORITHM			
Replay size	192000	Policy learning rate	$2e^{-4}$
Batch size	100	Q-function learning rate	$2e^{-4}$
The noise of the Actor model network	0.1	The noise of the Actor target network	0.2
Nb. of random actions for initial training	24000	Nb. of steps between networks update	8
SPECIFIC PARAMETERS FOR THE PPO ALGORITHM			
Nb. of gradient descent steps for policy/value function	20/20	Nb. of steps between networks update	96
Clip ratio	0.1	KL-divergence	0.01
Learning rate for policy optimizer	$1e^{-4}$	Learning rate for value function	$2e^{-4}$
COMMON PARAMETERS OF THE ALGORITHMS			
Hidden layers	32x32	α	5
ω	8	β	300
Activation function (Actor network)	Tanh	Activation function (Critic network)	ReLU

IV. MAIN RESULTS

A. *Control Performances for Constant Impedances*

The off-policy RL type of TD3PG allows multiple reuses of transition samples [s_t; s_{t+1}; a_t; r_t]. For this reason, TD3PG presents a faster convergence, reaching a VPI of 2.8% after 3600 epochs, while PPO needs 9000 epochs to achieve the same accuracy (Fig. 6). However, TD3PG has problems with further convergence due to the "Deadly Triad" phenomenon, which is common for off-policy algorithms with function approximations as neural networks [27]. This phenomenon occurs if s_t and s_{t+1} are sufficiently similar, which is the case for the voltage control over 30 minutes. As a result, the value function suffers from instabilities and/or divergence. PPO, on the contrary, due to its constraints on the newly learned policy avoids divergence and allows almost monotonic slow improvement, reaching a VPI of 1.85% after around 20,000 epochs for

known impedances [18]. In addition, its results for diverged impedances can be further improved by implementing an online learning stage, as presented in Section IV.C.

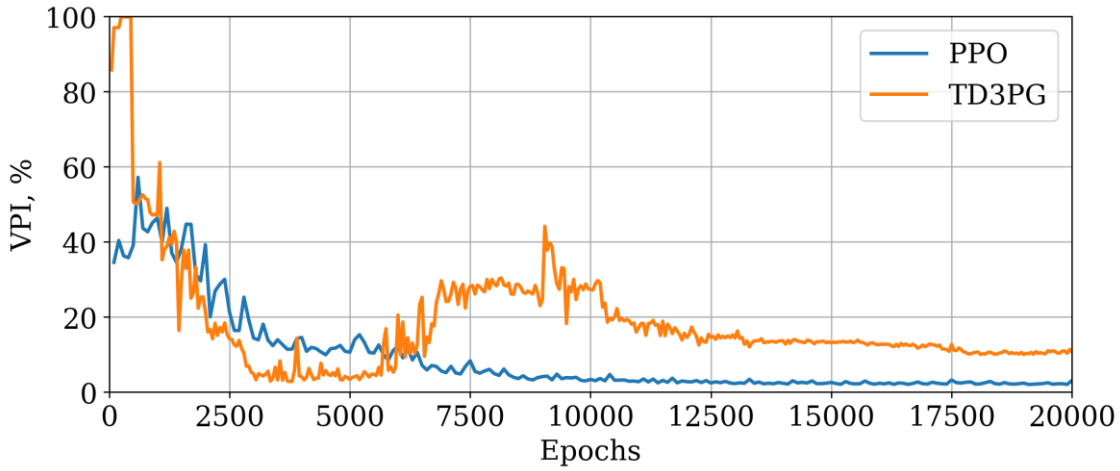

Fig. 6: Learning curves of TD3PG and PPO for offline training

Due to higher final accuracy, the voltage control results in this section are presented for the offline trained PPO algorithm. Impedances of the grid are fixed, i.e., the values over the test year are equal to the values of the simulated environment in the training year. For illustration, the voltage profile at node 3 is displayed in Fig. 7. For visibility purposes, only one representative day of each season is plotted. The voltage profile before optimization deviates from 0.87 p.u. to 1.12 p.u during the year. Undervoltage occurs due to high consumption in autumn to spring while overvoltages can be seen mostly during summer days due to greater levels of solar generation. However, thanks to controlled flexibility at the same node, the voltage is mostly maintained within the given limits once the RL-based control is applied.

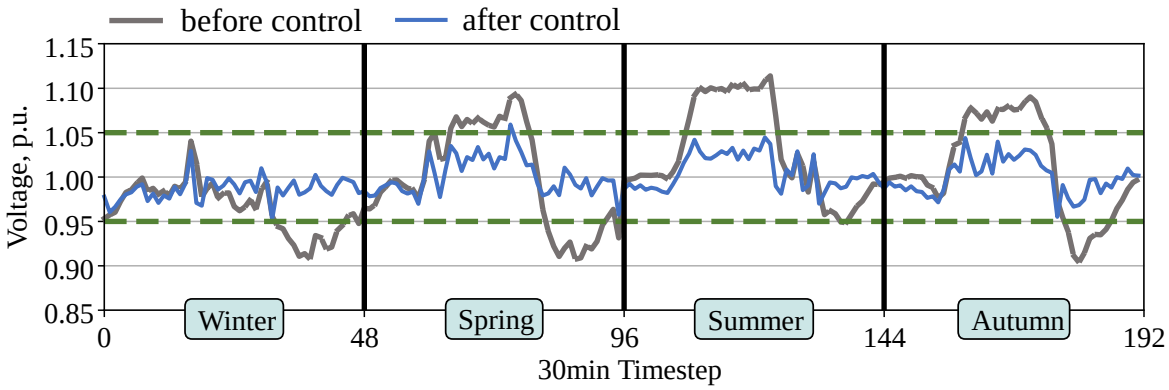

Fig. 7: Voltage profile at node 3 (grid displayed in Fig. 1) for four representative days of the test year with constant impedances and offline control (PPO)

Fig. 8 displays the voltage profiles at bus 9. Similar to the previous results for bus 3, the RL control mostly handles the problem of undervoltage thanks to flexibilities from neighboring nodes. However, overvoltage cannot be as efficiently mitigated as undervoltage. This is explained by the lack of flexibility in this node (i.e., no storage unit connected). Moreover, the controller cannot lower the voltage of node 9 using flexibilities in nodes 8 and 10, because it would lead to a voltage drop in these nodes significantly below 0.95 p.u.

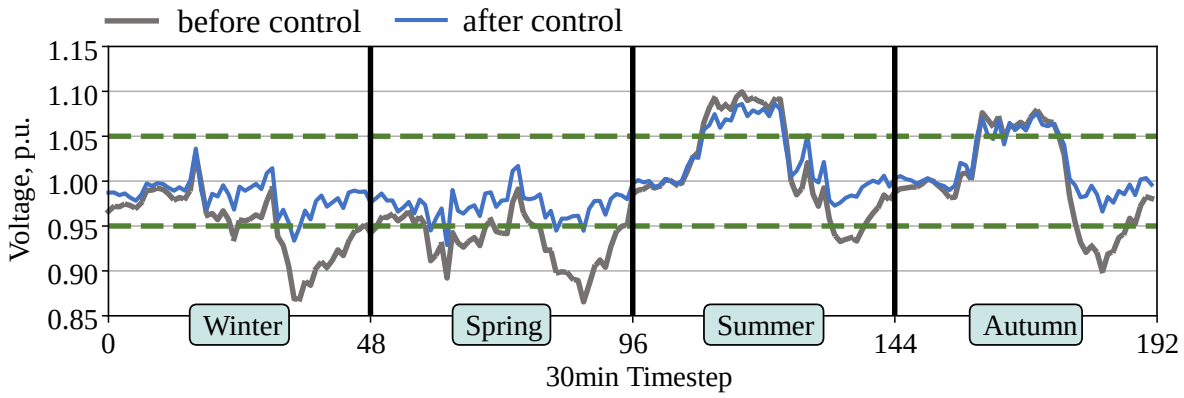

Fig. 8: Voltage profile at node 9 (grid displayed in Fig. 1) for four representative days of the test year with constant impedances and offline control (PPO)

Fig. 9 displays the injection of power from inverters 1 to 4, connected at nodes 3, 6, 8, and 10 respectively. It can be seen that injections from inverters 3 and 4 are higher than from inverters 1 and 2. This can be explained by the fact that they increase the voltage also in node 11, which does not present flexibility. In short, the contribution of each flexibility depends on its location relative to the consumption/generation profile dynamics and impedances of the neighboring lines.

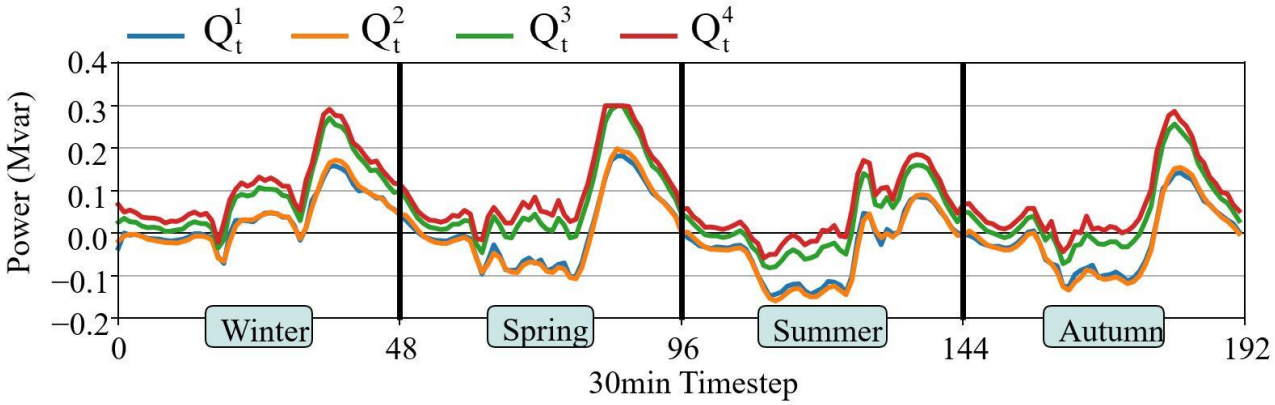

Fig. 9: Inverter powers for four representative days of the test year with constant impedances and offline control (PPO)

The state of charge profiles of the batteries are shown for the four representative days in Fig. 10. The SOC displays daily variations in the range of 0.12 (for battery 3) to 0.74 (for battery 4). Batteries are then never completely charged or emptied, thus preserving their availability, thanks to the considered reward function that induces SOC deviation around 0.5.

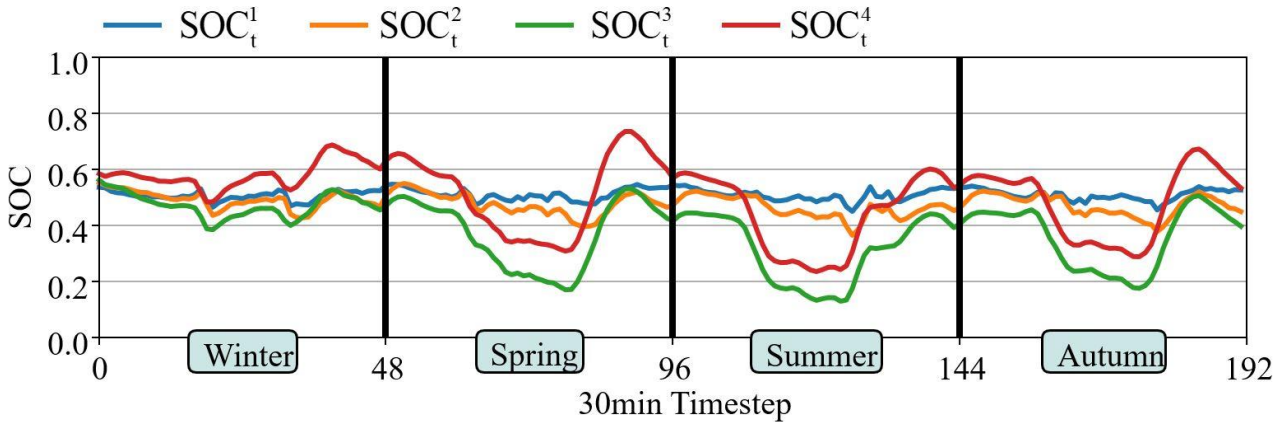

Fig. 10: State of charge of each battery for four representative days of the test year with constant impedances and offline control (PPO)

B. Impedance uncertainties mitigation

AI-based RL algorithms are relevant compared to traditional optimization-based controls when facing uncertainties. In this paper, grid uncertainties as investigated, focusing more precisely on the values of the impedances of the lines. To study the impact of these uncertainties on the performance of the voltage control, the VPI was calculated for a set of abrupt standard deviations of the lines' impedances ranging from 5 % to 40 % (scenario (1) in Fig. 3). Parameters deviations in the grid were modeled by changing the impedances values in the grid simulation tool, which allows testing multiple scenarios of different deviations over the years, that is not feasible with a real electrical grid. Different runs were executed over T=1344 timesteps corresponding to the four test weeks while changing the environment parameters (line impedances $Z_{ij}^{(k)}$ at the k^{th} sample test) around the original values with a normal law and standard deviation of σ% (refer to Fig. 11 for the synoptic). For each deviation, the VPI is calculated as the average VPI over K=30 independent samples.

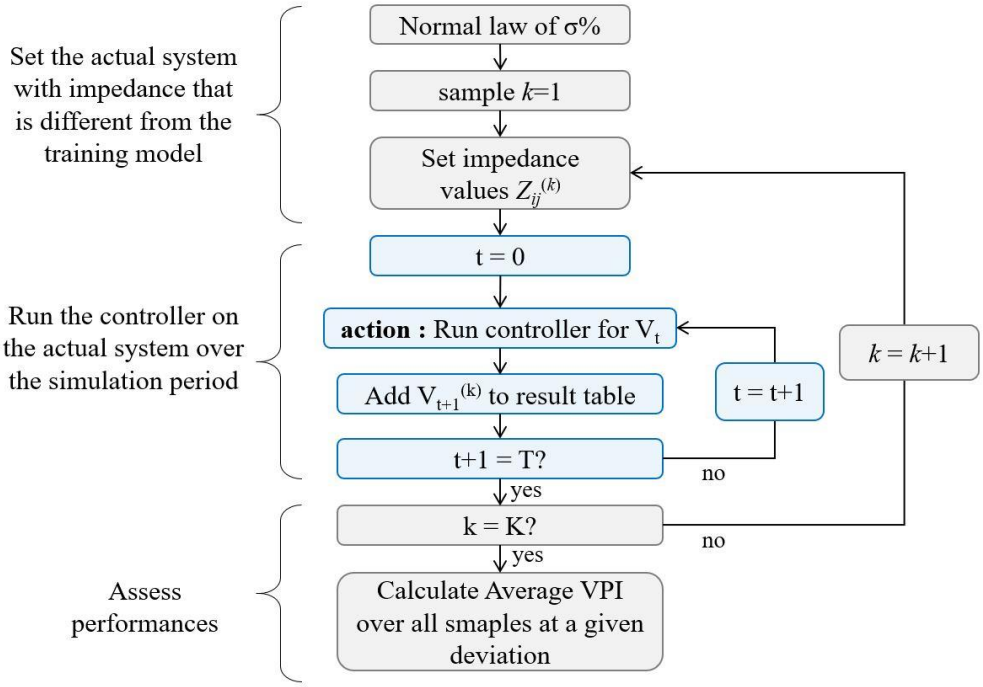

Fig. 11: Simulation framework to estimate the impact of impedances uncertainties

To compare the optimization and the AI-based controllers, initial measurements of load or impedances were given to the optimization-based control algorithm. Its control outputs were then sent to the system with distorted loads or impedances. Similarly, the RL algorithms were trained on the model of the grid with the same measurements, but its control setpoints were applied to the grid with deviated values. Remind that the presented RL algorithms are completely immune to any load errors because they rely only on measurements of voltage and flexibilities powers in the proposed implementation. On the other hand, the optimization-based control, on the contrary, directly depends on load values.

Note that on an actual distribution grid it is unlikely to have access to voltage measurements at all the nodes unless all the points of delivery are equipped with smart meters. In such a case, the RL-based controllers shall be fed with voltage values that could be outputted from a state estimator. Thus, the implementation of the RL-based controllers shall not be the best solution. However, the final performances will be somewhat limited by the accuracy of the state estimation as both reward and controls are directly computed from the voltage values. Also, without accurate voltage measurements, it would not be straightforward to correctly asses the performances of any control strategy.

The VPI comparison of TD3PG, PPO (both in offline training), and optimization-based controls for different impedance deviations (abrupt deviations, scenario (1) in Fig. 3) is presented in Fig. 12. The “best possible VPI” on the same plot represents the optimization-based control results where all deviated impedances are perfectly known.

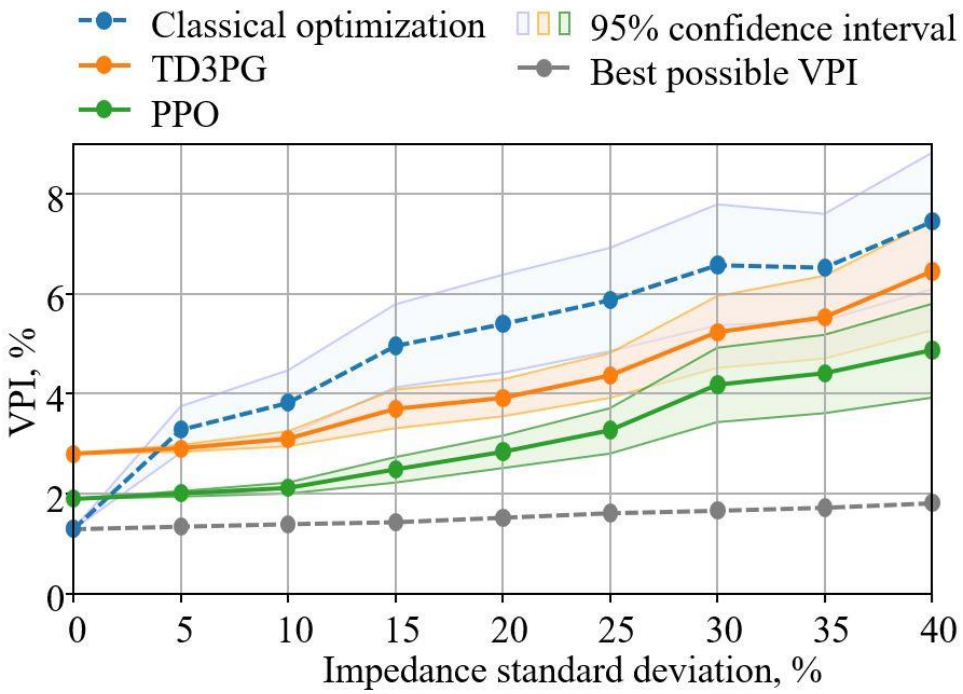

Fig. 12: VPI of PPO, TD3PG, and optimization-based algorithms in the case of impedance step variations – average values and corresponding confidence intervals of 95 %. The "best possible VPI" represents the optimization where all deviated impedances are known (baseline reference)

The optimization-based control performs better when there is no impedance deviation, i.e. when impedances are completely known (same impedance values in the model-based controller and the environment). The average VPI is 1.29 % vs 1.85% for PPO and 2.8 % for TD3PG. However, when the actual system does not correspond to the one considered during the training (i.e., starting from 5 % of impedance deviation), the RL algorithms perform better and the difference tends to maintain with the growth of standard deviation. Such results are explained by the fact that the proposed RL algorithms do not rely on impedance data directly but on voltage measurements, which partially depend on the impedance values. It can also be noted that PPO outperforms TD3PG for all deviations, mainly due to a more successful offline training, which allowed us to get lower VPI for known impedances.

It is worth mentioning that the implementation of the optimization is conducted with some advantageous conditions compared to TD3PG and PPO. Indeed, the optimization relies on perfect predictions of all load and generation for the whole considered period. RL algorithms, on the contrary, make their decisions for the next timestep based on the data from the previous timestep only. Thus, TD3PG and PPO implicitly embed load and PV generation predictors for the next timestep, and this uncertainty contributes to the final control error. This error can be reduced by decreasing the timestep.

Moreover, the proposed algorithms significantly outperform the optimization in terms of speed, once trained. The pretrained algorithms take less than a minute to calculate a full-year control of the flexibilities when almost 5 hours are needed for the optimization in similar conditions (around 300 times longer). Thus, the RL controllers require much fewer computational resources to run in the operational phase, assuming that the training is not repeated too often (but would not require to be conducted in real-time, thus allowing for flexibility).

C. Online training and drifting line impedances

To demonstrate the effectiveness of the proposed two-stage method, an online training of the PPO algorithm with continual σ deviation from 0% to 40% is simulated over 10 years (scenario (2) in Fig. 3). The average monthly VPI with the optimization-based controller, PPO trained only offline and both offline/online is calculated according to equation (7) with T=1460. The results are presented in Fig. 13, where "best possible values" represent the optimization-based control results if all impedances were perfectly known.

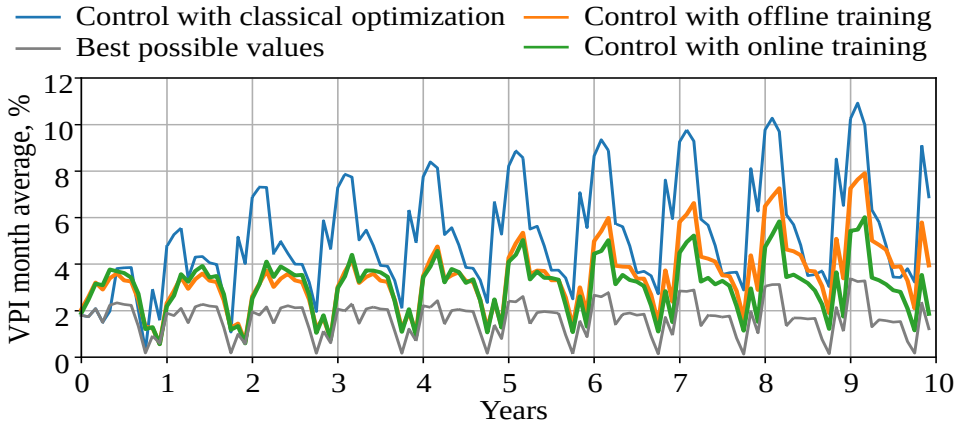

Fig. 13: Average monthly VPI over 10 years with continual drifting of impedances from 0% to 40%. The "best possible values" represent the optimization where all deviated impedances are known (baseline reference)

Over the first four years, online learning shows slightly worse results than offline one. This happens due to daily updates of the policy, which converge (based on last-day data only) to a local optimum. However, after four years the deviation becomes significant, and online learning is noticeably more robust to this variation of impedance than the PPO algorithm with offline training. The PPO with online learning outperforms the optimization-based control already after a few months and the gap between their performance is growing with time.

For illustration, the last year's results are presented in Fig. 14 with weekly average values of VPI. As can be seen, the online training shows better VPI during the whole year (the average yearly VPI is 3.43 %), compared to a case where the initial controller (issued from the offline training) remains unchanged (in that case, the VPI is of 5.33 %). The two-stage training solution significantly outperforms the more traditional optimization (whose VPI is of 6.99 %). Thus, the final VPI reached by the PPO with online training is more than 50 % smaller than the one obtained with the optimization, which is a quantification of its robustness. However, this is still far from the best possible value of 1.82 %, which can be explained by uncertainties in consumption and local generation that cannot be perfectly envisaged by the built-in PPO predictor.

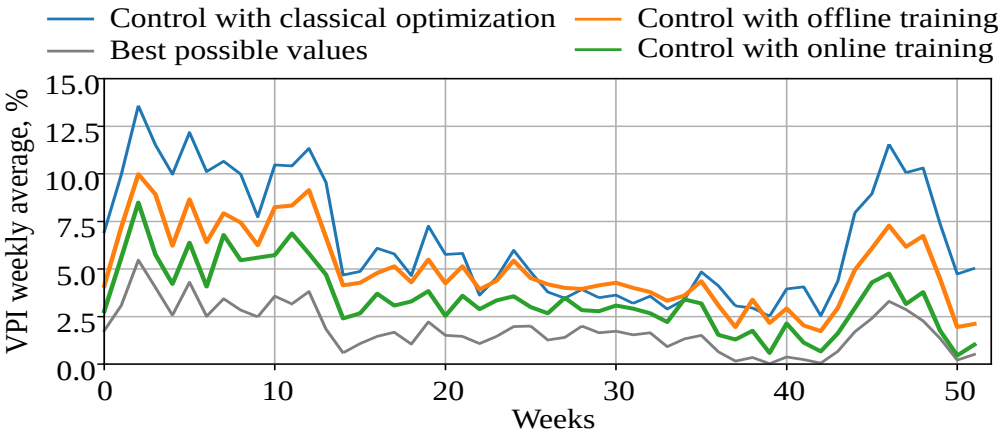

Fig. 14: Average weekly VPI for optimization-based, offline-trained PPO, and online-trained PPO algorithms for the last year of the 10 years simulation with continual drifting of impedances

D. Scalability Tests on a 55-bus System

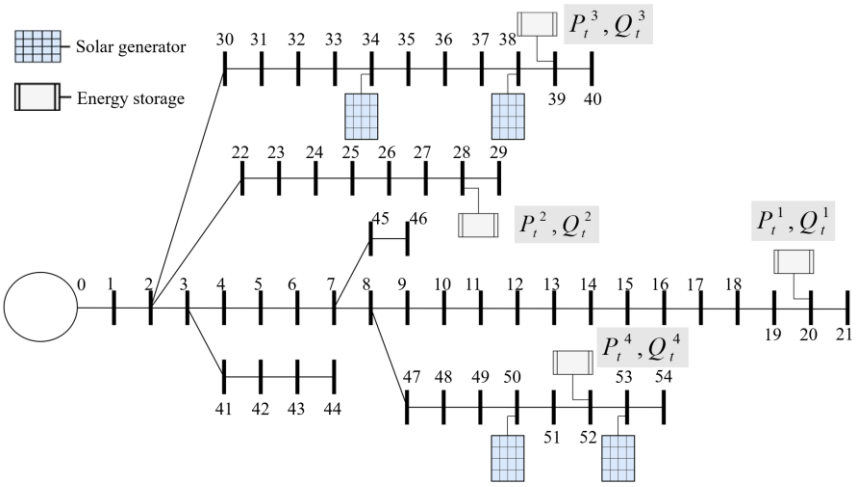

Fig. 15: Considered 55-bus system

The scalability of the proposed controller is finally tested on a 55-bus system that displays four flexibility to control (Fig. 15). At first no impedance uncertainties are considered. TABLE III then displays the results of the controller trained offline only and compares them with the baseline scenario (i.e. no control) and the performance of the model-based approach. Similarly, to section IV, those VPI results are given over a 4 weeks test period. With full knowledge of line impedance, the optimization controller returns the best theoretical VPI at 6.0 %, largely improved from a baseline case with no storage. Both PPO and TD3PG-trained controllers display similar performances with VPI of 8.0 % and 7.5 % respectively.

TABLE III VPI RESULTS ON THE 55-BUS SYSTEM				
	Baseline no control	Optimization-based Control	RL Control **PPO**	RL Control **TD3PG**
1 year	20.9 %	6.0 %	8.0 %	7.5 %

Like in the previous test on the simpler use case, the superiority of the proposed RL-based controller is highlighted when considering uncertain impedances. **Fig. 16** summarizes the results of the controller when applied to a system that displays different line parameters compared to the training phase (i.e. offline training only and run over the four weeks test period). As previously, the performances in terms of VPI tend to degrade with increasing digress of uncertainties on the impedances. Also, both RL controllers outperform significantly the reference model-based method as soon as the line parameters are not perfectly known (error ≥ 5 %).

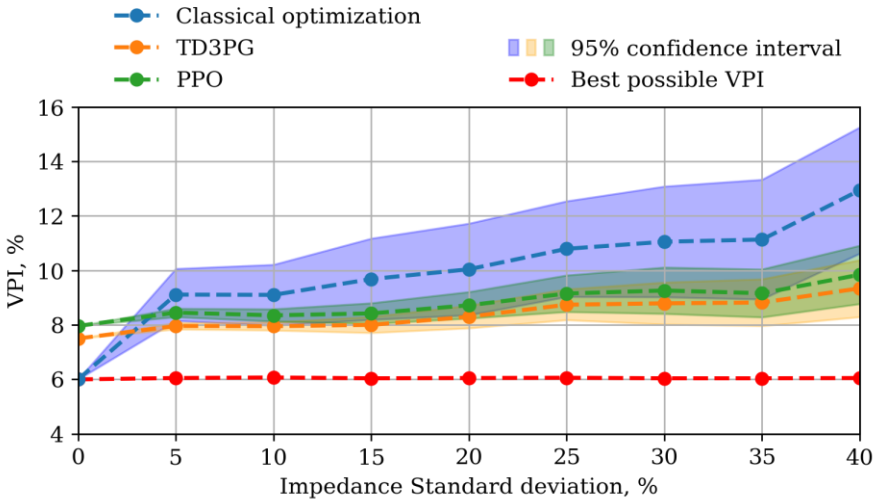

Fig. 16: ***VPI of offline trained TD3PG, PPO, and optimization-based algorithms for impedance step variations of 55-bus grid***

In the last validation test, the online is applied with a PPO-based controller that adapts itself to a network with impedance deviations of 40% compared to the values used in the offline training. VPI of the controller trained offline only is 9.9% (**Fig. 16**). As displayed in **Fig. 17**, with daily updates of the controller, this VPI is reduced to 8 % after 7 years, and then reaches the value obtained with the PPO when no uncertainties are considered (TABLE III). During the first four years, the adaptive controller performs slightly worse, because it needs time to adapt to new impedances. Fluctuations occur due to the daily update of the controller, which leads to the convergence of the algorithm to the local optimum (the best control for a given day) and may temporarily worsen the control for subsequent days. Finally, note that applying the RL-based controllers on the larger network required tuning the algorithm's hyperparameters in the offline phase similar to what is presented in section III. No details are given here for the sake of clarity.

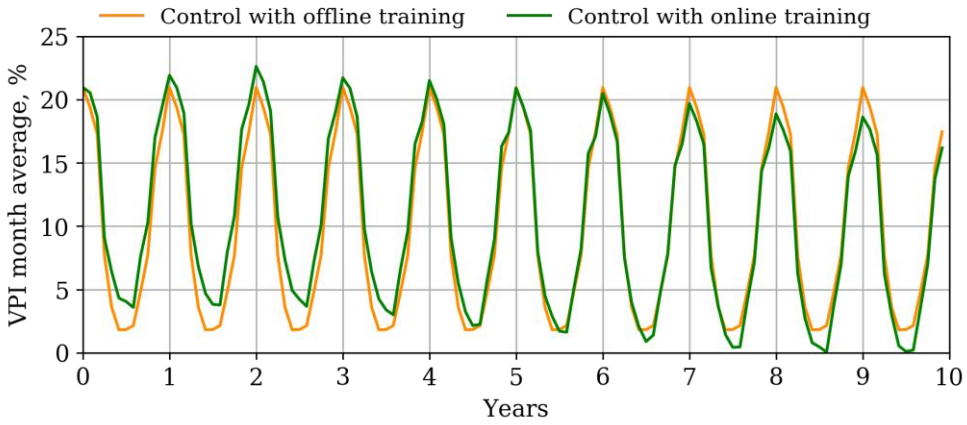

Fig. 17: ***Average monthly VPI over 10 years with abrupt impedance deviation***

V. CONCLUSION

An AI-based reinforcement learning method using the Twin Delayed Deep Deterministic Policy Gradient (TD3PG) and Proximal Policy Optimization (PPO) was proposed to control voltage profiles under load and impedance uncertainties in a distribution grid. A Two-stage training strategy was proposed that consisted of offline and online learning phases, compared to a more traditional optimization-based control formulated as a multi-objective optimal power flow problem with a second-order conic relaxation to account for the grid model.

A sensibility analysis of the main hyperparameters was presented. Selected values allowed the decrease of voltage violations from 19.5% to 1.85% of the time for the case where impedances are constant. Moreover, the algorithm implicitly includes a predictor part, by computing the next action based only on voltage measurements obtained 30 min before, i.e., it does not need external forecast tools such as conventional model predictive approaches.

A comparison of the two proposed algorithms with the optimization-based approach showed that the AI-based voltage control is strictly more robust to line impedance uncertainty after 5 % of deviations concerning theoretical values. In addition, the proposed two-stage control outperforms, in terms of accuracy, the algorithm with only offline training by 55% in the case of impedance uncertainty with a standard deviation of 40% after simulating 10 years of continual impedance deviation.

The large time resolution considered in this study (30 minutes) directly derives from the available time series in terms of load and solar generation profiles. Similar to the model-based controller, the RL works in a steady-state mode and could cope without any further modification with finer time resolutions (down to the minute scale). In such a case and as mentioned in the paper, the RL-based controller significantly outperforms optimization approaches as the running time is negligible (below the second) on the training is done. One other advantage of running at finer resolutions is that more samples can be available for the same period. This can then reduce the length of period for the training (e.g. weeks instead of a year) and speed up the online tuning (e.g. hourly update instead of daily updates). Finally, the proposed control scheme also presents the interest of being versatile in terms of control application. It would only require adapting the reward function

with any other considered objective for grid management operations. Future work will focus on evaluating the scalability of the proposed solution, allowing us to verify the feasibility of the proposed method not only at the feeder level but also for a larger distribution grid.

VI. ACKNOWLEDGMENT

This work is funded by MIAI@Grenoble Alpes (ANR-19-P3IA-0003) and the Enedis Industrial Chair on Smart Grids.

REFERENCES

[1] J. Morin, F. Colas, J. Dieulot, S. Grenard, X. Guillaud, "Embedding OLTC nonlinearities in predictive Volt Var control for active distribution networks," *Electr. Power Syst. Res.* 143, pp. 225–234, 2017.

[2] M.H.K. Tushar, C. Assi, "Volt-VAR control through joint optimization of capacitor bank switching, renewable energy, and home appliances," *IEEE Trans. Smart Grid* 9 (5), pp. 4077–4086, Sept. 2018.

[3] F.U. Nazir, B.C. Pal, R.A. Jabr, "A two-stage chance-constrained Volt/VAR control scheme for active distribution networks with nodal power uncertainties," *IEEE Trans. Power Syst.* 34 (1), pp. 314–325, Jan. 2019.

[4] V. Sarfi and H. Livani, ''Optimal Volt/VAR control in distribution systems with prosumer DERs,'' *Electr. Power Syst. Res.*, vol. 188, Nov. 2020, Art. no. 106520, doi: 10.1016/j.epsr.2020.106520.

[5] Y. P. Agalgaonkar, B. C. Pal and R. A. Jabr, "Stochastic Distribution System Operation Considering Voltage Regulation Risks in the Presence of PV Generation," *IEEE Trans. on Sustainable Energy*, vol. 6, no. 4, pp. 1315-1324, Oct. 2015.

[6] N.Chettibi, A.Massi Pavan, A.Mellita, A.J.Forsyth, R.Todd, "Real-time prediction of grid voltage and frequency using artificial neural networks: An experimental validation," *Sustainable Energy, Grids and Networks*, September 2021, DOI: 10.1016/j.segan.2021.100502.

[7] N. Kumar, B. Singh, B. K. Panigrahi, "PNKLMF-Based Neural network control and learning-based HC MPPT technique for multiobjective grid integrated solar PV based distributed generating system," *IEEE Trans. Ind. Inf.*, 15, pp. 3732–3742, June 2019.

[8] S. Li, Y. Sun, M. Ramezani, and Y. Xiao, "Artificial neural networks for Volt/VAR control of DER inverters at the grid edge," *IEEE Trans. on Smart Grid*, vol. 10, no. 5, pp. 5564–5573, Sep. 2019.

[9] Nakabi, T.A.; Toivanen, P. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, 2021, 25, doi:10.1016/j.segan.2020.100413

[10] C.H. Tai, J.H. Hong, D.Y. Hong, L.C. Fu, "A real-time demand-side management system considering user preference with adaptive deep Q learning in home area network," *Sustainable Energy, Grids and Networks*, 2022, 29, DOI: 10.1016/j.segan.2021.100572

[11] S. Wang, J. Duan, D. Shi, C. Xu, H. Li, R. Diao, and Z. Wang, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Trans. Power Syst.*, Nov. 2020.

[12] D. Cao, J. Zhao, W. Hu, N. Yu, F. Ding, Q. Huang, and Z. Chen, "Deep reinforcement learning enabled physical-model-free two-timescale voltage control method for active distribution systems," *IEEE Trans. on Smart Grid*, 2021

[13] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.* 35, pp. 814–817, Jan. 2020.

[14] D. Cao, J. Zhao, W. Hu, F. Ding, N. Yu, Q. Huang, Z. Chen. "Model-free voltage control of active distribution system with PVs using surrogate model-based deep reinforcement learning," *Applied Energy*. v. 306, Part A, January 2022.

[15] J-F. Toubeau, B. B. Zad, M. Hupez, Z. De Grève, F. Vallée, "Deep reinforcement learning-based voltage control to deal with model uncertainties in distribution networks," *Energies*, vol. 13, no. 15, 2020.

[16] X. Sun and J. Qiu, "A Customized Voltage Control Strategy for Electric Vehicles in Distribution Networks With Reinforcement Learning Method," in IEEE Transactions on Industrial Informatics, vol. 17, no. 10, pp. 6852-6863, Oct. 2021, doi: 10.1109/TII.2021.3050039.

[17] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," arXiv preprint arXiv:1802.09477, 2018.

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

[19] Smart meter energy consumption data in London households. [Online]. Available: http://data.Jondon.gov.uk/dataset/smartmeter-energy-usedata-in-Iondon-households.

[20] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The national solar radiation database (NSRDB)," *Renewable Sustain. Energy Rev.*, vol. 89, pp. 51–60, 2018.

[21] M. A. Putratama, R. Rigo-Mariani, V. Debusschere, Y. Besanger, "Parameter Tuning for LV Centralized and Distributed Voltage Control with high PV production," Powertech conference 2021, Madrid, Spain (Online), July 2021. DOI: 10.1109/PowerTech46648.2021.9494802

[22] R. S. Sutton and A. G. Barto, "Reinforcement learning," "Temporal-Difference Learning," in *Reinforcement Learning: an Introduction*. Second edition. Cambridge, MA: The MIT Press, 2018.

[23] R. Liu and J. Zou, "The effects of memory replay in reinforcement learning," 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), October 2018.

[24] N.S. Keskar, J. Nocedal, D. Mudigere, M. Smelyanskiy, and P.T.P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," ICLR 2017.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014

[26] L. P. Kaelbling, M. L. Littman, A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artif. Intelligence Research* 4, pp. 237-285, May 1996.

[27] H. Van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayilm “Deep reinforcement learning and the deadly triad,”. arXiv preprint arXiv:1812.02648, 2018