Community energy storage operation via reinforcement learning with eligibility traces

Salazar Duque, Edgar Mauricio; Giraldo, Juan S.; Vergara, Pedro P.; Nguyen, Phuong; van der Molen, Anne; Slootweg, Han

DOI
10.1016/j.epsr.2022.108515

Publication date
2022

Document Version
Final published version

Published in
Electric Power Systems Research

Citation (APA)
Salazar Duque, E. M., Giraldo, J. S., Vergara, P. P., Nguyen, P., van der Molen, A., & Slootweg, H. (2022). Community energy storage operation via reinforcement learning with eligibility traces. *Electric Power Systems Research*, *212*, Article 108515. https://doi.org/10.1016/j.epsr.2022.108515

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Community energy storage operation via reinforcement learning with eligibility traces

Edgar Mauricio Salazar Duque [a,*], Juan S. Giraldo [b,1], Pedro P. Vergara [c], Phuong Nguyen [a], Anne van der Molen [a], Han Slootweg [a]

[a] *Eindhoven University of Technology, The Netherlands*
[b] *University of Twente, The Netherlands*
[c] *Delft University of Technology, The Netherlands*

ARTICLE INFO

Keywords:
Battery management
Reinforcement learning
Operation under uncertainty
Temporal difference learning
Eligibility traces

ABSTRACT

The operation of a community energy storage system (CESS) is challenging due to the volatility of photovoltaic distributed generation, electricity consumption, and energy prices. Selecting the optimal CESS setpoints during the day is a sequential decision problem under uncertainty, which can be solved using dynamic learning methods. This paper proposes a reinforcement learning (RL) technique based on temporal difference learning with eligibility traces (ET). It aims to minimize the day-ahead energy costs while maintaining the technical limits at the grid coupling point. The performance of the RL is compared against an oracle based on a deterministic mixed-integer second-order constraint program (MISOCP). The use of ET boosts the RL agent learning rate for the CESS operation problem. The ET effectively assigns credit to the action sequences that bring the CESS to a high state of charge before the peak prices, reducing the training time. The case study shows that the proposed method learns to operate the CESS effectively and ten times faster than common RL algorithms applied to energy systems such as Tabular Q-learning and Fitted-Q. Also, the RL agent operates the CESS 94% near the optimal, reducing the energy costs for the end-user up to 12%.

1. Introduction and related work

A community energy storage system (CESS) is a mid-size battery within the 100 kWh–10 MWh range, connected to the distribution network installed near the residential areas. CESS promises to generate collective socio-economic benefits such as offering ancillary services for the system operator, reducing energy bills, and generating revenue offering demand flexibility in a scenario of dynamic prices during the day [1]. The optimal operation of a CESS is a challenging problem due to dynamic prices and the stochastic nature of the multiple variables influencing the state of the distribution network during the day. The grid inherently brings technical constraints for the operation of the CESS, e.g., networks with high photo-voltaic (PV) penetration may present over-voltage problems at noon, impeding the battery from selling energy. On the other hand, the battery might not fully charge because of a potential undervoltage even in the presence of low prices.

A range of techniques has been applied to the CESS operation under uncertainty. The first approach is to find the set-points of the battery, solving the problem using robust convex optimization [2] and stochastic mixed-integer linear programming (MILP) [3] methods over a finite time horizon. Additionally, the optimization can be combined with a rolling window as a model predictive control (MPC), whose stochastic variables are predicted individually [4]. A second approach is to cast the problem as a *Markov decision process* (MDP) and use dynamic programming (DP) techniques studied in [5,6], to optimize operational cost for residence with storage coupled with a PV system. When the complexity of the MDP problem is high, e.g., the number of stochastic variables and control decisions increases, adaptive dynamic programming (ADP) can be used to find computationally feasible sub-optimal solutions [7]. In line with the category dynamic methods, the third approach is the use of model-free RL techniques, which have been applied in different domains for power systems [8,9]. As opposed to the DP approach, RL does not require an explicit model for the stochastic transition dynamics of the MDP in order to find a solution. Instead, RL is able to find one solution while interacting with the system. Usually, optimization approaches focus on creating models and approximations to make the problem tractable and suitable for commercial convex

[*] Corresponding author.
E-mail address: e.m.salazar.duque@tue.nl (E.M.S. Duque).

[1] J. S. Giraldo was financially supported by The Netherlands Enterprise Agency (RVO) – DEI+ project 120037 "*Het Indië terrein: Een slimme buurtbatterij in de oude weverij*".

https://doi.org/10.1016/j.epsr.2022.108515
Received 3 October 2021; Received in revised form 14 April 2022; Accepted 2 July 2022
Available online 16 July 2022

programming solvers. For RL techniques, the task burden lies in the correct algorithm selection, state representation, and hyper-parameter tuning to obtain an optimal result. RL has the disadvantage of sampling inefficiency, requiring a significant amount of interaction trials with the MDP to learn a control policy. Nevertheless, it can be powerful when being used jointly with complex simulators that include non-linearities as a digital twin [10], since it can potentially find an optimal operational solution by interaction.

Studies related to the operation of CESS using RL approaches typically focus on reducing energy costs, neglecting network constraints [11,12], e.g., voltage violations. Alternatively, the battery is used only to provide solutions for technical problems on the network, e.g., voltage regulation, congestion management, without considering the energy costs for the end-user [13]. This work focuses on both sides, reducing users' energy costs while maintaining the distribution network's technical limits. Recent research of RL in energy systems has been concentrated on the use of deep neural networks, possibly inspired by the achievements of deep reinforcement learning (DRL) in other fields [14]. DRL has a problem of instability and brittleness due to algorithmic hyper-parameter selection and implementation details which significantly affect the algorithm's performance on specific tasks [15], making the tuning processes a challenge. On the other hand, RL techniques with simpler linear *function approximators* (FA) that have stronger mathematical guarantees of convergence, have not been widely explored in power systems problems [16]. In other research fields, RL algorithms with linear FA have been empirically tested and shown to be equally powerful in domains that DRL was thought to be the best approach [17].

The contribution in this paper are the following:

- We propose a linear temporal difference (TD) RL technique with eligibility traces for the day-ahead operation of a CESS in distribution systems. The optimal policy generalizes common stochastic variables that affect grid operation, i.e., load consumption and PV generation.
- Analysis of the selection process of the hyper-parameters of the RL algorithm in the context of CESS operation. The proposed method has a more straightforward tuning process than RL methods which use non-linear function approximators, e.g., neural networks, an ensemble of regression trees.

The solutions of our RL model are compared against the optimal operation using an mixed-integer second-order cone programming (MISOCP) formulation [2], which serves as an *oracle* with perfect information. Additionally, this work pursues a more fundamental view of common RL techniques used in battery control in the energy domain [16], stating the major differences and pointing clearly that RL eligibility traces are a strong candidate for the CESS operation problem, making learning more efficient and robust. To the best of the authors' knowledge, the use of RL with eligibility traces and linear FA on the CESS operation problem has not been explored in the literature before.

The paper is organized as follows. Section 2 describes the proposed approach to the CESS operation using RL. Section 3 defines the CESS operation problem as an MDP. Section 4 explains the solution of the MDP using RL, highlighting the advantages of eligibility traces on the CESS operation. Section 5 shows the case study and results of the proposed approach. Finally, Section 6 summarizes and concludes.

2. CESS operation with RL approach

The main advantage of using a control agent based on RL is that, from its perspective, the model representing distribution network can be seen as a black box process with unknown dynamics. Inside the black box, the power grid can be a complex non-linear model, and the RL algorithm finds an optimal solution based on just interactions with the system. This enables to use of RL with a digital twin, which emulates detailed real grid operation assisted by previous consumption, generation, meteorological and price databases. Fig. 1 shows the operational framework for the RL-based controller, which is training the RL agent at the end of the day, learning a control *policy* for the CESS, and deploy it for operation for the next day. The *policy* function $\pi(\cdot)$ maps each *state* (s) e.g., active power of the load, energy prices, voltage at the point of connection; to *actions* (a), i.e., power set-points of the battery. The digital twin serves as an emulator of real-life responses for different CESS set-points in a safe simulation environment. The policy relies on a small set of variables, describing the *state* of the system based on local measurements. The learned policy allows operating the CESS within grid's technical limits.

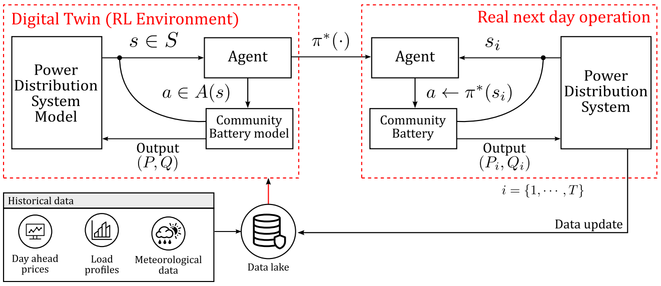

Fig. 1. RL agent learns to control a CESS under a digital twin network grid simulator, which uses actual historical data and current topology status of the grid. The learned policy $\pi^*(\cdot)$ is deployed in real life for the next-day operation.

The RL approach generalizes the set-point solutions for the CESS under uncertain load values for the next day, without requiring to compute the solutions for each time step as it would happen in an MPC. The MPC rolling horizon for the CESS control could be formulated as a MILP problem solved for every time step t, and its solution would generate a control action sequence over the battery $\{a_o, \dots, a_k, \dots, a_T\}$ for the next T periods, assuming a perfect forecast of stochastic variables, which is not realistic. On the other hand, policies in RL are more general than control sequences in the case of stochastic uncertainty; they can result in improved revenue because the choice of the control actions incorporate knowledge of the state s_t, which improves the generalization of optimal action over multiple possible states [18].

The day-ahead energy price is assumed to be known in advance, and the irradiance profile is assumed to be given by a prediction algorithm including an error to consider the variability of PV generation sources.

3. CESS operation as a Markov Decision Process

The operation of a CESS can be formalized as an MDP [19], which is described by a tuple $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, where $\mathcal{S}$ is the set of states; and $\mathcal{A}$ is the set of possible actions that the agent can perform on the CESS, both taken in discrete time steps $t = 0, 1, 2, \dots$. The function $p \doteq p(s_{t+1}|s_t, a_t)$ is a transition probability to the state s_{t+1} when the action $a_t \in \mathcal{A}$ is taken in the state s_t; the reward $r \doteq r(s_t, a_t, s_{t+1}) \mapsto \mathbb{R}$ is the expected reward that the agent obtains for transitioning from state s_t to s_{t+1} when the action a_t is executed. Finally, $\gamma \in (0, 1]$ is a discount factor which modulates the relevance of future rewards in time. The main objective of solving an MDP is to find an *optimal policy* function that maps each state to actions, i.e., $\pi^*(s_t) \mapsto a_t$, leading to the maximum possible reward of the process. The MDP for the CESS operation problem is defined as follows:

3.1. State

The state is defined by the information available at the point of connection (POC) of the CESS in the grid, and it is composed by the tuple: active, reactive power of the load, voltage, battery state of

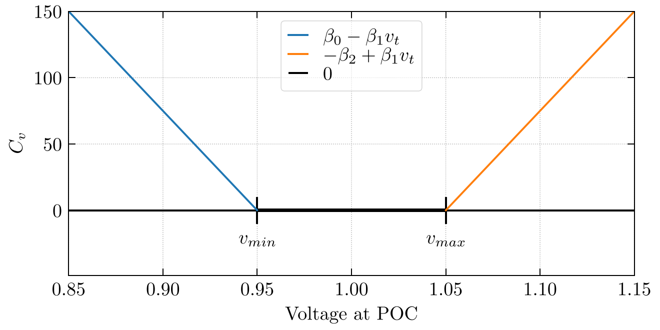

Fig. 2. Example plot of the Voltage penalty C_v, with parameters $\beta_0 = 1425$, $\beta_1 = 1500$, $\beta_2 = 1575$, $v_{min} = 0.95$ and $v_{max} = 1.05$. Values of voltage magnitudes outside technical limits at POC results in additional costs for the agent.

charge (SOC), global irradiance, and a time step of the day, denoted respectively as:

$$s_t = \langle P_t^{\mathrm{D}}, Q_t^{\mathrm{D}}, V_t, SOC_t, Q_t^{\mathrm{irr}}, t \rangle, \tag{1}$$

All variables are considered stochastic and continuous. Note that the price is not considered in the tuple as it is deterministic, assumed to be known, and the time-series profile does not change during the learning process of the policy for the next day of CESS operation.

3.2. Action

The action space $\mathcal{A}$ is the possible discrete battery's active power set-points u_i at step t, i.e., $a_t = u_{i,t} = P_t^{\mathrm{ess}}$. The set of possible actions is discretized in order to use an RL critic-only algorithm (discussed in Section 4). The action set is defined as $\mathcal{A} := \{u_i \mid u_i \in [-P_{min}^{\mathrm{ess}}, P_{max}^{\mathrm{ess}}],\ i = 1, \dots, N\}$, where P_{min}^{ess} and P_{max}^{ess} are the minimum and maximum power rate output of the CESS's inverter. It is convenient to have an symmetric range of power levels, positive and negative values available to charge/discharge, and one option to set the battery on idle mode. Therefore, N is an odd integer number for active power levels, and the action space starts from $u_1 = -P_{min}^{\mathrm{ess}}$ and subsequent values $u_{i+1} = u_i + (P_{max}^{\mathrm{ess}} - P_{min}^{\mathrm{ess}})/(N-1)$.

In practice, the actions affecting the battery's stored energy depend on the current SOC, e.g., charging actions when the battery is full will take no effect. The action space can be reduced by blinding those actions that do not affect the battery state, making the learning more efficient. Therefore, we determine a subset of valid actions that the agent can choose based on the current state, $\tilde{\mathcal{A}}(s_t) \subseteq \mathcal{A}$ dependent on the battery SOC, defined as $\tilde{\mathcal{A}} := \{u_i \mid SOC_{min} \le u_i \Delta t + SOC_t \le SOC_{max}\}$.

3.3. Reward

The reward function is composed by two components. First, a cost term $C_e(\cdot)$ for the energy from/to the grid, and the second term, $C_v(\cdot)$, is a penalization for voltage magnitudes outside the technical limits, defined as

$$r_t(s_t) = -[C_v(s_t) + C_e(s_t)]. \tag{2}$$

The reward function's negative sign is that the RL is defined as maximizing reward, which is equivalent to minimizing the costs. Specifically, each of the reward components is

$$C_e = \xi_t P_t^{\mathrm{net}} \Delta t, \tag{3}$$

$$C_v = \begin{cases} \beta_0 - \beta_1 v_t & \text{if } v_t \le v_{min} \\ -\beta_2 + \beta_1 v_t & \text{if } v_t \ge v_{max} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

On which ξ_t is the energy price, the net load is defined as $P_t^{\mathrm{net}} = P_t^{\mathrm{D}} - (P_t^{\mathrm{PV}} + P_t^{\mathrm{ess}})$, P_t^{PV} is PV generation at the POC. The parameters $\beta_0, \beta_1, \beta_2$ create a penalty function for the voltage technical limits at the POC of the battery (Fig. 2). The dead band between v_{min} and v_{max} allows the agent to focus on minimizing energy cost once the voltage values are within the voltage limits.

The slope β_1 controls the severity of the voltage penalty. The parameters β_0 and β_2 depend on the voltage technical limits, and they are calculated as $\beta_0 = \beta_1 v_{min}$ and $\beta_2 = \beta_1 v_{max}$. The value of β_1 is tuned by trial and error, but as a general guideline, its value should be high enough so the cost function C_v overcomes the cost of energy C_e at any time t if there is a voltage violation. Otherwise, the agent could learn a policy that reduces energy costs while violating technical limits. Additionally, the linear barrier function of C_v provides a smooth corrective signal for the agent, which is trained via gradient descent (Section 4).

3.4. Environment

The RL agent training process is data-intensive, and multiple episodes (daily scenarios) for learning $\pi^*(\cdot)$ should be performed. This creates a computational bottleneck for RL techniques in simulators for network grid dynamics. We selected an efficient AC power flow for radial distribution networks [20] to decrease simulation time. The power flow solutions provide us the transition function $f(\cdot)$ for the voltage at the POC, i.e., $V_{t+1} = f(P_{t+1}^{\mathrm{D}}, Q_{t+1}^{\mathrm{D}}, P_t^{\mathrm{ess}})$, as a response from the network grid when the set-points action a_t for the CESS is executed. The transition function defining the dynamics of the CESS is

$$\mathrm{SOC}_{t+1} = \mathrm{SOC}_t - \frac{\Delta_t}{\mathrm{EC}} P_t^{\mathrm{ess}}, \tag{5}$$

where EC is the energy capacity in kilowatt-hour, and Δ_t is the time interval in hours. Additionally, the PV generation at each step t is defined by $P_t^{\mathrm{PV}} = kQ_t^{\mathrm{irr}}$, which is directly proportional to the solar global irradiance Q_t^{irr}, and k is a factor proportional to the installed PV capacity. The prediction error over the global irradiance variable is σ_t^{irr}.

It should be mentioned that environment's complexity can be increased, including a detailed battery model e.g., efficiency curves, temperature, degradation, or generative models for load consumption [21]. However, since one of the objectives in this work is assessing the performance of the RL model, then the battery dynamics need to be simplified for the MISCOP model.

3.5. Naive charging policy

In this paper, we assume exact knowledge of day-ahead price. One can develop a simple but effective naive rule-based policy to charge when the current price is below the average price for the day ($\bar{\xi}$) and discharge when it is above.

$$\tilde{\pi}(s_t) = \begin{cases} P_{max}^{\mathrm{ess}} & \text{if } (\xi_t < \bar{\xi}) \wedge (SOC_t < SOC_{max}), \\ P_{min}^{\mathrm{ess}} & \text{if } (\xi_t \ge \bar{\xi}) \wedge (SOC_t > SOC_{min}), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

This policy only charges based on SOC and energy price data in the current time step. It only considers the maximum profit for the user neglecting the grid constraints, which could be the case of a private user.

4. Temporal difference learning and eligibility traces (λ)

The purpose of this subsection is to highlight the main differences between the critic-only RL techniques used in this paper, i.e., Tabular Q-learning, Fitted-Q iteration (FQI), and True-Online Sarsa(λ). The first two methods are commonly used in energy management systems [8]. Here we emphasize the latter RL technique's advantage, which uses eligibility traces on the CESS operation problem.

The reward collected from each time step onwards is called *return*, defined as:

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma G_{t+1} \\ &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-1} R_{t+T-1}, \end{aligned} \tag{7}$$

which is the discounted sum of stochastic reward variable R observed after the time t. Variable T denotes the time step at the end of the episode (decision horizon).

The action-value function $q_\pi(S_t, A_t) \mapsto \mathbb{R}$ is the expected *return* given that action a_t is taken in the state s_t and following the policy $\pi(\cdot)$ after that. Meaning that it quantifies how rewarding an action is for a specific state. The action-value function [22] can be expanded recursively based on the following state as:

$$\begin{aligned} q_\pi(s, a) &\doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]. \end{aligned} \tag{8}$$

Bellman's principle of optimality states that the optimal action-value function and optimal policy for an MDP has the recursive solution as

$$\begin{aligned} q_{\pi^*}(s, a) &= \mathbb{E}_\pi[R_{t+1} \\ &\quad + \gamma \max_{a_{t+1} \in \mathcal{A}} q^*_\pi(S_{t+1}, a_{t+1}) | S_t = s, A_t = a]. \end{aligned} \tag{9}$$

The solution of (9) can be obtained by using a TD algorithm known as Q-learning [23], which solves the following update rule iteratively as follows:

$$\begin{aligned} &\hat{q}_{k+1}(S_t, A_t) = \hat{q}_k(S_t, A_t) \\ &\quad + \underbrace{\alpha[R_{t+1} + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{q}_k(S_{t+1}, a_{t+1}) - \hat{q}_k(S_t, A_t)]}_{\delta_t:\ \text{TD Error}}, \end{aligned} \tag{10}$$

where $\hat{q}(\cdot)$ is a tabular representation of function $q_\pi(s, a)$ in (9). This table is built by combining all the states and possible actions in the MDP, and $\alpha \in (0, 1]$ is a learning rate. In problems where the state space is continuous, a typical approach is to coarsely discretize the state space and apply the update (10) iteratively until the temporal difference error (TD) converges, i.e., $\delta_t \to 0$. Other approaches have been proposed to also solve (9) for continuous states spaces, replacing the tabular representation of $\hat{q}(\cdot)$ for different types of non-linear FA such as an ensemble of regression trees (FQI) [24] or neural networks (Neural Fitted-Q) [25] as a form of iterative supervised learning problems using updates in batches of experiences, e.g., sets of transitions tuples.

The RL techniques based on action-value methods (i.e., critic-only methods) use the learned action-value function as a surrogate for the policy. Therefore, for the optimal value function $q_{\pi^*}(\cdot)$, the policy is simply:

$$\pi^*(s) \doteq \operatorname*{argmax}_{a \in \mathcal{A}} q^*_\pi(s, a). \tag{11}$$

The maximization of (11) can be computationally expensive or intractable if the action space is continuous. Therefore, the most straightforward approach is to discretize the action space in a set of possible values, as defined in Section 3.2.

Another approach to solve the MDP using action-value functions is to set up the problem as an error-minimization problem [26]

$$\mathcal{L}(\theta) \doteq \frac{1}{2} \sum_{s_i \in S} d_\pi(s_i)(q_\pi(s_t, a_t) - \hat{q}(s, a, \boldsymbol{\theta}))^2, \tag{12}$$

where $d_\pi(s_i)$ is the stationary probability distribution of the states while following the policy $\pi(\cdot)$, and $\hat{q}(s, a, \boldsymbol{\theta})$ is parametrized function by $\boldsymbol{\theta}$. The problem in (12) can be solved using gradient descent while sampling from the stationary distribution, where the updates of the parameter $\boldsymbol{\theta}$ follows

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(q_\pi(S_t, A_t) - \hat{q}(s, a, \boldsymbol{\theta}_t))\nabla \hat{q}(s, a, \boldsymbol{\theta}_t) \tag{13a}$$

$$= \boldsymbol{\theta}_t + \alpha(U_t - \hat{q}(s, a, \boldsymbol{\theta}_t))\nabla \hat{q}(s, a, \boldsymbol{\theta}_t) \tag{13b}$$

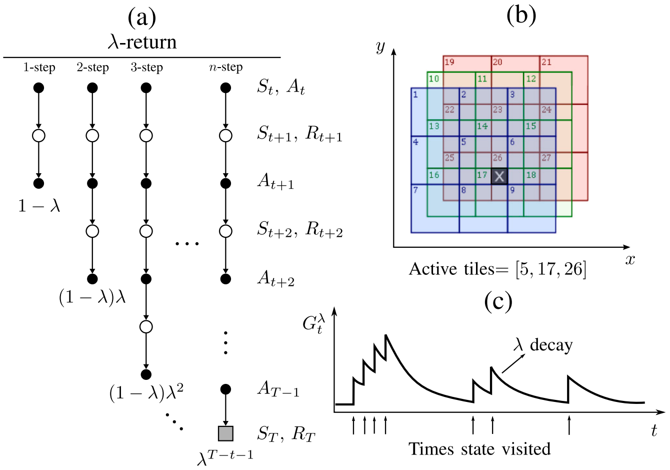

Fig. 3. (a) Multi-step update target: parameter λ weights each experience trajectory during training, making the learning more efficient. Adapted from [29]. (b) Example of the soft generalization in 2-D using tile coding. Three tiles are activate for a state visit depicted by an X. (c) Illustration of eligibility trace decay when the same state is visited during a training episode.

The actual function $q_\pi(S_t, A_t)$ is unknown and it is replaced by a target estimate U_t in (13b). Different types of targets can be used: (i) Monte-Carlo update target, $U_t = G_T$ where G_T is the total reward at the end of the episode, which is unbiased but shows large variance and slow convergence in practice. (ii) One-step update target, $U_t = R_{t+1} + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{q}_k(S_{t+1}, a_{t+1}, \boldsymbol{\theta}_t)$, which is the target function used in techniques that use non-linear FA of $\hat{q}(\cdot, \boldsymbol{\theta})$ such as DQN [14], Boltzmann Machines [27], where the parameter $\boldsymbol{\theta}$ is computed via stochastic gradient descent. (iii) Multi-step update target (λ-return) [28], $U_t = G_t^\lambda$, which is the proposed target to use in this paper is defined as

$$G_t^\lambda \doteq (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t-1} G_t \tag{14}$$

$$G_t^{(n)} \doteq \sum_{k=1}^{n} \gamma^{k-1} R_{t+k} + \gamma^n \hat{q}(s_{t+n}, a_{t+1}, \boldsymbol{\theta}_{t+n-1}). \tag{15}$$

The intuition of this target is shown in Fig. 3(a). Parameter $\lambda \in [0, 1]$ works as a weighted average for multiple trajectory paths experienced during training. In the extreme values of its interval, when $\lambda = 1$, expression (14) reduces to target option (i), and when $\lambda = 0$ it becomes option (ii). *It is crucial to notice that the methods*: Tabular Q-learning in (10), FQI that solves the problem in (9), and DQN that solves (13a) using option (ii) have a similar update in the way that only uses the return and information from the current step, and next transition $\langle s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1} \rangle$, i.e., 1-step return in Fig. 3(a). On the other hand, the multi-step update covers the spectrum from one-step to Monte-Carlo (n-steps) via the parameter λ. This makes it more effective in learning, especially for processes that intrinsically carry an internal state [30], such as the SOC.

The CESS operation is inherently a delayed reward problem, meaning that for an optimal operation, the RL agent should perform CESS charging actions earlier, buying energy at low peak prices (negative reward) preparing the battery for being in a good *state* i.e., high SOC before peak prices. This allows the CESS to sell back energy the grid (high reward). Eligibility traces assigns the credit to all the charging actions which lead to a high SOC before the peak price effectively, using parameter λ. The effective credit assignment reduces the training time considerably. The case of study brings further discussion on this matter.

The multi-step return has a convergence guarantee [26], when $\hat{q}_\pi(\cdot, \boldsymbol{\theta})$ is modeled using a linear FA of the form

$$\hat{q}(s, a, \boldsymbol{\theta}) = \boldsymbol{\theta}^\intercal \boldsymbol{\phi}(s, a) \in \mathbb{R}^d, \tag{16}$$

Algorithm 1: True online Sarsa(λ) for CESS operation

```
1: Input parameters: Step size $\alpha > 0$, trace decay $\lambda \in [0, 1]$
2: Initialize: $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ (e.g., $\boldsymbol{\theta}_0 = \mathbf{0}$)
3: repeat
4:     Initialize $s_0$ from the environment
5:     Choose $a_0 \sim \pi(\cdot|s_0)$
6:     $\boldsymbol{e}_{-1} \leftarrow \mathbf{0}$
7:     $\boldsymbol{\psi}_0 \leftarrow \boldsymbol{\phi}(s_0, a_0)$
8:     $q_{-1} \leftarrow \mathbf{0}$
9:     $t = 0$
10:    repeat
11:        Take action $a_t$ on environment, observe $r_t$, $s_{t+1}$
12:        Choose $a_{t+1} \sim \pi(\cdot|s_{t+1})$ or near greedily
13:        $\boldsymbol{\psi}_{t+1} \leftarrow \boldsymbol{\phi}(s_{t+1}, a_{t+1})$
14:                                    ▷ (if $s_{t+1}$ it Terminal, $\boldsymbol{\psi}_{t+1} \leftarrow \mathbf{0}$)
15:        $q_t \leftarrow \boldsymbol{\theta}_t^{\mathrm{T}} \boldsymbol{\psi}_t$
16:        $q_{t+1} \leftarrow \boldsymbol{\theta}_t^{\mathrm{T}} \boldsymbol{\psi}_{t+1}$
17:        $\delta_t \leftarrow r_t + \gamma q_{t+1} - q_t$
18:        $\boldsymbol{e}_t \leftarrow \gamma\lambda \boldsymbol{e}_{t-1} + (1 - \alpha\gamma\lambda \boldsymbol{e}_{t-1}^{\mathrm{T}} \boldsymbol{\psi}_t)\boldsymbol{\psi}_t$
19:        $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha(\delta_t + q_t - q_{t-1})\boldsymbol{e}_t - \alpha(q_t - q_{t-1})\boldsymbol{\psi}_t$
20:        $q_{t-1} \leftarrow q_{t+1}$
21:        $\psi_t \leftarrow \psi_{t+1}$
22:        $a_t \leftarrow a_{t+1}$
23:        $t \leftarrow t + 1$
24:    until $s_{t+1}$ is Terminal state
25: until finish the total number of episodes
```

where the base feature function $\phi(s, a)$ can have multiple representations such as radial basis functions, Fourier series, polynomial and binary representations (BR) [29]. We used the BR version named tile coding with displacement vectors [31] shown in Fig. 3(b). It provides a computationally efficient soft generalization between states, using tiles represented by a binary vector of dimension d, in which elements of the vector are set to 1 when the value of a state lies in the specific set of tiles. True-Online Sarsa(λ) (TOS(λ)) [28], shown in Algorithm 1, uses the G_t^λ target. It computes the update for $\boldsymbol{\theta}_t$ for each step *on-line* during the interaction with the environment, reducing training time. This can be done with the aid of the vector $\boldsymbol{e}$, i.e., line 6 and 18 in Algorithm 1, which is the *eligibility trace*. This vector works as a short-term memory vector, which parallels its update with the parameter vector $\boldsymbol{\theta}$. Fig. 3(c) shows an example of one eligibility trace which keeps propagating the reward to a state in a decaying fashion during the training process.

4.1. Optimal energy dispatch model (oracle)

An optimization model for the CESS dispatch is formulated as a MISOCP problem [32], which allows us to use commercial solvers for convex programming [33] to find a solution for the battery set-points during the day. The detailed base model can be found in [34], summarized and modified as:

$$\min \sum_{t \in \Omega_T} \xi_t \Delta_t \mathrm{P}_t^{\mathrm{D}} - (\mathrm{P}_t^{\mathrm{PV}} + \mathrm{P}_t^{\mathrm{ess}}) \tag{17}$$

$$\text{s.t.} \qquad \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{y} \tag{18}$$

$$\sum_{i \in |\mathcal{A}|} \boldsymbol{\Phi}_{i,t} u_{i,t} = P_t^{ess}, \quad \sum_{i \in |\mathcal{A}|} \boldsymbol{\Phi}_{i,t} = 1, \tag{19}$$

$$\boldsymbol{\Phi}_{i,t} \in \{0, 1\}, \qquad \forall t \in \Omega_{\mathrm{T}}, u_i \in \mathcal{A}$$

where the objective function is to minimize energy cost (17) for the time horizon defined by set Ω_T. Grid constraints and battery dynamics are represented by second-order inequalities (18). CESS set-points are discretized in (19) to have the same power values between RL, described in Section 3.2, and the MISOCP model. It should be reminded that this optimization model has a perfect prediction of the realization of stochastic variables.

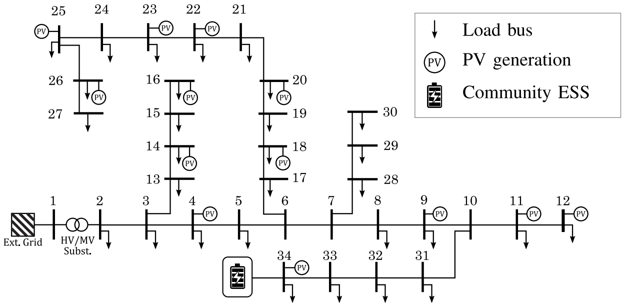

Fig. 4. Modified IEEE-34 Node bus test system with distributed PV generation. CESS placed in the longest feeder path as it is the worst-case scenario for grid's undervoltage problem.

Table 1
Summary — parameters for RL models and CESS.

RL algorithms	$\gamma = 0.995$ $\alpha = [0.025, 0.05, \dots, 0.175]$ $\lambda = [0.9, 0.8, \dots, 0.1]$ Batch size (FQI) $= [10, 100, \dots, 10000]$
Reward	$\beta_0 = 1425$, $\beta_1 = 1500$, $\beta_2 = 1575$
CESS	$N = 7$, $\mathrm{SOC}_{min} = 0.0$, $\mathrm{SOC}_{max} = 1.0$ $P_{min}^{\mathrm{ess}} = -150$ [kW], $P_{max}^{\mathrm{ess}} = 150$ [kW] $\mathrm{EC} = [300, 375, 500, 750, 1500, 2250, 3000]$ [kWh]
PV gen.	$k = \mathcal{U} \sim [0.1 - 0.25]$, $\sigma^{\mathrm{irr}} = 5\%$
Voltage limits	$\upsilon_{max} = 1.05$, $\upsilon_{min} = 0.95$

5. Case study

Tests are conducted in the 34-node IEEE test system shown in Fig. 4. The training data are Dutch market day-ahead prices and load measurements from 3 weeks before the day for CESS control, which a distribution network operator provides. The global solar irradiance comes from actual meteorological information adding normally distributed error with a 5% standard deviation. The parameters for the different tests and algorithms are summarized in Table 1. Here one episode is 24 h ($T = 24$). The number of tiles is set to 20 for each one of the stochastic variables.

5.1. Performance of RL algorithms

The RL agent training curves for one day of operation for different RL algorithms are shown in Fig. 5(a). The number of episodes to achieve a near-optimal performance of Tabular Q-learning (TQL) is ten times higher than TOS(λ). FQI shows a faster convergence in contrast to TOS(λ). Nevertheless, FQI is not close to the MISCOP solution, whereas TOS(λ) achieves near-optimal performance. The FQI agent learns how to stay between voltage limits in the grid, but it cannot optimize the energy costs for the day. TQL agent requires more iterations to be close to the optimal. The variance in the curve for the TQL agent is because the table has poor generalization due to the coarse discretization of state variables. Fig. 5(b) shows a specific case where SOC = 0.5 at the beginning of the day. The FQI agent sells the energy in the morning and sets the CESS on idle the rest of the day. This happens because action selection with the argmax($\cdot$) operator in (11) and the non-linear FA exacerbates the problem of a myopic one-step return. The FQI agent is biased to select the selling action for the next training episodes most of the time, locking the learning in a non-optimal solution. Increasing the exploration could solve the problem for FQI but requires additional techniques and more interaction episodes with the simulator. TOS(λ) is capable of selecting the correct action sequence to achieve high rewards.

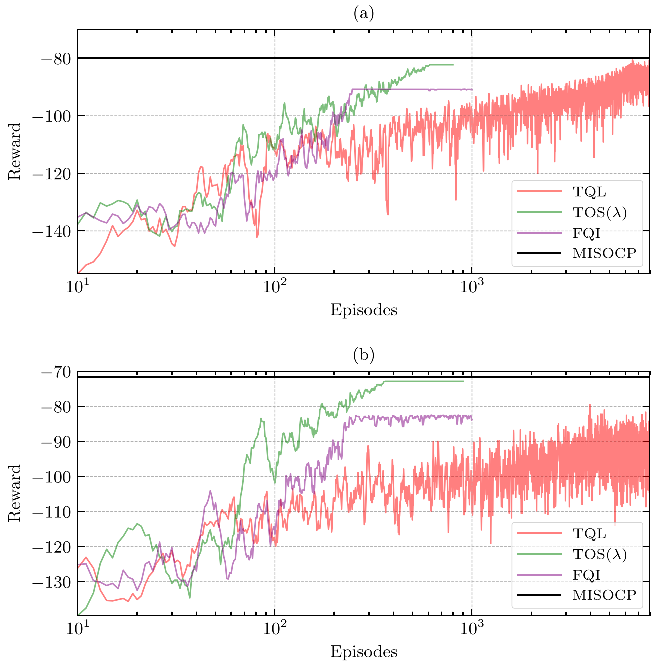

Fig. 5. Comparison of RL techniques learning curves for one day of operation. In this case, CESS has an EC = 500 [kWh]. Two different cases of SOC at the beginning of the day. (a) Case with SOC = 0 (b) Case with SOC = 0.5.

Actions that charge the CESS inflict a negative reward on the agent, as buying is penalized using (3). Nevertheless, those actions are necessary to increase the SOC to sell the energy at peak times (usually in the afternoon) and get a high positive reward. TD(0) techniques, i.e., FQI, TQL, only one state is associated with the high reward, e.g., CESS with high SOC in the afternoon, but it does not propagate the reward to all those state–actions that incurred the high SOC at the afternoon. TD(0) techniques require more iterations to backpropagate the reward. Hence, the longer training times for TQL. On the other hand, TD(λ) techniques, i.e., TOS(λ) uses the parameter λ helps to assign correct updates for the action-values for the states that helped to get a high SOC in the afternoon, e.g., all charging action since the early morning, helping the policy on selecting the charging from the first hours in the day, increasing the effectiveness on the speed of learning with fewer samples.

5.2. Hyperparameters analysis for True online Sarsa(λ)

A parameter sweep of λ and α for a battery capacity EC = 500 kWh are shown in Fig. 6(a) and (b). TOS(λ) has an optimal combination with $\alpha = 0.1$ and $\lambda = 0.7$. This shows that assigning a weighting average over different trajectories ($\lambda = 0.7$) boosts the learning rate for the RL agent. Fig. 6(c) shows the optimal parameters for different EC values and *constant* inverter output power. The EC directly affects the selection of λ but not on α. Parameter λ is inversely proportional to the EC. This is expected because, in the case of a small EC, the CESS can cycle more times for a day, creating more than one peak of high rewards. High parameter λ means that the reward of latter peaks can be propagated back to the state–action values related to the morning. High EC has lower cycling, requiring smaller λ to propagate the rewards effectively to previous state–action values.

The optimal combination of parameters is analyzed against a non-optimal in Fig. 7. The reward of Eq. (2) is showed in subplot (a). The components of the reward on expression (3)– are shown in subplots (b) and (c), respectively. The non-optimal combination of parameters leads to longer training times. Also, it shows higher variability on the δ_t error,

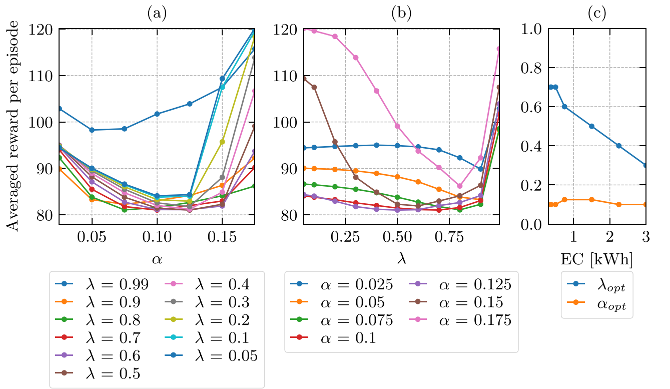

Fig. 6. Parameter sweep for Algorithm 1 for each combination of α and λ shown in Table 1. Results reported are the mean averaged reward for the first 1000 episodes for 50 independent runs. (a) Parameter sweep viewed as function of learning rate α. (b) Same parameter sweep viewed as function of decay rate λ. (c) Optimal α and λ values for repeated sweeps as function of the battery size.

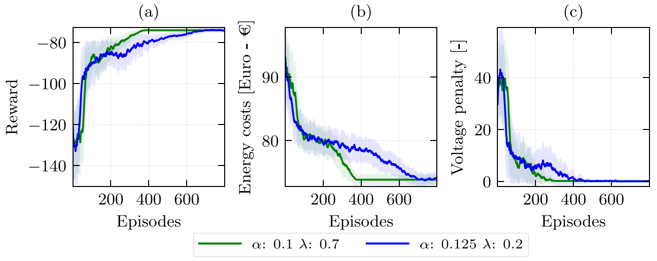

Fig. 7. Learning comparison between two sets of parameters of TOS(λ) for one day of operation. (a) Reward function in (2). (b) Energy cost $C_e(\cdot)$. (c) Voltage penalty $C_v(\cdot)$. Shadow areas represents one standard deviation for 50 independent runs.

represented by the shadowed areas. Interestingly, the agent learns first to stay between voltage limits before reducing the energy costs, which can be seen in a steep descent in subplot (c).

5.3. CESS continuous operation and cumulative costs

The proposed method was tested for a week of operation, and the time series results are shown in Fig. 8. The MISOCP optimal operation for the CESS output power P_t^{ess} and SOC$_t$ is shown in subplot (b). The response of our proposed model in (c) shows an agreement with the optimal solution. Nevertheless, the RL algorithm shows some bouncing response on the inverter's output power, attributed to variable uncertainties and FA error. The grid without a battery is operating close to an undervoltage problem, and the naive policy violates the grid's technical limits because it only considers energy prices for its policy. However, it reasonably captures the actions of the optimal battery setpoints. Fig. 9 shows the cumulative energy costs for a week. In optimal conditions, the end-users have an energy cost reduction of 12.7% compared to the case with no battery. The naive policy is 1.1% above the optimal costs, but it has voltage violations. The proposed method is 6% above the optimal. The 4.9% of cost difference between the naive and TOS(λ) policies is the trade-off between energy cost reduction and staying within the grid's technical limit.

6. Conclusion

This paper proposed an RL control agent based on temporal difference learning with eligibility traces for CESS operation. The agent is trained to learn an operation policy in a simulated scenario, with historical consumption data, next-day-ahead prices, and solar irradiance forecast. The trained policy is then deployed for the next-day operation.

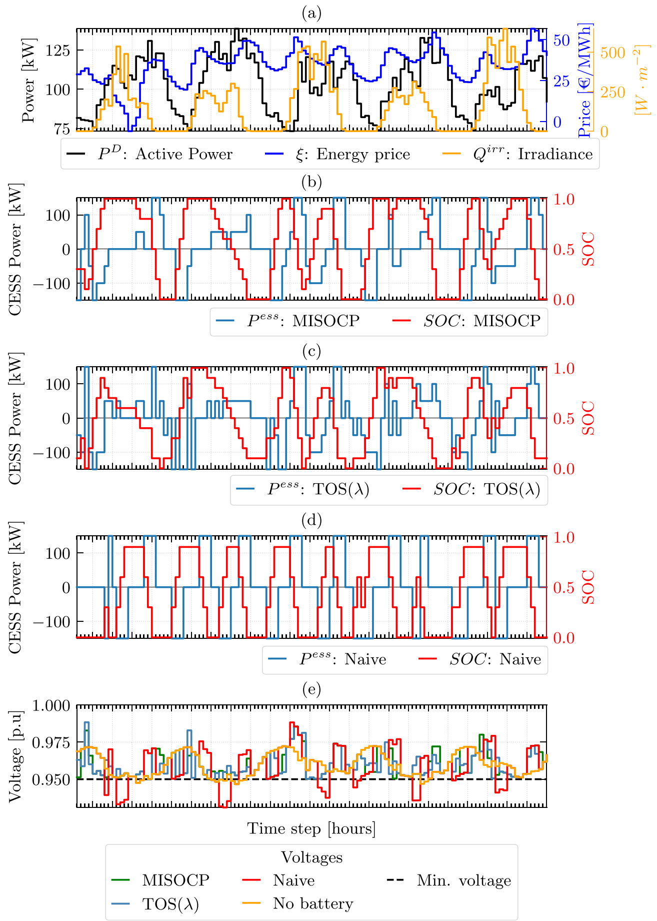

Fig. 8. CESS operation for a week with different methods. (a) Load consumption, solar irradiance, and energy prices at POC. CESS active power set points and SOC for: (b) MISOCP-Oracle, (c) TOS(λ), and (d) Naive charging policy. Subplot (e) show the voltages for the different solutions at POC.

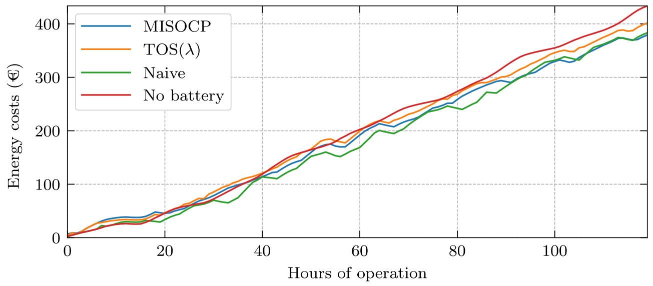

Fig. 9. Cumulative costs for a CESS of $EC = 500$ [kWh] operating for a week with different controller methods.

The agent can minimize the end user's energy costs and stay within the MV networks grid's technical limits, using sensor data at the POC. Popular RL algorithms used in energy applications, namely FQI and Tabular Q-learning, are analyzed in contrast to True Online Sarsa(λ). The latter shows that ET brings a faster learning rate for the RL agent. ET effectively assigns credit to the actions that lead to a high SOC before the energy price peaks to sell the stored energy at high prices. This leads to fewer training interactions with the simulator to obtain an optimal result.

The future research is to explore the ET for more data-efficient RL methods such as LSTD and LSPI, and analyze improvements for CESS operation.

CRediT authorship contribution statement

Edgar Mauricio Salazar Duque: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Juan S. Giraldo:** Software, Writing – review & editing. **Pedro P. Vergara:** Writing – review & editing. **Anne van der Molen:** Writing – review & editing. **Han Slootweg:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B.P. Koirala, E. van Oost, H. van der Windt, Community energy storage: A responsible innovation towards a sustainable energy system? Appl. Energy 231 (2018) 570–585.

[2] J.S. Giraldo, J.A. Castrillon, J.C. Lopez, M.J. Rider, C.A. Castro, Microgrids energy management using robust convex programming, IEEE Trans. Smart Grid 10 (4) (2019) 4520–4530.

[3] Z. Chen, L. Wu, Y. Fu, Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization, IEEE Trans. Smart Grid 3 (2012) 1822–1831.

[4] F. Sossan, E. Namor, R. Cherkaoui, M. Paolone, Achieving the dispatchability of distribution feeders through prosumers data driven forecasting and model predictive control of electrochemical storage, IEEE Trans. Sustain. Energy 7 (4) (2016) 1762–1777.

[5] I. Ranaweera, O.M. Midtgård, Optimization of operational cost for a grid-supporting PV system with battery storage, Renew. Energy 88 (2016) 262–272.

[6] I. Ranaweera, O.M. Midtgård, M. Korpås, Distributed control scheme for residential battery energy storage units coupled with PV systems, Renew. Energy 113 (2017) 1099–1110.

[7] D.F. Salas, W.B. Powell, Benchmarking a scalable approximate dynamic programming algorithm for stochastic control of grid-level energy storage, INFORMS J. Comput. 30 (2017) 106–123.

[8] Z. Zhang, D. Zhang, R.C. Qiu, Deep reinforcement learning for power system applications: An overview, CSEE J. Power Energy Syst. 6 (1) (2020) 213–225.

[9] P.P. Vergara, M. Salazar, J.S. Giraldo, P. Palensky, Optimal dispatch of PV inverters in unbalanced distribution systems using reinforcement learning, Int. J. Electr. Power Energy Syst. 136 (2022) 107628.

[10] W. Danilczyk, Y. Sun, H. He, ANGEL: An intelligent digital twin framework for microgrid security, in: 51st North American Power Symposium, NAPS 2019, Institute of Electrical and Electronics Engineers Inc., 2019.

[11] Q. Wei, D. Liu, G. Shi, A novel dual iterative Q-learning method for optimal battery management in smart residential environments, IEEE Trans. Ind. Electron. 62 (4) (2015) 2509–2518.

[12] B. Huang, J. Wang, Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system, IEEE Trans. Smart Grid 12 (3) (2021) 2272–2283.

[13] M. Al-Saffar, M. Al-Saffar, P. Musilek, Reinforcement learning-based distributed BESS management for mitigating overvoltage issues in systems with high PV penetration, IEEE Trans. Smart Grid 11 (4) (2020) 2980–2994.

[14] V. Mnih, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[15] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, A. Madry, Implementation matters in deep RL: A case study on PPO and TRPO, in: ICLR 2020, 2020.

[16] A. Perera, P. Kamalaruban, Applications of reinforcement learning in energy systems, Renew. Sustain. Energy Rev. 137 (2021) 110618.

[17] A. Rajeswaran, K. Lowrey, E. Todorov, S. Kakade, Towards generalization and simplicity in continuous control, Adv. Neural Inf. Process. Syst. 2017-December (2017) 6551–6562.

[18] D. Bertsekas, Reinforcement learning and optimal control, Athena Scientific, 2019, pp. 14–15.

[19] L. Busoniu, R. Babuska, B. De Schutter, D. Ernst, Reinforcement learning and dynamic programming using function approximators, CRC Press, 2017, pp. 11–41.

[20] J.S. Giraldo, O.D. Montoya, P.P. Vergara, F. Milano, Current injection power flow formulation for electric distribution systems using laurent series, in: 2022 Power Systems Computation Conference, PSCC, 2021, Manuscript submitted for publication, https://dx.doi.org/10.1016/j.epsr.2022.108326.

[21] E.M.S. Duque, P.P. Vergara, P.H. Nguyen, A. van der Molen, J.G. Slootweg, Conditional multivariate elliptical copulas to model residential load profiles from smart meter data, IEEE Trans. Smart Grid 12 (5) (2021) 4280–4294.

[22] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, MIT Press, 2018, p. 73.

[23] C.J. Watkins, P. Dayan, Technical note: Q-learning, Mach. Learn. 8 (3) (1992) 279–292.

[24] D. Ernst, P. Geurts, L. Wehenkel, Tree-based batch mode reinforcement learning, J. Mach. Learn. Res. 6 (2005) 503–556.

[25] M. Riedmiller, Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method, in: J.a. Gama, R. Camacho, P.B. Brazdil, A.M. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 317–328.

[26] J.N. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation, IEEE Trans. Automat. Control 42 (5) (1997).

[27] E. Mocanu, D.C. Mocanu, P.H. Nguyen, A. Liotta, M.E. Webber, M. Gibescu, J.G. Slootweg, On-line building energy optimization using deep reinforcement learning, IEEE Trans. Smart Grid 10 (4) (2019) 3698–3708.

[28] H. van Seijen, A.R. Mahmood, P.M. Pilarski, M.C. Machado, R.S. Sutton, van Seijen, R.S.S. van Seijen, True online temporal-difference learning, J. Mach. Learn. Res. 17 (2016) 1–40.

[29] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, MIT Press, 2018.

[30] R.S. Sutton, Learning to predict by the methods of temporal differences, Mach. Learn. 1988 3:1 3 (1988) 9–44.

[31] W.T. Miller III, F.H. Glanz, L.G. Kraft III, Application of a general learning algorithm to the control of robotic manipulators, Int. J. Robot. Res. 6 (2) (1987) 84–98.

[32] R.A. Jabr, Radial distribution load flow using conic programming, IEEE Trans. Power Syst. 21 (3) (2006) 1458–1459.

[33] J. Kronqvist, D.E. Bernal, A. Lundell, I.E. Grossmann, A review and comparison of solvers for convex MINLP, Optim. Eng. 2018 20 (2018) 397–455.

[34] J.S. Giraldo, M. Salazar, P.P. Vergara, G. Tsaousoglou, J.G. Slootweg, N.G. Paterakis, Optimal operation of community energy storage using stochastic gradient boosting trees, in: 2021 IEEE Madrid PowerTech, PowerTech 2021 - Conference Proceedings, Institute of Electrical and Electronics Engineers Inc., 2021.