Optimal Dispatch of PV Inverters in Unbalanced Distribution Systems using Reinforcement Learning

Pedro P. Vergara[a*], Mauricio Salazar[b], Juan S. Giraldo[c], and Peter Palensky[a],

[a]Intelligent Electrical Power Grids (IEPG) Group, Delft University of Technology, Delft 2628CD, The Netherlands.

[b]Electrical Energy Systems (EES) Group, Eindhoven University of Technology, Eindhoven 5612AE, The Netherlands.

[c]Mathematics of Operation Research Group, University of Twente, Enschede 7522NB, The Netherlands.

emails:p.p.vergarabarrios@tudelft.nl, e.m.salazar.duque@tue.nl, jnse@ieee.org,, p.palensky@tudelft.nl

*Corresponding author

Abstract

In this paper, a Reinforcement Learning (RL)-based approach to optimally dispatch PV inverters in unbalanced distribution systems is presented. The proposed approach exploits a decentralized architecture in which PV inverters are operated by agents that perform all computational processes locally; while communicating with a central agent to guarantee voltage magnitude regulation within the distribution system. The dispatch problem of PV inverters is modeled as a Markov Decision Process (MDP), enabling the use of RL algorithms. A rolling horizon strategy is used to avoid the computational burden usually associated with continuous state and action spaces, coupled with a computationally efficient learning algorithm to model the action-value function for each PV inverter. The effectiveness of the proposed decentralized RL approach is compared with the optimal solution provided by a centralized nonlinear programming (NLP) formulation. Results showed that within several executions, the proposed approach converges either to the optimal solution or to solutions with a PV curtailment excess of less than 2.5% while still enforcing voltage magnitude regulation.

Keywords: Distribution systems, optimal dispatch, PV systems, reinforcement learning, voltage regulation.

1. Introduction

According to the International Energy Agency, for the year 2020, a total addition of 107 GW to the global solar PV capacity was reached [1]. From this new PV capacity, approximately 36% comes from residential, commercial, and industrial projects, usually located at low voltage (LV) and medium (MV) voltage distribution networks [2]. Due to these constantly increasing levels of PV generation, Distribution System Operators (DSOs) are facing several technical and operational challenges, including overvoltage issues, increase in the

frequency of tap changes in distribution transformers as well as in power losses, violation of the thermal limits on the lines, among others [3, 4].

Various strategies can be found in the literature to cope with the technical issues on distribution networks due to a high PV penetration. These strategies can be grouped into coordinated and locally implemented strategies. Locally implemented strategies are easy to implement and do not require any type of communication infrastructure. Among these strategies, one can find those based on droop control, such as in [5] and [6]. In these droop-based control strategies, the PV inverters regulate their active and reactive power injection as a function of their voltage magnitude at the point of connection with the distribution system [7]. Despite their effectiveness to solve overvoltage issues, as curtailment decisions are made based only on local information, a larger amount of active power will be curtailed, especially when compared with coordinated strategies that consider the whole distribution network's operation. Moreover, they can be seen as unfair, as PV inverters located at the end of the feeders curtail more than those located closer to the distribution transformer [8]. This issue can be solved, for instance, if the operation of the droop control is coordinated among all PV inverters, as shown in [9].

In contrast to locally implemented strategies, coordinated strategies can ensure minimum PV power curtailment, but they require the deployment of either a centralized (e.g., [10]) or a distributed (e.g., [11, 12]) communication infrastructure. The dispatch of all PV inverters within the distribution system can be formulated as a nonlinear optimization problem to ensure minimum PV power curtailment, such as in [10] and [13]. Although optimality can be guarantee through convexification procedures, these centralized approaches show poor scalability features. To overcome this issue, works such as [12] and [14] have developed distributed strategies in which all the information required to perform coordination is shared either with a centralized operator or between PV inverters closely located. Nevertheless, due to their distributed nature, an online iterative procedure must be executed until a convergence criterion is reached. If such criterion is not met, optimality and feasibility cannot be guaranteed.

Recently, coordinated methods based on reinforcement learning (RL) have drawn much attention for their capacity to learn from historical data and/or from continuous interaction with an environment [15]. If properly designed, RL-based approaches offer multiple advantages when compared with other optimization-based methods. Such advantages include that distributed implementation is easier and straightforward; they can be used in real-time (they are usually trained offline); do not require an accurate physical model since they can be updated after interacting with the environment [16], among others. An updated review on the application of RL approaches for energy systems problems can be found in [17]. Regarding the dispatch problem of PV inverters, in [18], a centralized deep RL algorithm is implemented. Results showed that

once trained; the developed deep RL approach can successfully mitigate overvoltage issues with lower PV power curtailment when compared with a droop-based strategy. A similar centralized deep RL strategy is presented in [19]. An RL-based strategy based on a multi-agent approach is presented in [20] and [21] to enable distributed implementation. In these works, deep neural networks are also used to model the value function within the RL strategy. Nevertheless, although deep neural networks have shown promising results in several RL application areas [22, 23], as these are nonlinear parametric models, their convergence within RL frameworks is not guaranteed, difficulting its implementation [24]. Moreover, a general procedure to optimally define some intrinsic parameters (e.g., number of layers, number of units, types of activation functions) is not available yet. The value and/or action-value function can be easily approximated using linear models to overcome this issue [25]. In this sense, the main advantage of linear parametric models is that their convergence is theoretically guaranteed as long as enough exploration is ensured [26].

Based on the aforementioned, an RL-based approach to optimally dispatch PV inverters in unbalanced distribution systems is presented in this paper. The proposed approach exploits a decentralized architecture in which PV inverters are operated by agents that perform all computational processes locally; while communicating with a central agent to guarantee voltage magnitude regulation within the distribution system. Here, the PV inverters dispatch problem is modeled as a Markov Decision Process (MDP), enabling the use of RL algorithms. Within the proposed RL model, a computationally efficient learning algorithm to model the action-value function is used. The effectiveness of the proposed decentralized RL approach is compared with the optimal solution provided by a centralized nonlinear programming (NLP) formulation. The main contribution of this paper is as follows:

- A decentralized RL approach able to optimally dispatch PV inverters in an unbalanced distribution system considering voltage magnitude constraints is presented. The proposed RL approach uses a customized reward function and state definition that enables to reach the centralized optimal solution, while still enables all computational processes to be performed locally (also known as on-device machine learning).

The remainder of this paper is structured as follows: Sec. 2 presents a centralized NLP formulation model for the optimal dispatch problem of PV inverters. Later, Sec. 3 introduces Markov Decisions Processes (MDPs) and RL. Sec. 4 presents the optimal dispatch problem of PV inverters as an MDPs and the proposed RL approach, while Sec. 5 presents the simulation results used to validate the proposed approach. Finally, conclusions are drawn in Sec. 6.

2. Optimal Dispatch of PV Inverters

The optimal dispatch problem of PV inverters in unbalanced distribution networks can be modeled using the NLP formulation given by (1)–(13). The objective function in (1) aims at minimizing the total PV generation curtailment for the time horizon $\mathcal{T}$.

$$\min_{\Delta P^{PV}_{m,t}} \left\{ \sum_{t\in\mathcal{T}} \left[\sum_{m\in\mathcal{N}} \sum_{\phi\in\mathcal{F}} (P^{D}_{m,t,\phi} - P^{G}_{m,t,\phi})\Delta t \right] \right\}, \tag{1}$$

subject to:

$$\sum_{nm\in\mathcal{L}} I^{\text{re}}_{nm,\phi,t} - \sum_{mn\in\mathcal{L}} I^{\text{re}}_{mn,\phi,t} + I^{G\text{re}}_{m,\phi,t} = I^{D\text{re}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{2}$$

$$\sum_{nm\in\mathcal{L}} I^{\text{im}}_{nm,\phi,t} - \sum_{mn\in\mathcal{L}} I^{\text{im}}_{mn,\phi,t} + I^{G\text{im}}_{m,\phi,t} = I^{D\text{im}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{3}$$

$$V^{\text{re}}_{m,\phi,t} - V^{\text{re}}_{n,\phi,t} = \sum_{\psi\in\mathcal{F}} \left(R_{mn,\phi,\psi} I^{\text{re}}_{mn,\psi,t} - X_{mn,\phi,\psi} I^{\text{im}}_{mn,\psi,t} \right) \qquad \forall mn\in\mathcal{L}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{4}$$

$$V^{\text{im}}_{m,\phi,t} - V^{\text{im}}_{n,\phi,t} = \sum_{\psi\in\mathcal{F}} \left(X_{mn,\phi,\psi} I^{\text{re}}_{mn,\psi,t} + R_{mn,\phi,\psi} I^{\text{im}}_{mn,\psi,t} \right) \qquad \forall mn\in\mathcal{L}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{5}$$

$$P^{D}_{m,\phi,t} = V^{\text{re}}_{m,\phi,t} I^{D\text{re}}_{m,\phi,t} + V^{\text{im}}_{m,\phi,t} I^{D\text{im}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{6}$$

$$Q^{D}_{m,\phi,t} = -V^{\text{re}}_{m,\phi,t} I^{D\text{im}}_{m,\phi,t} + V^{\text{im}}_{m,\phi,t} I^{D\text{re}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{7}$$

$$P^{G}_{m,\phi,t} = V^{\text{re}}_{m,\phi,t} I^{G\text{re}}_{m,\phi,t} + V^{\text{im}}_{m,\phi,t} I^{G\text{im}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{8}$$

$$0 = -V^{\text{re}}_{m,\phi,t} I^{G\text{im}}_{m,\phi,t} + V^{\text{im}}_{m,\phi,t} I^{G\text{re}}_{m,\phi,t} \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{9}$$

$$P^{G}_{m,\phi,t} = P^{PV}_{m,t}(1 - \Delta P^{PV}_{m,t})/3 \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{10}$$

$$\underline{V}^2 \le \left(V^{\text{re}}_{m,\phi,t}\right)^2 + \left(V^{\text{im}}_{m,\phi,t}\right)^2 \le \overline{V}^2 \qquad \forall m\in\mathcal{N}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{11}$$

$$0 \le \left(I^{\text{re}}_{mn,\phi,t}\right)^2 + \left(I^{\text{im}}_{mn,\phi,t}\right)^2 \le \overline{I}^2_{mn} \qquad \forall m\in\mathcal{L}, \forall\phi\in\mathcal{F}, \forall t\in\mathcal{T} \tag{12}$$

$$0 \le \Delta P^{PV}_{m,t} \le 1 \qquad \forall m\in\mathcal{N}, \forall t\in\mathcal{T} \tag{13}$$

The unbalanced distribution network is modeled using the AC three-phase power flow formulation shown in (2)–(5)[27]. Constraints (2) and (3) model the real and imaginary line current balance, respectively. Constraints (4) and (5) model the real and imaginary voltage drop in lines, respectively. The active and reactive power consumption is modeled using (6) and (7), respectively, while the active and reactive PV power generation is modelling using (8) and (9), respectively. Notice in (9) that it is assumed that the PV inverter operates with unity power factor. Constraint (10) models the PV active power output of the PV

inverters as a function of the PV power curtailment percentage ($\Delta P^{PV}_{m,t}$). Finally, constraints (11) and (12) enforce the voltage magnitude limits and the thermal limits of lines, respectively, while (13) defines the limits for the PV power curtailment percentage. Notice that a centralized approach must be used to solve the above-presented NLP formulation. A central operator gathers the operational data (e.g., nominal capacity, long-term expected PV generation, etc.) to define the dispatch decisions for the PV inverters enforcing voltage magnitude limits. Simultaneously, the central operator defines the total amount of curtailed power.

3. Markov Decision Process and Reinforcement Learning

In this section, some background on Markov Decision Process (MDP) and the used Reinforcement Learning (RL) algorithm are provided.

3.1. Markov Decision Process (MDP)

In general, a MDP can be described by the 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a finite set of states $s \in \mathcal{S}$ (also know as state space), $\mathcal{A}$ is a finite set of actions $a \in \mathcal{S}$ (also know as action space), $\mathcal{P}$ is a Markovian transition model that states the probability of transitioning from one state to another state after taking an action; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a reward functions that maps from each state $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, $r = \mathcal{R}(s, a, s')$ is the reward obtained when the system transitions from state s to state s' after implementing action a; and $\gamma \in [0, 1)$ is a discount factor. For now on, we will refer to the 4-tuple (s, a, s', r) as a transition.

Let S_t and A_t denote the state and action at time t, respectively, and R_t the reward received after taking action A_t in state S_t. Let $\mathbb{P}$ denote the probability operator, then, $\mathcal{P}_t(s'|s, a) := \mathbb{P}\{S_{t+1} = s'|S_t = s, A_t = a\}$ is the probability of transitioning from state s to state s' after taking action a at time t. Thus, one can estimates the expected reward from an state-action pair (s, a), as

$$R(s, a) = \mathbb{E}[R|s, a] = \sum_{s' \in \mathcal{S}} \mathcal{R}(s, a, s')\mathcal{P}(s'|s, a), \tag{14}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. The total discounted rewards from time t until the system reach a terminal state at time T, denoted by G_t, and also known as the expected return, can be defined as

$$G_t = \sum_{t'=t+1}^{T} \gamma^{t'-t-1} R_{t'}. \tag{15}$$

Let define a deterministic policy π that maps from $\mathcal{S}$ to $\mathcal{A}$, such that $a = \pi(s), s \in \mathcal{S}, a \in \mathcal{A}$. Then, ones

can define an action-value function $Q_\pi(s,a)$ under policy π as follows

$$Q_\pi(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a; \pi], \tag{16}$$

where $Q_\pi(s,a)$ estimates the expected return when taking action a in state s, following policy π. In this sense, the action-value function $Q_\pi(s,a)$ estimates the quality of the state-action pair (s,a) for a give policy π. If the optimal value-function $Q^*(s,a)$ is known, then, an optimal policy can be derived as $\pi^*(s) = \arg\max_{a\in\mathcal{A}} Q^*(s,a)$. Then, it follows from (15) and (16) that $Q^*(s,a)$ satisfies the Bellman optimality equation (see [24]),

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \max_{a'\in\mathcal{A}} Q^*(s',a') \tag{17}$$

In this case, if $\mathcal{P}$ is known and both the state and the action spaces are finite, the action-value function can be exactly represented in a tabular form for all the pairs $(s,a) \in \mathcal{S}\times\mathcal{A}$ solving recursively the expression in (17). If $\mathcal{P}$ is not known, it can be approximated from a batch of transitions samples obtained by directly interacting with the system, using a type of RL algorithms known as batch RL, such as Q-Learning [28].

3.2. Reinforcement Learning and Action-Value Function Approximation

All RL algorithms follow a similar step-by-step procedure i.e., for a state $s \in \mathcal{S}$, take an action $a \in \mathcal{A}$ either randomly or using $Q(s,a)$, observe a new state $s' \in \mathcal{S}$ and a reward r, update the action-value function $Q(s,a)$, repeat until convergence. If the state $\mathcal{S}$, and action $\mathcal{A}$ spaces are finite, conventional Q Learning can be used, in which the state-action spaces are discretized (see e.g., [29]). However, this procedure may suffer from the *curse of dimensionality*, depending on the size of the discretization step. In practical applications, when the state space $\mathcal{S}$ is large or continuous, the action-value function $Q(s,a)$ can be approximated by any type of parametric function such as linear [30] and neural network [31], or non-parametric functions such as decision trees [32]. If a linear function approximation is used, $\hat{Q}(s,a)$ can be represented as,

$$\hat{Q}(s,a) = \boldsymbol{\omega}^\top \boldsymbol{\phi}(s,a), \tag{18}$$

where $\boldsymbol{\phi}(\cdot) : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^f$ is a feature function for (s,a), which is also referred as a basis function, and $\boldsymbol{\omega} \in \mathbb{R}^f$ is a parameter vector.

One of the most data efficient algorithms available in literature to estimate parameters ω, and thus approximate the action-value function $\hat{Q}(s,a)$, is known as Least Square Policy Iteration (LSPI) [26]. To be executed, the LSPI algorithm requires a collection of transition samples $\mathcal{D} = \{(s,a,s',r) : s,s' \in \mathcal{S}, a \in \mathcal{A})\}$

to iteratively estimate $\boldsymbol{\omega}$. To better understand the intuition behind the LSPI algorithm, define an error estimation function $J(\boldsymbol{\omega})$ as

$$J(\boldsymbol{\omega}) = \sum_{(s,a,s',r)\in\mathcal{D}} \Big(Q(s,a) - \boldsymbol{\omega}_k^\top \boldsymbol{\phi}(s,a)\Big)^2, \tag{19}$$

where $\boldsymbol{\omega}_k$ corresponds to the approximation of $\boldsymbol{\omega}$ at iteration k. Notice that $Q(s,a)$ is not known and can be replaced by the temporal-difference (TD) target $r + \gamma \boldsymbol{\omega}^T \phi(s',a')$, where $a' = \arg\max_{a\in\mathcal{A}} \boldsymbol{\omega}_k^\top \phi(s',a)$ is the optimal action taken for state s' based on the current approximation available of parameters $\boldsymbol{\omega}_k$. Thus, $J(\boldsymbol{\omega}_k)$ can be expressed as

$$J(\boldsymbol{\omega}_k) = \sum_{(s,a,s',r)\in\mathcal{D}} \Big(r + \gamma \boldsymbol{\omega}^T \boldsymbol{\phi}(s',a') - \boldsymbol{\omega}_k^\top \boldsymbol{\phi}(s,a)\Big)^2. \tag{20}$$

Therefore, at iteration $k+1$, $\boldsymbol{\omega}_{k+1}$ can be approximated by solving the next non-constrained optimization problem,

$$\boldsymbol{\omega}_{k+1} = \arg\min_{\boldsymbol{\omega}} J(\boldsymbol{\omega}_k). \tag{21}$$

As can be seen from (21), at each iteration, the LSPI algorithm finds parameters $\boldsymbol{\omega}$ that minimizes the mean squared error between the TD target and $\hat{Q}(s,a)$ over all transitions samples available in $\mathcal{D}$. This process is repeated until a convergence criterion, defined as $||\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k|| \leq \varepsilon$, is met, where $||\cdot||$ corresponds to the L_2 norm and ε is a small number.

The LSPI algorithm has multiple advantages: First, linear functions are used to approximate the action-value function $Q(s,a)$, which allows the algorithm to handle MDPs with large and continuous $\mathcal{S}$ as well as to guarantee learning convergence. Second, at each iteration, the whole available batch of transitions samples are used to approximate $\boldsymbol{\omega}$, thus, increasing data efficiency. Third, different from the classic Q Learning algorithm, there is no need to define a learning rate, thus fewer hyper-parameters are required to be tuned. Interested readers are referred to [26] for more details on convergence and performance guarantee.

4. PV Inverters Dispatch Problem as an MDP

The PV inverters dispatch problem is modelled as a MDP as in [19]. If $P_{m,t}^{PV}$ represents the PV generation of the PV inverter connected at node m at time t, with voltage magnitude $V_{m,\phi,t}$, and $P_{m,t}^{PV}(1 - \Delta P_{m,t}^{PV})$ represents the PV generation after curtailing $\Delta P_{m,t}^{PV}$, then, these can not be equal at the same time step t. There should be a time delay between applying the PV curtailment action and the distribution system reaching a new steady-state, in which the PV inverter m perceives $V_{m,\phi,t+1}$. In other words, $V_{m,\phi,t+1}$ is the result of the distribution system reaching a steady-state considering the current PV generation power for the PV inverter m as $P_{m,t}^{PV}(1 - \Delta P_{m,t}^{PV})$, instead of $P_{m,t+1}^{PV}$, as shown in Fig. 1. The described modelling

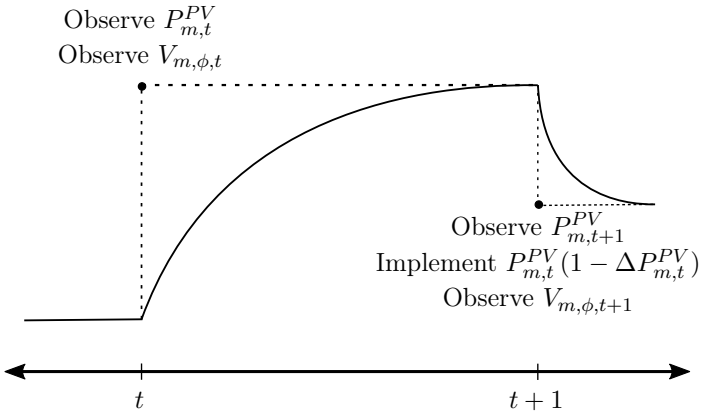

Figure 1: Transition representation used to model the PV inverters dispatch problem as a MDP as in [19]. Notice that $V_{m,\phi,t+1}$ is the result of the distribution system reaching steady-state considering the current PV generation power for the PV inverter m as $P^{PV}_{m,t}(1-\Delta P^{PV}_{m,t})$ instead of $P^{PV}_{m,t+1}$.

representation is the base for the definition of the transition model, later explained in Sec. 4.4.

An agent-based architecture is developed to facilitate the implementation of the proposed RL approach as in Fig. 2. Two types of agents are considered: PV Agents and a centralized Distribution System (DS) Agent. PV Agents are in charge of controlling the PV inverters, while the DS Agent is in charge of supervising the distribution network's operation, enforcing voltage magnitude constraints. Regarding these agents, the following assumptions that are assumed to hold:

1. The DS Agent is aware of the topology of the distribution network and can execute a power flow algorithm assuming the proposed curtailment actions by each PV Agent. After executing the power flow algorithm, the DS Agent shares with each PV Agent their expected voltage magnitude.

2. The PV Agents have enough computational resources to execute all the required processes of the LSPI algorithm locally, as explained in Sec. 4.5. Also, such agents only communicate and share data with the DS Agent and not between themselves, ensuring privacy. The shared data is limited to the proposed curtailment action and their PV power forecast.

The remaining definitions for the proposed RL approach, regarding state space, action space, reward function, and transition models, are presented next.

4.1. State Space

For the PV Agent m, connected to node $m \in \mathcal{N}$ of the distribution system at time t, define the state $s_{m,t} = (P^{PV}_{m,t}, \overline{V}_{m,t}) \,|\, s_{m,t} \in \mathcal{S}$, conformed by the tuple between the expected PV active power generation, $P^{PV}_{m,t}$, and the maximum voltage magnitude among the phases $\psi \in \mathcal{F}$ at time t, $\overline{V}_{m,t} = \max_{\psi\in\mathcal{F}}\{V_{m,\psi,t}\}$, containing only continuous values. Thus, the state space $\mathcal{S} \in \mathbb{R}^2$.

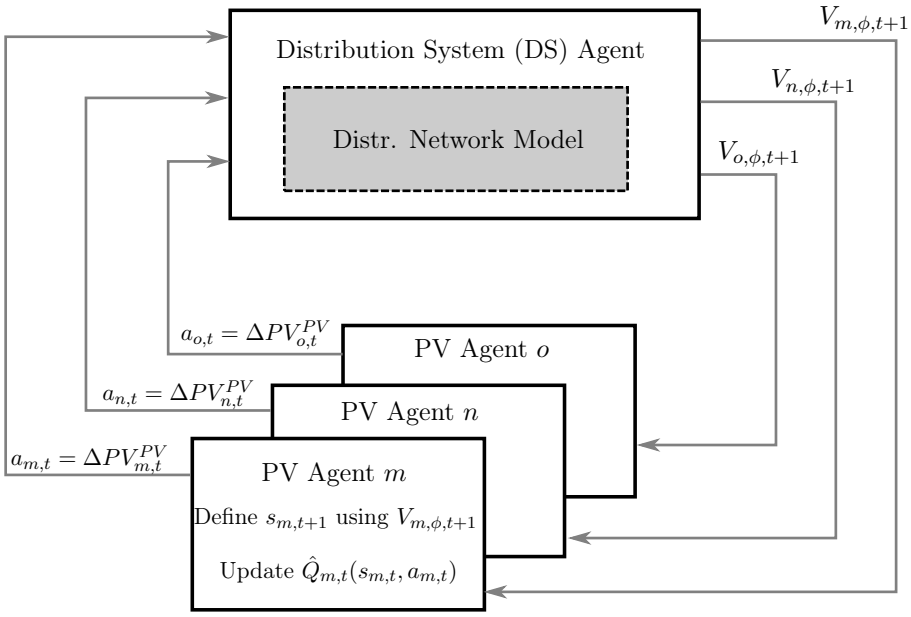

Figure 2: RL approach using an agent-based architecture. Two types of agents can be found: a DS Agent and the PV Agents. PV Agents share limited information with the DS Agent, while update of the $\hat{Q}(s,a)$ is done locally and in parallel.

4.2. Action Space

For the PV Agent m, actions are defined as a discrete PV power curtailment percentage of the expected PV power generation at time step t, i.e., $a_{m,t} = \Delta P_{m,t}^{PV}$. Thus, the action space is defined as $\mathcal{A} = \{0, \Delta, 2\Delta, ..., 1.0\}$, where Δ defines the discretization step used.

4.3. Reward Function

As discussed in Sec. 2, the centralized objective of the optimal dispatch of PV inverters problem is to solve local voltage issues while minimizing the total amount of PV active power curtailed. The centralized objective function in (1) can be translated as a local reward function for each PV Agent m, $\mathcal{R}_{m,t}(\cdot)$, as follows:

$$\mathcal{R}_{m,t}(s_{m,t}, a_{m,t}, s'_{m,t}) = -\delta_{\mathcal{A}} \Delta P_{m,t}^{PV} + \min\{0, \delta_V (\frac{\overline{V} - \underline{V}}{2} - |V_0 - \overline{V}_{m,t}|)\}, \tag{22}$$

where $\delta_{\mathcal{A}}$ and δ_V are positive penalty parameters. In expression (22), the first term corresponds to a penalty term proportional to the action taken. PV Agents need to chose lower value actions $\Delta P_{m,t}^{PV}$, thus, reducing the total amount of PV active power curtailed; while the second term penalizes actions that result in voltage magnitude violation. The second term of expression (22) in terms of the voltage magnitude is depicted in Fig. 3. Notice that if the maximum voltage magnitude over all phases of the PV Agent m, i.e., $\overline{V}_{m,t}$, is above $\underline{V}$ or below $\overline{V}$, the penalty term is equal to zero, otherwise, the penalty increases with slope $-\delta_V$.

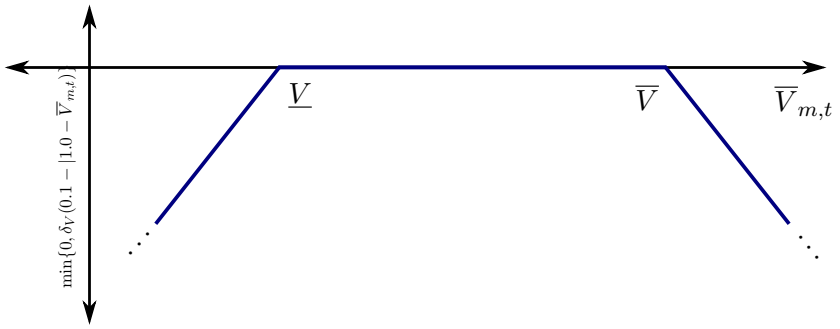

Figure 3: Representation of the second term of the reward function in (22) related to the penalty due to voltage magnitude violation as a function of the maximum voltage over the phases $\overline{V}_{m,t}$.

4.4. Transition Model

Once the PV Agents share the proposed PV curtailment percentages (actions) with the DS Agent, as explained on the MDP modelling in Sec. 4 and the state definition provided in Sec. 4.1, the DS Agent solves a nonlinear power flow to estimate $V_{m,\phi,t+1}$ for all the PV Agents m. This information is used by the PV Agents to define the values of the state transition, from $s_{m,t} = (P^{PV}_{m,t}, \overline{V}_{m,t})$ to $s'_{m,t} = (P^{PV}_{m,t+1}, \overline{V}_{m,t+1})$, knowing that $\overline{V}_{m,t+1}$ is a result of implementing actions $a_{m,t} = \Delta P^{PV}_{m,t}$ in the current state. The definition of states $s_{m,t}$ and $s'_{m,t}$ are needed in order to update the approximation of the action-value function $\hat{Q}_m(s, a)$.

4.5. Action-Value Function Approximation

The algorithm used to approximate (learn) the action value function $\hat{Q}_m(s, a)$ is based on the LSPI algorithm presented in Sec. 3.2. Although the LSPI is an efficient algorithm to handle data, it may become computationally intractable as the action space $\mathcal{A}$ increases. To overcome this issue, instead of approximating a general $Q(s, a)$, a separate approximation is defined for each action $a \in \mathcal{A}$ and for each time step $t \in \mathcal{T}$. Thus, each PV Agent m learns an approximated optimal action-value function

$$\hat{Q}_m(s_{m,t}, a^{(l)}_{m,t}) \;=\; \boldsymbol{\omega}^{(l)\top}_{m,t} \boldsymbol{\phi}^{(l)}(s_{m,t}, a^{(l)}_{m,t}), \tag{23}$$

where $a^{(l)}_{m,t}$ is the l-th component of the action space $\mathcal{A}$, $\boldsymbol{\omega}^{(l)}_{m,t}$ are the parameters associated with action $a^{(l)}_{m,t}$, and $\boldsymbol{\phi}^{(l)}(\cdot, \cdot)$ is a vector of basis functions. In this paper, we propose to use radial basis functions (RBFs) of the form $e^{(-(x-x_c)^2/\sigma^2)}$, where x is a generic variable[1] of the state s, x_c is a generic (and constant) center related to the generic variable x, and σ is the standard deviation of the RBFs, forming the following feature vector

$$\boldsymbol{\phi}^{(l)}(s_{m,t}, a^{(l)}_{m,t}) = \left[1, P^{PV}_{m,t}, e^{-(\overline{V}_{m,t}-V_{c_1})^2/\sigma^2}, e^{-(\overline{V}_{m,t}-V_{c_2})^2/\sigma^2}, \ldots, e^{-(\overline{V}_{m,t}-V_{c_\kappa})^2/\sigma^2}\right]^\top, \tag{24}$$

[1]For instance, in our state definition in Sec. 4.1, x can be either $P^{PV}_{m,t}$ or $\overline{V}_{m,t}$.

Algorithm 1: LPSI Algorithm used by PV Agent m to learn $\boldsymbol{\omega}_m$ and approximate the action-value function $\hat{Q}_m(s,a)$.

Input:

$\mathcal{D}_m$: Transition samples for PV Agent m
$\boldsymbol{\phi}(\cdot,\cdot)$: RBFs approximation
γ : Discount factor
ε : Small threshold value
c : Small number

Output:

$\boldsymbol{\omega}_m$: updated parameter vector to approximate $\hat{Q}_m(s,a)$ as $\boldsymbol{\phi}(s,a)^\top \boldsymbol{\omega}_m$.

Initialize $\boldsymbol{\omega}_{m,-1} = \mathbf{0}_f$ and $k = 0$
while $||\boldsymbol{\omega}_{m,k+1} - \boldsymbol{\omega}_{m,k}|| > \varepsilon$ **do**
  Initialize $\boldsymbol{B}_0 = c\boldsymbol{I}_{f\times f}, \boldsymbol{b}_0 = \mathbf{0}_f, i = 0$
  for $(s,a,r,s') \in \mathcal{D}_m$ **do**
    $a' = \arg\max_{a\in\mathcal{A}} \boldsymbol{\phi}(s,a)^\top \boldsymbol{\omega}_{m,k}$
    $\boldsymbol{B}_i = \boldsymbol{B}_{i-1} + \boldsymbol{\phi}(s,a)(\boldsymbol{\phi}(s,a) - \gamma\boldsymbol{\phi}(s',a'))^\top$
    $\boldsymbol{b}_i = \boldsymbol{b}_{i-1} + r\boldsymbol{\phi}(s,a)$
    $i = i + 1$
  $\boldsymbol{\omega}_{m,k} = \boldsymbol{B}^{-1}_{|\mathcal{D}_m|}\boldsymbol{b}_{|\mathcal{D}_m|}$
  $k = k + 1$
Set $\boldsymbol{\omega}_m = \boldsymbol{\omega}_{m,k}$

where κ corresponds to a positive number that indicates the total number of RBFs used. Based on this definition, notice that $\boldsymbol{\phi}^{(l)}(s_{m,t}, a^{(l)}_{m,t}) : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^{(\kappa+2)}$, and that $\phi(s,a) : \mathcal{S}\times\mathcal{A}\times\mathcal{T} \to \mathbb{R}^f$, where $f = (\kappa+2)\times|\mathcal{T}|\times|\mathcal{A}|$. Therefore, as the function approximation $\boldsymbol{\phi}(s,a)$ is a collection of feature vectors of the form shown in (24), specifically, when constructing $\boldsymbol{\phi}(s,a)$ for the pair $(s_{m,t}, a^{(l)}_{m,t})$, the remaining terms of $\boldsymbol{\phi}(s,a)$ for all $a \in \mathcal{A}$ different from $a^{(l)}_{m,t}$ and time steps different from $t \in \mathcal{T}$ are set to $\mathbf{0}_{\kappa+2}$, as shown next

$$\boldsymbol{\phi}(s,a) = \begin{pmatrix} \mathbf{0}_{\kappa+2} \\ \vdots \\ \boldsymbol{\phi}^{(l)}(s_{m,t}, a^{(l)}_{m,t}) \\ \vdots \\ \mathbf{0}_{\kappa+2} \end{pmatrix} \in \mathbb{R}^f. \tag{25}$$

Based on this definition, the LPSI algorithm shown in Algorithm 1 is used to learn parameters $\boldsymbol{\omega}_m \in \mathbb{R}^f$ to approximate $\hat{Q}(s,a)$. Notice that Algorithm 1 requires as input a collection of transition samples $\mathcal{D}_m$. The procedure used by each PV Agent m to obtain these collections of samples is explained next.

4.6. Overview of the Proposed RL Approach

The proposed RL approach builds on the agent-based architecture shown in Fig. 2 and follows the step-by-step procedure presented in Algorithm 2, implemented over a rolling time horizon strategy. Initially, the

time horizon is divided into larger size time steps $t_h \in \mathcal{T}_h$, e.g., $\mathcal{T}_h$ can be a set of time in hours, while $\mathcal{T}$ a granular partition of each hour. In other words, if the duration between two time steps $t \in \mathcal{T}$ is 15 min, i.e. $\Delta t = 0.25$ h, thus set $\mathcal{T}$ will have a total of three partitions, i.e., $\mathcal{T} = \{t_h, t_h + \Delta t, t_h + 2\Delta t, t_h + 3\Delta t\}$. By doing this, the RL algorithm is executed each $t_h \in \mathcal{T}_h$, while decisions are taken for the next time steps $\mathcal{T} = \{t_h, t_h + \Delta t, ..., t_h + (n-1)\Delta t\}$, where n is the number of partitions. This approach reduces the need of a long-term forecast of PV generation by PV Agents, as well as the need of advanced computational infrastructure to execute Algorithm 1. Additionally, limiting the LPSI algorithm to take decisions only for a few future time steps allow the proposed RL approach to adapt to changes in the system dynamics.

According to Algorithm 2, the following procedure is executed to learn parameters $\boldsymbol{\omega}_{m,t_h}$, which are used to take the optimal actions $a^*_{m,t}$ for $t \in \mathcal{T}$ as $a^*_{m,t} = \arg\max_{a \in \mathcal{A}} \boldsymbol{\phi}(s_{m,t}, a)^\top \boldsymbol{\omega}_{m,t_h}$. In t_h, for each time step $t \in \mathcal{T}$, each PV Agent m either chooses a random action from set $\mathcal{S}$ or the best action obtained using the current estimation of $\boldsymbol{\omega}_{m,t_h}$ (also known as ϵ-greedy). In a real implementation, this procedure is done in parallel by all PV Agents m, using only its local computational infrastructure, as shown in Fig. 2. Notice in Algorithm 2 that as the number of iterations increases, parameter ϵ_j decreases, reducing the chance of selecting random actions, thus allowing controlling the balance between exploration and exploitation. Once all PV Agents have individually proposed one action, they share this information with the DS Agent, which uses the transition model explained in Sec. 4.4. The output information from this step (see also Fig. 2) is shared with each PV Agent, and it is used to estimate the reward $r_{m,t}$ and construct the next state $s'_{m,t}$, as explained in Sec. 4.3 and Sec. 4.4. After all this, each PV Agent m updates its collection of samples $\mathcal{D}_m$ and improves its current approximation of $\boldsymbol{\omega}_{m,t_h}$ by executing Algorithm 1. This procedure is done until a maximum number iterations J is reached.

5. Results and Discussion

In this section, simulation results are presented. Comparisons with the optimal solution of the centralized PV dispatch formulation in Sec. 2 are also presented.

5.1. Simulation Setup

The proposed RL approach has been implemented in Python language and executed on a notebook with a processor Intel Core i7 and 16 RB RAM memory. The unbalanced 25-bus system shown in Fig. 4 is used, load consumption per node, as well as resistance and reactance data, can be found in [33][2]. The load level per

[2]To increase the number of voltage issues in the distribution system, the resistance/reactance ratio has been increased by a factor of 3.

Algorithm 2: RL-Based Approach used to define the optimal dispatch power of the PV Agents.

Input:

J : Maximum number of iterations

ϵ_0 : Parameter to control exploration

η : Decay rate to control exploration

Output:

$\boldsymbol{\omega}_{m,t_h}$: updated parameter vectors for each $t_h \in \mathcal{T}_h$

$a^*_{m,t}$: Optimal actions to implement for each $t \in \mathcal{T}$

Initialize $j = 0$, $\boldsymbol{\omega}_{m,t_h} = 0,\ \forall m \in \mathcal{N}, t_h \in \mathcal{T}_h$,

for $t_h \in \mathcal{T}_h$ **do**

  Approximate action-value function as follows: **while** $j < J$ **do**

    $\epsilon_j = \min\{0.05, \epsilon_0/(1 + j\eta)\}$

    for $t \in \mathcal{T}$ **do**

      for $m \in \mathcal{N}$ **do**

        if $\epsilon_j < \xi$ **then**

          $a_{m,t} = \text{random}(a \in \mathcal{A})$

        else

          $a_{m,t} = \arg\max_{a \in \mathcal{A}} \boldsymbol{\phi}(s_{m,t}, a)^\top \boldsymbol{\omega}_{m,t_h}$

      $\overline{V}_{m,t+1} \leftarrow$ DS Agent$(a_{m,t}, s_{m,t})$

      for $m \in \mathcal{N}$ **do**

        $r_{m,t} \leftarrow \mathcal{R}_{m,t}(s_{m,t}, a_{m,t}, s'_{m,t})\ \mathcal{D}_m = \mathcal{D}_m \cup (s_{m,t}, a_{m,t}, s'_{m,t}, r_{m,t})$

        $s_{m,t} = s'_{m,t}$

        $\boldsymbol{\omega}_{m,t_h} \leftarrow$ Execute Algorithm 1

  Define optimal actions as follows:

  for $t \in \mathcal{T}$ **do**

    for $m \in \mathcal{N}$ **do**

      $a^*_{m,t} = \arg\max_{a \in \mathcal{A}} \boldsymbol{\phi}(s_{m,t}, a)^\top \boldsymbol{\omega}_{m,t_h}$

time step is shown in Fig. 5. In total, three PV Agents are considered, located at nodes $m = 13, 17, 25$, with a nominal capacity of 1500 kW, 1800 kW, and 2000 kW, respectively. The irradiance profile used for simulations is shown in Fig. 5. All PV inverters are assumed to operate with a unity power factor. The nominal voltage of the distribution system in Fig. 4 is 4.16 kV, set up to a value of 1.03 p.u. to avoid undervoltage problems during the peak consumption, while the minimum and maximum voltage magnitude level have been defined as 0.90 p.u. and 1.10 p.u., respectively. The base power used for the power flow formulation and the state representations is 1000 kVA.

Regarding the RL algorithm, six RBFs are used for the voltage magnitude representation shown in (25), with centers located at $V_{c_\kappa} = \{0.90,\ 0.94,\ 0.98,\ 1.02,\ \ 1.06,\ 1.10\}$. The remaining parameters are set as $\gamma = 0.95$, $\epsilon_0 = 0.5$, $c = 0.1$, $\sigma = 0.1$, $\varepsilon = 1 \times 10^{-3}$, $\delta_V = 10 \times 10^5$ and $\delta_{\mathcal{A}} = 500$. The action space $\mathcal{A}$ is discretized using $\Delta = 0.05$. The maximum number of sample transitions that can be stored in $\mathcal{D}_m$ is set to 2000, while the maximum number of iterations for Algorithm 2 is set to $J = 1000$.

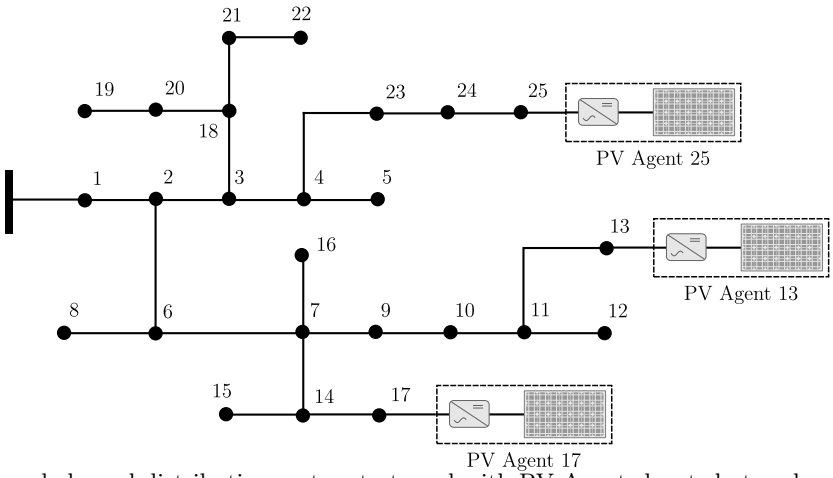

Figure 4: 25-node unbalanced distribution system test used with PV Agents located at nodes $m = 13, 17$ and 25.

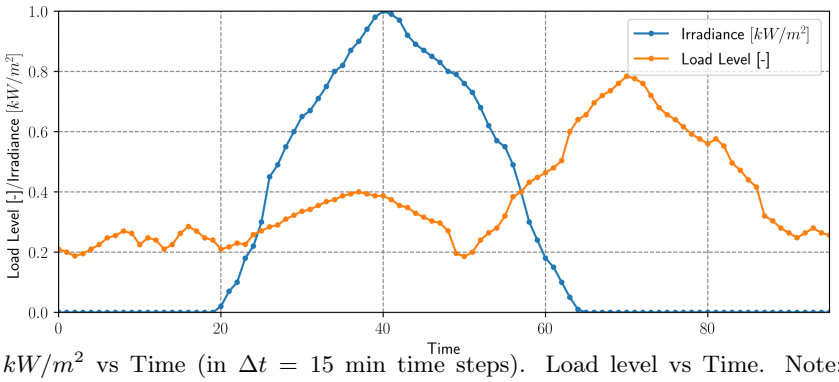

Figure 5: Irradiance in kW/m^2 vs Time (in $\Delta t = 15$ min time steps). Load level vs Time. Note: All active and reactive consumption power for each node is multiplied by this load level factor, where a load level factor of 1.0 is equivalent to the data provided in [33].

5.2. Validation and Comparison

Fig. 6 shows the learning results obtaining after executing Algorithm 2 for hour $t = 40$ (i.e., 12:00). As each hour is discretized into shorter time steps of $\Delta t = 0.25$ h, Algorithm 2 provides the optimal actions $\Delta PV_{m,t}^{PV}$ for time steps t =12:00, 12:15, 12:30, and, 12:45. Fig. 6(a) presents the typical learning curve obtained when training RL algorithms, in which it is possible to observe how the reward improves over the learning process. As described in Algorithm 2, at the beginning of the learning process, as the actions proposed by the PV Agent are defined randomly, lower (negative) reward values are obtained. Nevertheless, as more and more samples are added by the PV Agents to the set $\mathcal{D}_m$, the estimation of the parameters $\boldsymbol{\omega}_{m,t_h}$ improves, leading to the PV Agents to propose decisions that receive higher reward values.

Due to the randomness associated with the exploration of the state-action space during the learning process, different estimation of parameters $\boldsymbol{\omega}_{m,t_h}$ can be obtained after convergence in different executions. As a consequence, different optimal actions can also be defined. To assess this, Fig. 6$(b)-(c)$ shows the mean and the standard deviation of the rewards obtained by each of the PV Agents over five different executions. As expected, higher standard deviations are observed at the beginning of the learning process due to the exploration process. Nevertheless, as the learning process continues, convergence to higher reward actions is

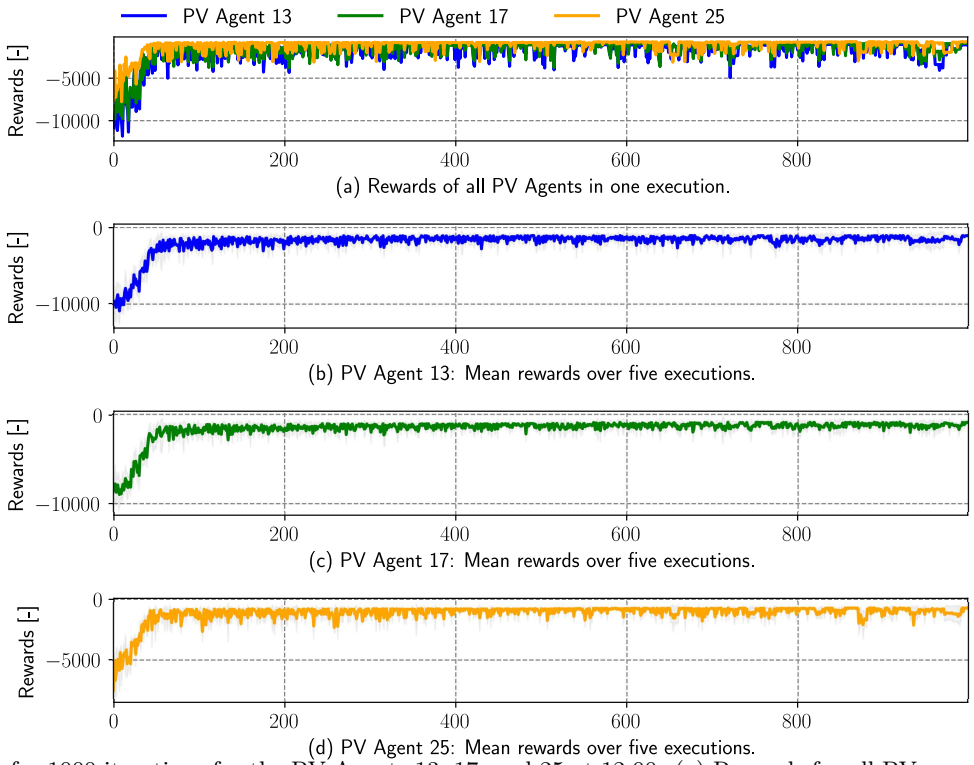

Figure 6: Rewards for 1000 iterations for the PV Agents 13, 17, and 25 at 12:00. (a) Rewards for all PV agents in one execution. (b), (c) and (d) Mean and standard deviation of rewards for PV Agents 13, 17, and 25, respectively, over five different executions.

attained. In this case, observe that even at the end of the learning process the mean does not converge to a single value (and thus the standard deviation is different than zero). This is due to the fact that exploration is never finished, i.e., ϵ_j never becomes zero. This is done to continuously improve the estimation of parameters $\boldsymbol{\omega}_{m,t_h}$, allowing the PV Agent to discover better solutions (exploitation process).

To assess the quality of the solutions obtained in different executions of the proposed RL approach, comparisons are presented in Table 1 for each PV Agent. In terms of actions $\Delta P^{PV}_{m,t}$, all PV Agents were able to achieve the optimal solution in at least one execution. The optimal solutions provided in Table 1 were obtained after solving the centralized model presented in Sec. 2 using a continuous and a discrete NLP formulation. As expected, the optimal solutions obtained by the proposed RL approach enforce the voltage magnitude within the required limits. Notice in Table 1 that although the optimal solution is attained by the PV Agents in at least one execution, different voltage magnitude results are obtained when compared with the optimal solution provided by the NLP formulation. This is due to the fact that, for instance, PV Agent 13 may have obtained the optimal solution, while the remaining PV Agents may have converged to a quasi-optimal solution. As a result, different voltage magnitude profiles are observed. In terms of the worst solutions obtained, notice that they differ from the optimal solution in curtailing more PV power than necessary to enforce voltage magnitude regulation. In this case, for the worst solutions, PV Agents 13, 17, and 25, curtail 1.23%, 3.65%, and 2.47%, respectively, more than the optimal solution. Nevertheless, when

comparing the centralized NLP formulation, the main advantage of the proposed RL approach relies on how good quality solutions can be attained in a distributed fashion, performing computations locally at the PV Agents and by sharing limited information.

Table 1: Comparison of the obtained results for the PV Agents over five executions for the time steps at 12:00, 12:15, 12:30, and, 12:45.

	PV Agent 13				PV Agent 17				PV Agent 25			
	Control Actions ($\Delta P^{\text{PV}}_{m,t}$) [%]											
t	12:00	12:15	12:30	12:45	12:00	12:15	12:30	12:45	12:00	12:15	12:30	12:45
Best	0.50	0.55	0.55	0.5	0.45	0.45	0.40	0.40	0.35	0.35	0.35	0.30
Worst	0.55	0.55	0.60	0.50	0.50	0.50	0.45	0.50	0.35	0.4	0.35	0.35
Optimal Control	0.50	0.55	0.55	0.5	0.40	0.45	0.45	0.40	0.35	0.35	0.35	0.30
Optimal Control*	0.46	0.51	0.50	0.47	0.39	0.42	0.41	0.38	0.31	0.35	0.33	0.30
	Voltage Magnitude ($\max_{\phi}\{V_{m,\phi,t}\}$) [p.u.]											
t	12:00	12:15	12:30	12:45	12:00	12:15	12:30	12:45	12:00	12:15	12:30	12:45
Best	1.0891	1.0916	1.0931	1.0910	1.0910	1.0932	1.0989	1.0960	1.0959	1.0977	1.0963	1.0987
Worst	1.0920	1.0897	1.0881	1.0960	1.0860	1.0905	1.0942	1.0876	1.0950	1.0934	1.0976	1.0940
No Control	1.1686	1.1753	1.1707	1.1633	1.1596	1.1656	1.1613	1.1544	1.1415	1.1453	1.1416	1.1361
Optimal Control	1.0945	1.0950	1.0931	1.0960	1.0960	1.0962	1.0942	1.0967	1.0957	1.0991	1.0969	1.0987
Optimal Control*	1.0998	1.1000	1.1000	1.0998	1.0994	1.1000	1.1000	1.0997	1.1000	1.1000	1.1000	1.0993
	Total PV Power Curtailed [%]											
Best	39.53				32.89				24.72			
Worst	40.76				35.78				27.19			
Optimal Control	39.53				32.13				24.72			
Optimal Control*	36.56				29.90				24.23			
	Total Reward [-]											
Best	-1050.0				-850.0				-680.0			
Worst	-1100.0				-975.0				-725.0			
Optimal Control	-1050.00				-850.0				-680.0			

* Optimal solution solving the continuous NLP formulation in Sec. 2.

5.3. Computational Time Assessment

In order to be able to implement the proposed RL approach, the total computational time required for the PV Agents to achieve convergence must fit within the time step discretization of the rolling horizon strategy used, which in this case is 1 hour ($\Delta t_h = 1$ h). To assess this, the wall-clock time of the proposed RL approach was measured, resulting in an average time per iteration (of Algorithm 2) lower than 2 s, and in an average total time lower than 32 min (all PV Agents perform computations in parallel). As these average results are way below the time step discretization of 1 hour, the proposed RL approach can be implemented to operate in real-time. Notice that the most computationally expensive operation within the proposed approach corresponds to the last step of Algorithm 1, in which the inverse of matrix $\boldsymbol{B}_{|\mathcal{D}_m|} \in \mathbb{R}^{|\mathcal{D}_m|}$ needs to be calculated to estimate parameters $\boldsymbol{\omega}_{m,k}$. The inversion of this matrix can be avoided by using the Sherman-Morrison formula, which allows to estimate it iteratively, as explained in [26].

5.4. Full-Time Horizon Operation

To assess the effectiveness of the proposed RL approach for different irradiance and consumption conditions, continuous simulations were executed for a time horizon of 24 h considering the irradiance and load level consumption data shown in Fig. 5. Obtained results are discussed based on Fig. 7, which presents the rewards for all PV Agents over the learning process for hours at 6:00, 7:00, and 8:00. These hours are selected as the irradiance increases as time passes. In terms of actions, as the irradiance is relatively low at 6:00, curtailment actions are not required to enforce voltage magnitude limits. This can be seen in Fig. 7 as the maximum reward obtained by the PV Agents is zero, which necessarily implies that no PV curtailment is performed. Nevertheless, as the irradiance conditions change during the next hours at 7:00 and 8:00, curtailment actions might be required to enforce voltage magnitude limits. In these cases, as shown in Fig. 7, the proposed RL approach is able to converge to good quality solutions when compared with the optimal reward (obtained using the centralized NLP formulation). In operational terms, the total PV curtailment for PV Agents 13, 17 and 25, was estimated to be 1.6 %, 2.13 %, and 0.76 %, respectively, higher than the centralized optimal solution, which validated the effectiveness of the proposed RL approach to obtain good quality actions during continuous operation. Notice that all the defined curtailment actions during continuous operation enforce all voltage magnitude constraints during the full-time horizon, as shown in Fig. 8.

Finally, notice in Fig. 7 that in case of continuous operations, once the optimal curtailment actions are defined for time step t_h, parameters $\boldsymbol{\omega}_{m,t_{h+1}}$ for time step t_{h+1} are initialized as zero. This is done as the LPSI algorithm is biased towards the current system's state, and thus, if $\boldsymbol{\omega}_{m,t_h}$ are used as an initial approximation for $\boldsymbol{\omega}_{m,t_{h+1}}$, exploration will be limited to the vicinity of the actions obtained after convergence in time step t_h, leading even to unfeasible solutions.

6. Conclusion

In this paper, an reinforcement learning (R)L-based approach to optimally dispatch PV inverters in distribution networks was presented. The proposed approach takes advantage of a decentralized architecture that enables all computational processes to be performed locally by the PV Agents. To avoid the computational burden usually associated with Markov Decision Processes (MDPs) with continuous state and action spaces, a rolling horizon strategy was used, together with a computationally efficient learning algorithm used to model the action-value function. Results showed that in several executions, the proposed RL approach converged to the optimal solution, and in the worst case, converged to solutions with an excess of PV curtailment lower than 2.5 %. However, in both cases, it was found that the solution still enforces voltage magnitude limits. Continuous operation of the proposed RL approach was also tested, obtaining similar results. Compared

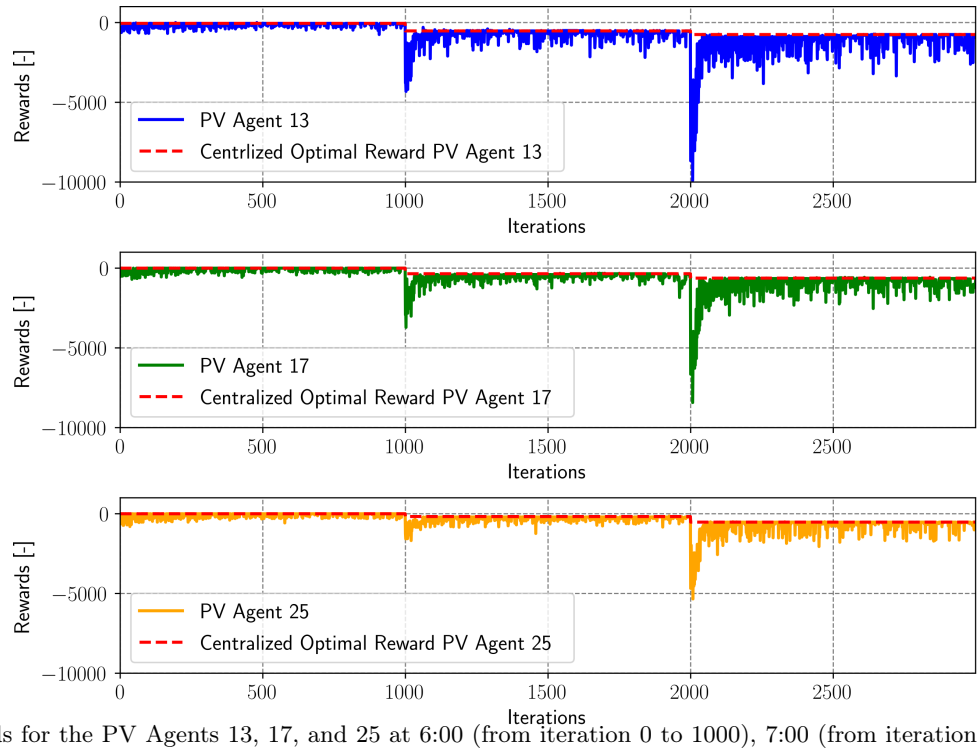

Figure 7: Rewards for the PV Agents 13, 17, and 25 at 6:00 (from iteration 0 to 1000), 7:00 (from iteration 1000 to 2000), and at 8:00 (from iteration 2000 to 3000), when executed in continuous operation for the full-time horizon of 24 h. The red dashed lined represents the optimal reward obtained when using the centralized NLP formulation.

with other distributed optimization-based approaches, RL approaches offer the advantage of straightforward implementation while guarantying convergence to good quality solutions. Moreover, RL approaches do not rely on strict convexity assumptions as long as linear parametric functions are used to model the action-value

References

[1] Renewables 2020, techreport, International Energy Agency, 2020. URL: `https://www.iea.org/reports/renewables-2020/solar-pv-abstract`.

[2] Y. Hu, W. Liu, W. Wang, A two-layer volt-var control method in rural distribution networks considering utilization of photovoltaic power, IEEE Access 8 (2020) 118417–118425. doi:10.1109/ACCESS.2020.3003426.

[3] C. Long, L. F. Ochoa, Voltage control of PV-rich LV networks: OLTC-fitted transformer and capacitor banks, IEEE Trans. Power Systems 31 (2016) 4016–4025.

[4] G. Tévar, A. Gómez-Expósito, A. Arcos-Vargas, M. Rodríguez-Montañés, Influence of rooftop PV generation on net demand, losses and network congestions: A case study, Int. J. of Electrical Power & Energy Systems 106 (2019) 68 – 86.

[5] R. Tonkoski, L. A. C. Lopes, T. H. M. El-Fouly, Coordinated active power curtailment of grid connected PV inverters for overvoltage prevention, IEEE Trans. Sustainable Energy 2 (2011) 139–147. doi:10.1109/TSTE.2010.2098483.

[6] S. Ghosh, S. Rahman, M. Pipattanasomporn, Distribution voltage regulation through active power curtailment with pv inverters and solar generation forecasts, IEEE Trans. Sustainable Energy 8 (2017) 13–22. doi:10.1109/TSTE.2016.2577559.

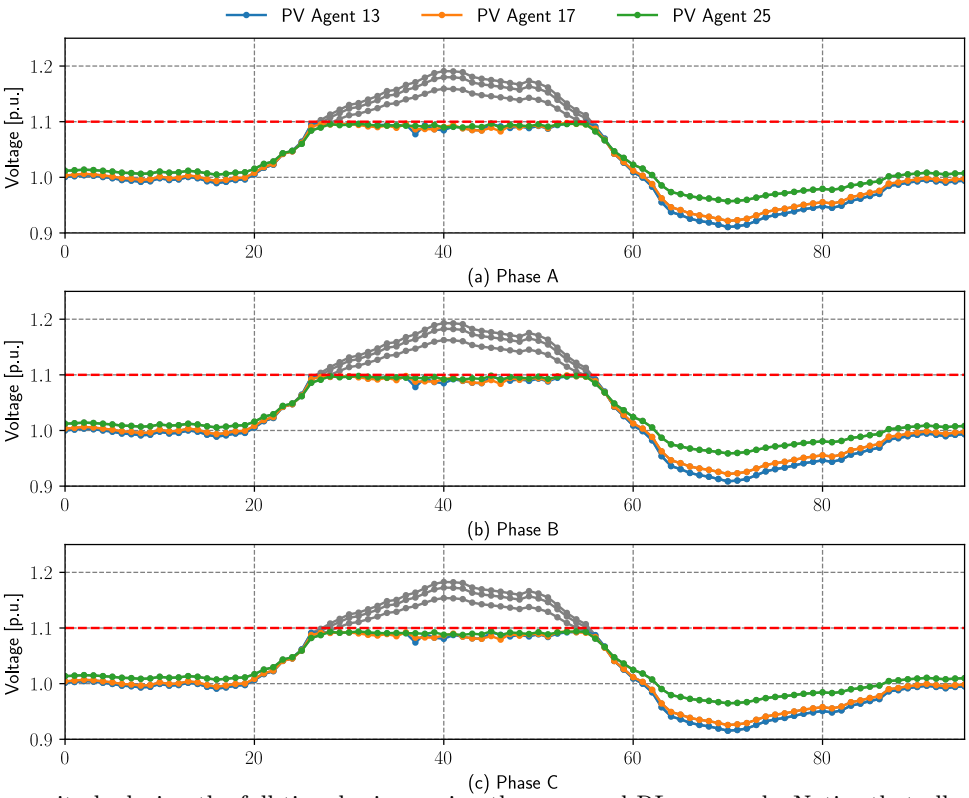

Figure 8: Voltage magnitude during the full-time horizon using the proposed RL approach. Notice that all actions implemented guaranteed that voltage magnitude constraints are enforced, when compared with the voltage magnitude profile when no control is applied (grey lines).

[7] P. P. Vergara, T. T. Mai, A. Burstein, P. H. Nguyen, Feasibility and performance assessment of commercial PV inverters operating with droop control for providing voltage support services, in: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2019, pp. 1–5.

[8] D. Gebbran, S. Mhanna, Y. Ma, A. C. Chapman, G. Verbič, Fair coordination of distributed energy resources with volt-var control and pv curtailment, Applied Energy 286 (2021) 116546.

[9] L. Wang, R. Yan, T. K. Saha, Voltage management for large scale pv integration into weak distribution systems, IEEE Trans. Smart Grid 9 (2018) 4128–4139. doi:10.1109/TSG.2017.2651030.

[10] E. Dall'Anese, S. V. Dhople, G. B. Giannakis, Optimal dispatch of photovoltaic inverters in residential distribution systems, IEEE Trans. Sustainable Energy 5 (2014) 487–497. doi:10.1109/TSTE.2013.2292828.

[11] E. Dall'Anese, S. V. Dhople, B. B. Johnson, G. B. Giannakis, Decentralized optimal dispatch of photovoltaic inverters in residential distribution systems, IEEE Trans. Energy Conversion 29 (2014) 957–967. doi:10.1109/TEC.2014.2357997.

[12] T. T. Mai, A. N. M. Haque, P. P. Vergara, P. H. Nguyen, G. Pemen, Adaptive coordination of sequential droop control for pv inverters to mitigate voltage rise in pv-rich lv distribution networks, Electric Power Systems Research 192 (2021) 106931.

[13] E. Dall'Anese, S. V. Dhople, B. B. Johnson, G. B. Giannakis, Optimal dispatch of residential photovoltaic inverters under forecasting uncertainties, IEEE J. of Photovoltaics 5 (2015) 350–359. doi:10.1109/JPHOTOV.2014.2364125.

[14] A. M. Howlader, S. Sadoyama, L. R. Roose, Y. Chen, Active power control to mitigate voltage and frequency deviations for the smart grid using smart pv inverters, Applied Energy 258 (2020) 114000.

[15] C. Feng, Y. Liu, J. Zhang, A taxonomical review on recent artificial intelligence applications to PV integration into power grids, International Journal of Electrical Power Energy Systems 132 (2021) 107176. URL: https://linkinghub.elsevier.com/retrieve/pii/S0142061521004154. doi:10.1016/j.ijepes.2021.107176.

[16] E. O. Arwa, K. A. Folly, Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review, IEEE Access 8 (2020) 208992–209007. doi:10.1109/ACCESS.2020.3038735.

[17] D. Cao, W. Hu, J. Zhao, G. Zhang, B. Zhang, Z. Liu, Z. Chen, F. Blaabjerg, Reinforcement learning and its applications in modern power and energy systems: A review, J. of Modern Power Systems and Clean Energy 8 (2020) 1029–1042. doi:10.35833/MPCE.2020.000552.

[18] C. Li, C. Jin, R. Sharma, Coordination of pv smart inverters using deep reinforcement learning for grid voltage regulation, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 1930–1937. doi:10.1109/ICMLA.2019.00310.

[19] R. E. Helou, D. Kalathil, L. Xie, Fully decentralized reinforcement learning-based control of photovoltaics in distribution grids for joint provision of real and reactive power, arXiv preprint arXiv:2008.01231 (2020).

[20] H. Liu, W. Wu, Online multi-agent reinforcement learning for decentralized inverter-based volt-var control, IEEE Transactions on Smart Grid (2021) 1–1. doi:10.1109/TSG.2021.3060027.

[21] Y. Zhang, X. Wang, J. Wang, Y. Zhang, Deep reinforcement learning based volt-var optimization in smart distribution systems, IEEE Transactions on Smart Grid 12 (2021) 361–371. doi:10.1109/TSG.2020.3010130.

[22] E. Samadi, A. Badri, R. Ebrahimpour, Decentralized multi-agent based energy management of microgrid using reinforcement learning, International Journal of Electrical Power Energy Systems 122 (2020) 106211. URL: https://linkinghub.elsevier.com/retrieve/pii/S0142061520304877. doi:10.1016/j.ijepes.2020.106211.

[23] C. Guo, X. Wang, Y. Zheng, F. Zhang, Optimal energy management of multi-microgrids connected to distribution system based on deep reinforcement learning, International Journal of Electrical Power Energy Systems 131 (2021) 107048. URL: https://linkinghub.elsevier.com/retrieve/pii/S0142061521002878. doi:10.1016/j.ijepes.2021.107048.

[24] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, Cambridge, MA, USA, 2018.

[25] H. Xu, A. D. Domínguez-García, P. W. Sauer, Optimal tap setting of voltage regulation transformers using batch reinforcement learning, IEEE Transactions on Power Systems 35 (2020) 1990–2001. doi:10.1109/TPWRS.2019.2948132.

[26] M. G. Lagoudakis, R. Parr, Least-squares policy iteration, J. of Machine Learning Research (2003) 1107–1149.

[27] P. Vergara, J. Lopez, M. Rider, L. Da Silva, Optimal Operation of Unbalanced Three-Phase Islanded Droop-Based Microgrids, IEEE Transactions on Smart Grid 10 (2019). doi:10.1109/TSG.2017.2756021.

[28] C. J. Watkins, P. Dayan, Q-learning, Machine learning 8 (1992) 279–292.

[29] J. G. Vlachogiannis, N. D. Hatziargyriou, Reinforcement learning for reactive power control, IEEE Trans. Power Systems 19 (2004) 1317–1325. doi:10.1109/TPWRS.2004.831259.

[30] L. Buşoniu, A. Lazaric, M. Ghavamzadeh, R. Munos, R. Babuška, B. De Schutter, Least-Squares Methods for Policy Iteration, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 75–109.

[31] M. Riedmiller, Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method, in: Machine Learning: ECML 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 317–328.

[32] D. Ernst, P. Geurts, L. Wehenkel, Tree-based batch mode reinforcement learning, J. of Machine Learning Research 6 (2005) 503–556.

[33] G. K. V. Raju, P. R. Bijwe, Efficient reconfiguration of balanced and unbalanced distribution systems for loss minimisation, IET Gen. Trans. Distr. 2 (2008) 7–12. doi:10.1049/iet-gtd:20070216.

Appendix A

The notation used throughout this paper is reproduced below for reference.

Notation

Sets:

$\mathcal{A}$	Set of actions (action space)
$\mathcal{F}$	Set of phases {A, B, C}.
$\mathcal{L}$	Set of lines.
$\mathcal{N}$	Set of nodes.
$\mathcal{S}$	Set of states (state space)
$\mathcal{T}$	Set of time-intervals.

Indexes:

ϕ, ψ	Phases ϕ, $\psi \in \mathcal{F}$.
km, mn	Line km, $mn \in \mathcal{L}$.
n, m	Nodes $n, m \in \mathcal{N}$.
t, t_h	Time steps $t \in \mathcal{T}$, $t_h \in \mathcal{T}_h$.

Parameters:

Δt	Time duration between two consecutive time steps.
$\overline{I}_{mn}$	Maximum line current limit.
$P^{PV}_{m,t,\phi}$	Expected active power generation of the PV systems.
$P^{D}_{m,\phi,t}$	Expected active power consumption.
$Q^{D}_{m,\phi,t}$	Expected reactive power consumption.
$\overline{V}, \underline{V}$	Maximum/minimum voltage magnitude.
$R_{mn,\phi,\psi}$	Resistance of the lines.
$X_{mn,\phi,\psi}$	Reactance of the lines.

Continuous Variables:

$\Delta P^{PV}_{m,t}$	PV power curtailment percentage.
$P^{G}_{m,t,\phi}$	Active power injection of the PV inverters (AC side).
$I^{G\mathrm{re}}_{m,\phi,t}$	Real part of the current injection of the PV inverters.
$I^{G\mathrm{im}}_{m,\phi,t}$	Imaginary part of the current injection of the PV inverters.
$I^{D\mathrm{re}}_{m,\phi,t}$	Real part of the current injection for the consumption.
$I^{D\mathrm{im}}_{m,\phi,t}$	Imaginary part of the current injection for the consumption.
$V^{\mathrm{re}}_{m,\phi,t}$	Real part of the voltage magnitude.
$V^{\mathrm{im}}_{m,\phi,t}$	Imaginary part of the voltage magnitude.