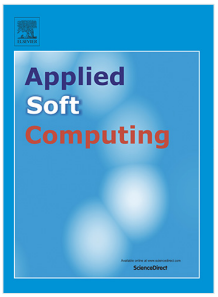
Journal Pre-proof



Deep reinforcement learning approach for MPPT control of partially shaded PV systems in Smart Grids

Luis Avila, Mariano De Paula, Maximiliano Trimboli,
Ignacio Carlucho

PII: S1568-4946(20)30649-9
DOI: https://doi.org/10.1016/j.asoc.2020.106711
Reference: ASOC 106711

To appear in: *Applied Soft Computing Journal*

Received date : 25 April 2020
Revised date : 1 September 2020
Accepted date : 6 September 2020

Please cite this article as: L. Avila, M. De Paula, M. Trimboli et al., Deep reinforcement learning approach for MPPT control of partially shaded PV systems in Smart Grids, *Applied Soft Computing Journal* (2020), doi: https://doi.org/10.1016/j.asoc.2020.106711.

Deep reinforcement learning approach for MPPT control of partially shaded PV systems in Smart Grids

Luis Avila[1,*], Mariano De Paula[2], Maximiliano Trimboli[3] and Ignacio Carlucho[4]

*Corresponding author

[1]Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC), CONICET-UNSL, Av. Ejército de los Andes 950, D5700HHW San Luis, Argentina.

[2]INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro CIFICEN – UNICEN – CICpBA – CONICET, Av. Del Valle 5737, 7400 Olavarría, Argentina.

[3]Laboratorio de Control Automático (LCA), CONICET-UNSL, Ruta Prov. Nº 55, D5730 San Luis, Argentina.

[4]Department of Mechanical Engineering, Louisiana State University, Baton Rouge, LA 70803, USA.

loavila@unsl.edu.ar, mariano.depaula@fio.unicen.edu.ar, mdtrimboli@unsl.edu.ar, ignacio.carlucho@fio.unicen.edu.ar

Competing interests: none

Abstract

Photovoltaic systems (PV) are having an increased importance in modern smart grids systems. Usually, in order to maximize the energy output of the PV arrays a maximum power point tracking (MPPT) algorithm is used. However, once deployed, weather conditions such as clouds can cause shades in the PV arrays affecting the dynamics of each panel differently. These conditions directly affect the available energy output of the arrays and in turn make the MPPT task extremely difficult. For these reasons, under partial shading conditions, it is necessary to have algorithms that are able to learn and adapt online to the changing state of the system. In this work we propose the use of deep reinforcement learning (DRL) techniques to address the MPPT problem of a PV array under partial shading conditions. We develop a model free RL algorithm to maximize the efficiency in MPPT control. The agent's policy is parameterized by neural networks, which take the sensory information as input and directly output the control signal. Furthermore, a PV environment under shading conditions was developed in the open source OpenAI Gym platform and is made available in an open repository. Several tests are performed, using the developed simulated environment, to test the robustness of the proposed control strategies to different climate conditions. The obtained results show the feasibility of our proposal with a successful performance with fast responses and stable behaviors. The best results for the presented methodology show that the maximum operating power point achieved has a deviation less than 1% compared to the theoretical maximum power point.

Keywords: MPPT, Deep RL, PV systems, OpenAI Gym.

1. Introduction

The increasing world energy demand together with a growing concern about environmental issues have generated enormous interest in the use of renewable energy sources through the development of smart grid systems. Today efforts aim at increasing the manageability and efficiency of smart grids by turning them into sub-systems called smart microgrids which consist of local energy charges and resources that act as an independent entity with respect to the general network. In this manner, they can be connected and disconnected from the utility network according to some energy optimization strategy. Furthermore, microgrids can integrate distributed energy networks -such as photovoltaic (PV) generation systems- that employ modern information, communication and control technologies to enhance economy, efficiency, reliability, and security of the electrical grid while allowing active participation of consumers into the energy market [1]. This renewable power is clean and free at the point of use but it cannot always be relied upon due to its uncertain nature, making the need for intelligent real time management to avoid the volatility of the whole power grid [2]. This emphasizes microgrids as an interesting field for the study and implementation of artificial intelligence based solutions.

Microgrid systems based on PV arrays are considered to be one of the most implemented and well accepted distributed generation sources [3]. To be efficient a PV array must constantly transmit the maximum power available to the load, regardless of climatic conditions, so this control problem is known as Maximum Power Point Tracking (MPPT) [4–6]. In the case of an MPPT control problem an action is considered as a change of the output voltage to affect the produced PV power. However, PV systems suffer from nonlinearity between the output voltage and current especially under partially shaded conditions (PSC), which can result in significant losses to the PV output power [7,8]. When PV modules belonging to the same string experience different insolation, the resulting power-voltage (P–V) relation becomes more complex and exhibits multiple peaks. As most conventional MPPT methods are based on the hill-climbing principle, i.e. moving to the next operating point in the direction in which power increases, the presence of multiple peaks reduces their effectiveness [9,10]. To this aim, several modifications of the traditional MPPT control techniques have been proposed, such as Perturb and Observe (P&O) [11–13], hill-climbing [14,15], incremental conductance [16,17], fuzzy-logic [18,19], and hybrid [20,21]. Also, a number of works have addressed the shading problem

using artificial intelligence algorithms, such as artificial neural networks [22–24], evolutionary algorithms [25] and particle swarm optimization techniques [26,27]. Notably, the majority of the MPPT control techniques are model-based and thus they make use of a model of the PV panel. However, obtaining an accurate model of the PV systems and its parameters can be a burden under dynamic environmental conditions. This is more challenging when PV panels are interconnected in series and parallel to form large arrays, which can be exposed to different irradiance conditions.

Reinforcement learning (RL) techniques are model-free and in consequence do not require system identification. Instead, they can build a closed-loop policy from a set of trajectories obtained from interactions with the real PV environment or from simulations. The overall objective of the RL agent is to maximize the accumulated value of the future rewards, where the reward is a numerical value given by the environment to the agent after each interaction representing how good, or bad, the action taken is with regards to the objective [28]. Noticeably, with the expansion of smart meters, data on electricity generation and its demand will be readily available making data-driven techniques more relevant. The RL paradigm is an unsupervised learning framework where an artificial agent continuously learns and adapts its behavior (commonly called policy) directly from raw interactions with its environment, i.e., with the PV system. MPPT control methods using Reinforcement Learning (RL) techniques have also been proposed for both uniform irradiance [29–31] and PSC [32,33]. In this type of controllers, the MPPT problem is seen as a Markov Decision Process (MPD) [34].

In order to reduce the computational cost of the RL techniques, often the state-action space has to be discretized. For example, Hsu et al. proposes [31] a RL MPPT control scheme based on only four states and four actions, which are defined according to the side and direction of movement of the operating point with respect to the MPP. Similarly, Kofinas et al. [29] defines a list of finite and discrete actions including positive and negative changes in the output voltage of the PV system. Nevertheless, to maximize the efficiency in MPPT control, it is necessary to work in a continuous state-action space. One of the main obstacles to RL formulations lies in the management of applications in continuous state and action spaces, thus the use of function approximators is required to estimate both the control policy and the value function [35,36].

Following the growing popularity of deep neural networks (DNN) [37,38], Mnih et al. [39] introduced the deep Q-Network (DQN) technique that uses convolutional neural networks (CNN) to approximate the value function for actions, stabilizing the training process. However, the DQN algorithm can only be applied to discrete systems, that is, systems with finite and discrete state / action spaces. Later, Lillicrap et al. [40] extended deep RL (DRL) formulations to continuous state spaces, using the deep deterministic policy gradient algorithm (DDPG) that incorporates the ideas of batch normalization [41] and repetition of experiences [39]. In this way, deep reinforcement learning is a modern subfield of machine learning which is positioning itself to tackle complex engineering problems. Recently, more approaches to address control problems dealing with continuous spaces have been proposed [42,43]. In addition, deep RL methods are powerful to deal with complex systems in a model-free way becoming attractive and advantageous to work with partial shading PV systems. Nevertheless, into the field of engineering applications several deep RL proposals have been developed and tested in applications that somehow involve image recognition. In particular, during the literature review, no previous works were found that used deep RL techniques for the management and control of photovoltaic systems, and even less with respect to PSC.

In this article we propose a deep learning model-free formulation to address the continuous MPPT control problem of PV systems under unpredictable environmental conditions, such as shading. The algorithm is based on the deterministic policy gradient theorem and uses neural networks to parameterize the policy. The state of the agent is then described directly by the sensor measurements, without the need of any preprocessing, while the continuous actions selected by the neural agent correspond to the control actions of the MPPT formulation. Additionally, a PV environment was developed to evaluate the efficiency of DRL algorithms for MPPT control under different operating conditions. In this way, we developed a framework compatible with the open source OpenAI Gym platform [44] which provides a standardized and fully parameterizable computing environment to test both our formulation and future developments coming from the machine learning community for the management of smart microgrids. In this way, the concept of environment allows the performance of different RL-based solutions to be directly compared to each other in a standard, controlled and well-defined environment. Also, this assures the reproducibility of the experiments allowing to further test future improvements in a fast way and in an open source way. Finally, the proposed RL control formulation is evaluated in the developed PV

environment. A number of several trials demonstrated successful results, showing that the maximum power operating point found is less than 1% compared to the theoretical maximum power point. These results show that the proposed deep RL algorithm is able to successfully address the MPPT control problem of PV systems under PSC.

2. MPPT for PV arrays

A typical PV array is made up of several PV modules connected in series and in parallel to provide the desired voltage and current. Each PV module is a package that consists of a connected assembly of photovoltaic solar cells. The voltage at which a PV cell can generate its maximum power is called maximum power point (MPP). Notably, the maximum power varies with solar radiation, ambient temperature and solar cell temperature.

When the PV array operates under uniform insolation, the resulting power-voltage (P–V) curves of the array exhibit a single MPP. However, when some modules of the array are shaded by clouds, for example, non-uniform insolation conditions force shaded modules to operate with a reverse bias voltage. This reverse voltage leads to modules consuming power instead of supplying it to the load and cause hotspots to appear in them [45].

The operating point of a PV array is defined as the power produced due to the current I_{PV} and the voltage V_{PV}. When a load is connected to a PV source, the operating point and the power produced are defined by the resistance of the electric load. For example, if the resistive value of an electrical load is equal to $R_L = V_{MPP}/I_{MPP}$, then the operating point will coincide with the MPP (V_{MPP}, I_{MPP}) and there is no need to track the MPP. When a different resistive load is connected, the operating point will be different from the MPP, i.e., $V_{PV} \neq V_{MPP}$ and $I_{PV} \neq I_{MPP}$. In this case, the PV source does not produce the maximum possible power and control actions must be applied to follow the MPP.

2.1 PV cells modeling

A PV photovoltaic array can be represented analytically by its electrical characteristic of current I_{PV} against voltage V_{PV}. Figure 1 shows the equivalent circuit of a given cell as a p-n junction diode.

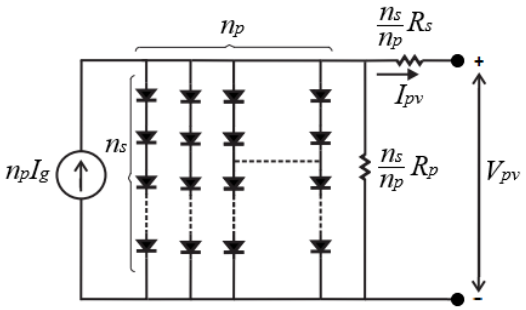


Fig. 1. Equivalent circuit for a PV cell

Denoting the number of PV cells in series and in parallel as n_s and n_p respectively, the output current I_{PV} is written as the difference between the generated photocurrent I_g and that of the diode current I_s:

$$I_{PV} = n_p I_g - n_p I_s \left(exp\left[\frac{q}{AkT} \left(\frac{V_{PV}}{n_s} + \frac{I_{PV} R_s}{n_p} \right) \right] - 1 \right) \tag{1}$$

where A is the diode ideal factor, k is the Boltzmann constant, q is the charge of electrons, T is the temperature in Kelvin and R_s is the equivalent series resistance. In turn, the photocurrent I_g generated by solar irradiance I_{rr} is:

$$I_g = \left(I_{sc} + \mathrm{k}_i (T - T_{ref}) \right) \frac{Irr}{1000} \tag{2}$$

here I_{sc} is the short-circuit current at reference temperature and radiation, T_{ref} is the reference temperature of the cell and k_i is the temperature coefficient for the short-circuit current.

The cell saturation current I_s, varies with the temperature according to the following relation:

$$I_s = I_{RS} \left[\frac{T}{T_{ref}} \right]^3 exp\left[\frac{qE_g}{Ak} \left(\frac{1}{T_{ref}} - \frac{1}{T} \right) \right] \tag{3}$$

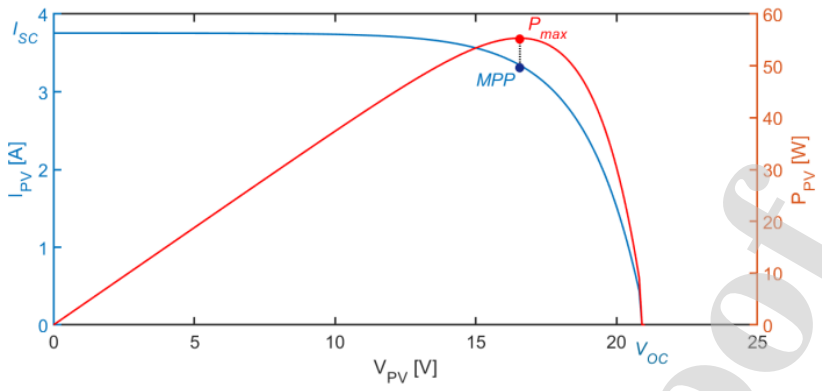
where I_{RS} is the reverse saturation current and E_g is the energy of the prohibited band of the semiconductor. Finally, the power P_{PV} delivered by the panel is calculated as:

$$P_{PV} = I_{PV} V_{PV} \tag{4}$$

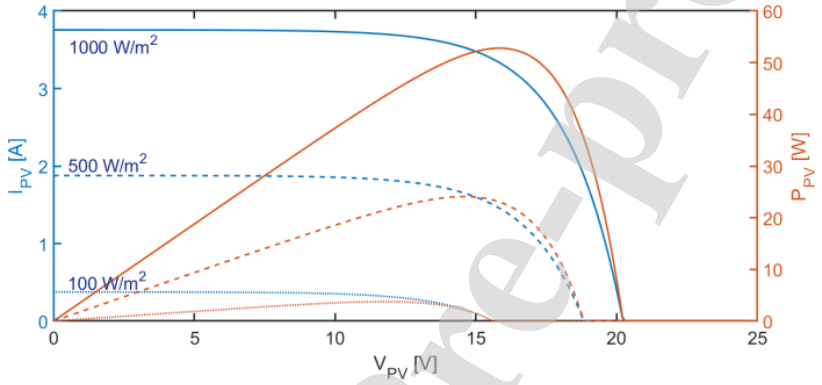
Relationships (1)-(3) clearly show the model's dependence on solar radiation conditions and temperature. From this mathematical representation, the power curve associated with a PV module is obtained by expressing the serial and parallel connection of all photovoltaic cells. If we consider an arrangement consisting of identical cells under uniform solar irradiation, the characteristic P-V curve will have a single peak, as shown in Fig. 2a. The figure illustrates the characteristic curve (I-V) for the open circuit voltage (V_{OC}), the short-circuit current (I_{SC}) and the maximum power operating point of the solar cell. The MPP is a unique point, as can be seen in Fig. 2a., where the power generated from the PV source is maximized. Note that, if the power required by the load increases, the operating point will move to the left of the MPP while if the operating point is reduced it will move to the right of the MPP. Therefore, it is necessary to track the MPP continuously.

On the other hand, the I-V curve of the source varies its characteristics according to the environmental conditions. A representation of the operation of a solar cell under different irradiations is shown in Fig. 2b. It can be seen that the output value corresponding to the MPP at an irradiation of 1000 [W/m^2] does not match the MPP at 500 [W/m^2]. Similarly, a change in temperature will affect the power delivered by the cells. The output voltage also depends largely on the temperature and an increase in temperature will decrease the value of V_{PV}. Since MPPT control can be achieved by regulating the output voltage of the V_{PV} system, this is considered as the optimization variable. Favorably, the MPP coincides with the point at which the derivative of the power P_{PV} with respect to the voltage V_{PV} is zero, that is:

$$\frac{dP_{PV}}{dV_{PV}} = 0 \tag{5}$$



a)



b)

Fig. 2. PV Source I-V and P-V curves: a) typical curves of a PV cell; b) curves for different solar irradiance.

2.2 PV Arrays under PSC

If several PV modules are connected in series, they conduct the same current, while the voltages across these modules are added to determine the resultant output voltage. On the other hand, if PV modules are connected in parallel to form a group, the voltage will be the same in each module, while the output current will be the sum of each individual current.

When the PV system is subjected to partial shading, series modules are assembled into groups having the same shading pattern. As shown in Fig. 3, bypass diodes are connected in parallel with each PV module to shunt the current around them, thus cells can continue supplying power at a reduced voltage rather than no power at all. Under normal operation, each solar cell is forward biased and bypass diodes are open-circuited. Under partially shaded conditions solar cells are reverse biased, and the bypass diode conducts (red path

in Fig. 3) allowing the current from the good solar cells to flow in the external circuit rather than forward biasing each good cell. Also, blocking diodes are connected to the PV panels to prevent current flowing back into them when the voltage produced by the panels is lower than that of the battery in the case of dark or shading conditions.

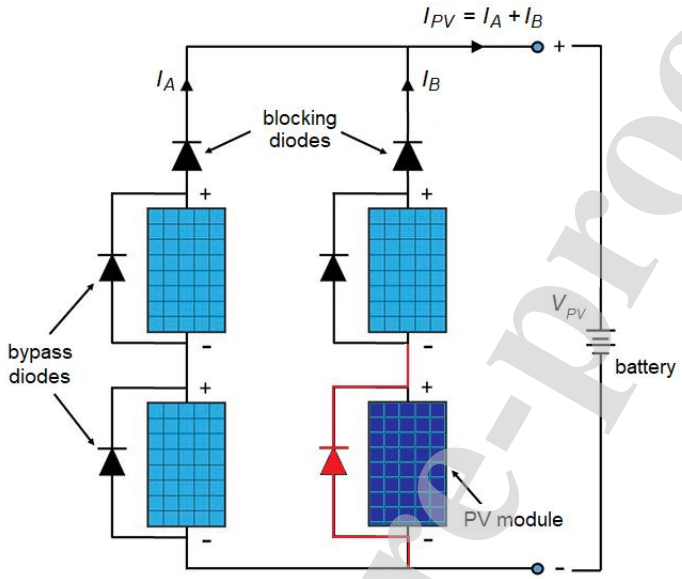


Fig. 3. PV array with bypass diodes under PSC.

The curves in Fig. 4, represent the I–V and P–V characteristics of the array under uniform insolation; under partially shaded condition and without diodes; and under partially shaded condition but with diodes connected in series with each of the series assemblies. Notice that the presence of bypass diodes allows the unshaded modules generate their maximum current at a given insolation level. When the bypass diodes are not present, the shaded modules will limit the current output of the series assembly and decrease the available output power from the PV array. The blocking diodes will prevent the reverse current through the series assemblies, which generate lower output voltage as compared to the others connected in parallel. This reverse current may cause excessive heat generation and thermal breakdown of PV modules.

Because the effect of diodes, P–V curves exhibit several local peaks and one global MPP (G_{MMP}), as shown in the dashed curve in Fig. 4. Unfortunately, presenting multiple maxima in the P-V characteristic is a crucial issue that most of the conventional MPPT algorithms may not be able to deal with. Most traditional MPPT techniques start searching in a selected region of the P–V curve. If this region is near a local peak, then those techniques would be unable to locate the global MPP because they would stop searching once the first

peak is located. Given this situation, conventional local search space MPPT techniques are not suitable for PV arrays that have PSC. Also, this is aggravated by the fact that the environmental conditions, i.e. temperature, shading and solar irradiance, may change from time to time during the day and consequently change the shape of the P-V curves.

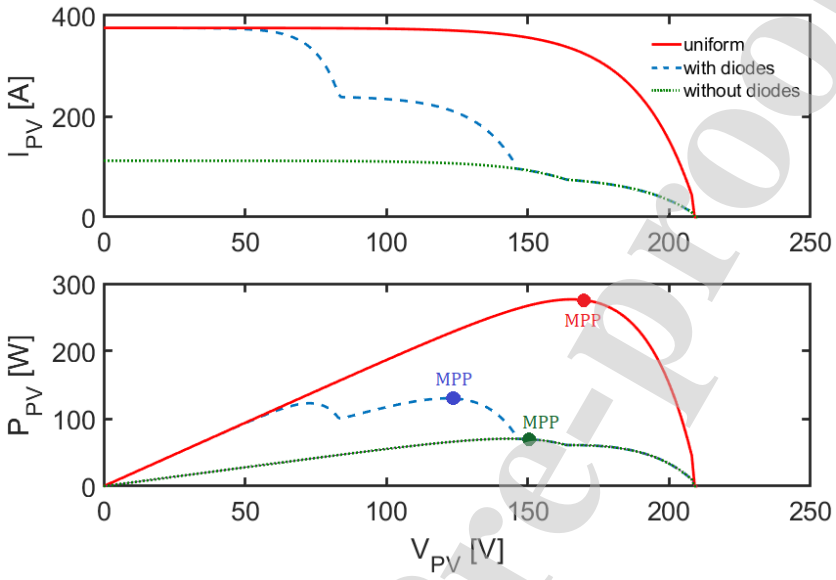


Fig. 4. Effect of bypass diodes on I-V and P-V curves

3. Deep-RL for MPPT control

This paper proposes a model-free RL approach to solve the continuous MPPT control problem in a PV array. An RL approach allows one to solve the problem without knowing the behavior of the PV source or predefine its dynamics. The RL algorithm aims to learn the behavior of system or its optimal configuration according to the responses of the interactions with the PV source. The formulation of a reinforcement learning approach for the operation of the PV arrangement, must be made in terms of a Markov decision problem.

3.1 The RL problem

Reinforcement learning assumes that there is an agent located in an environment and in each interaction with it, the agent takes an action and acts on the environment receiving a

reward in the form of a numerical signal. An RL algorithm will then seek to maximize the total reward received by the agent, for which the RL problem must be formalized as a Markov Decision Process (MDP) [28].

Commonly, in RL formulations, the control problem is defined by four elements: the state space $\mathbb{X}$, the action space $\mathbb{U}$, the probability of state transition p and the reward function r. For the MPPT control problem, in time t an action is taken that corresponds to a value u_t for the manipulated variable V_{PV}. During the learning process, the agent interacts with the system by applying an action $u_t \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ and, after that, the system evolves from the state $\mathrm{x}_t \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ to a successor state x_{t+1} and the agent receives a numerical signal r_t called reward (or punishment) that provides a measure of how good (or bad) the chosen action u_t was. Rewards act as "clues" about achieving goals or optimal behavior. Therefore, the objective of the RL methods is to find an optimal policy π^* that satisfies

$$J^* = \max_\pi J_\pi = \max_\pi \mathrm{E}_\pi\{R_t|\mathrm{x}_t = \mathbf{x}\} \tag{6}$$

where J_π corresponds to the expected total reward given the control policy π.

Let us assume that under a given policy π, the expected cumulative reward $V^\pi(\mathbf{x})$, or value function over a certain time interval, is a function of $\mathbf{x}^\pi$, where $\mathbf{x}^\pi = \{\mathbf{x}_t\}_{t=1}^{t=n}$ are the corresponding state values and $\mathbf{k}^\pi = \{\mathbf{k}_t\}_{t=1}^{t=n}$ defines the policy-specific sequence of the agent's actions. The sequence $\mathbf{x}^\pi$ of state transitions gives rise to rewards $\{r_t\}_{t=1}^{t=n}$. Robot control is a continuous task without a single final state therefore the discounted sum of future rewards $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{t=o}^{\infty} \gamma^k r_{t+k+1}$ is used to define the (discounted) expected state-value function for a policy π from the state $\mathbf{x}$:

$$V^\pi(\mathbf{x}) = E_\pi\{R_t|\mathbf{x}_t = \mathbf{x}\} = E_\pi\{\sum_{t=o}^{n} \gamma^k r_{t+k+1} \,|\mathbf{x}_t = \mathbf{x}\} \tag{7}$$

where $\gamma \in (0,1]$ is the discount factor which weights future rewards. $V^*(\mathbf{x})$ is used to denote the maximum discounted reward obtained when the agent starts in state $\mathbf{x}$ and executes the optimal policy π^*. Thus, the associated optimal state-value function that satisfies the Bellman's equation for all state $\mathbf{x}$ is:

$$V^*(\mathbf{x}_t) = \arg\max_{\mathbf{k}}\{r_t + \gamma E_{\mathbf{x}_{t+1}}[(V^*(\mathbf{x}_{t+1})|\mathbf{x}_t, \mathbf{k}_t)]\} \tag{8}$$

where $\mathbf{k}_t = \pi^*(\mathbf{x}_t)$. Similarly, the state–action value function Q^* is defined by:

$$Q^*(\mathbf{x}_t, \mathbf{k}_t) = r_t + \gamma E_{\mathbf{x}_{t+1}}[(V^*(\mathbf{x_{t+1}})|\mathbf{x}_t, \mathbf{k}_t)] \quad (9)$$

such that $V^*(\mathbf{x}) = \max_{\mathbf{k}} Q^*(\mathbf{x}, \mathbf{k})$ for all $\mathbf{x}$. Once Q^* is known through interactions, then the optimal policy can be obtained directly through:

$$\pi^*(\mathbf{x}) = \arg\max_{\mathbf{k}} Q^*(\mathbf{x}, \mathbf{k}) \quad (10)$$

3.2 Deep-RL

One of the biggest challenges that RL techniques have faced since its inception has been how to deal with spaces of continuous action. If the space of actions is too discretized, it ends with a dimensionality problem. But insufficient discretization of the space of action could disregard valuable information about the geometry of the domain of actions. Consequently, RL algorithms have been limited to small and discrete grid environments, which detract from their feasibility in their application for most dynamic systems.

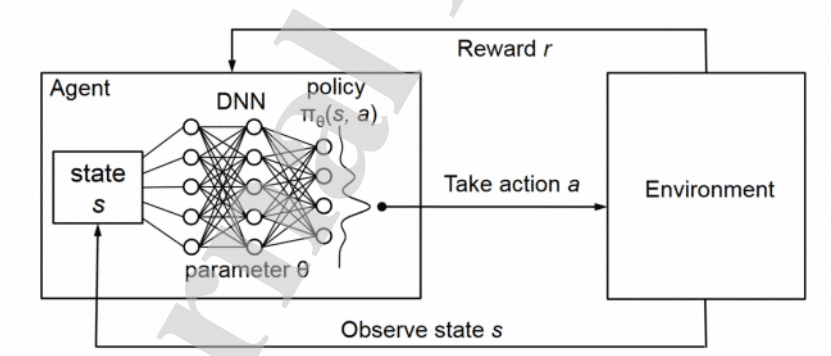


Fig. 5. The reinforcement learning framework with a deep network, in which an agent takes an action that generates a reward and a new state.

The success and rapid acceptance of the Deep Q-Networks approach [39], generated an expansion of the study and implementation of RL techniques to solve high dimensional problems within the dynamic systems control area [40,46], refer to Fig. 5.

Based on the previous work presented by Silver et al. [47], in relation to the deterministic policy gradients, Lillicrap et al. [40] developed an actor-critical approach called a deep

deterministic policy gradient (DDPG) that has the characteristic of being off-policy and model-free. RL approaches based on the policy gradient use an iterative method in which they evaluate the policy and then follow its gradient to maximize performance. DDPG uses a stochastic policy to achieve a good exploration but estimates a deterministic objective policy that is much simpler to learn. On the other hand, DDPG is based on an actor-critical approach, so it uses two deep neural network models. These networks calculate the prediction of the next action for the current state and generate a time difference (TD) error signal at each step. The input to the actor network is the current state and the output is a single real value that represents an action chosen from the continuous action space. The output of the network that models the critic is simply the estimated Q value for the current state and the action given by the actor.

More recently, Hausknecht et al. [48] proposed the use of deep neural networks in a structured (parametrized) space with continuous actions to delimit the gradients of the space of action suggested by the critic. This approach focuses on learning a small set of discrete actions, each of which is parameterized with continuous variables. This allows the use of RL to be extended to the Markov Decision Processes (MDP) class with continuous and parameterized action spaces. On the other hand, this approach reduces gradients as hyperparameters approach the limits of their ranges and are reversed if they exceed the range of values (hence their denomination inverted gradients). This allows the agent to keep the parameters within the limits, minimizing the problems of overestimation.

3.3 Deep RL tracking control algorithm

In this section the algorithm proposal for our Deep RL control formulation is introduced, as can be seen in Algorithm 1. As previously stated, the RL agent is based on the DDPG algorithm. Thus, the agent consists of a neural network that parameterizes the actor policy (μ) and a network for the critic (Q). In addition, there are two more networks, called targets, that are used for stabilizing the learning procedure. The MMPT control algorithm starts by initializing those networks, together with the replay buffer R in line 1. The replay buffer is used for storing experience, and subsequently training the agent with it.

The algorithm continues in line 2, where the main training loop starts. This loop is performed for the selected numbers of episodes (M). The following line is where the Noise

process utilized for exploration is initialized (line 3), followed by the initial state observation. This gives way to the second loop that starts in line 5 and is performed for a predefined number of timesteps (T). In each execution of this loop, the agent selects an action based on the current state (line 6) and performs that action in the environment. As a result, the environment transitions to a new state and a reward is provided based on the quality of the action taken (line 8). Finally, the state, action, reward and future state are stored in the replay buffer as is demonstrated in line 9.

In the following line of the algorithm, line 10, if enough transitions have been stored within the Buffer, the training process of the agent is started. First, a minibatch of size N with random transitions are extracted from the buffer (line 11). With this minibatch of experience, the critic network is updated in line 12, followed by an update of the actor network by means of the deterministic policy gradient. The training step is then finished when both target networks are updated via a soft update rule.

Algorithm 1 MPPT RL control

1: Initialize/Load Q, Q', μ and μ' networks and replay buffer R
2: **for** j = 1 to M **do**
3: Initialize a random noise process for exploration
4: Get initial state $\mathbf{s}_0$
5: **for** t = 1 to T **do**
6: Select action $\mathbf{a}_t = \mu_t(\mathbf{s}_t|\theta) + \text{noise}$
7: Execute action $\mathbf{a}_t$
8: Get new state $\mathbf{s}_{t+1}$ and reward r_t
9: Store the transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in R
10: **if** $|R| > N$ **then**
11: Sample a random minibatch r of N transitions
12: Update the critic $L(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q^{\mathbf{w}}(s_i, a_i))^2$
13: Update the actor using the deterministic policy gradient:
$\nabla_{\boldsymbol{\theta}} J = \mathbb{E}_{s_t \sim \rho^{\pi}}[\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\mathbf{s}) \nabla_{\mathbf{a}} Q^{\pi}(\mathbf{s}, \mathbf{a})]$
14: Update the actor and critic target network μ' and Q'
15: **end if**
16: Set $\mathbf{s}_t = \mathbf{s}_{t+1}$
17: **end for**
18: **end for**
19: $Q(\cdot, \cdot|\mathbf{w}), \mu(\cdot|\boldsymbol{\theta}), R$

The algorithm continues in line 16 when the current state of the system is updated. Finally, in line 19 the trained networks for the actor and critic are returned together with the replay buffer. In the following section, results obtained with the proposed algorithms and the developed MPPT gym environment are provided.

4. Results and discussion

In this section we present and discuss the obtained results when using our proposed control algorithm for MPPT control of the partial shading PV system. To assess the performance of the proposed DRL method to solve the MPPT control problem in the continuous action space, a number of test scenarios have been simulated. Each scenario represents the behavior of a PV array under different shading conditions. To this end, a complete PV environment was developed in the open source OpenAI Gym platform. OpenAI Gym [44] is an open source platform implemented in Python language where you can train, test and evaluate RL algorithms under a variety of environments.

Every environment comes with action and observation space that defines the system attributes, i.e. describes the format of valid actions and observations, the PV environment was implemented in Gym following this guideline and using the model set out in Section 2. The major advantage of using this platform is that it allows to compare the performance of different control techniques for the problem of maximum power tracking of a photovoltaic system under different climate conditions.

In the following subsection, we outline the PV simulation environment as well as the problem formulation as a Markov decision problem. Then, we detail the training stage for the proposed deep reinforcement learning algorithm for the MPPT control of the partial shading PV system. After that, we show different testing scenarios and finally a comparative section is presented aiming to highlight the performance of our proposal.

4.1 PV environment

A Gym environment basically consists of four functions. The first is the initialization function of the “init” class, which also establishes the initial state of our RL problem. The second function is that of step “step”, which receives data from the next action and returns a list of four elements: the next state, the reward resulting from the last action, a Boolean value that informs whether the current episode has ended and extra information about the state of the system. The "reset" function restores the state and other environment

variables to the start state and the "render" rendering function that provides relevant information about the behavior of our environment so far. Once the MPPT environment is complete with the PV information, we can create an instance of it. In this way, this environment can be easily used to test our control algorithm as well as any other RL algorithms developed under this architecture. For further details, the developed simulation environment is available in the following link: https://github.com/loavila/mppt-gym. All indications are given in the corresponding readme file.

The special categorization and terminology to model the shading pattern on that array is given in Patel [49]. For example, Fig. 6 shows a PV array solar system of 100 modules divided into ten series assemblies of 10 modules each, connected in parallel. The shading configuration of Fig. 6 gives three groups where the first group containing 40 assemblies in parallel has six shaded modules; the second group of 38 assemblies in parallel has three shaded modules whereas the series assemblies of the last group of 22 assemblies has no shaded module.

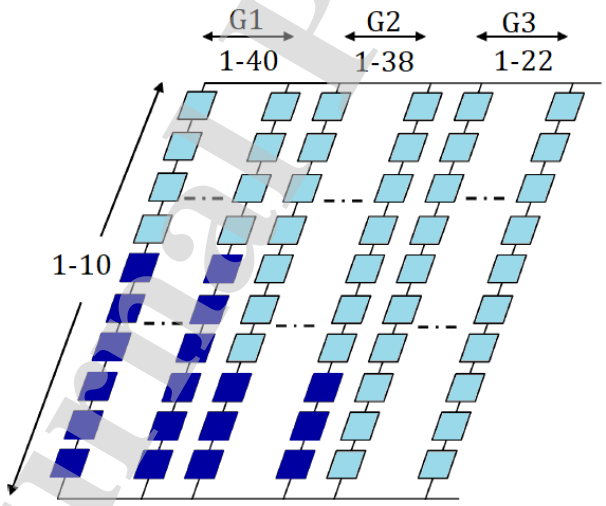


Figure 6. PV array configuration (from Patel [49]).

To formulate the MPPT control problem as a Markov decision problem into the RL framework, we must first define its main three elements:

State space: in the MPPT control problem, the efficiency of the task is defined according to how far from the MPP a PV panel is operating under specific environmental conditions. In general, the RL approaches applied to the MPPT use qualitative characteristics of the I-V

and P-V curves, such as the trend of power change and / or define discrete states [29,31,50]. These approaches have the advantage of forming a small state space, but are unable to describe the operation of the PV array under varying operating conditions. In this work a continuous state space is defined that corresponds to the current values $[V_{PV}, P_{PV}, \Delta P_{PV}]$. The ΔP_{PV} value allows on the one hand to determine on which side of the MPP the PV array is working and, on the other hand, it gives a better understanding of the Markovian return to the system since it allows to define if the algorithm is increasing the output voltage or reducing it.

Action space: the action space applied to the MPPT control problem is continuous, so it contains all the actions that can be applied in a PV array to generate a change in system operation. While this guarantees high precision in the magnitude of the action, it requires powerful learning techniques to make the approach computationally efficient. The action of the RLMPPT agent here is defined as the desired disturbance ΔV_{PV} applied to the controllable variable V_{PV}.

Reward function: for each action chosen by the agent and applied to the system, it reacts and evolves into a successor state generating a response in the form of a reward that goes from the environment to the agent. Intuitively, the simplest but most effective reward derivation could be a type of successful or fail function [31]. Keeping in mind that our control formulation is model free, this means that any knowledge about the system behavior is given to the deep RL MMPT control algorithm. Thus, the reward function could be thought proportional to the instantaneous power obtained at each sampling time due to the applied control action. A simple reward function should provide a capacity for generalization and adaptability for the model. For this reason, to dispense a priori information about the system, and consequently of the region of maximum power, the following reward function is defined:

$$r_t = \begin{cases} P/c\,, & \text{if } P > 0, \\ -1, & \text{if } P \leq 0, \end{cases} \tag{11}$$

where c is a normalization factor. To limit the reward signal, unless otherwise stated, hereafter we have taken $c = 50000$ for all presented experiments. As can be seen, with this simple function the reward obtained is directly proportional to the power and no prior knowledge about the system is needed to define it, which facilitates agent learning.

4.2 Training setup

The training of our proposed control algorithm as well as the test trials were carried out using the simulator developed in the Open AI Gym platform, described in the previous section. In Fig. 7 the characteristic curves of the model are presented, where the temperature is set at 25°C and the solar irradiance has been varied from 100 W/m² to 1000 W/m². The curves show the behavior of the PV system under different shading conditions. This figure shows the characteristic V-P curves for only ten different partial shading conditions and, on each of them, the corresponding maximum power point is indicated with a filled circle. As we can see, each of them has different local maximum and sometimes these local maximums are near the global maximum which increases the complexity of the control problem.

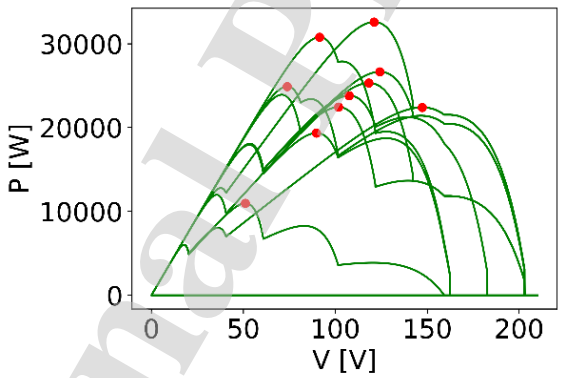


Figure 7. V-P characteristic curves for ten different shading conditions.

Taking into account that the partial shading PV system has a configuration as in Fig. 6, i.e. with three groups where each group can have from one to ten shaded modules, we can have up to 1000 different shading configurations for the PV system. It is noteworthy that the proposed model-free control algorithm does not receive any additional information to the system state and the reward signal, as was explained above. Thus, during the training phase the shaded configuration is randomly selected for each learning epoch and it is kept until each epoch ends. Fig. 8 shows the evolution of a projection of the value function along 5000 epochs, each of them with a different initialization seed (random seed). As can be

seen, the algorithm evolves towards operation zones with high values which, in turn, are consistent with zones of high power.

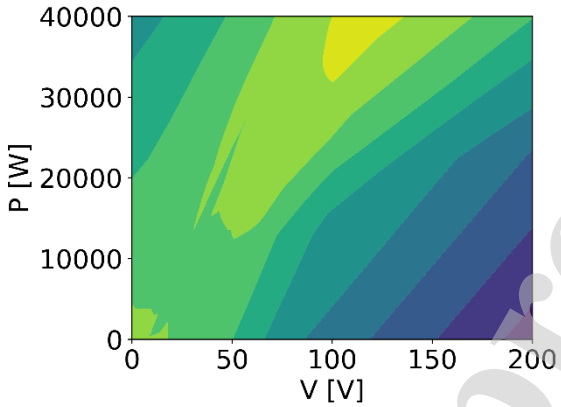


Figure 8. Value function evolution during the training phase.

The algorithm was implemented in Python using Tensorflow and trained during 30000 episodes. Each training episode consists of a maximum established duration of 100 samples, each sample consisting of one second. For all cases, an ϵ-greedy exploration strategy was used, with linear decay throughout the episodes. A repeat experience buffer with maximum size fixed in 50000 samples ($|R| = 50000$) was established with a random selection lot size of 64 samples ($M = 64$). The characteristic state vector of the system consists of the voltage and power at each sampling moment, as well as the last power variation such that $\mathbf{x}_t = [V_{PV}, P_{PV}, \Delta P_{PV}]$; and the reward function that was used is that described in Eq. (7). The developed deep RL algorithm is fully available in the GitHub repository https://github.com/marianodepaula/mppt_ddpg and the corresponding indications are given in the readme file.

4.3 Testing the MPPT control algorithm

In this section we show a number of cases where the obtained optimal policy, after the training stage, is tested under different shading conditions in an off-line way, i.e. the training phase is stopped.

In the first testing case we follow the array configuration given in Fig. 6, here the first group has 8 shaded modules; the second group has 6 shaded modules; whereas the series

assemblies of the last group have 5 shaded modules. Figure 9 shows the theoretical V-P profile for this shading condition and the red filled circle indicates the maximum power point for this case, which is 30784 W. This characteristic V-P profile was obtained with the developed PV environment (Section 2), varying the tension from 0 to 210 volts in an open loop way.

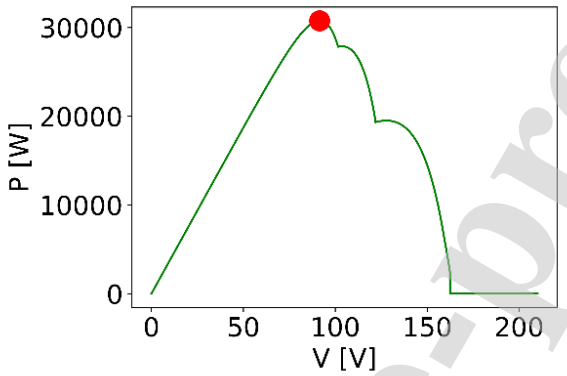


Figure 9. V-P curve for a partial shading condition with 8, 6 and 5 shaded modules in the PV system

Keeping the shading conditions of Fig. 9 during 1000 samplings (each sampling time is of one second), when the optimal control policy is used to track the MPP the results showed in Fig. 10 are obtained. In Fig. 10a tension profile is shown, where it can be seen that only around 100 sampling times were necessary to successfully drive the system to the optimal operation condition. In Fig. 10b the current profile of the PV system is showed whilst Fig. 10c shows the power evolution. As can be seen, the operative condition is achieved in a short time and, in this case, the obtained maximum power is 30576 W, that is just 0.68% less than the maximum achievable power point for the given shading conditions. Finally, Fig. 10d shows the V-P profile obtained when the system is controlled by the learned optimal policy under the given shading conditions.

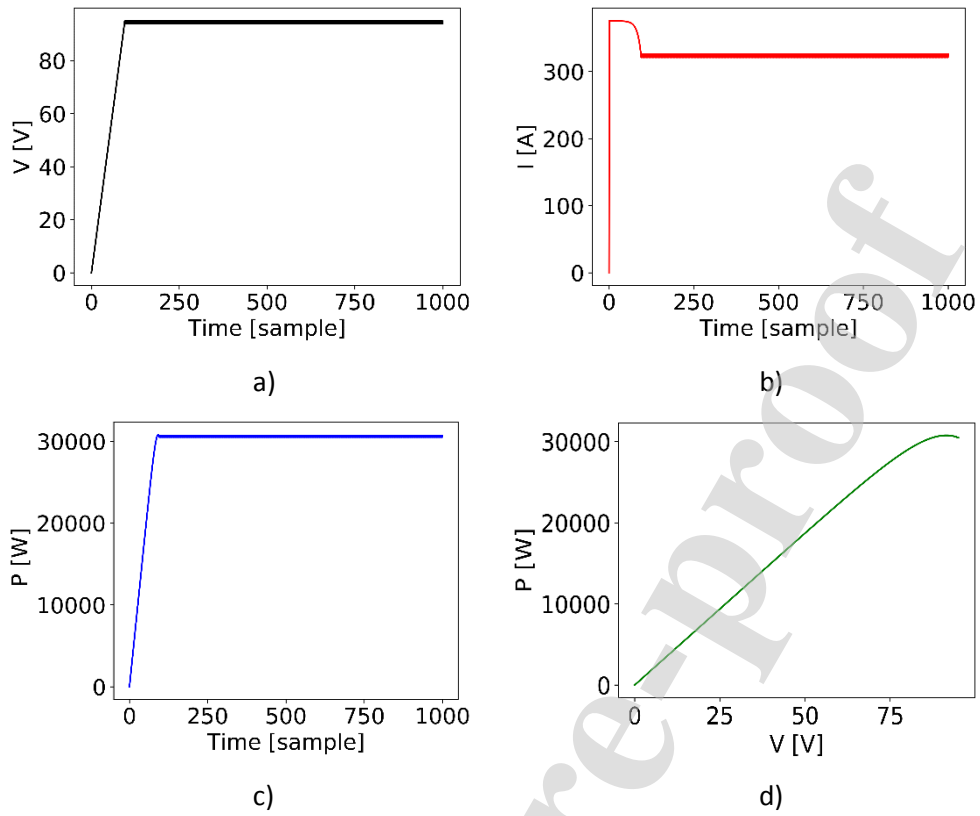


Figure 10. Results obtained when using the trained DDPG algorithm for the partial shading conditions given for Fig. 9. a) Output Voltage b) Output current c) Output power d) V-P curve.

Analogous to the previous testing case, Fig. 11 shows the theoretical V-P profile for another shading pattern, with 5, 10 and 4 shaded modules in the PV system. As can be seen, the behavior of the PV system is completely different to the previous case and the maximum power point is in a completely different operation zone, being the maximum achievable power of 24908 W. Thus, when we used the learned optimal control policy to control the PV system under the previously mentioned shading conditions, the results showed in Fig. 12 were obtained. As can be seen, the obtained control policy drives the system successfully in a similar smooth behavior pattern as in the previous testing case. In this case, the maximum operative power point is 24699 W, which is just 0.89 % less than the theoretical maximum power point.

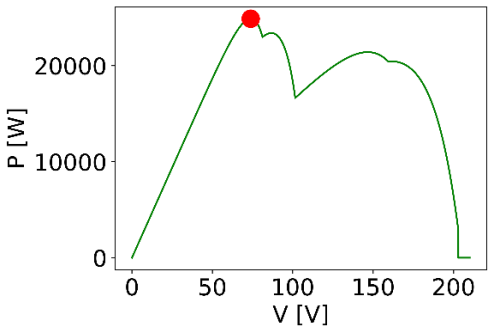


Figure 11. V-P curve for a partial shading condition with 5, 10 and 4 shaded modules in the PV system.

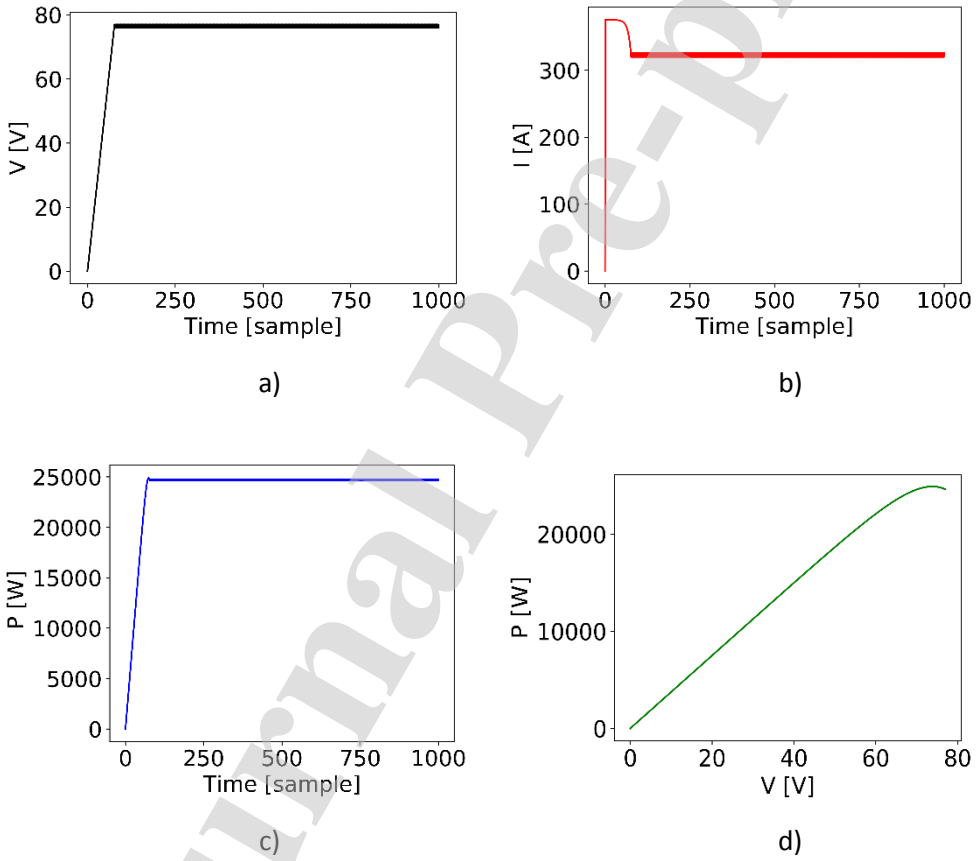


Figure 12. Results obtained when using the trained DDPG algorithm for the partial shading conditions given for Fig. 11. a) Output Voltage b) Output current c) Output power d) V-P curve.

In order to further demand, the adaptive ability of the obtained control policy we set a test where the shading conditions are suddenly changed a number of times. We initially set a shading pattern, with 2, 10 and 7 shaded modules. Then, after the first 1000 seconds, we change the shading conditions to another with 9, 3 and 6 shaded modules in the PV system. Finally, for the last 1000 seconds we set a shading pattern with 5, 9 and 10 shaded modules in the PV system. Note that these testing patterns are quite different to those used in the previous test cases. Directly, in Fig. 13 are shown the obtained results of this trial. As it can be seen the control policy drives the system successfully and, even more, doing it in a fast way achieving a smooth behavior. In addition, it is noteworthy that there are no significant overshooting and unstable behaviors at the times shading conditions change. In this case, the mean difference against the theoretical power maximums for each shading condition is less than 1%.

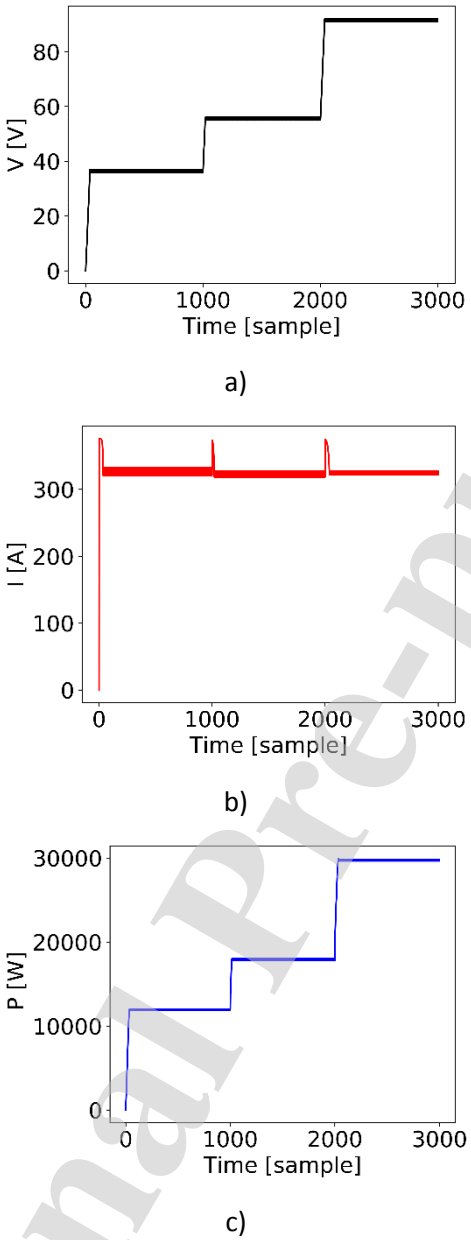


Figure 13. Testing scenario for changing partial shading conditions: first with 2, 10 and 7 shaded modules; secondly with 9, 3 and 6 shaded modules; lastly with 5, 9 and 10 shaded modules. a) Output Voltage b) Output current c) Output power.

Similar to the previous case, in Fig. 14 are the results obtained when the control policy was used to control the partial shaded PV system according to the following shading schedule: at the beginning 3, 8 and 5 shaded modules are taken into account during 1000 seconds; then, for the next 1000 seconds, 1, 5 and 2 shaded modules were considered and, finally, the PV system configuration was with 4, 7 and 10 shaded modules. Note that in this case the shading patterns are significantly different to those used in the previous case. As can be seen in Fig. 14, the shading schedule does not follow an upward behavior in terms

of the achievable maximum power as in the previous case. However, the learned control policy timely acts, driving the system in a successful way being the mean difference of the total acquired power, regarding the maximum theoretical achievable power, around 1.15%.

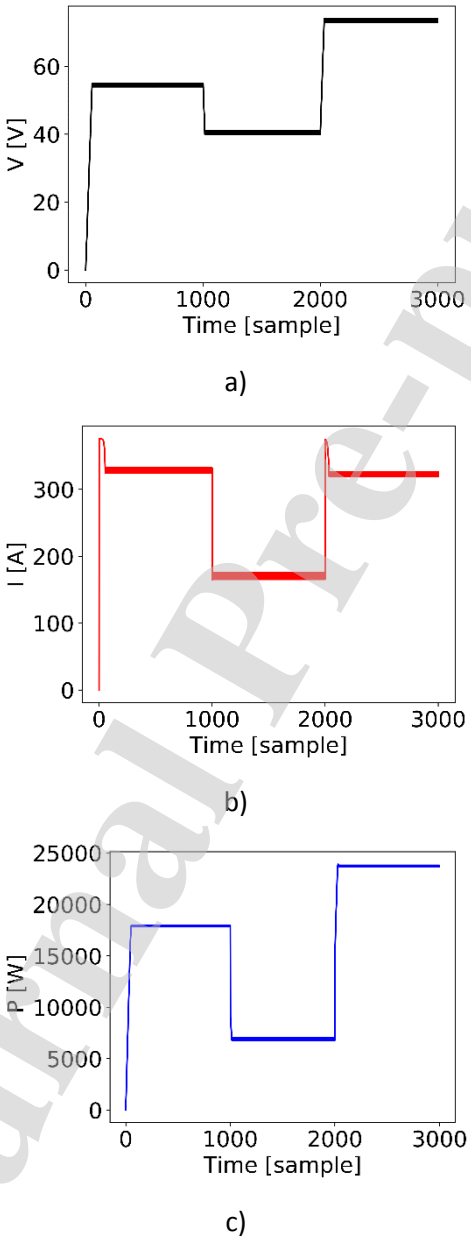


Figure 14. Testing scenario for the DDPG algorithm with changing partial shading conditions: first with 3, 8 and 5 shaded modules; second with 1, 5 and 2 shaded modules; lastly with 4, 7 and 10 shaded modules. a) Output Voltage b) Output current c) Output power.

4.4 Performance comparison

In order to make a performance comparison with other comparable methodologies, in this section, we present a comparison between the proposed algorithm and the Twin Delayed DDPG (TD3) [51], which has been gaining an important place within the deep RL community. The TD3 is an algorithm that incorporates three critical modifications with respect to the DDPG: first, TD3 learns two Q functions instead of one (hence the term twin) and uses the smallest of the two Q values to compute Bellman's function; second, TD3 updates the policy less frequently than the Q function (approximately one policy update for every two updates of the Q function); and finally, TD3 adds some noise to the action in order to make it difficult for the policy to learn the errors of the Q function. Since, previous works have reported a successful performance of this algorithm, even improving those reached by the DDPG, for several applications in different research fields we chose it to compare our deep RL formulation for the MPPT control of a partial shading PV system. It is noteworthy that for a fair comparison we used the same training conditions, i.e. same training epochs, buffer size, exploration rate, learning rate, etc. For further details about the TD3 algorithm implementation refers to the following link: https://github.com/marianodepaula/mppt_td3.

Figure 15 shows the results when the TD3 algorithm is used to control a partial shading PV system with 8, 6 and 5 shaded modules. In other words, this condition is the same as those assumed in the first testing case presented in the previous sections. Keeping in mind the shading conditions stated in Fig. 9, the results showed in Fig. 10 and the timely comments, we can see in Fig. 15 that although the TD3 algorithm presents a correct performance, the behavior is somewhat degraded, i.e. more rippled in the steady operation condition. Moreover, in this case the mean difference against the theoretical power maximum (showed in Fig. 9) for this shading condition is around 3.7%.

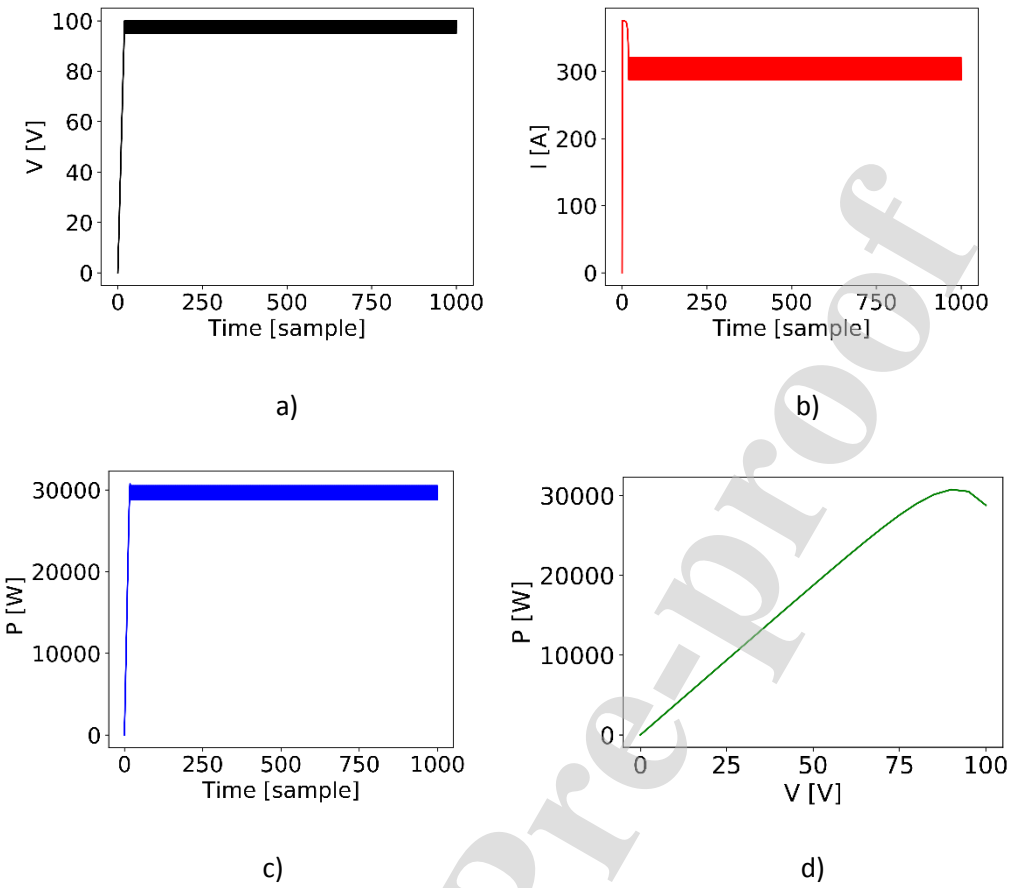


Figure 15. Results obtained when using the trained TD3 algorithm for the partial shading conditions given in Fig. 9. a) Output Voltage b) Output current c) Output power d) V-P curve.

Similar to the previous comparison case, Fig. 16 shows the result obtained when the optimal control policy learned by the TD3 algorithm is used to control the PV system under the same shading conditions set for Fig. 11. In this way, this result is comparable with that showed in Fig. 12. Here, again we can qualitatively see that despite the good performance of the TD3 algorithm, our proposal overcomes the performance of it. Also, from a quantitative point of view, our proposal has better performance that the TD3, in this case, the mean difference against the theoretical power maximum (showed in Fig. 9) for this shading condition is around 2.6%.

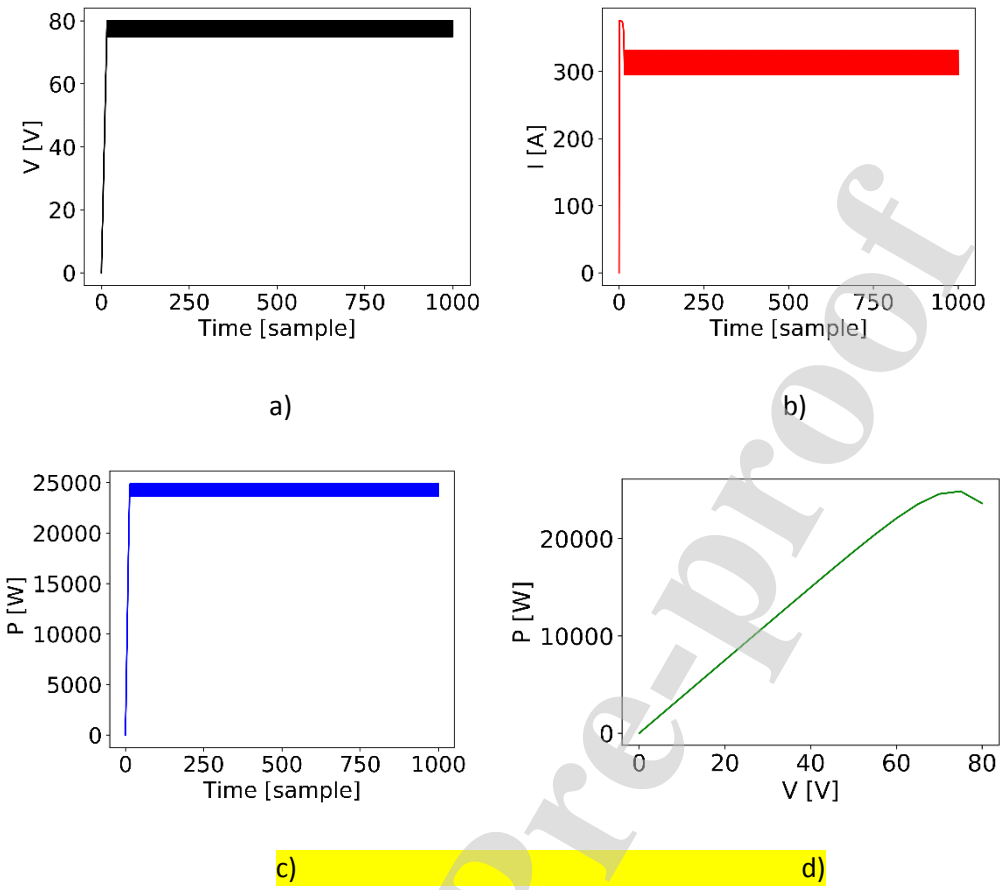


Figure 16. Results obtained when using the trained TD3 algorithm for the partial shading conditions given in Fig. 11. a) Output Voltage b) Output current c) Output power d) V-P curve.

Although, in general terms, the TD3 algorithm shows an acceptable behavior, although inferior to that obtained using our proposal, we have also tested the TD3 for changing shading conditions like those given for Fig. 13. Directly, Fig. 17 shows the results obtained when the control policy, obtained using the TD3 algorithm, is used to control the PV system under these changing shading conditions. Here it is notable the rippling behavior introduced by the control policy and also the total power obtained is less and also less stable than when using our proposed control algorithm.

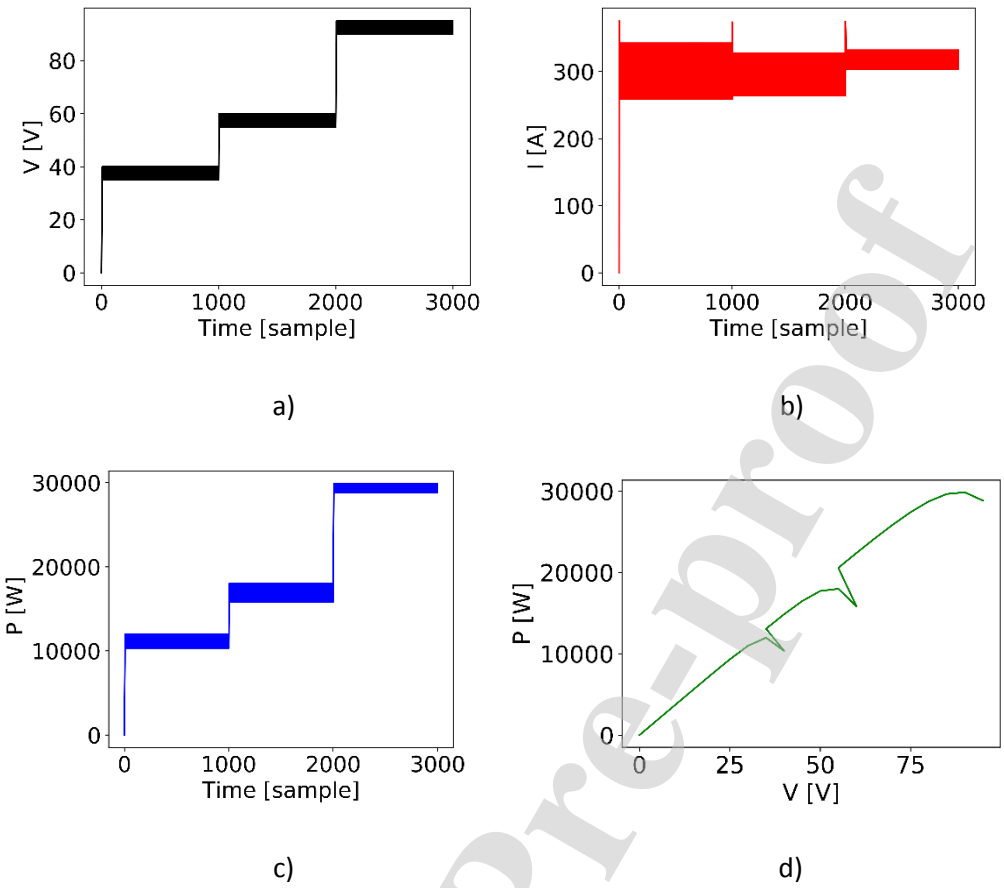


Figure 17. Results obtained when using the trained TD3 algorithm for the partial shading conditions given in Fig. 13.

5. Concluding remarks

This work proposes a deep reinforcement learning formulation to address the MPPT control problem of a PV array under partial shading conditions in a continuous state-action domain. An exhaustive performance study was made, and several testing cases were presented. In addition, a comparative study was carried out between our proposed algorithm and comparable methodology with similar features. The obtained results demonstrated a successful performance of DRL to manage a complex PV system under partial shading conditions.

The proposed algorithms demonstrated high precision to obtain the global maximum power value while maintaining the state and action space tractable. Furthermore, the

developed strategy is able to learn a highly efficient policy from scratch. In other words, it can be implemented without any additional configuration and previous knowledge of the system. Because of the model-free feature of RL, we rely solely on sampling to estimate the required actions, thus we do not need to model the inner operation of the system. This is a valuable feature since no expert knowledge is required to learn an optimal control policy by means of the proposed algorithm. The only exception is the reward function, that needs to be defined beforehand, but this is common to all RL formulations. However, we demonstrated that using a reward function defined in the simplest possible way causes the algorithm to converge to an optimal policy, even more, it does it in a successful way showing outstanding capabilities of generalization and adaptability of such policy. All this is of paramount importance and represents a distinctive advantage for the management of complex systems operating under uncertain conditions, i.e. like PV systems, especially when they are exposed to shading conditions.

Another valuable contribution of this work is the modeling and development of a partial shading photovoltaic system as a Gym environment in the wide spread open source OpenAI platform. This environment allows setting shading conditions as well as other system features by setting up its parameters. In this sense, this environment will facilitate the testing for future developments of MPPT control algorithms mainly those based on machine learning techniques.

Finally, it is worth noting the lack of bibliography on deep RL formulations for the MPPT control problem under PSC. The best results show that the maximum operating power point using a continuous control policy is less than 1% compared to the theoretical maximum power point. Therefore, the obtained results open a promising avenue for future works and developments in the area of machine learning -especially in the modern deep reinforcement learning community- to address the complex MPPT problem of PV systems under variable environmental conditions. In a broad sense, our work is a seminal contribution for the development of artificial intelligence applications in the field of smart grids.

References

[1] S. Leonori, M. Paschero, F.M. Frattale Mascioli, A. Rizzi, Optimization strategies for Microgrid energy management systems by Genetic Algorithms, Appl. Soft Comput. J. 86 (2020) 105903. doi:10.1016/j.asoc.2019.105903.

[2] K. Moharm, State of the art in big data applications in microgrid: A review, Adv. Eng. Informatics. 42 (2019) 100945. doi:10.1016/j.aei.2019.100945.

[3] N.L. Panwar, S.C. Kaushik, S. Kothari, Role of renewable energy sources in environmental protection: A review, Renew. Sustain. Energy Rev. 15 (2011) 1513–1524. doi:10.1016/j.rser.2010.11.037.

[4] M.A. Danandeh, S.M. Mousavi G., Comparative and comprehensive review of maximum power point tracking methods for PV cells, Renew. Sustain. Energy Rev. 82 (2018) 2743–2767. doi:10.1016/j.rser.2017.10.009.

[5] H. Islam, S. Mekhilef, N. Shah, T. Soon, M. Seyedmahmousian, B. Horan, A. Stojcevski, Performance Evaluation of Maximum Power Point Tracking Approaches and Photovoltaic Systems, Energies. 11 (2018) 365. doi:10.3390/en11020365.

[6] Z. Salam, J. Ahmed, B.S. Merugu, The application of soft computing methods for MPPT of PV system: A technological and status review, Appl. Energy. 107 (2013) 135–148. doi:10.1016/j.apenergy.2013.02.008.

[7] A. Fathy, Reliable and efficient approach for mitigating the shading effect onphotovoltaic module based on Modified Artificial Bee Colony algorithm, Renew. Energy. 81 (2015) 78–88. doi:10.1016/j.renene.2015.03.017.

[8] Y.I.A. Osman, J. Li, X. Zhen, A. Yang, Experimental Study of the Cloud Influence on PV Grid Connected System, Smart Grid Renew. Energy. 09 (2018) 1–15. doi:10.4236/sgre.2018.91001.

[9] D. Verma, S. Nema, A.M. Shandilya, S.K. Dash, Maximum power point tracking (MPPT) techniques: Recapitulation in solar photovoltaic systems, Renew. Sustain. Energy Rev. 54 (2016) 1018–1034. doi:10.1016/j.rser.2015.10.068.

[10] K.M. Abo-Al-Ez, S.S. Kaddah, S. Diab, E.H. Abdraboh, Performance analysis of maximum power point tracking (MPPT) for PV systems under real meteorological conditions, in: Green Energy Technol., Springer Verlag, 2020: pp. 199–228. doi:10.1007/978-3-030-05578-3_7.

[11] M.H. Osman, A. Refaat, Adaptive multi-variable step size P&O MPPT for high tracking-speed and accuracy, IOP Conf. Ser. Mater. Sci. Eng. 643 (2019) 012050. doi:10.1088/1757-899X/643/1/012050.

[12] J. Ahmed, Z. Salam, An improved perturb and observe (P&O) maximum power point tracking (MPPT) algorithm for higher efficiency, Appl. Energy. 150 (2015) 97–108. doi:10.1016/j.apenergy.2015.04.006.

[13] M. Karabacak, A new perturb and observe based higher order sliding mode MPPT control of wind turbines eliminating the rotor inertial effect, Renew. Energy. 133 (2019) 807–827. doi:10.1016/j.renene.2018.10.079.

[14] L. Liu, C. Liu, J. Wang, Y. Kong, Simulation and hardware implementation of a hill-climbing modified fuzzy-logic for maximum power point tracking with direct control method using boost converter, J. Vib. Control. 21 (2015) 335–342. doi:10.1177/1077546313486912.

[15] M. Lasheen, M. Abdel-Salam, Maximum power point tracking using Hill Climbing and ANFIS techniques for PV applications: A review and a novel hybrid approach, Energy Convers. Manag. 171 (2018) 1002–1019. doi:10.1016/j.enconman.2018.06.003.

[16] S. Motahhir, A. El Ghzizal, S. Sebti, A. Derouich, Modeling of Photovoltaic System

with Modified Incremental Conductance Algorithm for Fast Changes of Irradiance, Int. J. Photoenergy. 2018 (2018). doi:10.1155/2018/3286479.
[17] J.Y. Shi, L.T. Ling, F. Xue, Z.J. Qin, Y.J. Li, Z.X. Lai, T. Yang, Combining incremental conductance and firefly algorithm for tracking the global MPP of PV arrays, J. Renew. Sustain. Energy. 9 (2017). doi:10.1063/1.4977213.
[18] K. Punitha, D. Devaraj, S. Sakthivel, Development and analysis of adaptive fuzzy controllers for photovoltaic system under varying atmospheric and partial shading condition, Appl. Soft Comput. J. 13 (2013) 4320–4332. doi:10.1016/j.asoc.2013.06.021.
[19] X. Li, H. Wen, Y. Hu, L. Jiang, A novel beta parameter based fuzzy-logic controller for photovoltaic MPPT application, Renew. Energy. 130 (2019) 416–427. doi:10.1016/j.renene.2018.06.071.
[20] H.M. El-Helw, A. Magdy, M.I. Marei, A Hybrid Maximum Power Point Tracking Technique for Partially Shaded Photovoltaic Arrays, IEEE Access. 5 (2017) 11900–11908. doi:10.1109/ACCESS.2017.2717540.
[21] M. Kermadi, Z. Salam, J. Ahmed, E.M. Berkouk, An Effective Hybrid Maximum Power Point Tracker of Photovoltaic Arrays for Complex Partial Shading Conditions, IEEE Trans. Ind. Electron. 66 (2019) 6990–7000. doi:10.1109/TIE.2018.2877202.
[22] L. Bouselham, M. Hajji, B. Hajji, H. Bouali, A New MPPT-based ANN for Photovoltaic System under Partial Shading Conditions, in: Energy Procedia, Elsevier Ltd, 2017: pp. 924–933. doi:10.1016/j.egypro.2017.03.255.
[23] V.R. Kota, M.N. Bhukya, A novel global MPP tracking scheme based on shading pattern identification using artificial neural networks for photovoltaic power generation during partial shaded condition, IET Renew. Power Gener. 13 (2019) 1647–1659. doi:10.1049/iet-rpg.2018.5142.
[24] S.A. Rizzo, G. Scelba, ANN based MPPT method for rapidly variable shading conditions, Appl. Energy. 145 (2015) 124–132. doi:10.1016/j.apenergy.2015.01.077.
[25] S. Titri, C. Larbes, K.Y. Toumi, K. Benatchba, A new MPPT controller based on the Ant colony optimization algorithm for Photovoltaic systems under partial shading conditions, Appl. Soft Comput. J. 58 (2017) 465–479. doi:10.1016/j.asoc.2017.05.017.
[26] T.S. Babu, J.P. Ram, T. Dragičević, M. Miyatake, F. Blaabjerg, N. Rajasekar, Particle swarm optimization based solar PV array reconfiguration of the maximum power extraction under partial shading conditions, IEEE Trans. Sustain. Energy. 9 (2018) 74–85. doi:10.1109/TSTE.2017.2714905.
[27] K. Sundareswaran, V. Vignesh kumar, S. Palani, Application of a combined particle swarm optimization and perturb and observe method for MPPT in PV systems under partial shading conditions, Renew. Energy. 75 (2015) 308–317. doi:10.1016/j.renene.2014.09.044.
[28] R. Sutton, A. Barto, Introduction to reinforcement learning, MIT press Cambridge, 1998.
[29] P. Kofinas, S. Doltsinis, A.I. Dounis, G.A. Vouros, A reinforcement learning approach for MPPT control method of photovoltaic sources, Renew. Energy. 108 (2017) 461–473. doi:10.1016/j.renene.2017.03.008.
[30] Chou, Yang, Chen, Maximum Power Point Tracking of Photovoltaic System Based on Reinforcement Learning, Sensors. 19 (2019) 5054. doi:10.3390/s19225054.
[31] R.C. Hsu, C.T. Liu, W.Y. Chen, H.I. Hsieh, H.L. Wang, A Reinforcement Learning-Based Maximum Power Point Tracking Method for Photovoltaic Array, Int. J. Photoenergy. 2015 (2015). doi:10.1155/2015/496401.
[32] X. Zhang, S. Li, T. He, B. Yang, T. Yu, H. Li, L. Jiang, L. Sun, Memetic reinforcement learning based maximum power point tracking design for PV systems under partial shading condition, Energy. 174 (2019) 1079–1090. doi:10.1016/j.energy.2019.03.053.

[33] M. Ding, D. Lv, C. Yang, S. Li, Q. Fang, B. Yang, X. Zhang, Global Maximum Power Point Tracking of PV Systems under Partial Shading Condition: A Transfer Reinforcement Learning Approach, Appl. Sci. 9 (2019) 2769. doi:10.3390/app9132769.
[34] G.E. Monahan, A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms, Manage. Sci. 28 (1982) 1–16. doi:10.1287/mnsc.28.1.1.
[35] M. Riedmiller, Neural fitted Q iteration - First experiences with a data efficient neural Reinforcement Learning method, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2005: pp. 317–328. doi:10.1007/11564096_32.
[36] T. Degris, P.M. Pilarski, R.S. Sutton, Model-Free reinforcement learning with continuous action in practice, in: Proc. Am. Control Conf., 2012: pp. 2177–2182. doi:10.1109/acc.2012.6315022.
[37] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature. 521 (2015) 436–444. doi:10.1038/nature14539.
[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, (2012) 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks (accessed August 28, 2018).
[39] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature. 518 (2015) 529–533. doi:10.1038/nature14236.
[40] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., International Conference on Learning Representations, ICLR, 2016.
[41] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: 32nd Int. Conf. Mach. Learn. ICML 2015, International Machine Learning Society (IMLS), 2015: pp. 448–456.
[42] J. Schulman, S. Levine, P. Moritz, M. Jordan, P. Abbeel, Trust region policy optimization, in: 32nd Int. Conf. Mach. Learn. ICML 2015, 2015: pp. 1889–1897. http://proceedings.mlr.press/v37/schulman15.html (accessed January 8, 2020).
[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, (2017). http://arxiv.org/abs/1707.06347 (accessed January 8, 2020).
[44] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, OpenAI Gym, (2016). http://arxiv.org/abs/1606.01540 (accessed January 16, 2020).
[45] M. Dhimish, Assessing MPPT techniques on hot-spotted and partially shaded photovoltaic modules: Comprehensive review based on experimental data, IEEE Trans. Electron Devices. 66 (2019) 1132–1144. doi:10.1109/TED.2019.2894009.
[46] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, G.G. Acosta, Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning, Rob. Auton. Syst. 107 (2018) 71–86. doi:10.1016/j.robot.2018.05.016.
[47] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: 31st Int. Conf. Mach. Learn. ICML 2014, International Machine Learning Society (IMLS), 2014: pp. 605–619.
[48] M. Hausknecht, P. Stone, Deep Reinforcement Learning in Parameterized Action Space, (2015). http://arxiv.org/abs/1511.04143 (accessed January 16, 2020).
[49] H. Patel, V. Agarwal, MATLAB-based modeling to study the effects of partial shading on PV array characteristics, IEEE Trans. Energy Convers. 23 (2008) 302–310. doi:10.1109/TEC.2007.914308.

[50] A. Youssef, M.E. Telbany, A. Zekry, Reinforcement Learning for Online Maximum Power Point Tracking Control, J. Clean Energy Technol. (2015). doi:10.7763/jocet.2016.v4.290.

[51] S. Fujimoto, H. van Hoof, D. Meger, Addressing Function Approximation Error in Actor-Critic Methods, (2018). http://arxiv.org/abs/1802.09477 (accessed January 16, 2020).

Research highlights for the manuscript:

“Deep reinforcement learning approach for MPPT control of partially shaded PV systems in Smart Grids”

- Control method for a partially shaded photovoltaic system for Smart grids.
- Model free deep RL approach for maximum power point tracking control.
- PV array under partially shaded conditions was modeled as an OpenAI Gym environment.
- Our approach finds a control policy without prior knowledge of the system behavior.

CRediT author statement

Luis Avila: Conceptualization, Methodology.

Mariano de Paula: Conceptualization, Methodology.

Ignacio Carlucho: Methodology, Software.

Maximiliano Trimboli: Software, Validation.

Declaration of interests

☒ 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.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

None