Technical University of Crete

School of Electrical and Computer Engineering

Circuits, Sensors and Renewable Energy Sources Laboratory

Diploma thesis

Electronic system for the control of the power production of photovoltaic arrays under partial shading

Author: Emmanouil Lioudakis

Examination committee: Professor Eftichios Koutroulis (Supervisor)

Professor Georgios Stavrakakis

Associate Professor Fotios Kanellos

Chania, June 2023

Contents

Abstract

Photovoltaic (PV) systems constitute a significant percentage of the worldwide installed electrical power plants. They exploit a renewable energy source and they do not require frequent maintenance, since they do not contain mechanical parts. These, combined with the fact that their components get cheaper over time, explain the increasing penetration of PV power plants in modern electricity grids. To avoid frequency disturbance and to improve the grid's power quality, a way of controlling the output power of PV systems is the Flexible Power Point Tracking (FPPT), where a reference value is configured for the PV array's output power. Since PV modules are usually shaded, conventional FPPT algorithms cannot operate efficiently, thus the concept of the Global FPPT (GFPPT) algorithm was introduced. In this thesis, two novel GFPPT algorithms based on machine learning are proposed. To evaluate their effectiveness, an experimental system was developed. One GFPPT algorithm from the literature and the two new ones were tested experimentally on varying operating conditions, as it could happen in real world applications. The experimental results demonstrated that the two new algorithms developed within the framework of this thesis converged in significantly less time, while they achieved an almost same steady-state tracking error.

Περίληψη

Τα φωτοβολταϊκά (Φ/Β) συστήματα αποτελούν ένα σημαντικό ποσοστό των παγκοσμίως εγκατεστημένων μονάδων παραγωγής ηλεκτρικής ενέργειας. Εκμεταλλεύονται μια ανανεώσιμη πηγή ενέργειας και δεν απαιτούν συχνή συντήρηση, αφού δεν περιέχουν μηχανικά μέρη. Αυτά, σε συνδυασμό με το γεγονός ότι τα εξαρτήματά τους γίνονται φθηνότερα με την πάροδο του χρόνου, εξηγούν την αυξανόμενη διείσδυση των Φ/Β σταθμών παραγωγής ισχύος στα σύγχρονα δίκτυα παραγωγής και διανομής ηλεκτρικής ενέργειας. Για να αποφευχθεί η διαταραχή της συχνότητας και να βελτιωθεί η ποιότητα ισχύος του δικτύου, ένας τρόπος ελέγχου είναι η εφαρμογή της διαδικασίας Flexible Power Point Tracking (FPPT), όπου η ισχύς του Φ/Β συστήματος ρυθμίζεται σε μια τιμή αναφοράς. Εφόσον τα Φ/Β πάνελ συνήθως σκιάζονται, οι συμβατικοί αλγόριθμοι FPPT δεν λειτουργούν αποδοτικά, επομένως αναπτύχθηκε η κατηγορία των Global FPPT (GFPPT) αλγορίθμων. Στην εργασία αυτή προτείνονται δυο νέοι GFPPT αλγόριθμοι, βασισμένοι στη μηχανική μάθηση. Για να αξιολογηθεί η αποτελεσματικότητά τους, αναπτύχθηκε ένα πειραματικό σύστημα. Ένας GFPPT αλγόριθμος από τη βιβλιογραφία και οι δυο νέοι δοκιμάστηκαν πειραματικά υπό μεταβαλλόμενες συνθήκες λειτουργίας, όπως θα μπορούσε να συμβεί και σε πραγματικές εφαρμογές. Τα πειραματικά αποτελέσματα έδειξαν ότι οι δυο νέοι αλγόριθμοι που αναπτύχθηκαν στο πλαίσιο αυτής της εργασίας συνέκλιναν σε σημαντικά μικρότερο χρονικό διάστημα, ενώ πέτυχαν ένα σχεδόν ίδιο σφάλμα παρακολούθησης στη μόνιμη κατάσταση.

Acknowledgements

I would like to thank my supervisor, Professor Eftichios Koutroulis, for his unceasing support and his truly excellent cooperation in conducting this thesis. Furthermore, I am grateful to:

- Professor Georgios Stavrakakis, not only for his admirable way of teaching about renewable energy sources, but also for participating in this thesis' examination committee,
- Associate Professor Fotios Kanellos, for his great lectures on power electronics and for the time he spent to examine this thesis,
- Emeritus Professor Kostas Kalaitzakis, for his incredible explanatory examples while teaching about electric measurements and sensors,
- Professor Apostolos Dollas, for the way he introduced me to the world of embedded systems,
- Professor Dionysios Christopoulos, for his valuable advice on technical writing, and
- Dr. Eleftheria Sergaki for her helpful tutoring on electrical power systems and the corresponding laboratory exercises.

Last, but not least, I want to thank my parents for their inexhaustible support and encouragement during the five-year period of my undergraduate studies.

Emmanouil Lioudakis

Chania, June 2023

1. Introduction

As of 2023, renewable energy sources accounted for 40% of the globally installed power plants. By the end of 2022, the globally installed photovoltaic system capacity reached approximately 1.1 TW [1]. The continuously increasing penetration of PV systems to the modern electricity transmission grids introduces the imperative need to control the PV plants' output power.

On each electricity transmission grid, an imbalance between the generated power and the power that is demanded by the loads can cause a fluctuation on the its frequency [2]. This fluctuation can cause a blackout, if the frequency does not recover to its normal level soon. Since everything nowadays is based on electricity, from communications to medical instruments, it is crucial to keep the grid operating at its normal condition.

Traditionally, when a PV power plant (either in the form of a field-installed plant, or in the form of a small rooftop system) is connected to the grid, it operates under Maximum Power Point Tracking (MPPT) mode, and outputs the maximum available power. When the grid operator needs to change the generated power, its only option is to connect or disconnect this PV plant.

Since this is not an effective technique, the concept of Flexible Power Point Tracking (FPPT) [3] was introduced, where instead of tracking the maximum available power, a specific value of power is tracked. This value is called reference power, and only if the reference power is greater than the available, then the system operates under MPPT mode. In that way, the grid operator is able to keep the generated power in balance to the demanded power.

When a PV array is installed outside, some of the PV modules may be shaded. This can happen because of neighboring buildings, trees and clouds or even because of the accumulated dust on their surface. Under partial shading conditions, the power-voltage characteristic of the PV array is changing shape and any calculations based on it become more complex. In that case, the MPPT and FPPT processes are called Global MPPT (GMPPT) and Global FPPT (GFPPT), respectively, since the aforementioned characteristic curve exhibits one or more local maximum power points. Many algorithms for the FPPT problem can be found in the literature, where the solar irradiance on the PV modules is uniform, however only a few exist for the GFPPT problem.

A GFPPT algorithm which regulates the output power of the PV array to a specific reference power value (an operation known as Constant Power Generation Control (CPGC) [3]), is reported in [4] and is based on a GMPPT scanning method of the power-voltage characteristic curve, with some modifications to reduce its convergence time. In order to avoid spending time on scanning the curve, which may happen frequently, if the solar irradiance changes frequently, this thesis introduces two new algorithms to solve the GFPPT problem, based on a machine learning

technique, called Q-learning. The first one is based on a Q-learning GMPPT algorithm [5], and after finding the GMPP, it limits the PV array's power to the reference value. To avoid the process of fine tuning the PV array's output power, which may take a lot of time if these two values differ significantly, the last algorithm, also based on Q-learning, aims exactly to the reference power. In that way, the convergence time is reduced even more.

This thesis is structured as follows:

- in Chapter 2, the basic principles of PV systems and their parts (such as DC/DC power converters) are explained,
- in Chapter 3, the experimental system that was implemented to test the algorithm (and its various components) is presented,
- in Chapter 4, the three GFPPT algorithms are analyzed in detail,
- in Chapter 5, the experimental procedure is shown, to compare the algorithms' performance, and
- in Chapter 6, the conclusions are presented, and possible future work is discussed.

2. Principles of PV systems

2.1. PV modules

A PV module (also called PV panel) is constructed by connecting multiple solar cells (in series and parallel). A solar cell is a semiconducting device that operates according to the PV effect. It absorbs photons from the solar irradiation, which lead to the creation of electron-hole pairs, resulting in an electric current flow (if the terminals of the device are connected to an electric circuit). A simplified equivalent circuit of a solar cell [6] is depicted in Fig. 2.1.

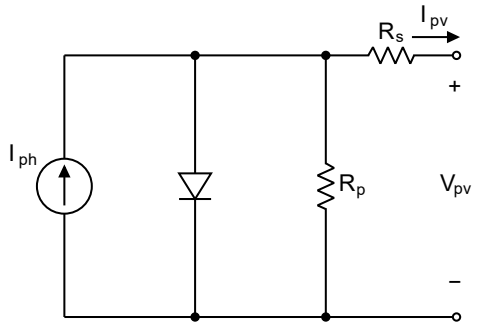

Figure 2.1: The equivalent circuit of a solar cell.

Based on the above circuit, the solar cell's current can be described by the following equation:

$$I_{pv} = I_{ph} - I_{r,sat} \cdot \left[e^{\frac{q_e \cdot (V_{pv} + I_{pv} \cdot R_s)}{n \cdot k \cdot T}} - 1 \right] - \frac{V_{pv} + I_{pv} \cdot R_s}{R_p} \qquad (2.1)$$

where:

- I_{ph} is the photocurrent,
- $I_{r,sat}$ is the reverse saturation current,
- q_e is the electron charge ($q_e = 1.6 \times 10^{-19}\ C$),
- V_{pv} is the voltage across the terminals of the solar cell,
- n is a factor expressing the quality of the semiconducting material,
- k is the Boltzmann constant ($k = 1.38 \times 10^{-23}\ J/K$),
- T is the absolute temperature of the solar cell,
- R_s is the series resistance of the solar cell, and
- R_p is the parallel resistance of the solar cell.

The output power of a solar cell can be expressed by:

$$P_{pv} = V_{pv} \cdot I_{pv} \qquad (2.2)$$

Based on (2.1) it is obvious that the power-voltage and the current-voltage curves of a solar cell are nonlinear [6], as it can be shown in Fig. 2.2. The solid line represents the current, while the dashed line represents the power. Three useful quantities can be extracted from theses curves:

- I_{sc}, the solar cell's short-circuit current,
- V_{oc}, the solar cell's open-circuit voltage, and
- The power at the MPP (Maximum Power Point), which is the maximum power that can be produced by the solar cell.

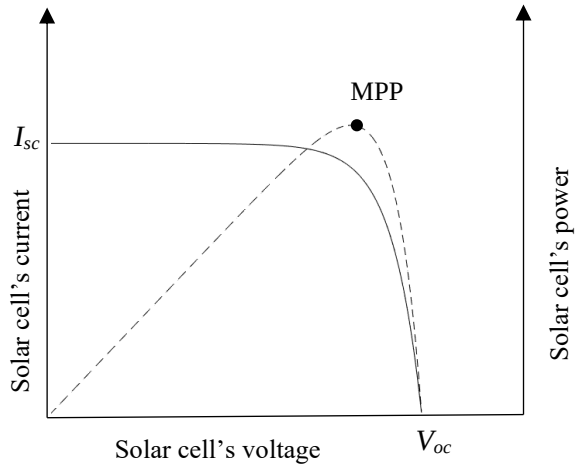

Figure 2.2: Current-voltage and power-voltage curves of a solar cell.

Since a PV module is constructed by connecting multiple solar cells in series and parallel, its power-voltage and current-voltage curves follow the same pattern as shown in Fig. 2.2. When the solar irradiance increases (while the temperature remains constant), the module's output power increases proportionally, while it decreases when the solar irradiance is reduced.

2.2. PV arrays

When multiple PV modules are connected in series and in parallel, a PV array is formed. The majority of PV power plants use more than one PV modules. To increase the PV array's output voltage the PV modules are connected in series, while to increase its current, they are connected in parallel. If the solar irradiance is equal for all the modules, then the array will operate efficiently, and its output power value will be proportional to the solar irradiance, when a circuit is connected to its output terminals (instead of a simple load, a DC/DC converter could be connected, or a grid-connected DC/AC inverter).

However, when the solar irradiance on each module is not the same, then the array will not operate efficiently. Supposing that there is a PV array with two panels connected in series (as shown in Fig. 2.3), where the upper one's (PV1) irradiance is 500 W/m^2 and the other's (PV2) is 1000 W/m^2. PV2 can generate a current flow

greater than PV1, but the two modules are connected in series, thus the same amount of current is will flow through them, the low current of PV1. In this case, the excess energy that is produced by PV2 will be consumed by PV1 and heat will be dissipated resulting in a “hot spot” [7]. As a result, the PV array will output only a part of the energy it can produce, but also the shaded panel may be damaged.

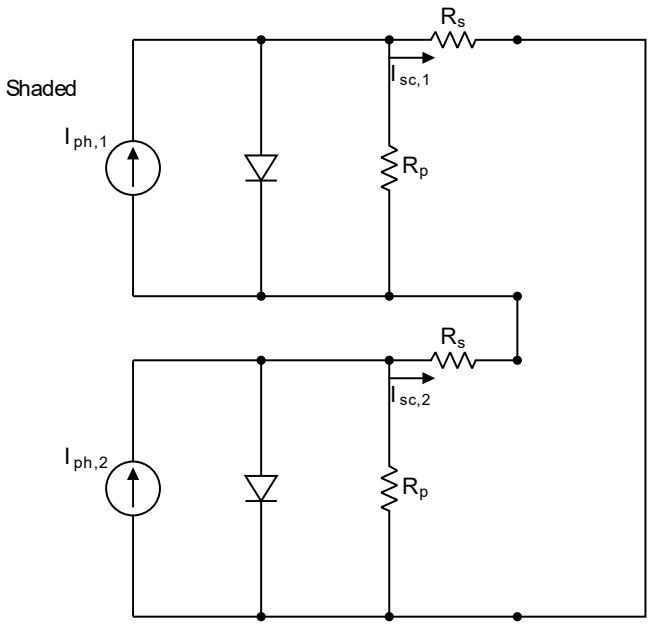

Figure 2.3: Series connection of two PV modules, without bypass diodes.

To overcome this issue, a diode can be added in parallel to each module, which is called bypass diode [7], as shown in Fig. 2.4. Physically, it is placed in the terminal box at the back side of the module. When the two panels receive the same amount of solar irradiance, their bypass diodes are reverse biased and act as an open-circuit, without affecting the PV system’s operation. On the contrary, when PV1 is partially shaded (thus receiving less solar irradiance than PV2), and produces a low current flow, its internal diode will be reverse biased and its bypass diode will conduct. A low voltage drop will occur, but the current flow is not going to reduce on PV2. Consequently, the creation of a “hot spot” is avoided and the PV system’s efficiency will increase.

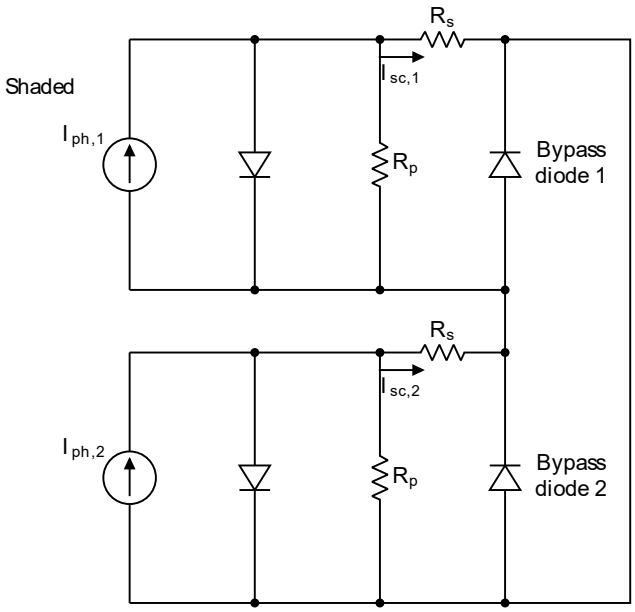

Figure 2.4: Series connection of two PV modules, with added bypass diodes.

After adding the bypass diodes, the PV array's power-voltage and current-voltage curves will have a form like this of Fig. 2.5 and 2.6 respectively, with two maximum power points, one local (LMPP) and one global (GMPP).

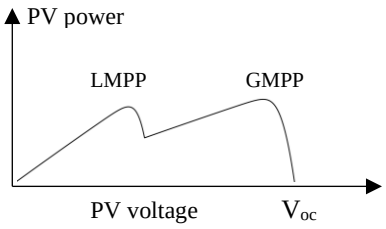

Figure 2.5: The PV array's P-V curve under partial shading conditions.

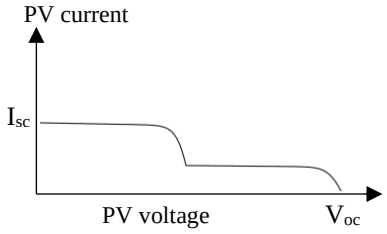

Figure 2.6: The PV array's I-V curve under partial shading conditions.

2.3. DC/DC Boost converters

2.3.1. Principles of operation

The circuit of a typical DC/DC boost converter is depicted in Fig. 2.7. It is a circuit whose output voltage is greater or equal than its input voltage, depending on the duty cycle of the electronic switch. Assuming that between the switch and the diode, only one will conduct each moment, the converter's operation can be divided in

two phases. During the first phase the switch is on (and the diode does not conduct), while during the second phase the switch is off (and the diode is forward biased).

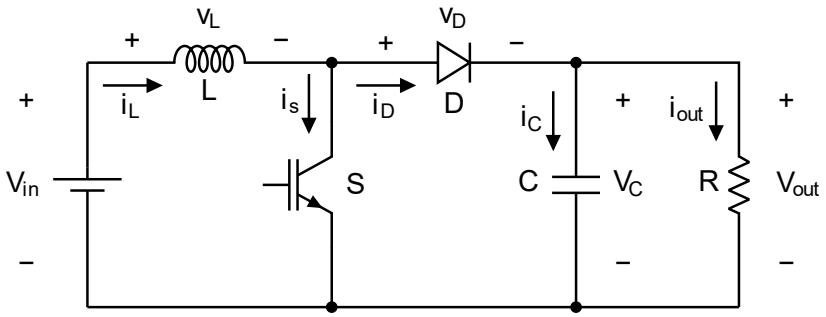

Figure 2.7: The Boost-type DC/DC converter's circuit.

The converter's equivalent circuit while the switch is on is shown in Fig. 2.8. Since the switch acts as a short-circuit, the diode is reverse biased and the input voltage is applied across the inductor. During this phase, the inductor is being charged from the input source and its current rises until reaching its maximum value, $I_{L,max}$. On the output side, the capacitor (which has been charged during the previous phase) discharges onto the load.

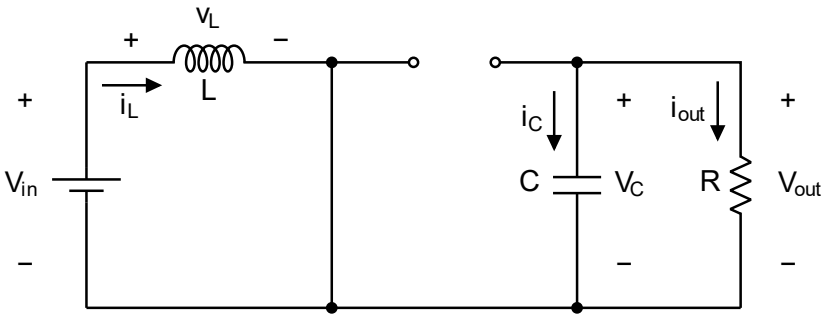

Figure 2.8: The Boost-type DC/DC converter’s equivalent circuit when the switch is on.

While the switch is off (Fig. 2.9), the energy stored in the inductor should discharge in some way. To achieve that, the inductor biases the diode forwardly. Thus, the inductor charges the capacitor and feeds the load. Subsequently, the inductor's current slowly decreases, until reaching its minimum value, $I_{L,min}$.

$$v_L(t) = L\frac{di_L(t)}{dt} \tag{2.3}$$

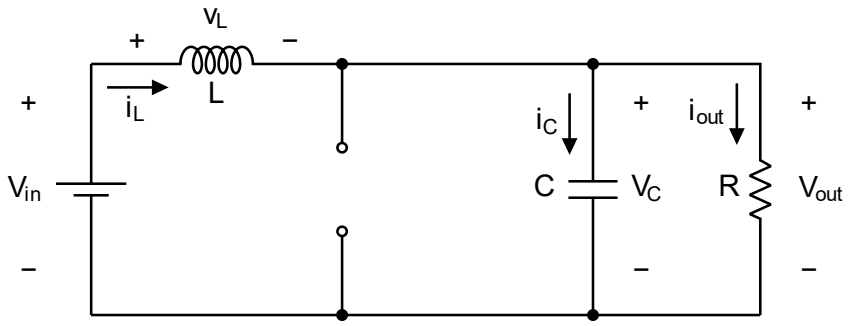

Figure 2.9: The Boost-type DC/DC converter's equivalent circuit when the switch is off.

From the two equivalent circuits (Fig. 2.8 and 2.9), the following can be derived for the component's voltages and currents:

$$v_{L,on}(t) = V_{in} \tag{2.4}$$

$$i_{L,on}(t) = \int_{t_0}^{t} V_{in}\, dt \; + \; i_L(t_0) \tag{2.5}$$

$$v_{L,off}(t) = V_{in} - v_c(t) = V_{in} - v_{out}(t) \tag{2.6}$$

$$i_{L,off}(t) = \frac{1}{L}\int_{t_0}^{t} \big(V_{in} - v_{out}(t)\big)\, dt \; + \; i_L(t_0) \tag{2.7}$$

$$v_c(t) = \frac{1}{C}\int_{t_0}^{t} i_c(t)\, dt \; + \; v_c(t_0) \tag{2.8}$$

$$i_c(t) = i_D(t) - i_{out}(t) \tag{2.9}$$

$$i_D(t) = i_{L,off}(t), \qquad \text{when it conducts} \tag{2.10}$$

$$i_s(t) = i_{L,on}(t), \qquad \text{when it is on} \tag{2.11}$$

The waveforms of $i_L(t)$, $i_s(t), i_D(t)$, $v_L(t)$ and $v_{out}(t)$ [8] are shown in Fig. 2.10. The minimum and maximum value of the inductor's current are described by the following equations:

$$I_{L,min} = \overline{I_L} - \frac{\Delta I_L}{2} \tag{2.12}$$

$$I_{L,max} = \overline{I_L} + \frac{\Delta I_L}{2} \tag{2.13}$$

where the inductor's current fluctuation ΔI_L can be described as:

$$\Delta I_L = I_{L,max} - I_{L,min} = \frac{V_{in}}{L} \cdot t_{on} \tag{2.14}$$

and t_{on} is the time period when the switch is on.

During the first phase, the amount of energy stored in the inductor can be described by the following equation:

$$W_{on} = V_{in} \cdot \overline{I_L} \cdot t_{on} = V_{in} \cdot \overline{I_L} \cdot D \cdot T_s \tag{2.15}$$

while during the second phase, the amount of energy that is discharged from the inductor is:

$$W_{off} = \left(V_{in} - \overline{V_{out}}\right) \cdot \overline{I_L} \cdot t_{off} = \left(V_{in} - \overline{V_{out}}\right) \cdot \overline{I_L} \cdot (1 - D) \cdot T_s \tag{2.16}$$

Supposing that there are no power losses in the converter, W_{on} and W_{off} should sum to zero.

$$W_{on} + W_{off} = 0 \Rightarrow \overline{V_{out}} = V_{in} \cdot \frac{T_s}{T_s - t_{on}} \Rightarrow V_{out} = \frac{V_{in}}{1 - D} \tag{2.17}$$

It is noted that for the last step of (2.17), the output capacitor is assumed to be large enough, in order for V_{out} to be almost stable (any ripple is negligible).

Additionally, because of the lossless circuit's assumption:

$$P_{in} = P_{out} \Rightarrow V_{in} \cdot \overline{I_L} = V_{out} \cdot \overline{I_{out}} \Rightarrow \overline{I_L} = \frac{V_{out} \cdot \overline{I_{out}}}{V_{in}} \Rightarrow \overline{I_L} = \frac{\overline{I_{out}}}{1 - D} \tag{2.18}$$

The average value of the inductor's current can be expressed by combining (2.17) and (2.18).

$$\overline{I_L} = \frac{\overline{I_{out}}}{1 - D} = \frac{V_{out}}{(1 - D)R} = \frac{V_{in}}{(1 - D)^2 R} \tag{2.19}$$

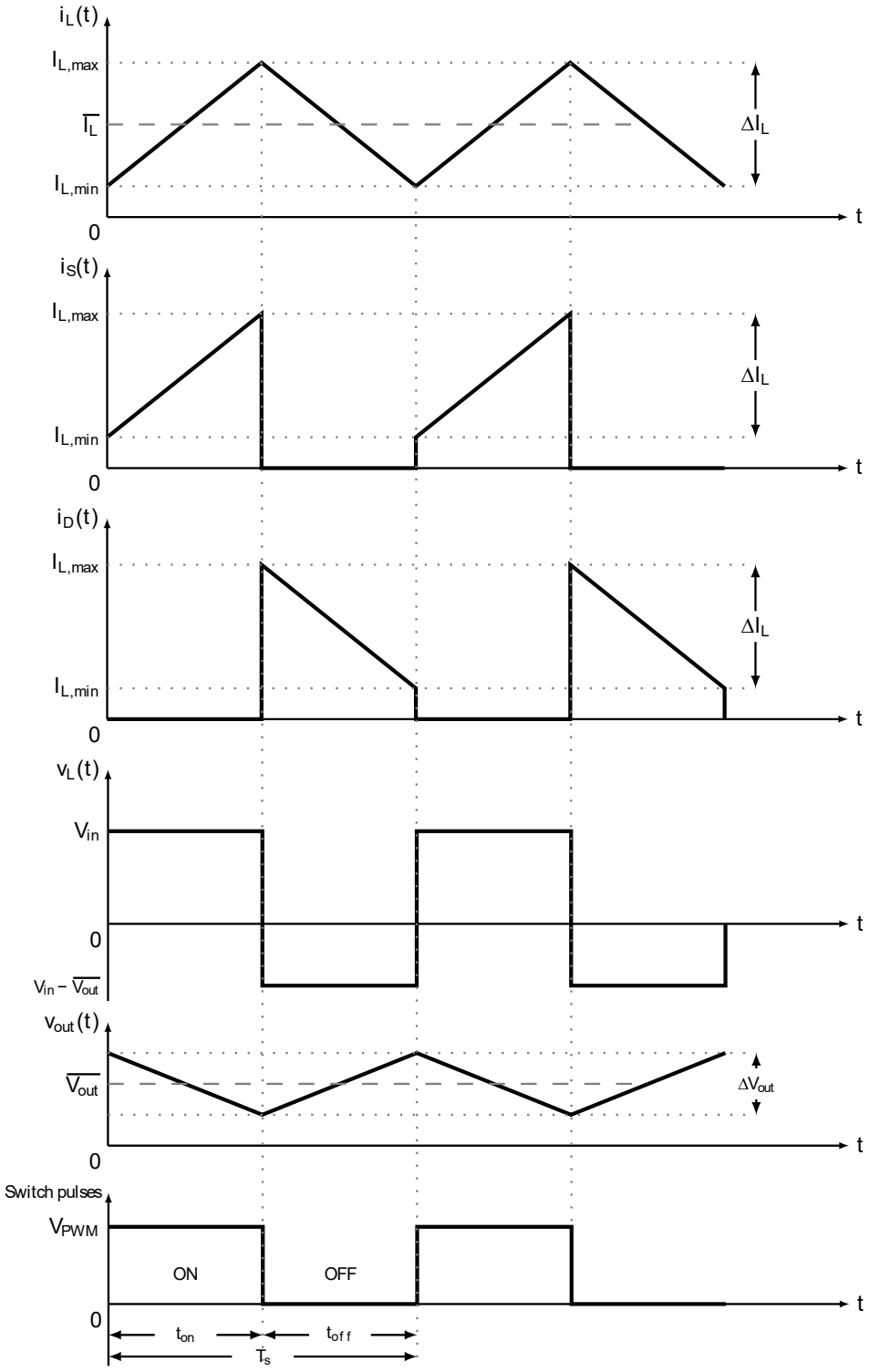

Figure 2.10: DC/DC boost converter waveforms.

In some cases, Boost converters are designed to have discontinuous inductor current. This means that during the second phase of their operation, the inductor will completely discharge, and its current will be zero until the next switching pulse. However, in applications where PV modules and batteries are used, it is preferred to obtain an intermittent current flow through the inductor, because it offers improved

control behavior and stability [8]. Thus, to obtain continuous inductor current, based on (2.12), it should be ensured that:

$$\overline{I_L} \geq \frac{\Delta I_L}{2} \tag{2.20}$$

At the boundary condition between continuous and discontinuous mode of operation, the equality gives:

$$\overline{I_L} = \frac{\Delta I_L}{2} = \frac{V_{in}}{2L} \cdot t_{on} = \frac{V_{in}}{2L} \cdot D \cdot T_s = \frac{V_{out} \cdot (1-D) \cdot D}{2L \cdot f_s} \tag{2.21}$$

Thus, the minimum inductance that can be used to ensure continuous inductor current flow can be expressed by rearranging the terms of (2.21).

$$L_{min} = \frac{V_{out} \cdot (1-D) \cdot D}{2\overline{I_L} \cdot f_s} \tag{2.22}$$

The voltage ripple on the capacitor is:

$$\Delta V_{out} = \frac{\Delta Q_{out}}{C} = \frac{\overline{I_{out}} \cdot t_{on}}{C} = \frac{\overline{I_L} \cdot D^2 \cdot T_s}{C} = \frac{\overline{I_L} \cdot D^2}{C \cdot f_s} \tag{2.23}$$

As a result, given a desired output voltage fluctuation ΔV_{out}, the capacitance value that should be used is:

$$C = \frac{\overline{I_L} \cdot D^2}{\Delta V_{out} \cdot f_s} \tag{2.24}$$

2.3.2. Electronic switches

Every DC/DC converter needs an electronic switch to operate. This can be either:

- a BJT (Bipolar Junction Transistor), or
- a MOSFET (Metal Oxide Semiconductor Field Effect Transistor), or
- an IGBT (Insulated Gate Bipolar Transistor).

There are also other types of switches, suitable for high voltage and high current applications, that will not be considered on this thesis. A comparison between the aforementioned types of switches can be summarized as follows:

- A BJT is current controlled and it is mainly used in applications that use moderate switching frequencies, up to 20-100 kHz. However, since it is current controlled, it usually requires complex circuits to drive it base. Additionally, is shows high switching power losses.
- A MOSFET is voltage controlled and has low switching power losses. It can operate in switching frequencies up to 500 kHz, and its gate driving circuit is simple.

- An IGBT is also voltage controlled and was designed to keep the advantages of both a BJT and a MOSFET and it can operate in low switching frequencies, up to 20-50 kHz. Moreover, like the MOSFET, its gate driving circuit is simple. As a result, through the last decade, IGBT's are extensively used in industrial products such as DC/AC inverters, UPS (Uninterruptible Power Sources) and AC motor drives [9].

3. Experimental setup

To evaluate the performance of the three PV GFPPT algorithms in real operating conditions, an experimental system has been implemented, composed of:

- 2 PV panels connected in series, to form an array,
- a DC/DC Boost converter,
- a battery bank,
- a voltage and a current sensor for the PV array, and
- the control unit.

The block diagram of the system is presented in Fig. 3.1.

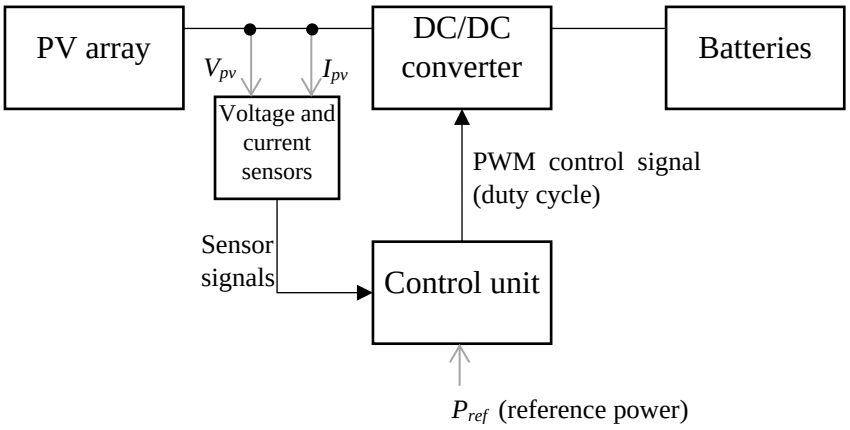

Figure 3.1: Block diagram of the developed PV system.

The experimental setup, installed at the testing field, is depicted in Fig. 3.2. Its parts will be explained in detail on this chapter.

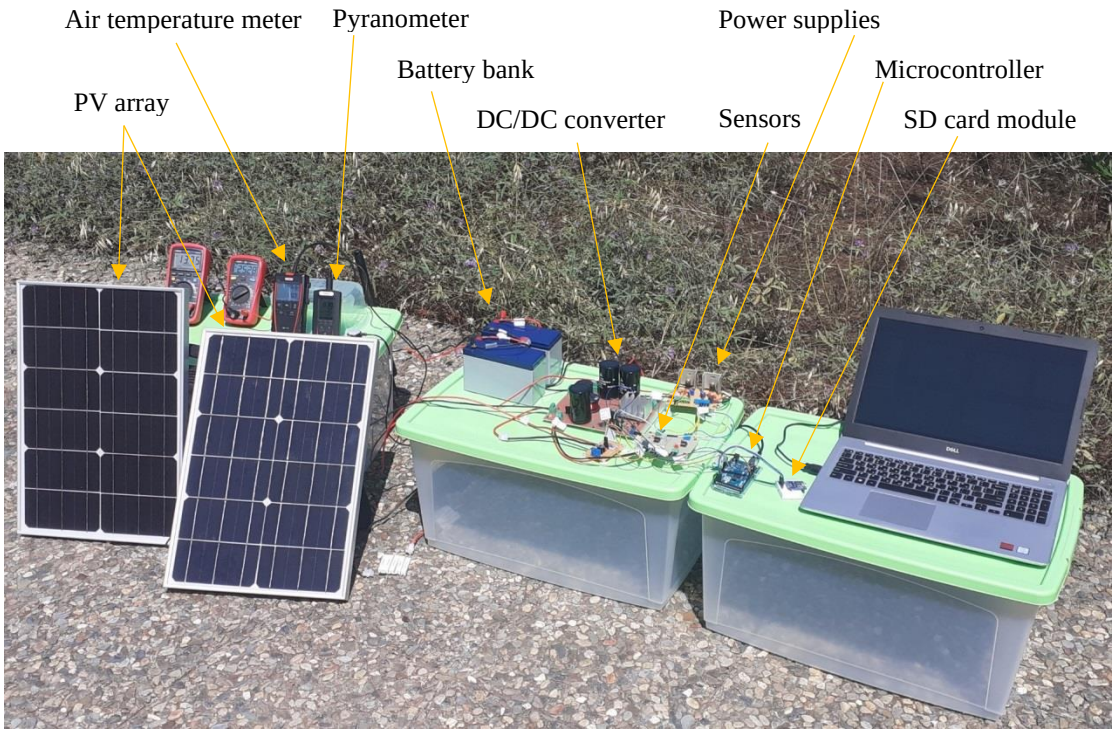

Figure 3.2: Photo of the experimental setup.

3.1. The PV array

Two PS-20W PV panels were used. Their technical data under STC (Standard Testing Conditions of 1000 W/m^2 irradiance and 25°C temperature) is presented in Table 3.1. To form a PV array and study the algorithm's performance under partial shading conditions, the two panels were connected in series and a 6A10 diode was connected in parallel to each PV panel, to act as a bypass diode. As it can be seen in Fig. 3.2, the two panels' tilt angle was different, to obtain the partial shading conditions.

Table 3.1: PS-20W PV panel technical specifications at STC.

Maximum power	20 W
Maximum power point current	3.33 A
Maximum power point voltage	6.00 V
Short-circuit current	6.13 A
Open-circuit voltage	9.00 V

3.2. The DC/DC converter

For the requirements of previous research work in the university's laboratory, a DC/DC converter board was deployed. However, since the values of the capacitors

and the inductors were not suitable for this thesis' system, the board outline was kept, but the components were changed, as described in section 3.3.2. Instead of a simple electronic switch and a diode, the board featured an IPM (Integrated Power Module), which contained 6 IGBTs with anti-parallel diodes (providing the ability to be used in 3-phase alternating current applications, such as 3-phase inverters). This thesis' system is operating with direct current, consequently only one of the phases was used (2 out of the 6 IGBTs), as shown in Fig. 3.3. The upper IGBT (S_{up}) acts as the converter's diode, thus its gate is connected to the ground and its antiparallel diode is going to conduct. On the other side, the lower IGBT (S_{down}) plays the role of the converter's switch and the PWM signal of the control unit is applied to its gate. As shown in the following schematic, the gray parts will never conduct.

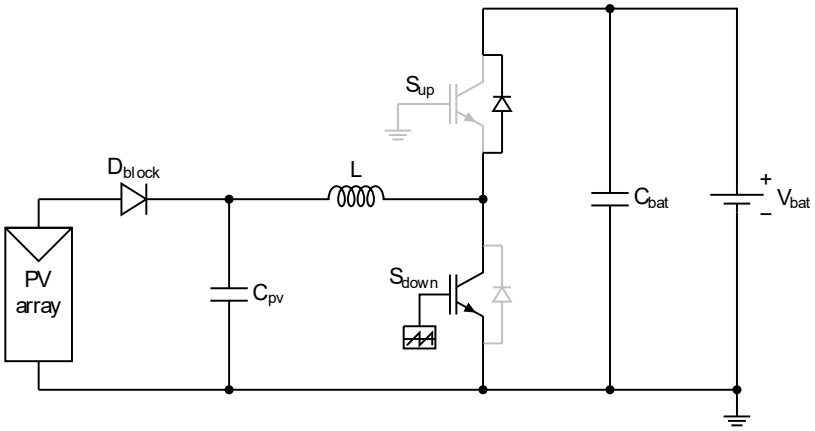

Figure 3.3: The DC/DC converter using an IPM.

3.2.1. DC/DC converter voltage control

One of the algorithms that will be analyzed in Section 4, outputs a reference voltage value to control the PV array's voltage. This value cannot be directly converted into a PWM signal. For this conversion, usually PI controllers are used [10]. However, these controllers need tuning of their parameters. To avoid any overshoot that may occur during the tuning process, a simpler type of controller was employed, where the PV array voltage is compared to the reference voltage each 25 msec and the duty cycle of the PWM signal is updated as follows:

$$D = \begin{cases} D + D_{step}, & \text{if } V_{pv} \geq V_{ref} \\ D - D_{step}, & \text{otherwise} \end{cases} \tag{3.1}$$

where the step of the duty cycle, D_{step}, is defined as follows:

$$D_{step} = \begin{cases} 0.025, & |V_{pv} - V_{ref}| > 2.5\ V \\ 0.01, & 1\ V \leq |V_{pv} - V_{ref}| \leq 2.5\ V \\ 0.0025, & |V_{pv} - V_{ref}| < 1\ V \end{cases} \tag{3.2}$$

On the contrary, the two last algorithms output a value for the duty cycle. However, until their convergence, the maximum duty cycle change that can occur between two calls is 28%. A sudden change to the inductor's current can cause an overvoltage at its terminals, which may damage the DC/DC converter's components. Consequently, as before, the duty cycle was changed in smaller steps each 25 msec, based on the following rules:

$$D = \begin{cases} D_{old} - D_{step}, & \text{if } D \geq D_{old} \\ D_{old} + D_{step}, & \text{otherwise} \end{cases} \tag{3.3}$$

where the step of the duty cycle, D_{step}, is defined as follows:

$$D_{step} = \begin{cases} 0.025, & |D - D_{old}| > 0.2 \\ 0.01, & 0.1 \leq |D - D_{old}| \leq 0.2 \\ 0.0025, & |D - D_{old}| < 0.1 \end{cases} \tag{3.4}$$

3.2.2. Sizing the converter's passive components

The diode that was already placed on the DC/DC converter was the RFV15TJ6S, which was utilized as the PV array's blocking diode, to prevent any current flow from the system subcircuits to the panels.

To size the inductor for the converter, (2.22) is used, with $D = 0.5, f_s = 20\,kHz$, $\overline{I_L} = 1\,A$ and $V_{out} = 24\,V$:

$$L_{min} = \frac{1}{2}\frac{24 \cdot 0.5 \cdot 0.5}{1 \cdot 20000} = 150\,\mu H \tag{3.5}$$

A safety margin of 20% is added, resulting to L = 180 μH. The Coilcraft PCV-2-184-10L inductor was used for that purpose.

To size the input and output capacitors, the theoretical values of $\overline{I_L}$ and $\overline{I_{out}}$ should be calculated again, using the new inductance value:

$$\overline{I_L} = \frac{1}{2}\frac{24 \cdot 0.5 \cdot 0.5}{(180 \times 10^{-6}) \cdot 20000} = 830mA \tag{3.6}$$

$$\overline{I_{out}} = \frac{1}{2}\frac{24 \cdot 0.5^2 \cdot 0.5}{(180 \times 10^{-6}) \cdot 20000} = 420mA \tag{3.7}$$

The voltage ripple on the input capacitor is described by the following equation:

$$\Delta V_{in} = \frac{\Delta Q_{in}}{C_{pv}} = \frac{\overline{I_L} \cdot t_{on}}{C_{pv}} = \frac{\overline{I_L} \cdot D \cdot T_s}{C_{pv}} = \frac{\overline{I_L} \cdot D}{C_{pv} \cdot f_s} \tag{3.8}$$

Thus, to obtain an input voltage ripple of 20 mV (with $D = 0.5, f_s = 20$ kHz and $\overline{I_L} = 830\,mA$), the capacitance of C_{pv} should be 1 mF. One aluminum electrolytic capacitor B43516-B9108-M 1000μF was used.

The voltage ripple on the output capacitor is described by (2.24). To obtain an output voltage ripple of 8 mV (with $D = 0.5, f = 20$ kHz and $\overline{I_{out}} = 420$ mA), the capacitance of C_{bat} should be 1.3 mF. Two aluminum electrolytic capacitors B43516-B9108-M 1000μF were connected in parallel.

The board of the DC/DC converter in its final form is depicted in Fig. 3.4.

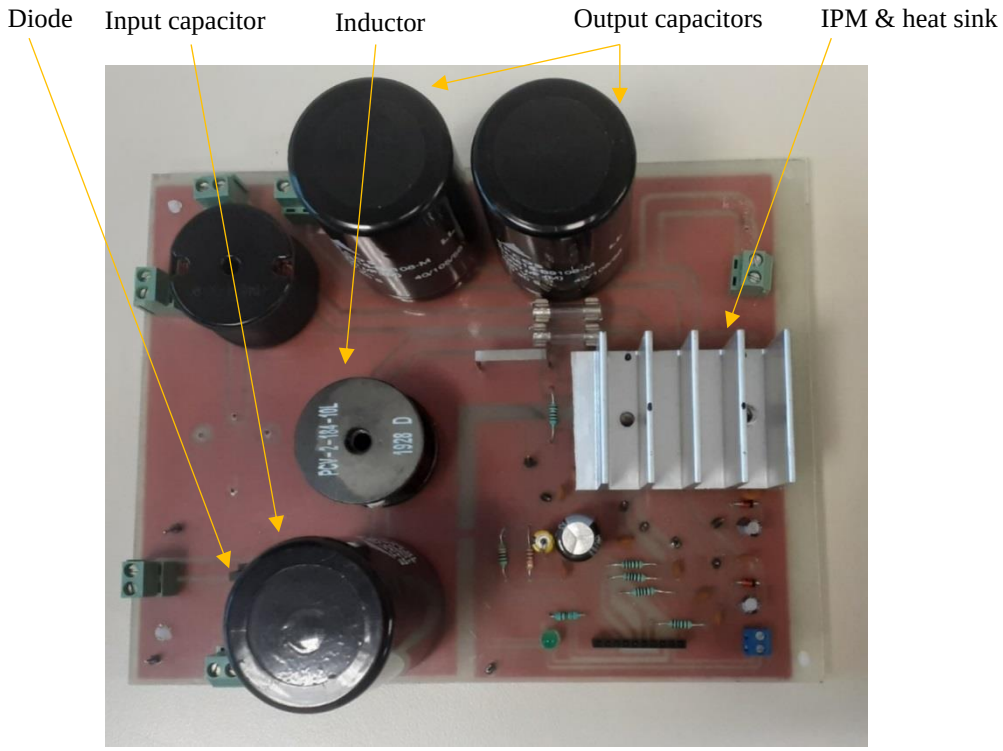

Figure 3.4: The DC/DC converter's board.

3.2.3. The IPM's fault signal

The IPM Infineon IKCM30F60GD which was used for the DC/DC converter, features a pin called VFO which can be used to detect any module faults during its operation. It is triggered when there is an undervoltage at the supplying pin of the IPM (because supplying the IPM with an out of bounds voltage may damage it) and when overcurrent occurs (to avoid burning the IPM). When a fault occurs, the IPM automatically shuts down internally all the IGBT gates in 380 nsec (typically), and does not follow the PWM pulses supplied to the control pins. When the current drops to its normal level, the gates' control is enabled again.

Furthermore, the VFO pin is also internally connected to a NTC thermistor, giving the ability to measure the IPM's temperature. For the current application, 80°C is considered as the threshold between normal operation and overheating. The pin needs an external pull-up resistor, which was connected to the +5V rail, in order to be on the same voltage level with the microcontroller. The thermistor's resistance at 80°C based on the datasheet [11] is presented in Table 3.2.

Table 3.2: The thermistor's resistance.

Temperature	R_{min}	R_{typ}	R_{max}	**Tolerance**
80°C	9.745 kΩ	10.169 kΩ	10.593 kΩ	4.2%

Based on the R_{typ} value, it is considered that the thermistor's resistance is R_{therm} = 10kΩ. The pull-up resistor's value is $R_{pull-up,IPM}$ = 3.3kΩ, thus when the temperature reaches 80°C, the voltage at the VFO pin will be:

$$V_{80°}^{VFO} = \frac{R_{therm}}{R_{therm} + R_{pull-up,IPM}} \cdot 5\ V = \frac{10\ k\Omega}{10\ k\Omega\ +\ 3.3\ k\Omega} \cdot 5\ V = 3.76\ V \qquad (3.9)$$

In case of undervoltage or overcurrent, the VFO pin's voltage will drop from 5 V to almost 0 V, while in case of overtemperature, it will drop below 3.76 V. As a result, in order to detect any faults in the system, an analog comparator can be used. While the VFO pin's voltage is above 3.76 V, the system is operating normally and if it drops above this level, a fault has occurred, so the PWM pulses should stop immediately.

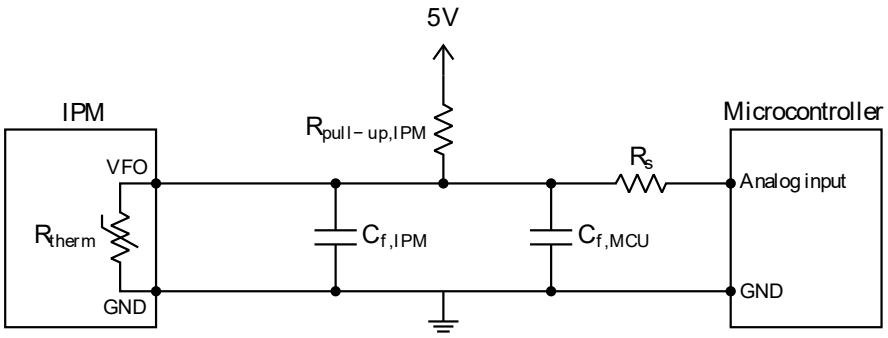

Figure 3.5: The VFO signal interface circuit, based on the IPM's datasheet.

Although the microcontroller used on the Arduino board (Microchip ATmega 2560) contains an analog comparator [12], only one of its two input pins is connected on the Arduino board [13] (Fig. 3.6). As a result, the topology of Fig. 3.5 cannot be used directly, and an external circuit has been designed, using the analog comparator LM393 (Fig. 3.7).

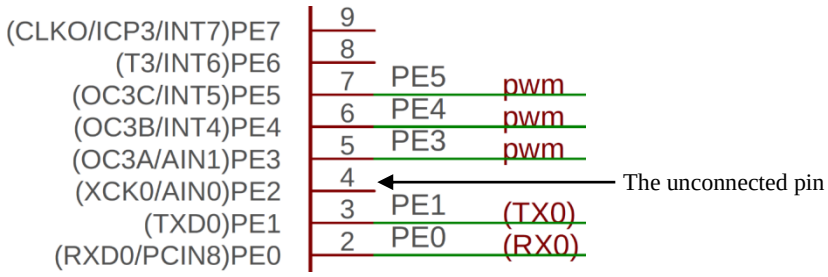

Figure 3.6: The analog comparator's pin on Arduino Mega.

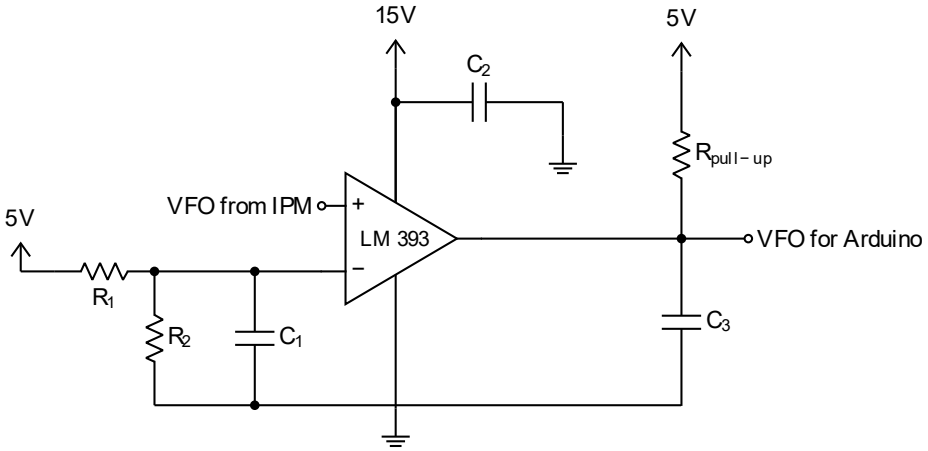

Figure 3.7: The analog comparator's circuit.

The resistors R_1 and R_2 form a voltage divider in order to supply the negative input pin of the comparator with 3.76 V. Their values were calculated using (3.10):

$$V_{comp} = \frac{R_2}{R_1 + R_2} \cdot 5\ V \Rightarrow \frac{R_2}{R_1 + R_2} = \frac{V_{comp}}{5\ V} = \frac{3.76\ V}{5\ V} \Rightarrow R_2 = 3.03R_1 \qquad (3.10)$$

and they were chosen to be $R_1 = 3.3\ k\Omega$ and $R_2 = 10\ k\Omega$.

Since the comparator features an open-drain output, a pull-up resistor should be used, in order to be able to drive a digital input pin of the microcontroller, whose value was chosen to be $R_{pull-up} = 10\ k\Omega$. The capacitors C_1 and C_3 were added to filter noise, while C_2 decouples the power supply pin of the comparator. Based on the comparator's datasheet [14], its input voltage should not exceed $V_{cc} - 2\ V$, where V_{cc} is its supplying voltage. Since the VFO ranges between 0 and 5 V, the comparator should be supplied with at least 7 V, thus it had been connected to the 15 V supply. The analog comparison circuit is depicted in Fig. 3.8.

Table 3.3: The components used for the analog comparator's circuit.

Item	Description
R_1	3.3 kΩ resistor
R_2 $R_{pull-up}$	10 kΩ resistors
C_1 C_2 C_3	0.1 μF polyester capacitor

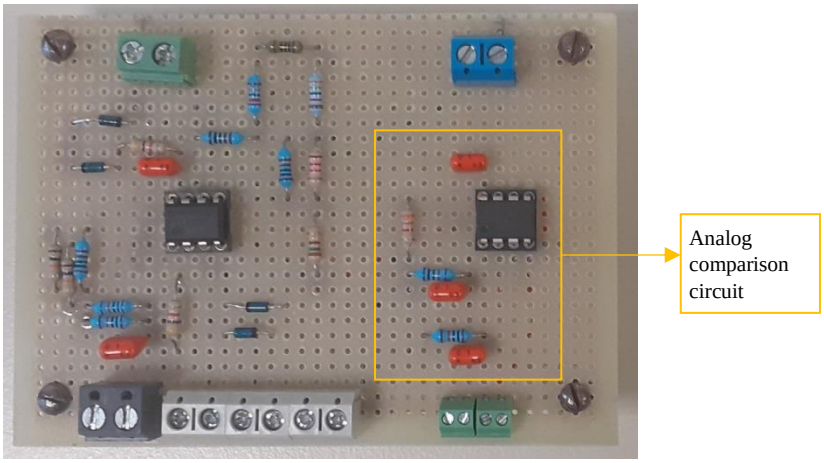

Figure 3.8: The analog comparator's circuit experimental prototype.

3.3. The battery bank

A battery bank consisting of two 12 V, 7.2 Ah batteries connected in series was used as a load for the DC/DC converter. Furthermore, the battery bank was used to supply the subcircuits, via the power supply board (presented in Section 3.7). To prevent the batteries' overcharging during the experiments, a 50 Ω resistive load was connected in parallel to them. Assuming that the bank has a stable voltage of 24 V, 480 mA of current will flow through the load, consuming constantly 11.52 W of power.

3.4. The microcontroller

In order to control the DC/DC converter and execute the algorithms, the Arduino Mega 2560 microcontroller unit was used, which features the Microchip ATmega 2560 microcontroller and has the following specifications:

- 5 V operating voltage

- 256 KB flash memory
- 8 KB SRAM
- 4 KB EEPROM
- 16 MHz CPU clock speed
- 54 digital I/O pins (15 of them providing PWM output)
- 16 analog input pins with 10-bit ADC
- SPI (Serial Peripheral Interface) connectivity
- 4 serial communication ports

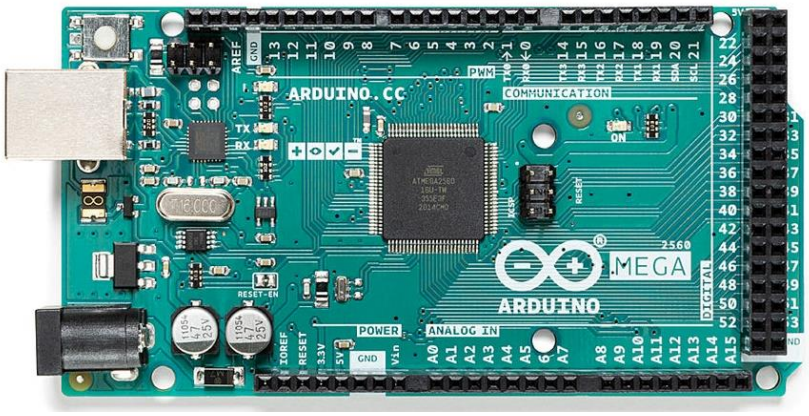

Figure 3.9: The Arduino Mega 2560 Rev3 board.

3.4.1. Generating the PWM signal

The 16-bit Timer/Counter 1 was used to produce the PWM signal to control the Boost converter's switch. To set the PWM signal's frequency, the appropriate value should be written to the register ICR1. Based on the microcontroller's datasheet [12], this value is given by:

$$f_{PWM} = \frac{f_{CPU}}{N \cdot (1 + \text{ICR1})} \Rightarrow \text{ICR1} = \frac{f_{CPU}}{N \cdot f_{PWM}} - 1 \tag{3.11}$$

where N is the prescaler divider value and f_{CPU} the microcontroller's CPU clock frequency, 16 MHz.

In order to achieve a frequency of $f_{PWM} = 20$ kHz (which is the maximum allowed for the IPM's normal operation [15]), the prescaler was set to $N = 1$ and ICR1 was set to 799. To set a specific value of duty cycle for the PWM signal, the value of the OCR1A register should be set, based on the following equation:

$$\text{OCR1A} = D \cdot \text{ICR1} \tag{3.12}$$

where D is the duty cycle, as a decimal number ranging from 0 to 1.

3.4.2. Serial port configuration

When operating normally (for instance during the execution of an algorithm), the controller sends a measurement packet (in the form: "M,<V_{pv}>,<I_{pv}>,<D><CR><LF>"), which consists of 26 ASCII characters in the worst case. One package like that is sent to the host computer each 1 second. The configuration of the serial port's packet is 8 data bits, no parity bit and 1 stop bit. Thus, the controller's transmission rate will be 234 bauds per second. As a result, the serial port's speed was set to the typical value of 9600 bauds per second, which also allows to send a few informative messages per second if needed.

3.4.3. External memory

As it will be mentioned in Section 4, the two last algorithms that are based on reinforcement learning, need to store two multidimensional arrays, named Q and N, in order to operate normally. Therefore, the state-space for both algorithms is extensive. However, the internal SRAM of Arduino Mega is only 8 KB, which is not enough. In detail, the Q arrays store floating point numbers (which occupy 4 bytes of memory), while the N ones store unsigned integers not greater than 2^{16}, thus a uint16_t data type can be used (which occupies 2 bytes of memory). Thus, the total memory usage is approximately 1.1 MB, as shown in Table 3.4.

Table 3.4: Detailed memory usage for the reinforcement learning data.

Algorithm	**State space**	**Q array size**	**N array (does not include the action field)**
2	P_{pv}: 12 states D: 12 states D_{old}: 6 states action: 7 states	6048 floating point values 24192 bytes ≈ 24 KB	864 integer values 1728 bytes ≈ 2 KB
3	P_{ref}: 10 states P_{pv}: 20 states D: 20 states D_{old}: 10 states action: 7 states	280000 floating point values 1120000 bytes ≈ 1 MB	40000 integer values 80000 bytes ≈ 78 KB

External memory – Hardware

In order to store the data on Table 3.4, an external memory module was used. The ATmega microcontroller features an SPI (Serial Peripheral Interface) to provide connectivity with peripheral devices. SPI is a synchronous serial data transmission protocol based on four main lines [16]:

- COPI (Controller Output - Peripheral In): the line through which data is sent from the controller to the peripheral device.
- CIPO (Controller Input - Peripheral Out): the line through which data is sent from the peripheral device to the controller.
- SCK (Serial Clock): since SPI is based on synchronous transmission, the controller has to generate clock pulses to synchronize the data transmission between itself and the devices.
- CS (Chip Select): a controller can be connected with many peripheral devices (which all use the same COPI, CIPO and SCK lines). However, each moment it can communicate with only one of them. Using the CS pin of each device, the controller can enable or disable it (when CS is low, then the device can communicate with the controller, whilst if CS is high, any data sent from the device to the controller are not going to be received by the former).

For this thesis, the Adafruit 254 SD Card Module was chosen, which features a microSD card slot and the required circuitry to be interfaced with either a 5 V or a 3.3 V Arduino board. It is depicted in Fig. 3.10. Except for a status indicating LED and a few filtering/decoupling capacitors, it contains:

- A 3.3 V voltage regulator (LP298x), since an SD card needs a stable power supply of 3.3 V to operate properly.
- A logic level shifter (74HC4050), which converts the transmitted data from 3.3-5 V to 3.3 V, since a 5 V Arduino board will produce 5 V clock and data pulses on the SPI bus, and the card would be damaged.

The pin connection between the Arduino and the module is shown on Table 3.5.

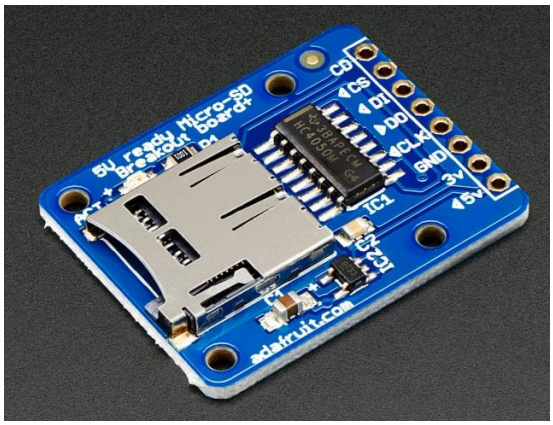

Figure 3.10: The Adafruit 254 SD Card Module.

Table 3.5: Adafruit 254 SD Card Module connection.

Adafruit 254 pin	Arduino Mega 2560 pin
5V	5V
GND	GND
DO	CIPO (Digital Pin 50)
DI	COPI (Digital Pin 51)
CLK	SCK (Digital Pin 52)
CS	CS (Digital Pin 53)

External memory – Software

In order to be stored in a file, a multidimensional array should be converted to a serial file (resembling a unidimensional array). This was achieved by storing the data of each dimension in successive positions to the file after proceeding to the next dimension. The process is presented for the 2nd algorithm's N array, which has only three dimensions, for simplicity purposes. Obviously, the conversion of four and five dimensional arrays is based on the same principle.

The array has the following form:

$$\mathrm{N\,[dim1][dim2][\,dim3]}$$

and after the conversion it will be converted to a serial file, where the line number is calculated as follows:

The line number if the array had only two dimensions would be:

$$line1 = dim1$$

The maximum line number of the first dimension is:

$$dim1_{max} = (\text{number of } P_{pv} \text{ states})$$

The line number if the array had only two dimensions would be:

$$line2 = dim2 \times dim1_{max} + dim2 + line1$$

The maximum line number of the second dimension is:

$$dim2_{max} = (\text{number of } D \text{ states}) \times dim1_{max} + (\text{number of } D \text{ states}) + dim1_{max}$$

Finally, the line number of the third dimension is:

$$index = dim3 \times dim2_{max} + dim3 + line2$$

The data will be stored in a binary file, in successive positions. Consequently, to access a specific entry of the N array, the seek function should be called on the file pointer (fp):

```
fp.seek(index*sizeof(uint16_t))
```

Now that the pointer has been set to the desired position, the cell's value can be read or modified using the following functions:

```
fp.read((uint16_t*)&read_value, sizeof(uint16_t))
fp.write((const uint8_t *)&write_value, sizeof(uint16_t))
```

It is worth noting that in order to open the file, the "(O_READ|O_CREAT)" mode is used when the file is opened for reading, and the "(O_READ|O_WRITE|O_CREAT)" mode when it is opened for writing.

3.4.4. The user interface

As described in Section 3.4.2, during the controller's operation, measurements and informative messages are sent to the computer from the controller. Additionally, the computer can send specific commands to control the microcontroller's behavior. Using a simple terminal application, the measurements will be visible in their raw form, for example "M,12.5858,1.2222,0.5050", which is difficult to understand without being visualized. Another approach to the problem was to use the Serial Plotter which is integrated into the Arduino IDE. However, it offers only one axis system on which all the variables would be plotted, and more than that, it does not support saving the measurements (thus, they would be stored in the SD card and plotted after turning of the Arduino and connecting the SD card to the computer). However, this is still ineffective, thus a new application would be developed to fulfill the requirements.

Using Microsoft Visual Studio 2022 and the C# (ver. 11) programming language on a Microsoft Windows 11 PC, a graphical user interface (GUI) application (based on the Windows Forms UI framework) was deployed, which as shown in Fig. 3.11:

- detects the available serial ports, provides the option to select one of them and configure its baud rate.
- logs the serial port's incoming and outcoming data and saves them to a text file.
- plots on real time four graphs, for the PV array's power, voltage and current and for the DC/DC converter's duty cycle (and provides the option to save them to a CSV file).
- plots on real time the PV array's P-V and I-V curves (and provides the option to save them to a CSV file).
- offers the ability to send commands to the controller.

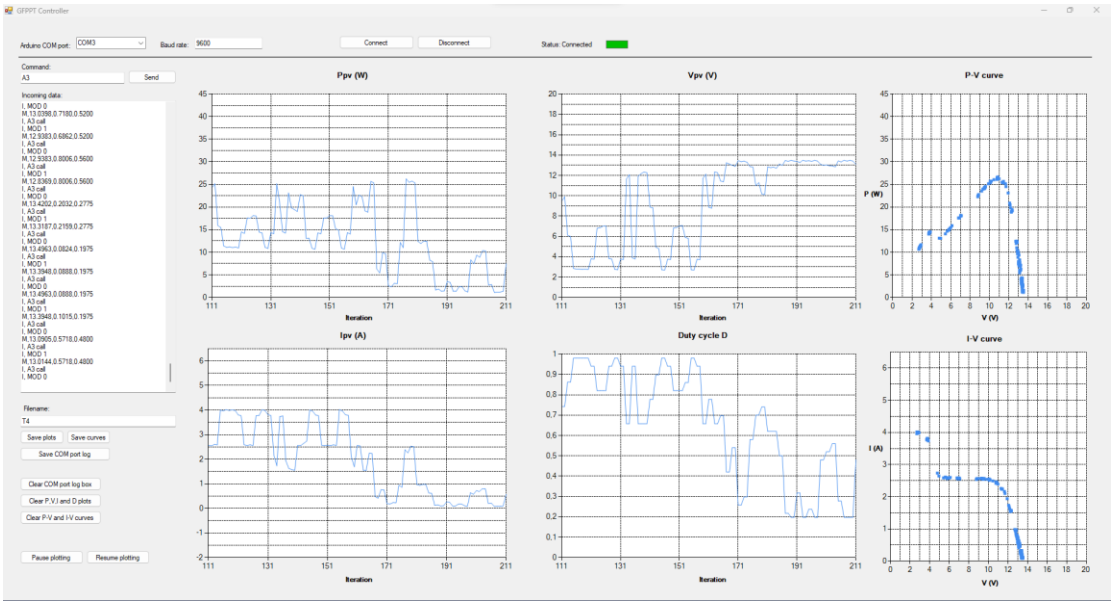

Figure 3.11: Screenshot of the developed UI during the controller's operation.

3.4.5. Communication protocol

In order to communicate with the controller, not only to receive measurements but also to send commands to it, the protocol presented in Table 3.6 was developed.

Table 3.6: Commands and their responses.

Command from PC	**Response from MCU**
AT<CR><LF>	OK_AT<CR><LF>
M<CR><LF>	M,<V_{pv}>,<I_{pv}>,<D><CR><LF>
D<VALUE><CR><LF>	OK_D_<VALUE><CR><LF>
V<VALUE><CR><LF>	OK_V_<VALUE><CR><LF>
C<VALUE><CR><LF>	OK_C<VALUE><CR><LF>
T<CR><LF>	OK_T<CR><LF>
A<VALUE><CR><LF>	OK_A<VALUE><CR><LF>
P<CR><LF>	OK_P<CR><LF>
B<CR><LF>	OK_B<CR><LF>
K<CR><LF>	OK_K<CR><LF>
R<VALUE><CR><LF>	OK_R_<VALUE><CR><LF>
S<CR><LF>	OK_S<CR><LF>

To identify the commands and their arguments, since the whole command set is based on ASCII characters, a simple state machine was implemented.

Detailed description of each command

AT The controller simply responds and performs no other action. This command is used to test the connection. It is also used by the user interface in order to check if the controller is operating properly, to update the "Connection Status" indicator.

M Although the controller sends measurements data each 1 second to the computer automatically, the user may acquire a set of measurements at a specific time using this command. The response contains the values of $V_{pv,}$ I_{pv} and D in ASCII format with 4 decimal digits each, for instance "M,12.5858,1.2222,0.5050<CR><LF>".

D Used to enter the forced duty cycle mode. The computer imposes a specific value for the duty cycle and stops whichever algorithm was running on the controller. The value is given as 2 ASCII characters declaring a value between 00 and 99.

V Used to enter the forced voltage mode. The computer imposes a specific value for the PV array's voltage and stops whichever algorithm was running on the controller. The value is given as 2 ASCII characters declaring a value between 00 and 99.

C Used to clear the Q and N matrix files for the algorithms 2 and 3. The value can take the value of either 2 or 3, to select whose algorithms files will be cleared. If the files do not exist, they are created.

T Traces the P-V and I-V curves of the PV array. This is achieved by setting the duty cycle to 5% and increasing it up to 98%, using a step of 1%. On each step, a measurement packet is sent to the computer, to visualize the curves.

A Executes one of the three algorithms. The value can take one of the values 1, 2 or 3.

P Executes a conventional MPPT Perturb and Observe (P&O) algorithm.

B Breaks whichever algorithm was being executed and jump to the idle mode.

K Prints information about the SD card.

R Used to declare a value for the reference power. The value is given as 2 ASCII characters declaring a value between 00 and 99.

S Used to pause the execution of whichever algorithm was being executed. Functions as a toggle, thus it can be used to both pause and resume the execution.

It is noted that the symbols <CR> and <LF> refer to the ASCII characters for Carriage Return (0x0D) and Line Feed (0x0A) respectively.

3.5. The PV array voltage sensor

Since the Arduino's analog input pins can handle voltages between 0 and 5 V, and the PV array's voltage can vary in the range from 0 to 18 V (based on the PV panels' open-circuit voltage on Standard Test Conditions), a circuit to scale the PV array's voltage was designed, using a voltage divider. Since the obtained signal needs filtering, and to ensure that there is isolation between the PV panels and the microcontroller input pins, a voltage follower was added as part of the circuit. The whole circuit is presented in Fig. 3.12.

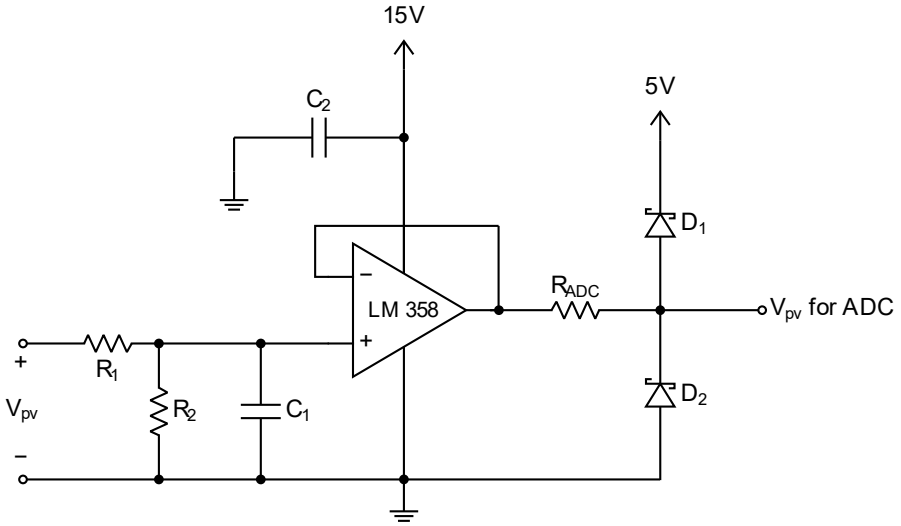

Figure 3.12: The PV array voltage sensor's circuit.

The sensor was designed to measure voltages in the range from 0 to 26 V. When the input voltage reaches its maximum value (26 V), the output signal should be at the level of 5 V, the maximum allowed for the Arduino analog input pins. Thus, the following equation is formed:

$$V_{out,max} = \frac{R_2}{R_1 + R_2} \cdot V_{in,max} \Rightarrow \frac{R_2}{R_1 + R_2} = \frac{V_{out,max}}{V_{in,max}} \Rightarrow \frac{R_1}{R_2} = \frac{21}{5} \qquad (3.13)$$

Based on the result, they were chosen to be R_1 = 21kΩ and R_2 = 5kΩ.

Table 3.7: The components used for the PV array voltage sensor's circuit.

Item	Description
$\boldsymbol{R_1}$	21 kΩ resistor (10 kΩ + 10 kΩ + 1 kΩ)
$\boldsymbol{R_2}$	5 kΩ resistor (10 kΩ ‖ 10 kΩ)
$\boldsymbol{R_{ADC}}$	4.7 kΩ resistor

C_1 C_2	0.1 μF polyester capacitor
D_1 D_2	BAT48RL small signal Schottky diodes

Based on the operational amplifier's datasheet [17], its input voltage should not exceed $V_{cc} - 2\,V$, where V_{cc} is its supplying voltage. Since the voltage at the output of the voltage divider ranges between 0 and 5 V, the amplifier should be supplied with at least 7 V, thus it had been connected to the 15 V supply. The capacitor C_1 was placed in the input for noise filtering purposes, while C_2 decouples the amplifier's supply pin. The resistor R_{ADC} is added to reduce any current flow to the pins of the ADC, in case of fault, and the two diodes clamp the output voltage between 0 and 5.25 V. They were chosen to be Schottky diodes on purpose, because of their low voltage drop. If typical silicon diodes were used, the circuit's output would range between 0 and 5.7 V, which may cause damage to the ADC. The implementation of the voltage sensor's circuit is depicted in Fig. 3.13.

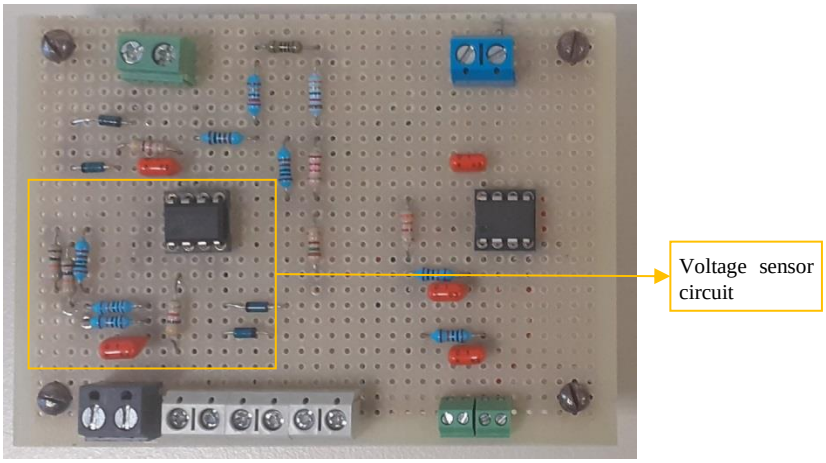

Figure 3.13: The voltage sensor's circuit experimental prototype.

Calibrating the voltage sensor

When the output of the voltage sensor's circuit is connected to the microcontroller's pin, the result of the analog conversion (the return value of the Arduino's analogRead function) will be a number between 0 and 1023, which is directly mapped to the input voltage range of the ADC (0 to 5 V). In order to obtain the PV array's voltage value, the sensor should be calibrated. This process is done by connecting a variable voltage source (a bench power supply) to the input terminals of the voltage sensor's circuit and logging pairs of two values, the input voltage and the ADC's output (as the ones plotted with black dots in Fig. 3.14). After that, using the

MATLAB's polyfit function, a linear equation between those two had been formed, to map the ADC's value to a specific PV voltage value:

$$V_{sensor} = 0.025362818210810 \cdot (\text{ADC value}) + 0.053996513434496 \quad (3.14)$$

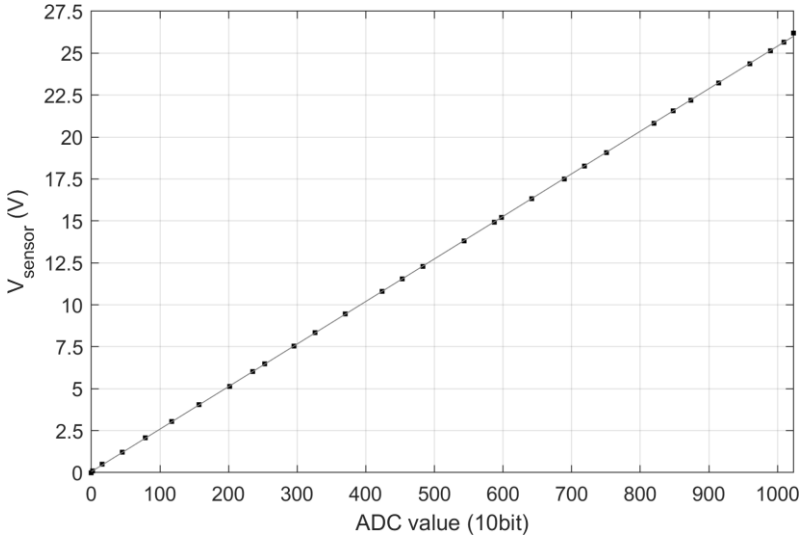

Figure 3.14: Calibration data for the PV array voltage sensor.

3.6. The PV array current sensor

The Allegro ACS712 (ACS712ELCTR-20A-T) Hall-effect based current sensor can measure currents ranging from -20 A to 20 A, and produces an output in the range from 0 to 5 V, offering a sensitivity of 100 mV/A. The sensor requires a single supply of V_{cc} = 5 V and when no current is flowing through its internal path, it produces an output of $V_{cc}/2$ = 2.5 V. Consequently, in this thesis' experiments, where the measured current ranged between 0 A and 6.13 A (the PV array's short-circuit current at STC), the sensor's output voltage would range between 2.5 V and 3.113 V. If this output had been directly connected to the Arduino's ADC, only a part of its input range (0 … 5 V) would be utilized. Since the Arduino features a 10-bit ADC, only values between 512 and 638 would be used and that would lead to a sensitivity of 49 mA per ADC step (which corresponds to 5 mV). Given that the ADC has an accuracy of ±1 digit, and that Hall sensors are easily affected by noise, this would downgrade the measurements' quality.

In order to overcome this issue, a circuit to scale the sensor's output to the ADC's full input range was designed. This circuit produces a 0 V output when there is no current flow and 5 V when 6.5 A of current are flowing through it (Table 3.8). A value slightly greater than the nominal PV array's short-circuit current was chosen, since the latter was measured by the manufacturer on STC conditions, and the solar irradiance can be higher than 1000 W/m^2 in real operating conditions.

Table 3.8: The PV array current sensor's output and its processed value, for the minimum and maximum current flow.

PV array current I_{pv}	**Sensor output** V_{sensor}	**Processed output** $V_{processed}$
0 A	2.5 V	0 V
6.5 A	3.15 V	5 V

An operational amplifier was used, and the circuit was designed based on the equation:

$$V_{processed} = a \cdot V_{sensor} - b \tag{3.15}$$

where a and b are positive constants, whose values will be calculated by forming the following system of equations:

$$\begin{cases} 0 = a \cdot 2.5 - b \\ 5 = a \cdot 3.15 - b \end{cases} \tag{3.16}$$

Solving this system results to:

$$a = 7.7 \text{ and } b = 19.25$$

The circuit which implements the desired equation is the following:

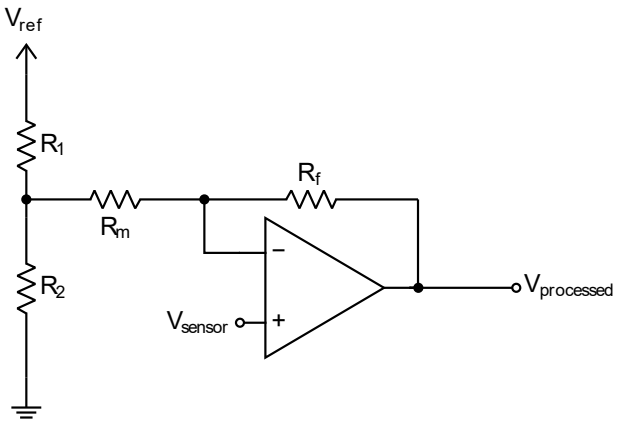

Figure 3.15: Initial form of the current sensor's signal processing circuit.

Being a linear circuit, it can be analyzed using the superposition rule. By setting $V_{sensor} = 0$, the circuit becomes an inverting amplifier with the following output voltage:

$$V^{-}_{processed} = -\left(\frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot V_{ref} \tag{3.17}$$

Also, by setting $V_{ref} = 0$, it becomes a non-inverting amplifier whose output voltage equals:

$$V^{+}_{processed} = \left(1 + \frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot V_{sensor} \tag{3.18}$$

Combining (3.17) and (3.18) results to the final transfer function of the circuit:

$$V_{processed} = V^{+}_{processed} + V^{-}_{processed}$$
$$= \left(1 + \frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot V_{sensor} - \left(\frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot V_{ref}$$
$$= \left(\frac{R_f + R_m + R_1 \parallel R_2}{R_m + R_1 \parallel R_2}\right) \cdot V_{sensor} - \left(\frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot V_{ref} \quad (3.19)$$

The values of a and b can be extracted by comparing (3.15) and (3.19):

$$a = \left(\frac{R_f + R_m + R_1 \parallel R_2}{R_m + R_1 \parallel R_2}\right) \quad (3.20)$$

$$b = \left(\frac{R_f}{R_m + R_1 \parallel R_2}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot V_{ref} \quad (3.21)$$

Given that a = 7.7, assuming that $R_1 \parallel R_2 \gg R_m$ gives:

$$a = \frac{R_f + R_m}{R_m} \Rightarrow R_f = 6.7R_m \quad (3.22)$$

Selecting $R_m = 33\ k\Omega$ gives $R_f = 221.1\ k\Omega$.

The reference voltage V_{ref} is chosen to be 5 V. Using (3.21), the following can be extracted for the values of R_1 an R_2.

$$b = \left(\frac{R_f}{R_m}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot V_{ref} \Rightarrow 19.25 = \left(\frac{221.1}{33}\right) \cdot \left(\frac{R_2}{R_1 + R_2}\right) \cdot 5 \Rightarrow R_2 = 1.35R_1 \quad (3.23)$$

Selecting $R_1 = 1\ k\Omega$ gives $R_2 = 1.35\ k\Omega$.

It is worth noting that assuming $R_1 \parallel R_2 \gg R_m$ results to $a = 7.585$ (1.5% divergence from the desired 7.7) and $b = 18.915$ (1.7% divergence from the desired 19.25), thus introduces an error to the final transfer function. However, since the resistors are not ideal and show some tolerance (the ones used had either 1% or 5% tolerance), the error will be nearly insignificant finally, because during the sensor's calibration process, any offset will be absorbed by the transfer function.

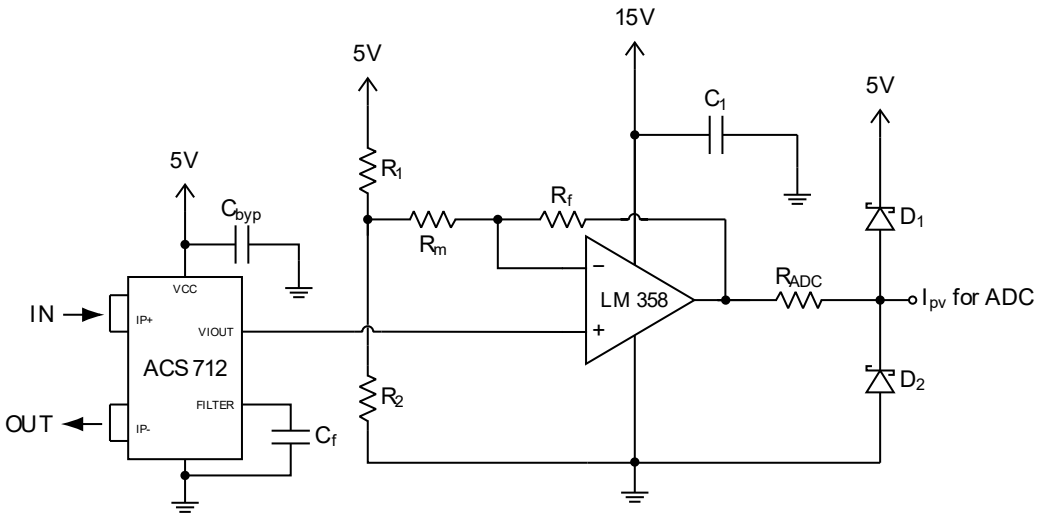

Figure 3.16: The current sensor's circuit.

Table 3.9: The components used for the current sensor's circuit.

Item	Description
$\boldsymbol{R_1}$	1 kΩ resistor
$\boldsymbol{R_2}$	1.35 kΩ resistor (1.2 kΩ + 150 Ω)
$\boldsymbol{R_m}$	33 kΩ resistor
$\boldsymbol{R_f}$	221.1 kΩ resistor (220 kΩ + 1 kΩ + 100 Ω)
$\boldsymbol{R_{ADC}}$	4.7 kΩ resistor
$\boldsymbol{C_1}$ $\boldsymbol{C_{byp}}$ $\boldsymbol{C_f}$	0.1 μF polyester capacitor
$\boldsymbol{D_1}$ $\boldsymbol{D_2}$	BAT48RL small signal Schottky diodes

Following the same procedure as for the voltage sensor's circuit, in order to protect the ADC, the output is clamped between 0 and 5.25 V by using two Schottky diodes and the protective resistor R_{ADC} . Since the LM358 chip contains two operational amplifiers, the decoupling capacitor C_1 refers to the same physical component as the decoupling capacitor used for the voltage sensor. The capacitor C_{byp} decouples the current sensor's supply pin, while the filtering capacitor C_f was sized according to the response time data provided on the current sensor's datasheet [18]. Choosing a capacitance of 0.1 μF provided a satisfying noise filtering and a rise time of the output signal of approximately 0.3 msec, which satisfies the requirements of the developed system.

The sensor is produced in a surface mount SOIC8 (Small Outline Integrated Chip) package, thus it had to be soldered on a breakout board. The Adafruit 1212 SMT (Surface Mount Technology) SOIC breakout board was used (Fig. 3.17), combined with the necessary pin headers, to be placed on a breadboard. However,

during the calibration of the current sensor, a current flow of 6.2 A for 1 minute resulted to overheating. In particular, before starting the tests, the board's temperature was 22°C, while one minute later it reached 86°C and the breadboard started melting.

On the sensors datasheet [18], it is referred that the conductive path inside the SOIC chip shows a typical resistance of 1.2 mΩ, which means that either the chip was defective or another part of the circuit introduced a non-negligible resistance to the current's path. Since the sensor was operating normally for lower values of current for larger time periods, it was considered that the breakout board introduced the unwanted resistance. To overcome this issue, two wires have been soldered directly onto the chip's pins (Fig. 3.18) and after that the sensor's temperature barely increased.

Figure 3.17: The Adafruit 1212 SMT SOIC breakout board.

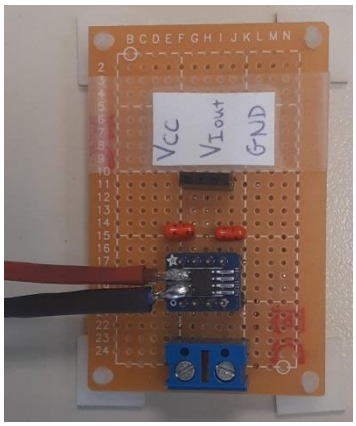

Figure 3.18: The final form of the PV array current sensor.

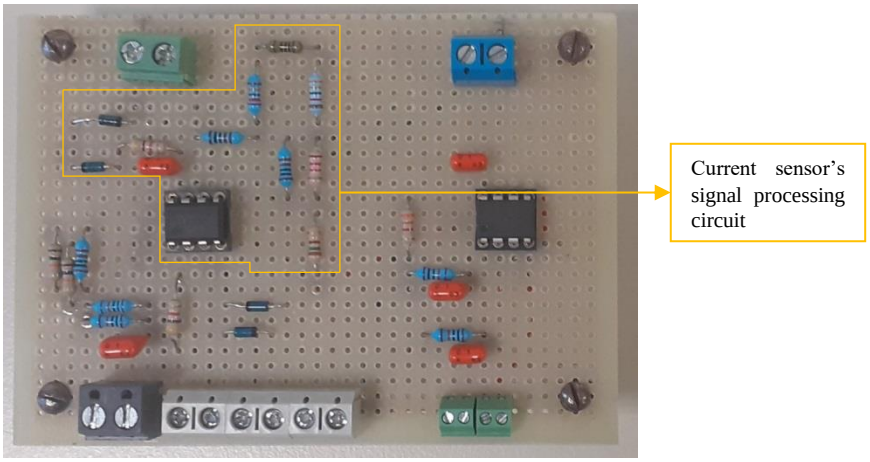

Figure 3.19: The PV array current sensor's circuit experimental prototype.

Calibrating the current sensor

Following the same process as for the voltage sensor, the bench supply was short-circuited, with the cable passing through the sensor and its voltage setting knob was set zero. After that, using the current limit knob, the ADC's output was related to the current flow by the equation:

$$I_{sensor} = 0.006355954103740 \cdot (\text{ADC value}) - 0.114634035038596 \quad (3.24)$$

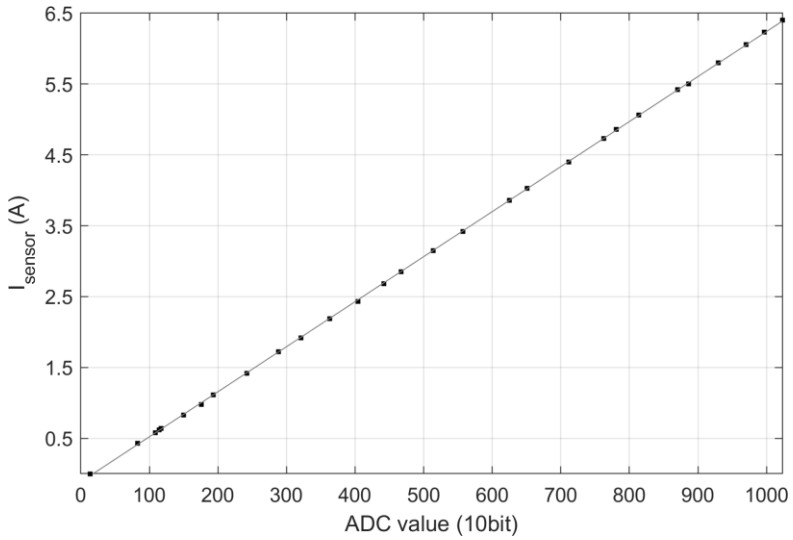

Figure 3.20: Calibration data for the PV array current sensor.

3.7. The power supplies

In order to supply the various subcircuits with the required voltage levels (+5 V and +15 V), two power supplies were designed (Fig. 3.21), using the LM317 adjustable voltage regulator.

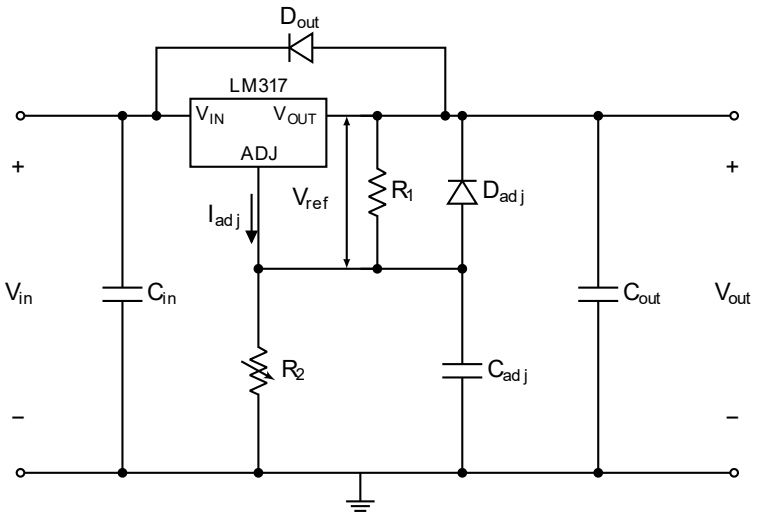

Figure 3.21: The power supplies' circuit.

The LM317 device contains an internal voltage reference of 1.25V, which is used to adjust the output voltage, using two external resistors, noted R_1 and R_2 on the circuit, according to the following equation:

$$V_{out} = V_{ref} \cdot \left(1 + \frac{R_2}{R_1}\right) + I_{adj} \cdot R_2 \tag{3.25}$$

The current I_{adj} is in the range of μA (typically around 50 μA) [19], thus considered as negligible, simplifying the above equation to:

$$V_{out} = V_{ref} \cdot \left(1 + \frac{R_2}{R_1}\right) \tag{3.26}$$

The input bypass capacitor C_{in} is placed to filter the noise from the regulator's input. The adjustment terminal is also bypassed to the ground via the capacitor C_{adj} in order to prevent the amplification of any voltage ripple. A third capacitor, C_{out}, was added in order to reduce the output ripple.

After adding those capacitors, it should be ensured that they will not discharge into the regulator and damage it. In order to avoid this from happening, two protection diodes, D_{adj} and D_{out} were added. If the input is shorted to the ground, C_{out} would discharge into the regulator's output. In this case, D_{out} provides to C_{out} a discharge path to the (grounded) input. Additionally, when either the input or the output is shorted, the capacitor C_{adj} will discharge into the regulator. D_{adj} ensures that C_{adj} is going to discharge either into the grounded input or the grounded output respectively.

Table 3.10: The components used for the power supplies.

Item	Description
R_1	240 Ω resistor
R_2	5V regulator: 1 kΩ trimmer (adjusted to ≈720 Ω) 15V regulator: 5 kΩ trimmer (adjusted to ≈2640 Ω)
C_{in}	0.1 μF polyester capacitor
C_{adj}	10 μF electrolytic capacitor
C_{out}	1 μF polyester capacitor
D_{adj} D_{out}	1N4002 general purpose diodes

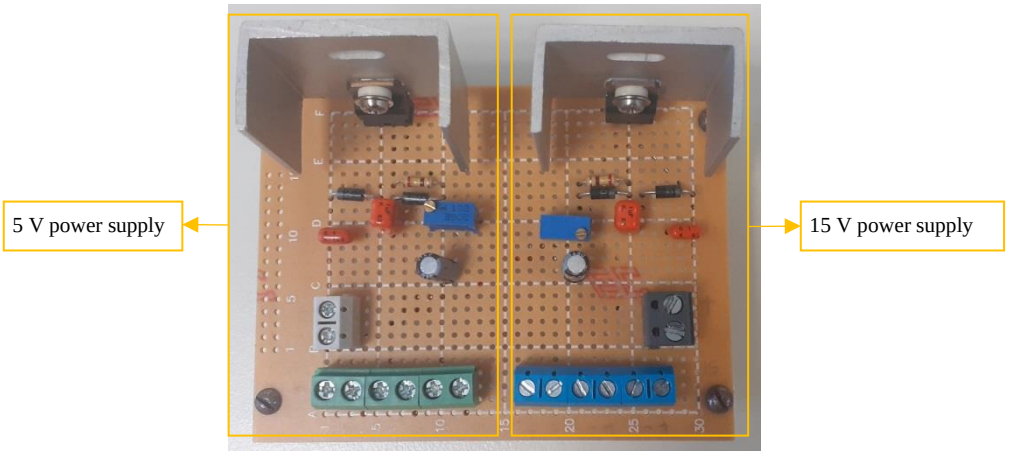

Figure 3.22: The power supplies' board.

3.8. The environmental measurement devices

In order to measure the irradiance on each PV panel, the pyranometer Kimo SL100 [20] was used. The measurement was taken before each experiment, by placing the probe (which contains a solar cell) on each PV panel. To measure the environment's temperature, the air quality meter Kimo AQ110S [21] was used, which features a thermistor on its probe to calculate the air temperature.

Figure 3.23: The Kimo SL100 pyranometer.

Figure 3.24: The Kimo AQ110S air quality meter.

4. Flexible Power Point Tracking Algorithms

4.1. The necessity of controlling the output power of PV arrays

On each electrical power grid, frequency is directly related to the balance between the active power that is generated by the power plants and the active power that is demanded by the various loads [2]. When the demanded power is higher than the generated one, the frequency drops proportionally to their difference. Respectively, when the generated power is higher than the demanded power, the frequency is going to rise. Each connection or disconnection of a load or a generator, results to a frequency disturbance. In particular, during an increase of the demanded power, the frequency will inevitably drop until the generated power matches the demanded. On the other hand, if a few loads switch off, then the frequency is going to rise, until the generated power reduces enough to match the new demanded power.

Frequency stability is an indicator of the power quality for an electrical grid. On each country's power system, strict limitations are defined for those two quantities' fluctuation from their nominal values. For instance, in Greece, the Independent Power Transmission Operator (IPTO) states that the Hellenic Electricity Transmission Systems (HETS) grid's nominal frequency is 50 Hz and under normal operation conditions, it may fluctuate between 49.85 Hz and 50.15 Hz [22].

If the frequency drops under a defined threshold value (typically 47 Hz for 50 Hz systems [2]), any power plant should be disconnected from the grid, since it is designed to operate under a specific frequency range. Similarly, there is also an upper limit for the grid frequency on which a power plant can operate normally. By each plant's disconnection, the total available power on the grid decreases, and the frequency drops even more. Consequently, the power plants will start disconnecting themselves from the grid one after another and this will lead to a complete collapse of the grid, a situation known as blackout.

In a grid with high PV penetration, if all the PV power plants' controllers are designed only for MPPT operation, then in case the grid's frequency is out of bounds, one way to stabilize it is by the PV plant's connection or disconnection. Another way would be to keep the PV plant's on MPPT mode and control other types of power plants, such as coal power plants. However, coal power plants (and in general power plants that are based on electromechanical systems, like synchronous generators) have a high response time to the control command, compared to the instantaneous response of semiconductor-based systems, such as the PV plants.

However, in the same grid, if even a few of the PV power plant's controllers were designed for FPPT operation, then the grid operator would be able to regulate its frequency by sending an updated reference power value to each one of them. In that way, on each load change, the power demand is going to be satisfied by the power generation instantaneously and the frequency will recover to its normal level.

This method to control the power output of a PV power plant is called Constant Power Generation Control (CPGC) [4]. As its name implies, a specific power

reference value is extracted from the plant, except if this value is greater than the plant's GMPP, therefore the maximum available energy will be extracted, as shown in Fig. 4.1. This method is also referred as "absolute active power control" or "power limiting control" in the literature [3].

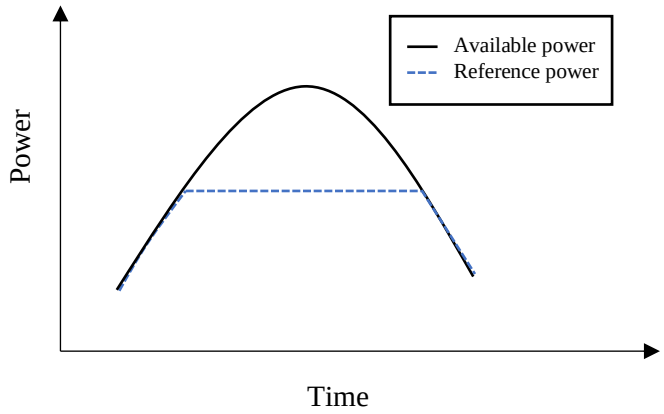

Figure 4.1: Constant Power Generation Control over a long time period.

Under uniform irradiation conditions, as described in Section 2.2, the P-V curve of the PV array will have one MPP, as shown in Fig. 4.2. Each reference power value P_{ref} lower than the MPP power P_{mpp}, can be obtained in two operation points, called Flexible Power Points (FPP). Those two FPPs are placed on both sides of the MPP. An FPP algorithm designed to operate under such conditions is going to select one of them to track the reference power. This procedure is called Flexible Power Point Tracking (FPPT).

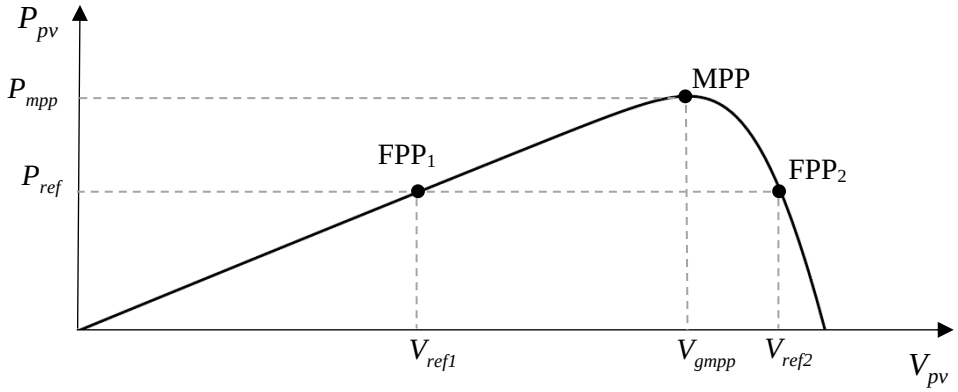

Figure 4.2: Principles of Flexible Power Point Tracking.

On the other hand, under partial shading conditions, the PV array's P-V curve is going to have more than one Local MPPs (LMPP), one of which being the Global MPP (GMPP). For simplicity, an example curve with two LMPPs is presented in Fig.

4.3, but the same principle holds for curves with any number of LMPPs. Each power reference value P_{ref} lower than the GMPP power P_{gmpp}, can be obtained in two or more operation points, noted as FPPs. Assuming a power reference value lower than the LMPPs power, there will be four FPPs, as shown in the figure. An FPP algorithm designed to operate under partial shading conditions is going to select one of them to track the reference power. This procedure is called Global Flexible Power Point Tracking (GFPPT).

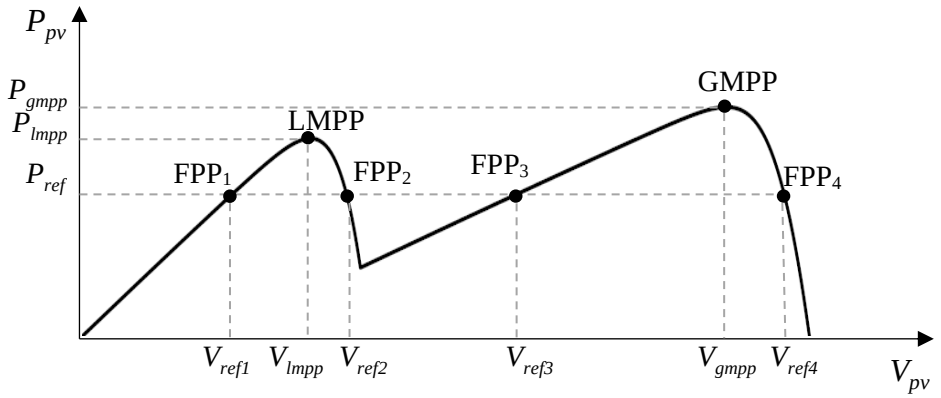

Figure 4.3: Principles of Global Flexible Power Point Tracking.

Numerous algorithms have been presented in the literature for the FPPT, where the P-V curve has only one MPP. However, only one is reported for the GFPPT problem for CPGC, where the reference power can be found on multiple operation points on the P-V curve. This is the algorithm presented in [4], which will be referred as algorithm #1 for the rest of this document. Its performance will be compared with the two new algorithms developed within the framework of this thesis.

There are also other techniques to control the output power of a PV power plant, such as the Power Reserve Control (PRC) [3], which extracts a percentage of the available power from the plant and keeps a defined amount of power as reserve, to support the grid in case of sudden load changes, providing inertia to it. In this type of control, the reference power is expressed as a percentage of the GMPP power. Consequently, to implement this type of algorithms, the GMPP should be tracked firstly, which can slow down the system's dynamic response under rapidly changing environmental conditions [23]-[25].

4.2. Algorithm #1 – GFPPT using a search-skip-judge GMPPT algorithm

The first algorithm is presented in [4] and its flowchart is presented in Fig. 4.4. It utilizes a SSJ (Search-Skip-Judge) GMPPT algorithm [26] which reduces its convergence time by skipping some parts of the PV array's P-V curve on runtime.

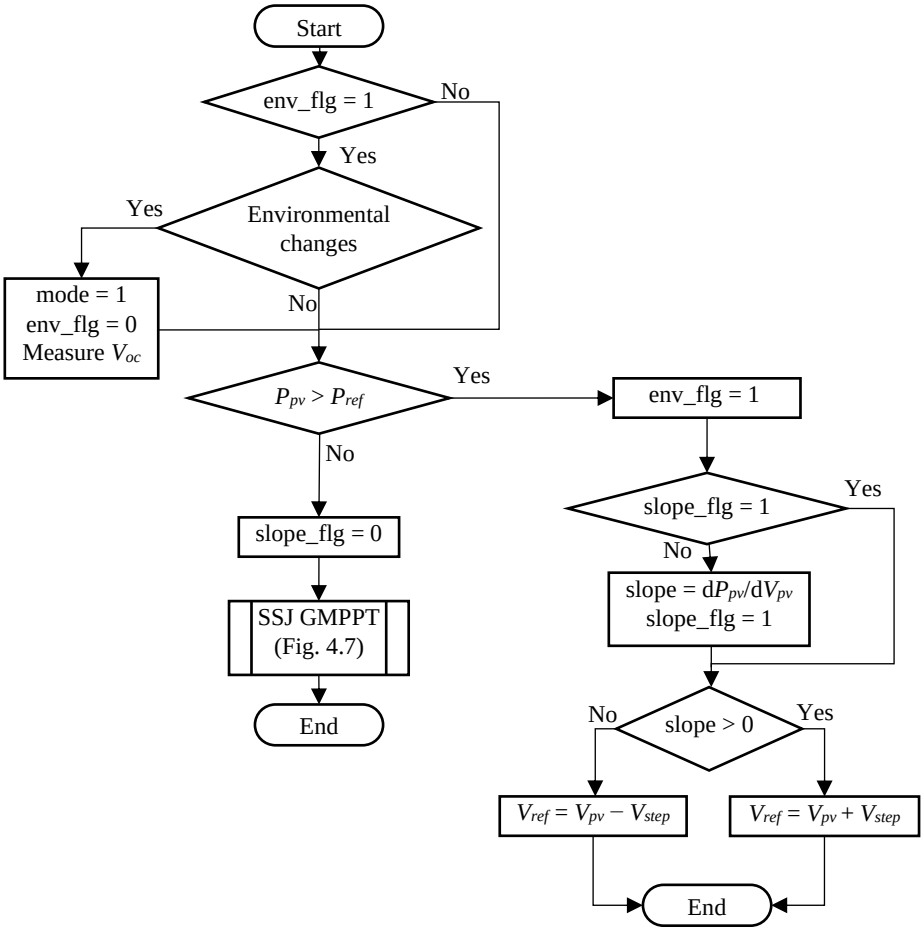

Figure 4.4: The first GFPPT algorithm's flowchart.

Assuming that there are not environmental changes, if the PV array's power is greater than the reference power, it is reduced based on a fine-tuning method and the algorithm operates under GFPPT mode. The slope (derivative) of the P-V curve is calculated. In case it is positive (region 1 in Fig. 4.5), the PV array's voltage should be increased in order to limit the output power. On the other hand, if it is negative (region 2 in Fig. 4.5), the PV array's voltage should be reduced to track the reference power. In both cases, the same voltage step V_{step} is used.

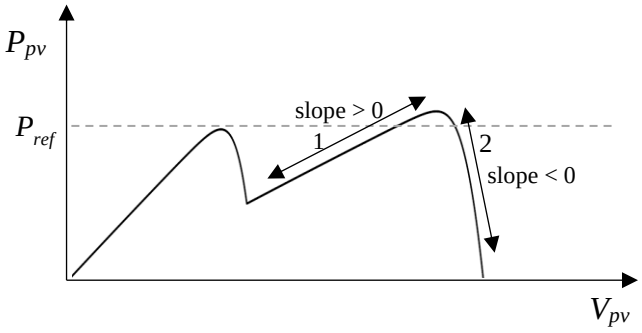

Figure 4.5: The slope on different regions of the P-V curve.

The flag's "slope_flg" usefulness is shown in Fig. 4.6. The PV system was operating at point 3 and the reference power reduced, thus the operation point should move to 4. The algorithm is going to reduce the PV array's voltage, based on the previous description. However, during this transition on the P-V curve, there it will find an SPD (Section Divider Point), around which the slope changes sign. Consequently, the operation point is going to oscillate indefinitely around the SDP and it will never reach the desired point 4. To overcome this issue, the slope's value is calculated once at the start of the transition (one time step before moving the operating point from 3). In that way, though the transition from 3 to 4, the algorithm has the information that the slope is positive and it will reduce the PV array's voltage until successfully tracking the new reference power.

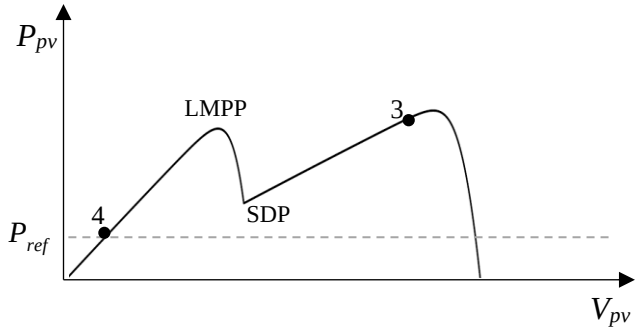

Figure 4.6: The usefulness of the "slope_flg" flag.

If the reference power is higher than the PV array's output voltage, the algorithm is going to operate under GMPPT mode. The SSJ-GMPPT algorithm's flowchart is presented in Fig. 4.7. Each time the algorithm enters the GMPPT mode, the "slope_flg" is set to 0. Three example cases for the GMPPT mode will be analyzed in detail, to explain its principles of operation.

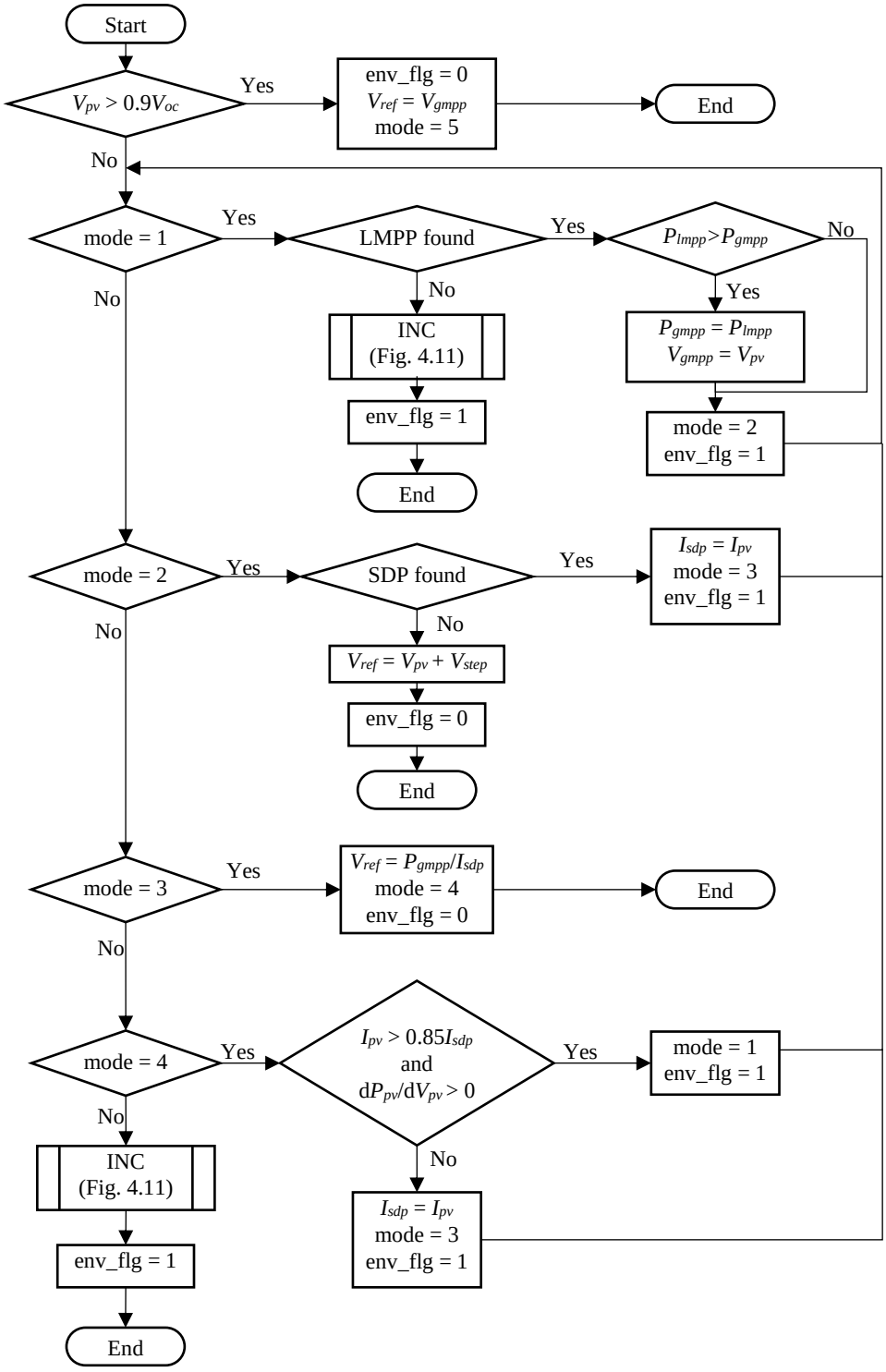

Figure 4.7: The SSJ-GMPPT algorithm's flowchart.

Case 1: The corresponding LMPP can satisfy the reference power

As shown in Fig. 4.8, the system was operating at point 5, and given the new value of the reference power, the operation point should move to 6. After setting "slope_flg" to zero, the GMPPT algorithm is called with mode = 1. Being on mode 1,

the GMPPT algorithm is going to call the Incremental-Conductance algorithm to find the LMPP. Before reaching the LMPP, the PV array's output power will exceed the reference power at point 6 and the operation point will be set to point 6.

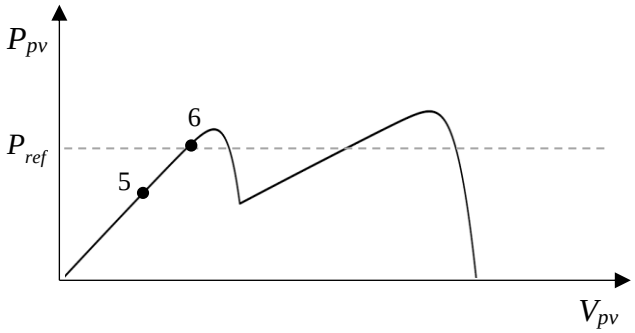

Figure 4.8: The PV array's P-V curve for case 1.

Case 2: The corresponding LMPP cannot satisfy the reference power

As shown in Fig. 4.9, the system was operating at point 5, and given the new value of the reference power, the operation point should move to 7. After setting "slope_flg" to zero, the GMPPT algorithm is called with mode = 1. Being on mode 1, it will search for the first LMPP, noted LMPP1 on the figure. After reaching it, since it cannot satisfy the power reference, it will continue searching. Moving to mode 2, it searches for the SDP. By transiting to mode 3, a part of the P-V curve is skipped and the reference voltage is set to $V_{ref} = P_{LMPP1}/I_{SDP}$, as the curved arrow shows. After checking the operating point in mode 4, it reverts to mode 1 and searches for the next LMPP, which happens in the example to be the GMPP. Before reaching it, the power reference is met and the operating point will be set to point 7.

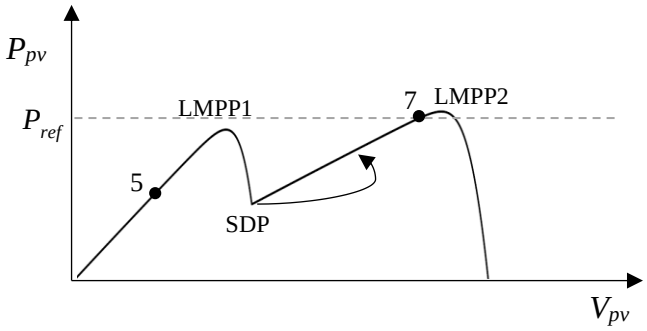

Figure 4.9: The PV array's P-V curve for case 2.

Case 3: The GMPP cannot satisfy the reference power

As shown in Fig. 4.10, the system was operating at point 5, and given the new value of the reference power, the operation point should move to 9, since it cannot

produce more power than the GMPP. After setting "slope_flg" to zero, the GMPPT algorithm is called with mode = 1. Being on mode 1, it will start searching for the first LMPP, noted LMPP1 on the figure. After reaching it, moving to mode 2, it searches for the SDP. By transiting to mode 3, a part of the P-V curve is skipped and the reference voltage is set to $V_{ref} = P_{LMPP1}/I_{SDP}$, as the black arrow shows. After checking the operating point in mode 4, it reverts to mode 1 and searches for the next LMPP (i.e., LMPP2), which in this example is the GMPP. After reaching the GMPP, at point 9, since the power reference is not satisfied, the mode is set to 2 and its starts searching for the next SDP. When the PV array's voltage reaches 90% of its open-circuit value, it is considered that no LMPPs exist in the last 10% region [26]. Thus, the mode is set to 5, and the incremental-conductance algorithm is called to oscillate the operating point around the GMPP, i.e., point 9.

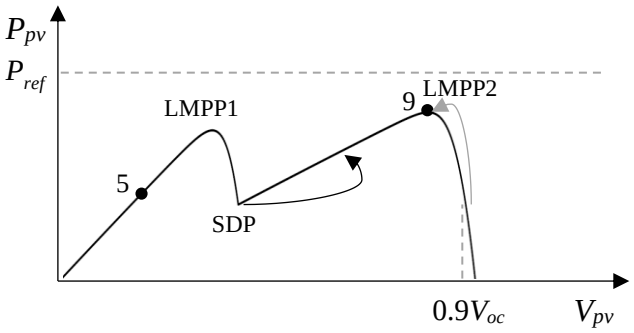

Figure 4.10: The PV array's P-V curve for case 3.

The flowchart of the incremental-conductance (INC) MPPT algorithm [27] which is used during the GMPP tracking process in modes 1 and 5, is presented in Fig. 4.11. At the MPP of the PV array, based on the I-V curve, the following equation applies:

$$\frac{\partial P_{pv}}{\partial V_{pv}} = 0 \Rightarrow \frac{\partial\left(V_{pv} \cdot I_{pv}\right)}{\partial V_{pv}} = 0 \Rightarrow I_{pv} + V_{pv} \cdot \frac{\partial I_{pv}}{\partial V_{pv}} = 0 \Rightarrow \frac{\partial I_{pv}}{\partial V_{pv}} = -\frac{I_{pv}}{V_{pv}} \qquad (4.2)$$

Based on the shape of the I-V curve, it can be observed that

- if $\frac{\partial I_{pv}}{\partial V_{pv}} < -\frac{I_{pv}}{V_{pv}}$, the system is operating at the right side of the MPP, thus the PV array's voltage should be reduced.
- if $\frac{\partial I_{pv}}{\partial V_{pv}} > -\frac{I_{pv}}{V_{pv}}$, the system is operating at the left side of the MPP, thus the PV array's voltage should be increased.
- if $\frac{\partial I_{pv}}{\partial V_{pv}} = -\frac{I_{pv}}{V_{pv}}$, the system is operating at the MPP, and the PV array's voltage is not modified.

It is noted that in order to calculate the aforementioned derivative, the change in the PV array's voltage and current between two consecutive calls of the algorithm is used, as it is shown in the flowchart.

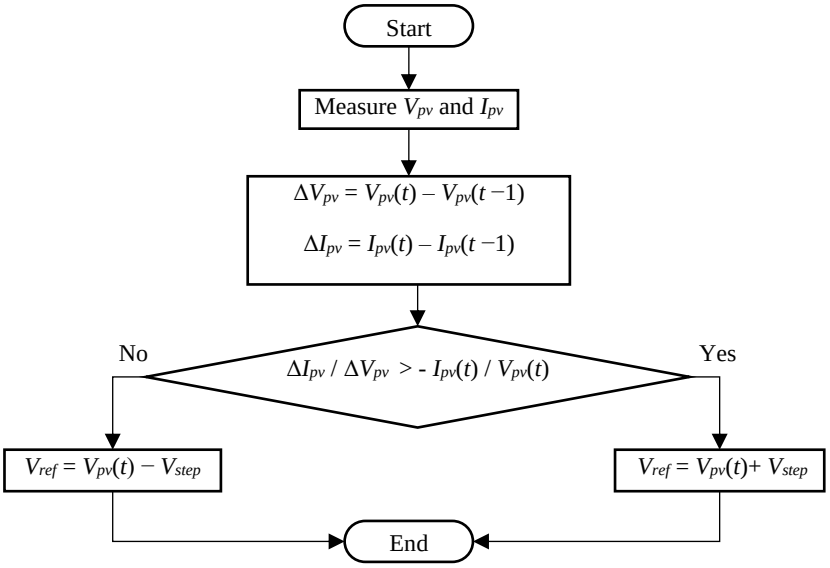

Figure 4.11: The incremental-conductance (INC) algorithm's flowchart.

To detect any environmental changes, such as irradiance or temperature changes, the following rule is applied:

$$\Delta p = \left| \frac{P_{pv} - P_{pv,old}}{P_{pv,old}} \right| > 0.15 \rightarrow \text{environmental changes} \qquad (4.1)$$

If the PV array's output power changes more than 15% between two successive algorithm calls, then environmental changes are considered to occur. This means that the P-V curve of the PV array has changed and the algorithm needs to reset, to track efficiently the reference power. Using the "env_flg" flag, the check for environmental changes is done only when the algorithm operates under GFPPT mode or when the GMPPT mode is not doing any rapid changes in the PV array's voltage. For instance, when the system is operating under GMPPT's mode 3, it should not check for environmental changes on the next algorithm call, since power change may be more than 15%, but not because of any irradiance changes.

To improve the algorithm's dynamic response, on each detection of environmental changes, the reference voltage is not immediately set to its starting value (which is a low value, used to scan the P-V curve). Instead of that, it keeps its previous value and the mode is set to 1, in order to detect the corresponding LMPP. If its power can satisfy the reference power, then the algorithm tracks successfully the GFPP in a few steps. If not, then the reference voltage will be set to its minimum value, since scanning the curve from its beginning is inevitable in order to track the GFPP.

It is worth noting that since the algorithm uses the value of the PV array's open-circuit voltage, after each environmental change it should be measured. This is the main drawback of this GFPPT algorithm, since it produces a large oscillation at the system for a few time-steps.

4.3. Algorithm #2 – GFPPT based on Q-learning GMPPT

The second GFPPT algorithm utilizes a GMPPT algorithm based on a machine learning technique called Q-learning [5]. After converging to the GMPP, it tunes the output power to the desired value. At first, the GMPPT algorithm will be presented.

4.3.1. Principles of Q-learning

Q-learning is a model-free algorithm based on reinforcement learning [28]. Being a model-free algorithm, it can solve a problem without using the probability distribution which is associated with the Markov decision process that models the problem. Consequently, it is considered an algorithm which performs a trial-and-error process to achieve its goal.

In reinforcement learning, the agent is exploring a discrete world with a finite number of states. On each step, it selects one action from a set of finite actions thus can be described as the controller of a Markov decision process (MDP), which is a discrete-time stochastic control process. The MDP models the interaction between the agent and the environment and consists of:

- the set of states, $\boldsymbol{S}$,
- the set of the possible actions, $\boldsymbol{A}$, and
- the reinforcement reward function, which represents the reward when selecting action α while being in state S, leading to a transition to state S'.

MDPs are usually used on optimization problems, such as the GMPPT problem. To solve this problem, a temporal difference Q-learning algorithm is used, whose goal is to find an action selection policy, that is going to maximize the total discounted rewards it will receive in the future. Discounted is called the reward which will be received by the agent after k time steps. It is used to keep a balance between immediate and future rewards. How much the future reward is going to worth in relation to the immediate one depends on the discount factor, γ. In k time steps, the future reward will be multiplied by γ^k. A simplified representation of the Q-learning GMPPT algorithm is shown in Fig. 4.12.

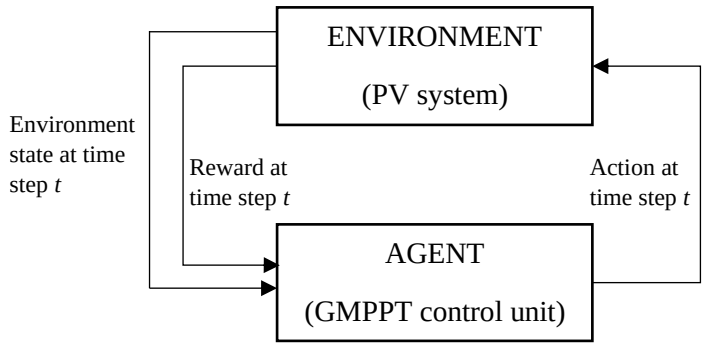

Figure 4.12: Simplified representation of the Q-learning GMPPT algorithm.

In Q-learning, the agent is obtaining knowledge through its interaction with the environment. It aims to learn the Q-values of state-action pairs, which, in the long term, represent the total discounted reward. The agent's knowledge can be described in terms of samples of the form $(S_t, \alpha_t, R_t, S_{t+1})$, meaning that at a specific time-step t, when its state was S_t, the agent chose to execute the action α_t. This caused the state's transition to S_{t+1} and as a result of this transition, the agent received the reward R_t. The Q-learning values are updated at each time-step according to (4.3):

$$Q(S_t, \alpha_t) = Q(S_t, \alpha_t) + l_t \cdot \left[R_t + \gamma \cdot \max_{\alpha_{t+1}} Q(S_{t+1}, \alpha_{t+1}) - Q(S_t, \alpha_t)\right] \quad (4.3)$$

where $Q(S_t, \alpha_t)$ is the Q-value of the current state-action pair (which is going to be updated), $Q(S_{t+1}, \alpha_{t+1})$ is the Q-value of the next state-action pair, l_t is the learning rate, R_t is the immediate reward for the transition and γ is the discount factor.

In each time-step the agent observes the current state S_t and selects an action α_t according to a well-defined action selection policy π^*. After performing the action, it observes the environment's future state S_{t+1} and receives the corresponding immediate reward R_t and chooses the future Q-value which offers the maximum value over all the possible actions, $\max_{\alpha_{t+1}}\{Q(S_{t+1}, \alpha_{t+1})\}$. The learning rate defines how much the new knowledge obtained by the agent is going to alter the existing estimate in the update of $Q(S_t, \alpha_t)$.

4.3.2. Action selection policy

To ensure that the Q-value function is optimally designed (i.e., it corresponds to the most effective action policy), all the possible pairs of state-action should be evaluated for each state that the agent has visited. To achieve this, the Boltzmann exploration policy was used, which defines that each possible action has a probability to be selected by the agent, which is given by the following equation:

$$p(S_t, \alpha_i) = \frac{e^{\frac{Q(S_t, \alpha_i)}{\tau}}}{\sum_{i=1}^{|A|} e^{\frac{Q(S_t, \alpha_i)}{\tau}}} \quad (4.4)$$

where τ is the Boltzmann exploration "temperature", α_i the i-th possible action, $|\mathbf{A}|$ the total number of available actions for each state and $Q(S_t, \alpha_i)$ is the Q-value for the i-th action while being in state S_t.

The temperature function is defined as:

$$\tau = \begin{cases} \tau_{min} + \left(1 - \frac{N}{N_{max}}\right) \cdot (\tau_{max} - \tau_{min}), & if\ N \leq N_{max} \\ \tau_{min}, & otherwise \end{cases} \quad (4.5)$$

where τ_{min} and τ_{max} are the minimum and maximum, respectively, "temperature" values, N is the number of the agent's visits in the specific state and N_{max} is the maximum number of agent visits on each state.

The "temperature" is a positive number and affects the action selection's randomness. When its value is high (during the first visits of the agent to each state), then the probabilities for all the actions will be almost equal, based on (4.4). Consequently, the action will be selected in a random way. After visiting a specific state a few times, its number of visits, N, increases and τ reduces and as a result, the randomness of the action selection also reduces. After N_{max} visits on the state, τ has reached its minimum value and the agent is going to choose the action with the highest Q-value for this state. The exploration process for this state is completed.

4.3.3. State space

To define the state space of the GMPPT algorithm, the following three values were used:

- the PV array's output power, P_{pv},
- the duty cycle of the DC/DC converter, D, and
- the previous value of the DC/DC converter's duty cycle, D_{old}.

The discretized state space is described as follows:

$$S_t = \{\boldsymbol{S} |\ S_{x,y,z} = (P_{pv,x}, D_y\ , D_{old,z}), x \in [1,2,3,\dots,m], y \in [1,2,3,\dots,n], z \in [1,2,3,\dots,j]\} \quad (4.6)$$

where m is the number of equally quantized steps of the PV array's output power range, n is the number of equally quantized steps of the duty cycle range and j is the number of the equally quantized steps of the duty cycle range for the previous time-step. Choosing high values for m, n and j is going to result in a higher accuracy of detecting the MPP, but the learning time and storage requirements will increase. On the contrary, low values for those parameters can result to a shorter learning time, but the MPP's detection accuracy is going to drop. A trade-off is formed between the learning process time and its accuracy.

To describe each state of the state space, all its three components should be defined. For instance, assume that $m = 10, n = 5, j = 5$, the maximum value of P_{pv} is 50 W, the duty cycle ranges between 0 and 1, the PV array's output power is 42 W,

the duty cycle equals 0.25 and its previous value was 0.52. In this case, the environment's state equals (9, 2, 3). This tuple occurred because 42 W belongs to the 9th quantization step of the PV power range, 0.25 belongs to the 2nd quantization step of the duty cycle's range and 0.52 belongs to the 3rd quantization step of the previous time-step's duty cycle's range.

4.3.4. Action space

The seven available actions for the GMPPT controller can be expressed by the following set:

$$\boldsymbol{A} = \{\alpha | + \Delta D_{low}, -\Delta D_{low}, +\Delta D_{mod}, -\Delta D_{mod}, +\Delta D_{high}, -\Delta D_{high}, 0\} \quad (4.7)$$

where ΔD expresses a change of the duty cycle, which may be an increase, a decrease or no-change. By selecting no-change, the duty cycle of the next time-step will be equal to the current duty cycle. The increase and decrease actions are further classified based on the change level, as low, high and moderate. This is done to ensure that all the values of duty cycle are going to be selected during the agent's learning process, thus the agent will converge to the global MPP and not to a local one. This selection is done following the Boltzmann exploration policy, as described in Section 4.3.2 and after exploring all the states, the one with the highest Q-value is selected for each state. When the last action, which stands for no-change to the duty cycle is the one with the highest Q-value, it can be derived that the GMPP is detected, and convergence has been achieved. Until convergence, the duty cycle is updated based on the following equation:

$$D_{t+1} = D_t + \alpha \quad (4.8)$$

4.3.5. Reward

After selecting an action, the agent receives a reward, in order to evaluate the selected action, according to the following equation:

$$R_t = \begin{cases} +r, & if \ P_{pv}(t+1) - P_{pv}(t) > +\delta \\ 0, & if \ |P_{pv}(t+1) - P_{pv}(t)| \leq +\delta \\ -r, & if \ P_{pv}(t+1) - P_{pv}(t) < -\delta \end{cases} \quad (4.9)$$

where $P_{pv}(t+1)$ and $P_{pv}(t)$ are the PV array's output power at two successive time steps, δ is a small positive number (the reward's power threshold) and r is a small reward value. If the power is increased more than δ, a positive reward r will be given to the agent, to "encourage" it to continue selecting such actions, whereas if the power is reduced more than $-\delta$, the reward will be negative, $-r$, to "persuade" the agent to avoid actions that reduce the output power. If the power change is less than the threshold δ, it is considered negligible and no reward is going to be received. By defining reward in that way, during the exploration process, the agent will try to select actions that lead to a power increment.

4.3.6. Discount factor and learning rate

The discount factor γ, as described in Section 4.3.1, is used to balance the immediate and future rewards. In Q-learning, the agent tries to maximize the total future rewards, instead of the immediate one. Based on its value, the significance of

future rewards can be indicated. It is necessary in order to help the agent find optimal sequences of future actions that lead to high rewards soon and not only a currently optimal action. In that way, it is ensured that the agent will converge to the global MPP, and not to a local one.

To ensure the agent's convergence to the GMPP, a criterion should be considered when selecting the learning rate l_t in (4.3), based on the visit frequency of each state. The agent should consider that knowledge obtained from states with a small number of visits is more important than the knowledge acquired from states with a higher number of visits. In that way, states with a small number of visits are going to have a greater learning rate that states that have been visited more times. Based on that, the learning rate l_t is defined as follows:

$$l_t = \frac{k_1}{k_2 + k_3 \cdot N} \tag{4.10}$$

where k_1, k_2 and k_3 are factors that determine the learning rate's decrease for each state, as its number of visits, N, increases.

4.3.7. The Q-learning GMPPT algorithm's flowchart

In order for the algorithm to track the GMPP successfully, two tables should be initialized at its first call. The first one, the Q-table, is a four-dimensional table created based on (4.6). The first three dimensions refer to the state, while the last one refers to the action and its cell contain the Q-value for each state-action pair. Since on the beginning of the exploration there is no prior knowledge, the Q-table is filled with zeros. The second one, the N-table, is a three-dimensional table based also on (4.6), which stores the number of visits for each state (the N values). Finally, an initial value, D_{init}, is assigned to the duty cycle.

After initialization, the flowchart of the Q-learning GMPPT algorithm is presented in Fig. 4.13. The learning process is performed as follows:

Step 1: The PV array's voltage and current are sampled, to calculate its output power. According to the latter, the current duty cycle's value and its previous one, the state S_t is determined using (4.6).

Step 2: The learning rate is calculated using (4.10).

Step 3: The temperature is calculated using (4.5). Additionally, for the current state, the probability of each possible action is calculated using (4.4).

Step 4: If the current state's number of visits is greater than or equal to N_{max} and the action with the highest Q-value for this state is the "no-change" ($\alpha = 0$, based on (X)), then it is considered that the agent achieved convergence to the GMPP or at a point close to it, avoiding all the local MPPs. Then, the Perturb and Observe (P&O) MPPT algorithm is going to be executed, in order to fine-tune the PV array's output power to the GMPP.

Step 5: If the current state's number of visits is less than N_{max} and the action with the highest Q-value for this state is not the "no-change" ($\alpha = 0$, based on (4.7)), a random number in the range $(0,1)$, called N_r is generated. Then, N_r is compared to

the probability of each possible action for the current state. Each probability encodes a specific action, based on the following equation:

$$\begin{gathered}
\text{If } 0 < N_r \leq p(S_t, \alpha_1) \rightarrow \text{Action } 1, \alpha = +\Delta D_{low} \\
\text{Else if } p(S_t, \alpha_1) < N_r \leq p(S_t, \alpha_1) + p(S_t, \alpha_2) \rightarrow \text{Action } 2, \alpha = -\Delta D_{low} \\
\text{Else if } p\,(S_t, \alpha_1) + p(S_t, \alpha_2) < N_r \leq \sum_{i=1}^{3} p(S_t, \alpha_i) \rightarrow \text{Action } 3, \alpha = +\Delta D_{mod} \\
\text{Else if } \sum_{i=1}^{3} p(S_t, \alpha_i) < N_r \leq \sum_{i=1}^{4} p(S_t, \alpha_i) \rightarrow \text{Action } 4, \alpha = -\Delta D_{mod} \\
\text{Else if } \sum_{i=1}^{4} p(S_t, \alpha_i) < N_r \leq \sum_{i=1}^{5} p(S_t, \alpha_i) \rightarrow \text{Action } 5, \alpha = +\Delta D_{high} \\
\text{Else if } \sum_{i=1}^{5} p(S_t, \alpha_i) < N_r \leq \sum_{i=1}^{6} p(S_t, \alpha_i) \rightarrow \text{Action } 6, \alpha = -\Delta D_{high} \\
\text{Else } \rightarrow \text{Action } 7, \alpha = 0
\end{gathered} \tag{4.11}$$

After performing the change to the DC/DC converter's duty cycle, the agent should wait for the PV array's output power to reach the steady-state. Thus, it returns and the learning process will continue on the next call.

Step 6: When the PV array's output power has reached its steady-state, its value is calculated and the reward is determined using (4.9).

Step 7: The future state is determined and the Q-table is updated, as defined by (4.3).

Since on a real system some time is required for the DC/DC converter to reach the steady-state, the steps 1-5 are executed during one call of the algorithm and the steps 6-7 are executed during the next call, as shown in the flowchart of Fig. 4.13.

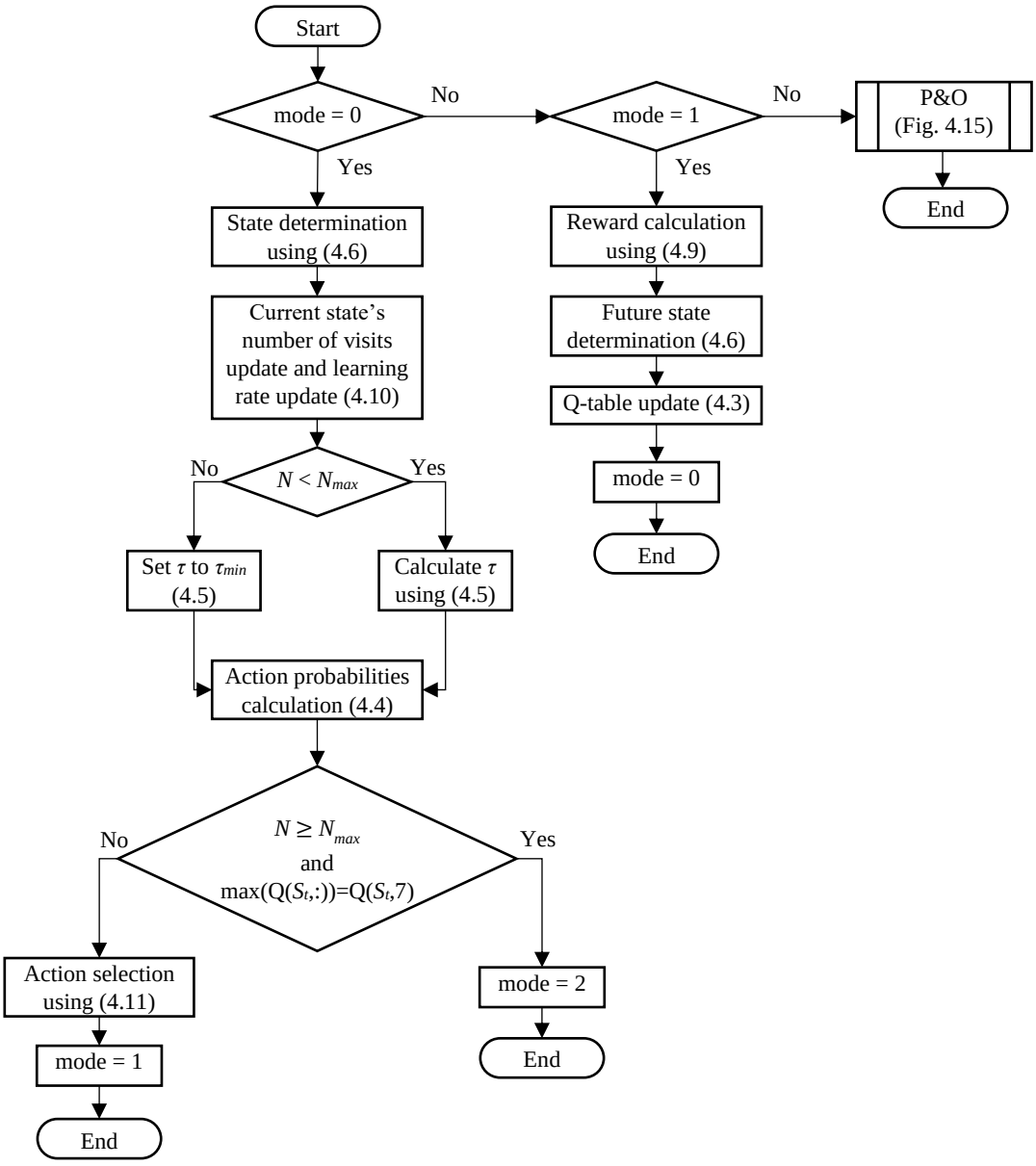

Figure 4.13: The Q-learning GMPPT algorithm's flowchart.

When convergence is achieved, the PV array's output power may be slightly less than the GMPP value, because the state space is quantized in steps that cover a range of values, and not a discrete one. However, to decide if the controller should stay at the GMPPT mode, or if it should tune the PV array's output power to a lower value, the GMPP should be reached. For this reason, when the agent decides that the GMPP is found, the P&O algorithm is executed for the next 10 calls, to ensure the GMPP's tracking. During these calls, if the PV array's power is higher than the Q-learning reported value, it is updated.

After the 10 calls, if the reference power is higher than the GMPP power, then the DC/DC converter will inevitably stay at the GMPP mode, since it is impossible to output the reference power. Otherwise, it will fine-tune the duty cycle to find the first

point at the left side of the GMPP, where the PV array's output power will be equal to the reference power.

To detect any environmental changes, such as irradiance or temperature changes, the same rule as for the previous algorithm is applied:

$$\Delta p = \left|\frac{P_{pv} - P_{pv,old}}{P_{pv,old}}\right| > 0.15 \rightarrow \text{environmental changes}$$

If the PV array's output power changes more than 15% between two successive algorithm calls, then environmental changes are considered to occur. This means that the P-V curve of the PV array has changed and the algorithm needs to reset in order to track efficiently the reference power. It is worth noting that since during the learning process the agent imposes high duty cycle steps to the DC/DC converter, the criterion for the detection of environmental changes is evaluated only after the agent's convergence.

The flowchart of the GFPPT algorithm based on Q-learning GMPPT is presented in Fig. 4.14.

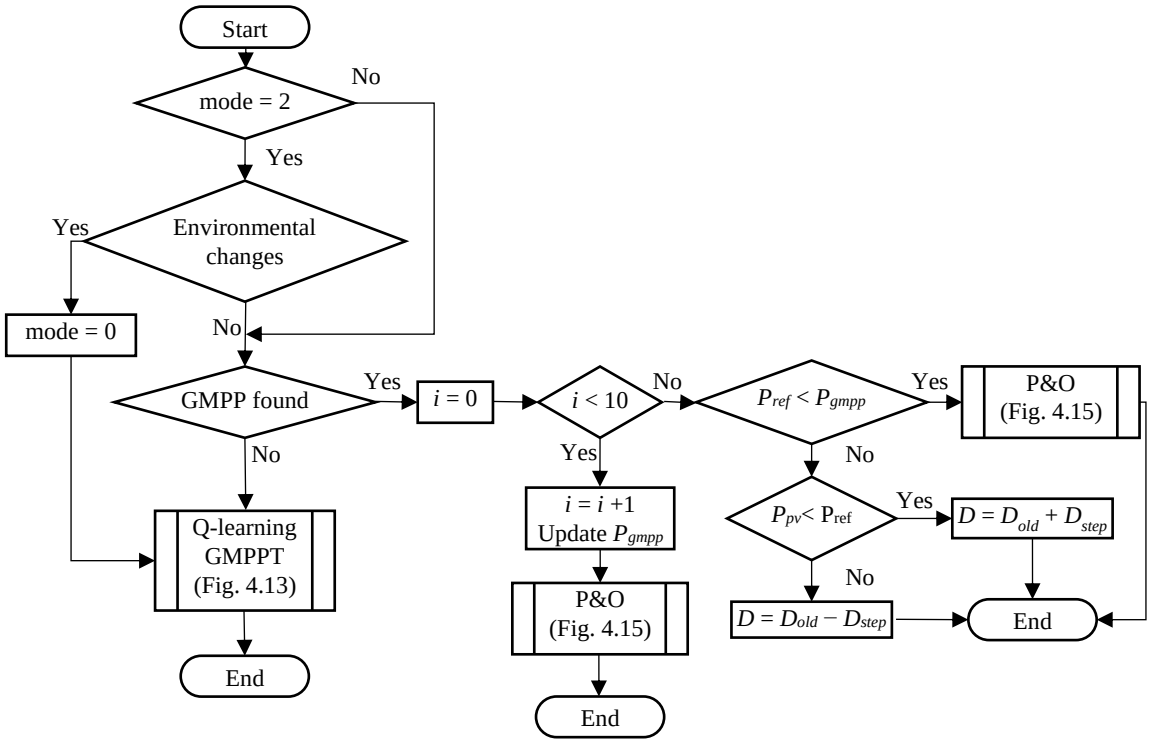

Figure 4.14: Flowchart of the GFPPT algorithm based on Q-learning GMPPT.

The flowchart of the Perturb and Observe (P&O) MPPT algorithm [27] which is used during the fine tuning is presented in Fig. 4.15. Based on the sign of derivative of the PV array's P-V curve, it modifies the duty cycle of the DC/DC converter according to the following rule:

- if $\frac{\partial P_{pv}}{\partial V_{pv}} < 0$, the system is operating at the right side of the MPP, thus the PV array's voltage should be reduced.
- if $\frac{\partial P_{pv}}{\partial V_{pv}} > 0$, the system is operating at the left side of the MPP, thus the PV array's voltage should be increased.
- if $\frac{\partial P_{pv}}{\partial V_{pv}} = 0$, the system is operating at the MPP, and the PV array's voltage is not modified.

It is noted that in order to calculate the aforementioned derivative, the change in the PV array's power and voltage difference between two consecutive calls of the algorithm is used, as it is shown in the flowchart. Additionally, since this algorithm outputs a duty cycle value, the P&O algorithm is adapted to this by modifying the duty cycle (by a small value of D_{step}) instead of outputting a reference voltage value.

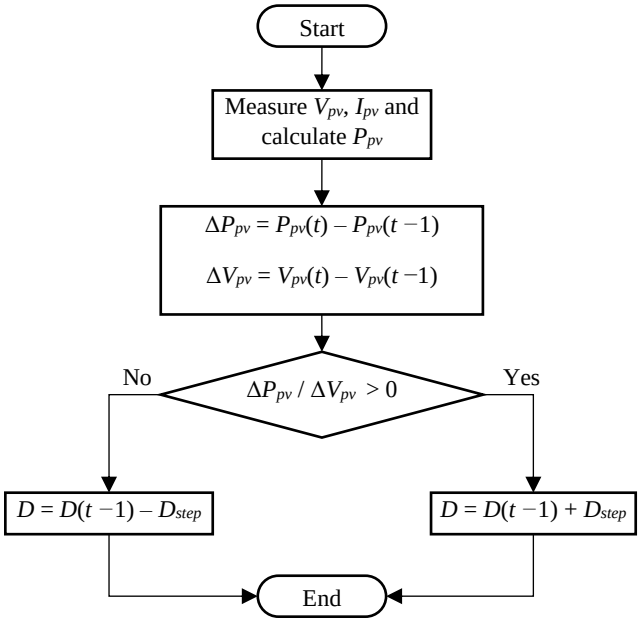

Figure 4.15: The Perturb and Observe (P&O) algorithm's flowchart.

4.4. Algorithm #3 – Q-learning GFPPT

Although the previous algorithm (GFPPT based on Q-learning GMPPT) shows great performance after appropriate training, with its convergence time being significantly reduced after training on a few shading patterns, it has a serious drawback. If the reference power is much lower than the GMPP power, then a lot of time will be spent for the fine-tuning process. Moreover, if the reference power is changing frequently (which can easily happen on a grid-connected PV system), and if there are not environmental changes, the algorithm is going to fine-tune the duty cycle, until achieving the desired power.

To overcome this issue, instead of searching the GMPP and then, if needed, limiting the PV array's output power to match the reference power, a new algorithm based on Q-learning was developed, which aims to regulate the PV array's output

power to the reference value, and only if this cannot be satisfied, it will output the maximum available power. This can be achieved by keeping the main structure of the Q-learning algorithm and modifying some of its components.

4.4.1. The state space

The state space of the GFPPT algorithm is defined using the following four values:

- the reference power, P_{ref},
- the PV array's output power, P_{pv},
- the duty cycle of the DC/DC converter, D, and
- the previous value of the DC/DC converter's duty cycle, D_{old}.

The discretized state space is described as follows:

$$S_t = \{\boldsymbol{S} |\, S_{w,x,y,z} = (P_{ref,w}, P_{pv,x}, D_y\,, D_{old,z},), w \in [1,2,3,\dots,q], \\ x \in [1,2,3,\dots,m], y \in [1,2,3,\dots,n], z \in [1,2,3,\dots,j]\,\} \tag{4.12}$$

where q is the number of equally quantized steps of the reference power, m is the number of equally quantized steps of the PV array's output power range, n is the number of equally quantized steps of the duty cycle range and j is the number of the equally quantized steps of the duty cycle range for the previous time-step. Choosing high values for q, m, n and j will result in a higher accuracy of tracking the flexible power point, but the learning time and storage requirements will increase. On the contrary, low values for these parameters can result to a shorter learning time, but the flexible power point's tracking accuracy is going to drop. A trade-off is formed between the learning process time and its accuracy.

To describe each state of the state space, all its four components should be defined. For instance, assume that $q = 10$, $m = 10$, $n = 5$, $j = 5$, the maximum reference power and the maximum value of P_{pv} is 50 W, the duty cycle ranges between 0 and 1, the reference power is 16 W, the PV array's output power is 42 W, the duty cycle equals 0.25 and its previous value was 0.52. In this case, the environment's state equals (4, 9, 2, 3). This tuple occurred because 16 W belongs to the 4th quantization step of the reference power range, 42 W belongs to the 9th quantization step of the PV power range, 0.25 belongs to the 2nd quantization step of the duty cycle's range and 0.52 belongs to the 3rd quantization step of the previous time-step's duty cycle's range.

4.4.2. Action space

The seven available actions for the GFPPT controller can be expressed by the following set:

$$\boldsymbol{A} = \{\alpha | + \Delta D_{low}, -\Delta D_{low}, +\Delta D_{mod}, -\Delta D_{mod}, +\Delta D_{high}, -\Delta D_{high}, 0\,\} \tag{4.13}$$

where ΔD expresses a change of the duty cycle, which may be an increase, a decrease or no-change. By selecting no-change, the duty cycle of the next time-step will be equal to the current duty cycle. The increase and decrease actions are further

classified based on the change level, as low, high and moderate. This is done to ensure that all the values of duty cycle are going to be selected during the agent's learning process, thus the agent will converge to the reference power's point with the highest voltage. This selection is done following the Boltzmann exploration policy, as described in Section 4.3.2. and after exploring all the states, the one with the highest Q-value is selected for each state. When the last action, which stands for no-change to the duty cycle is the one with the highest Q-value, it can be derived that the flexible power point (or the GMPP in case the reference power is greater than that) is detected and convergence has been achieved.

Until convergence, the duty cycle is updated based on the following equation:

$$D_{t+1} = D_t + \alpha \tag{4.14}$$

4.4.3. Reward

After selecting an action, the agent receives a reward, in order to evaluate the selected action. In the previous algorithm, the agent had one single objective, the maximization of the PV array's output power. However, this algorithm's agent aims to minimize the error and since there are many points where the error minimizes, it simultaneously aims to find the one with the highest voltage. The former algorithm belongs to the class of the Single Objective Reinforcement Learning (SORL) algorithms, while the latter belongs to the class of the Multi Objective Reinforcement Learning (MORL) algorithms. Since the problem of GFPPT includes two objectives, a separate reward can be introduced for each objective.

For the error minimization, the error's absolute value is defined as follows:

$$e(t) = \left|P_{pv}(t) - P_{ref}(t)\right| \tag{4.15}$$

and the agent should receive a reward R_e, proportional to how much the error was reduced between two successive time steps.

$$R_e = -\left(\frac{e(t+1) - e(t)}{C_e}\right) \tag{4.16}$$

where C_e is a factor used to normalize the reward's value. The minus sign is added, in order to achieve the error's minimization.

To encourage the agent to select the point with the greatest voltage value, it receives a reward R_V, which is calculated as follows:

$$R_V = \frac{V_{pv}(t+1) - V_{pv}(t)}{C_V} \tag{4.17}$$

where C_V is a factor used to normalize the reward's value.

To define an overall reward function, the weighted sum approach [29] was used. The overall reward is defined as the weighed combination of R_e and R_V, as follows:

$$R = W_e R_e + W_V R_V \tag{4.18}$$

where W_e and W_V are the weights assigned to the two parts of the reward.

Additionally, it was observed during the simulation of the algorithm that a few times it would get stuck to the minimum or the maximum value of the duty cycle for six to ten successive calls. To prevent this from happening, one more reward was defined:

$$R_d = \begin{cases} -1, & \text{if } D(t+1) = D(t-1) = D_{min}, \text{or } D(t+1) = D(t-1) = D_{max} \\ 0, & \text{otherwise} \end{cases} \tag{4.19}$$

After being assigned a weight constant W_d, it was combined with the reward in (4.18), to give the total reward function of the GFPPT controller:

$$R_t = W_e R_e + W_V R_V + W_d R_d \tag{4.20}$$

By defining reward in that way, during the exploration process, the agent will try to select actions that lead to the minimization of the power error and the maximization of the PV array's voltage simultaneously.

4.4.4. Discount factor and learning rate

The definitions for the discount factor and the learning rate remain the same as in the previous algorithm, described in Section 4.3.6.

4.4.5. The Q-learning GFPPT algorithm's flowchart

In order for the algorithm to track the GFPP successfully, two tables should be initialized at its first call. The first one, the Q-table, is a five-dimensional table created based on (4.12). The first four dimensions refer to the state, while the last one refers to the action and its cell contain the Q-value for each state-action pair. Since on the beginning of the exploration there is no prior knowledge, the Q-table is filled with zeros. The second one, the N-table, is a four-dimensional table based also on (4.12), which stores the number of visits for each state (the N values). Finally, an initial value, D_{init}, is assigned to the duty cycle.

After initialization, the flowchart of the Q-learning GFPPT algorithm is presented in Fig. 4.16. The following process is performed during the algorithm's training period:

Step 1: The PV array's voltage and current are sampled, to calculate its output power and the error, using (4.15). According to the error, the PV array's power, the current duty cycle's value and its previous one, the state S_t is determined using (4.12).

Step 2: The learning rate is calculated using (4.10).

Step 3: The temperature is calculated using (4.5). Additionally, for the current state, the probability of each possible action is calculated using (4.4).

Step 4: If the current state's number of visits is greater than or equal to N_{max} and the action with the highest Q-value for this state is the "no-change" ($\alpha = 0$, based on (4.13)), then it is considered that the agent achieved convergence to the GFPP or at a

point close to it, avoiding all the other FPP points with lower voltage. Then, the duty cycle is fine-tuned, to track the GFPP or the GMPP (when the reference power is greater than the GMPP).

Step 5: If the current state's number of visits is less than N_{max} and the action with the highest Q-value for this state is not the "no-change" ($\alpha = 0$, based on (4.13)), a random number in the range (0,1), called N_r is generated. Then, N_r is compared to the probability of each possible action for the current state. Each probability encodes a specific action, based on (4.11).

After performing the change to the DC/DC converter's duty cycle, the agent should wait for the PV array's output power to reach the steady-state. Thus, it returns, and the learning process will continue on the next call.

Step 6: When the PV array's output power has reached its steady-state, the new value or error is calculated, and the reward is determined using (4.20).

Step 7: The future state is determined using (4.6), and the Q-table is updated, as defined by (4.3).

Similarly to the previous algorithm, since on a real system some time is required for the DC/DC converter to reach the steady-state, the steps 1-5 are executed during one call of the algorithm and the steps 6-7 are executed during the next call, as shown in the flowchart of Fig. 4.11.

To detect any environmental changes, such as irradiance or temperature changes, the same rule as for the previous algorithms is applied:

$$\Delta p = \left| \frac{P_{pv} - P_{pv,old}}{P_{pv,old}} \right| > 0.15 \rightarrow \text{environmental changes}$$

If the PV array's output power changes more than 15% between two successive algorithm calls, then environmental changes are considered to occur. This means that the P-V curve of the PV array has changed and the algorithm needs to reset, to track efficiently the reference power. It is worth noting that since during the learning process the agent imposes high duty cycle steps to the DC/DC converter, the criterion for the detection of environmental changes is evaluated only after the agent's convergence.

By defining the state space as described by (4.12), if the reference power between two consecutive calls changes enough to transit to a different quantization level, the algorithm resets. The reset is done by setting mode to 0.

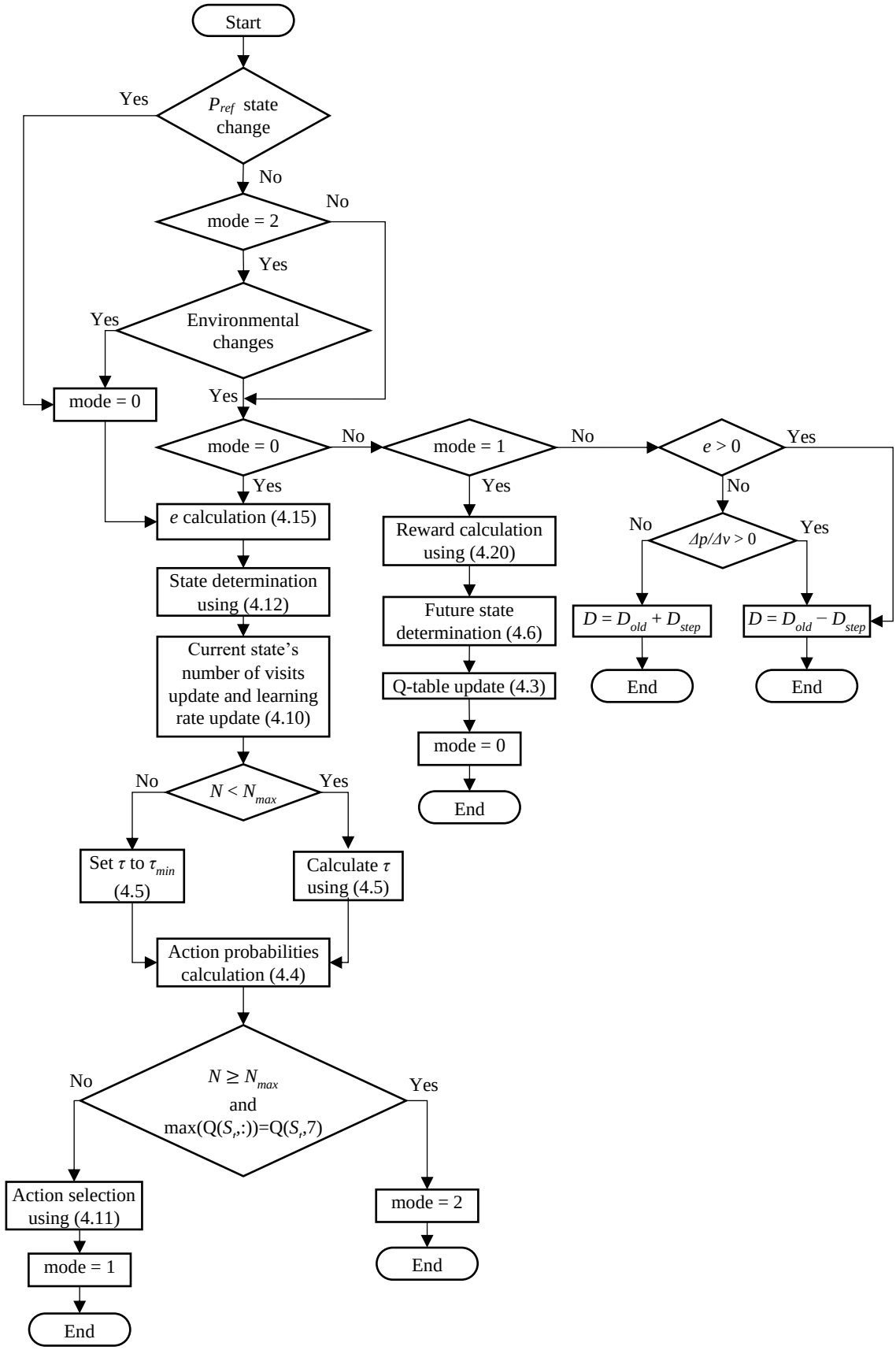

Figure 4.16: Flowchart of the Q-learning GFPPT algorithm.

4.5. Comparison of the three PV GFPPT algorithms

Based on the analysis that was done for each algorithm in the previous sections, their advantages and disadvantages that can be derived are presented below.

Algorithm #1

Advantage: It is based on simple algebraic calculations and can be implemented on the commercially available microcontrollers.

Disadvantages: Instead of a value for the duty cycle, it outputs a reference voltage value. This means that more complexity is added to the system, in order to control the DC/DC converter's duty cycle. Additionally, it needs the PV array's open-circuit voltage value to operate. Obtaining this value results to a rapid oscillation for the system, after each environmental change.

Algorithm #2

Advantage: Being based on a machine learning technique, after a relatively short training period, its convergence time to the GMPP can reduce significantly.

Disadvantages: Since it converges to the GMPP, further fine-tuning of the duty cycle is needed to track the reference power. If these two values have a great difference, a significant amount of time is going to be spent on fine-tuning. Furthermore, a microcontroller which supports an external memory module is needed (this can be a flash memory or an SD card module), in order to store the Q and N tables.

Algorithm #3

Advantages: Aiming directly to the reference power and based on a machine learning technique, after a relatively short training period, its convergence time to the reference power can reduce significantly. Moreover, by maximizing the PV array's voltage, its current is reduced and consequently the power losses are minimized.

Disadvantage: Similarly to algorithm #2, it also requires a microcontroller capable of interfacing an external memory module.

5. Experimental results

5.1. Configuration of the three algorithms

In order to obtain a fair comparison between the algorithms, each one's parameters had been tuned in order to obtain optimal performance. An algorithm's performance is considered optimal when convergence to the reference power is achieved in the smallest time interval with the lowest possible steady-state tracking error. Based on the trial-and-error method, after numerous tests for each algorithm, the following sets of parameters was chosen, as shown in Tables 5.1, 5.2 and 5.3 respectively. The sampling time for all the algorithms was set to 1 sec.

The tracking error is calculated over a time period from $t = t_0$ to $t = t_1$ as follows:

$$TE = \frac{\int_{t_0}^{t_1} \left| P_{pv} - P^* \right| dt}{\int_{t_0}^{t_1} |P^*| \, dt} \tag{5.1}$$

where P_{pv} is the PV array's output power and P^* is defined by the following formula:

$$P^* = \begin{cases} P_{ref}, & \text{if } P_{ref} < P_{gmpp} \\ P_{gmpp}, & \text{otherwise} \end{cases} \tag{5.2}$$

It is worth noting that many sets of parameters for the Q-learning procedure can result to a satisfactory system performance. Based on the system's performance observation, the sets of Tables 5.2 and 5.3 were used.

Finally, it should be mentioned that the voltage and duty cycle steps cannot be extremely low, since the voltage and current measurements are inevitably affected by noise. Even if the noise's effect on the measurements is negligible, the system will have a slow transient response. On the other hand, relatively large steps for these parameters may lead to instability on the steady-state, since large voltage oscillations may occur. A tradeoff is performed between the system's dynamic response and its reference power tracking accuracy. Based on experimental observations, the voltage step for algorithm #1 was set to be 0.15 V and the fine-tuning duty cycle steps to be 1% and 0.5% for algorithms 2 and 3, respectively. A lower step was used for the last algorithm, since it operates at the left part of an LMPP, where the derivative of the P-V curve is higher than on the right.

Table 5.1: Operational parameters for the first algorithm.

Algorithm #1 – Experimental evaluation parameters	
Voltage step V_{step}	0.15 V
Minimum reference voltage	3 V

Table 5.2: Operational parameters for the second algorithm.

Algorithm #2 – Experimental evaluation parameters	
Quantization levels of P_{pv}	12
Quantization levels of D	12
Quantization levels of D_{old}	6
Nominal maximum system power	48 W
Duty cycle range	0.2 – 0.98
Initial duty cycle value	0.2
Discount factor γ	0.75
Learning rate factors	k_1 = 10, k_2 =25, k_3 = 0.6
Max. number of state visits N_{max}	20
Actions (duty cycle changes)	0, ±0.4. ±0.12. ±0.28
Temperature limit values	τ_{min} = 0.08, τ_{max} = 0.8
Immediate reward value r	1
Power threshold for the reward δ	1 W
Duty cycle step during the fine tuning	0.01

Table 5.3: Operational parameters for the third algorithm.

Algorithm #3 – Experimental evaluation parameters	
Quantization levels of P_{ref}	10
Quantization levels of P_{pv}	20
Quantization levels of D	20
Quantization levels of D_{old}	10
Nominal maximum system power	48 W
Duty cycle range	0.2 – 0.98
Initial duty cycle value	0.2
Discount factor γ	0.75
Learning rate factors	k_1 = 10, k_2 =25, k_3 = 0.6
Max. number of state visits N_{max}	20
Actions (duty cycle changes)	0, ±0.4. ±0.12. ±0.28
Temperature limit values	τ_{min} = 0.08, τ_{max} = 0.8
Reward normalization factors	C_e = 16, C_v = 6
Reward component weights	W_e = 2, W_v = 1, W_d = 3
Duty cycle step during the fine tuning	0.005

5.2. Training the machine-learning-based algorithms

As described in Section 4, the first algorithm does not include any form of machine learning. Thus, its convergence time is not affected by any previous calls.

However, the two last ones obtain knowledge for each shading pattern they face. As a result, on their first executions they will show a slow transient response, which is going to accelerate as the number of calls increases. To bring out their advantages when comparing the three algorithms, they were trained as described in the following two sections.

5.2.1. Algorithm #2 training process

To train the agent for the second algorithm (GFPPT based on Q-learning GMPPT), a set of 15 different shading patterns were used, as shown on Table 5.4. In order to achieve an effective training, for some of the patterns the GMPP is located at the left region on the P-V curve, while for some others it is located at the right region of the P-V curve. It is noted that during the training process the reference power was set to 48 W, thus the GMPP was tracked on each experiment. The tracking error (TE) is calculated according to (5.1).

Table 5.4: Algorithm #2 training overview.

Shading pattern	PV1 (+) Irradiance (W/m^2)	PV2 (-) Irradiance (W/m^2)	Air temperature (°C)	GMPP power (W)	Number of convergence steps	TE_{ss} (%)
2.1	805	570	24.8	24.64	220	1.03
2.2	965	265	25.3	15.05	34	0.84
2.3	560	1020	25.4	25.05	10	1.43
2.4	420	900	25.8	18.53	12	1.42
2.5	550	550	25.3	22.07	2	1.13
2.6	740	560	25.2	24.05	12	2.63
2.7	700	400	25.3	25.05	10	1.43
2.8	800	600	25.5	26.52	42	0.82
2.9	620	320	26.1	12.46	16	4.09
2.10	760	460	25.9	20.49	8	1.19
2.11	940	540	25.2	24.07	8	0.74
2.12	810	810	25.8	33.63	30	0.58
2.13	950	570	25.9	23.13	8	0.88
2.14	900	700	26.0	29.45	18	1.46
2.15	900	600	25.9	25.48	8	1.17

To show the algorithm's advantage on gaining knowledge over time to converge faster to the GMPP, the experimental results for shading patterns 2.1 and 2.13 are analyzed next. For shading pattern 2.1, the agent had no prior knowledge of where the GMPP could be located. As a result, it had to explore many states, using the seven possible duty cycle steps, as described by (4.7). The steps it followed are shown in Fig. 5.1-5.6.

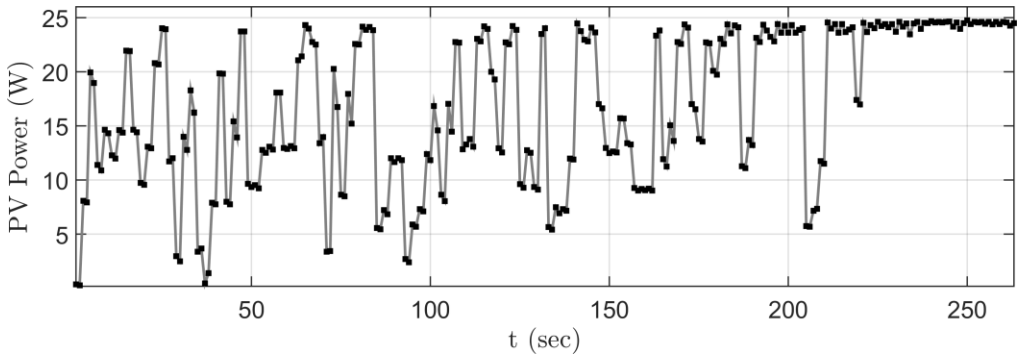

Figure 5.1: The PV array's power (shading pattern 2.1).

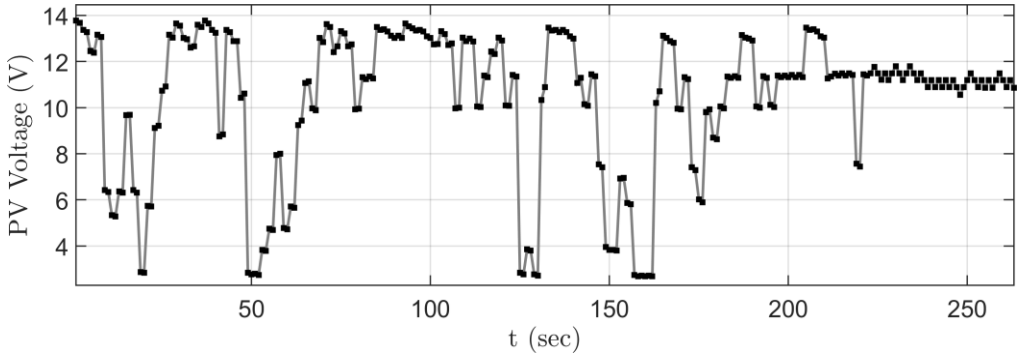

Figure 5.2: The PV array's voltage (shading pattern 2.1).

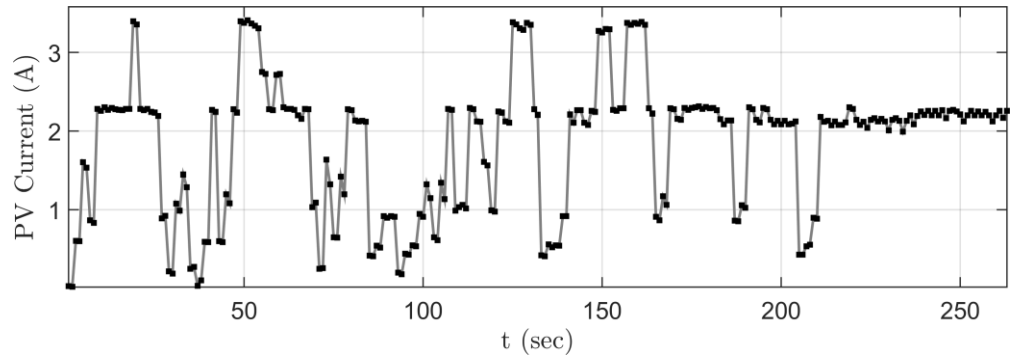

Figure 5.3: The PV array's current (shading pattern 2.1).

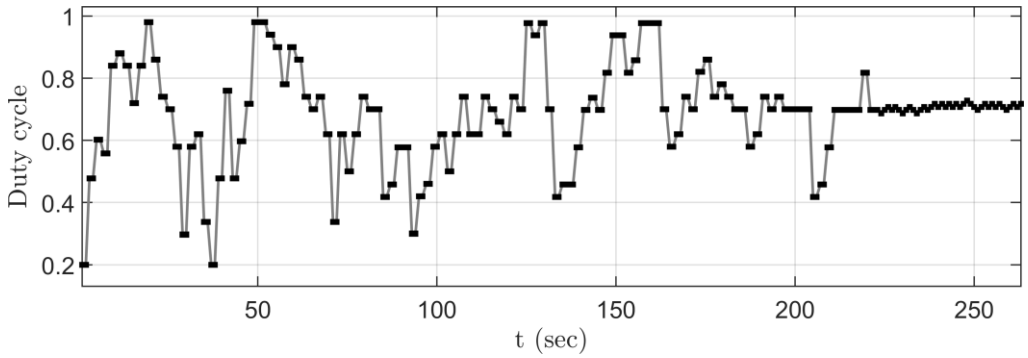

Figure 5.4: The DC/DC converter's duty cycle (shading pattern 2.1).

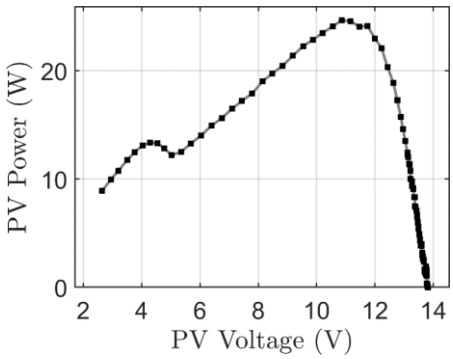

Figure 5.5: The PV array's P-V curve (shading pattern 2.1).

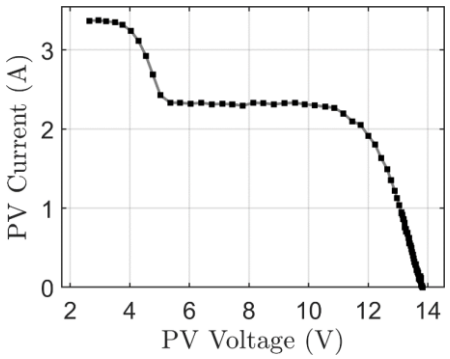

Figure 5.6: The PV array's I-V curve (shading pattern 2.1).

After being trained on 12 shading patterns, when it was called for shading pattern 2.13, it could estimate where the GMPP might be located. Its steps were not that random as in the first experiment, and as a result, it converged to the GMPP in only 8 steps, as shown in Fig. 5.7-5.12.

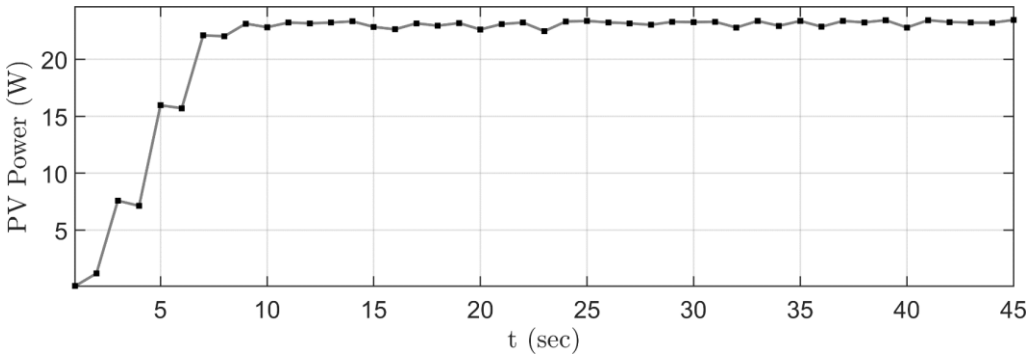

Figure 5.7: The PV array's power (shading pattern 2.13).

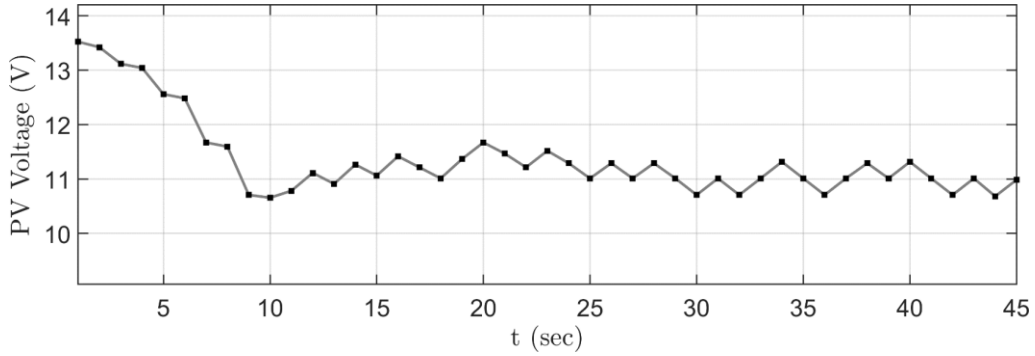

Figure 5.8: The PV array's voltage (shading pattern 2.13).

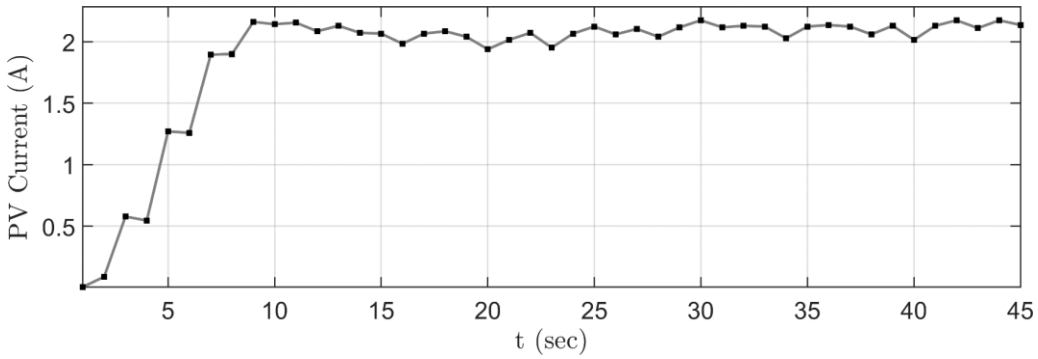

Figure 5.9: The PV array's current (shading pattern 2.13).

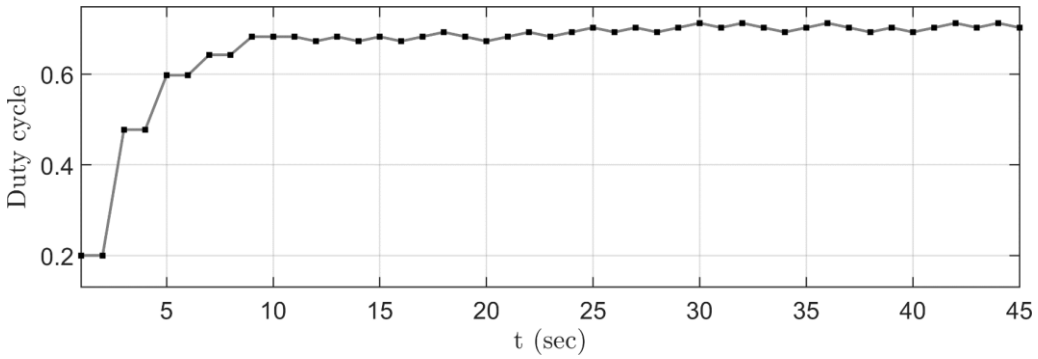

Figure 5.10: The DC/DC converter's duty cycle (shading pattern 2.13).

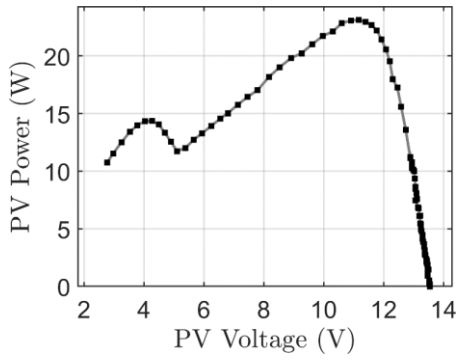

Figure 5.11: The PV array's P-V curve (shading pattern 2.13).

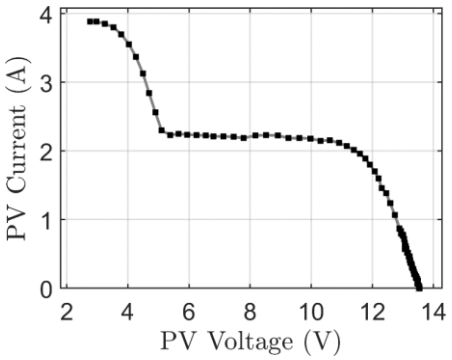

Figure 5.12: The PV array's I-V curve (shading pattern 2.13).

5.2.2. Algorithm #3 training process

Unlike the second algorithm, the third one's state-space contains one more attribute, the reference power. Thus, in order to train its agent, a set of different reference power values should be used, ranging from 12 W to 30 W, as shown in Tables 5.5 – 5.8. The tracking error (TE) is calculated according to (5.1).

Table 5.5: Algorithm #3 training overview (for 20 W reference power).

Shading pattern	PV1 (+) Irr. (W/m^2)	PV2 (-) Irr. (W/m^2)	Air Temp. (°C)	Ref. power (W)	GMPP power (W)	Number of convergence steps	TE_{ss} (%)
3.20.1	530	845	25.1	20	22.02	468	1.72
3.20.2	660	880	25.1	20	27.94	12	2.47
3.20.3	765	865	24.2	20	32.71	118	2.86
3.20.4	620	920	24.1	20	28.02	32	2.49
3.20.5	630	980	24.4	20	27.11	228	2.92
3.20.6	530	1020	24.7	20	22.60	22	1.93
3.20.7	360	1060	24.6	20	15.73	78	2.11
3.20.8	270	1040	24.6	20	15.30	112	2.32
3.20.9	500	1000	23.9	20	21.14	2	1.30
3.20.10	520	900	24.9	20	22.70	8	1.76
3.20.11	425	810	24.9	20	19.34	14	1.96
3.20.12	380	620	24.5	20	15.93	12	3.29
3.20.13	550	740	23.8	20	22.51	50	2.16
3.20.14	700	730	23.4	20	29.12	88	3.02
3.20.15	530	845	25.1	20	32.88	26	3.02

Table 5.6: Algorithm #3 training overview (for 25 W reference power).

Shading pattern	PV1 (+) Irr. (W/m^2)	PV2 (-) Irr. (W/m^2)	Air Temp. (°C)	Ref. power (W)	GMPP power (W)	Number of convergence steps	TE_{ss} (%)
3.25.1	925	600	24.8	25	25.27	398	1.08
3.25.2	760	650	24.9	25	26.62	66	1.33

3.25.3	740	430	24.9	25	16.31	210	0.80
3.25.4	640	500	25.0	25	18.33	16	1.58
3.25.5	880	650	25.0	25	27.79	158	1.94
3.25.6	1015	640	24.9	25	27.81	14	1.68
3.25.7	705	530	24.7	25	20.96	92	0.55
3.25.8	840	480	27.1	25	18.09	30	1.52
3.25.9	810	360	27.7	25	14.45	58	1.54
3.25.10	720	470	27.5	25	17.61	26	1.14
3.25.11	820	720	27.3	25	29.54	42	1.76
3.25.12	975	700	27.8	25	29.66	58	1.97
3.25.13	720	510	27.5	25	18.67	16	1.32
3.25.14	910	640	27.4	25	26.73	50	1.58
3.25.15	590	320	27.1	25	12.09	108	2.95

Table 5.7: Algorithm #3 training overview (for 30 W reference power).

Shading pattern	**PV1 (+) Irr. (W/m^2)**	**PV2 (-) Irr. (W/m^2)**	**Air Temp. (°C)**	**Ref. power (W)**	**GMPP power (W)**	**Number of convergence steps**	**TE_{ss} (%)**
3.30.1	850	545	26.1	30	23.31	378	1.73
3.30.2	785	710	25.7	30	28.57	226	2.70
3.30.3	800	865	28.1	30	32.95	146	2.44
3.30.4	755	960	26.5	30	31.76	36	1.91
3.30.5	730	1010	28.3	30	30.94	22	1.47
3.30.6	865	990	29.0	30	36.35	138	1.44
3.30.7	640	985	29.4	30	28.71	22	1.56
3.30.8	535	1010	29.4	30	29.08	22	2.52
3.30.9	250	1005	29.3	30	14.41	144	1.66
3.30.10	280	860	29.2	30	12.51	46	0.84
3.30.11	565	815	28.8	30	25.55	8	1.80
3.30.12	405	615	24.0	30	15.76	28	3.99
3.30.13	570	345	24.7	30	10.57	40	8.97
3.30.14	875	775	24.8	30	32.62	40	1.55
3.30.15	950	660	24.7	30	26.96	10	0.41

Table 5.8: Algorithm #3 training overview (for 12 W reference power).

Shading pattern	**PV1 (+) Irr. (W/m^2)**	**PV2 (-) Irr. (W/m^2)**	**Air Temp. (°C)**	**Ref. power (W)**	**GMPP power (W)**	**Number of convergence steps**	**TE_{ss} (%)**
3.12.1	600	1000	25.2	12	25.78	148	7.62
3.12.2	200	1000	26.1	12	13.52	112	3.88
3.12.3	300	800	25.7	12	11.75	18	0.53
3.12.4	900	700	26.1	12	31.60	14	2.11

For the shading pattern 3.20.1, the agent had no prior knowledge of where the GFPP could be located. As a result, it had to explore many states, using the seven

possible duty cycle steps, as described by (4.13). The steps it followed are shown in Fig. 5.13-5.18.

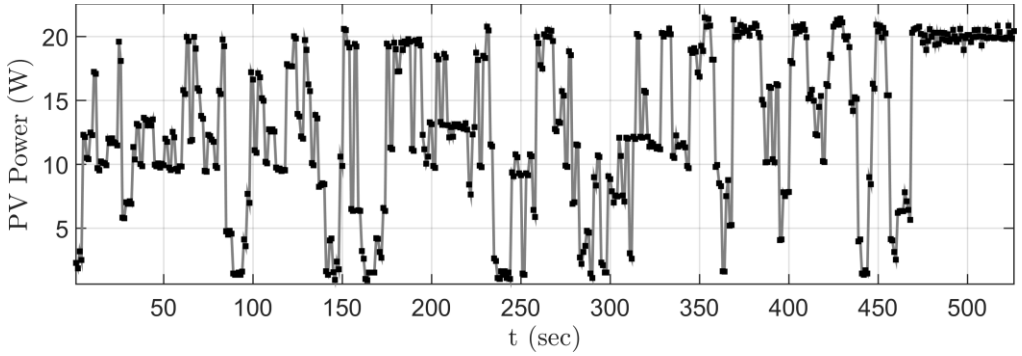

Figure 5.13: The PV array's power (shading pattern 3.20.1).

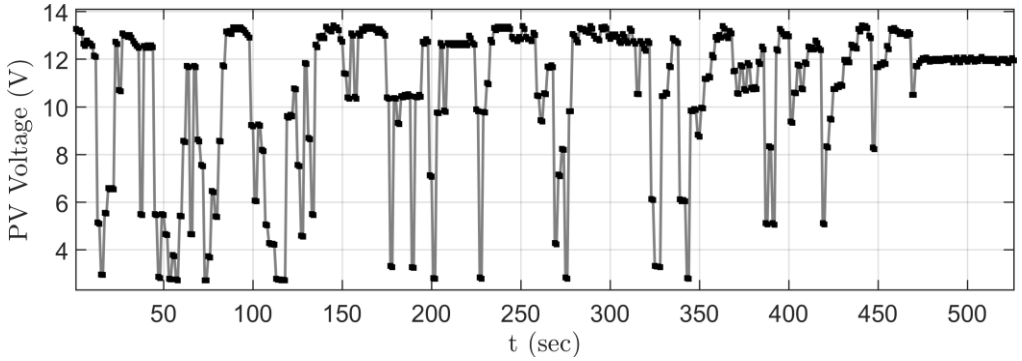

Figure 5.14: The PV array's voltage (shading pattern 3.20.1).

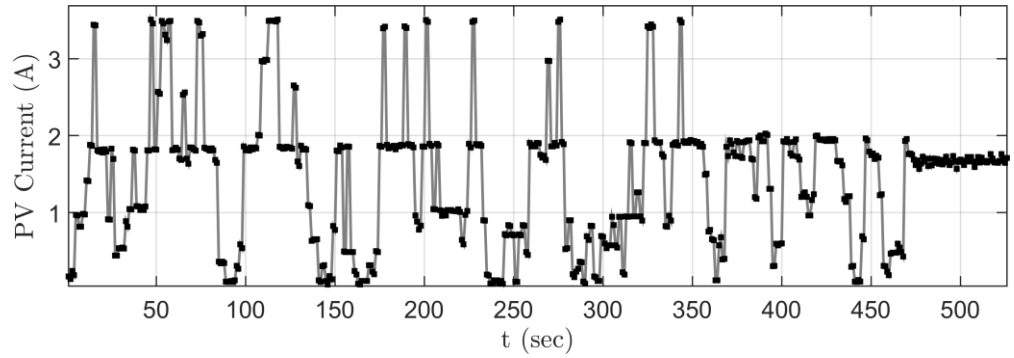

Figure 5.15: The PV array's current (shading pattern 3.20.1).

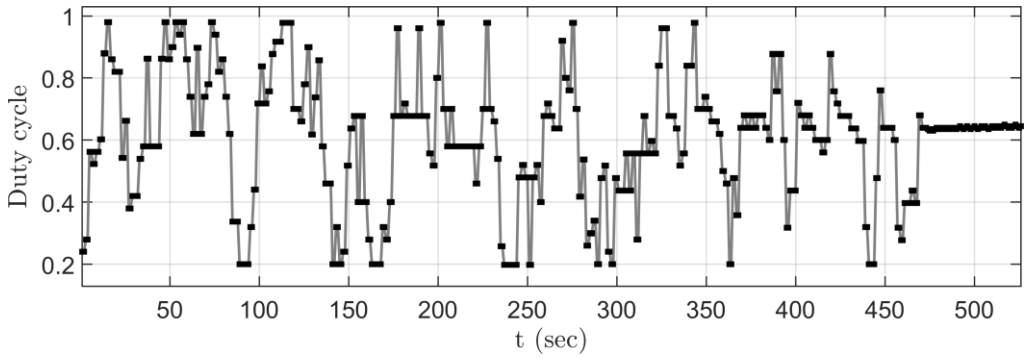

Figure 5.16: The DC/DC converter's duty cycle (shading pattern 3.20.1).

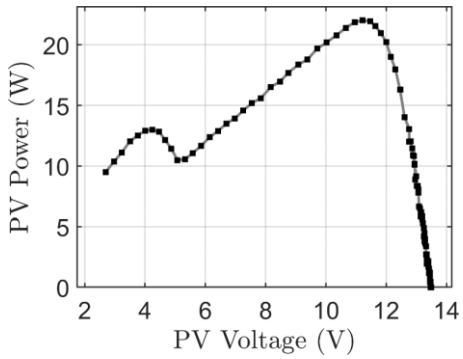

Figure 5.17: The PV array's P-V curve (shading pattern 3.20.1).

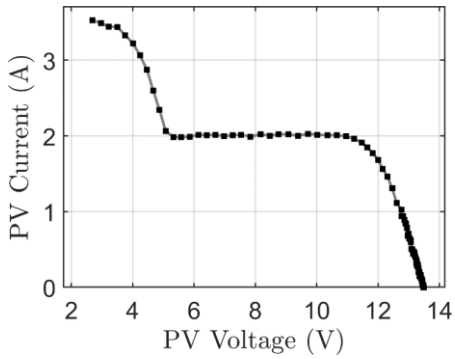

Figure 5.18: The PV array's I-V curve (shading pattern 3.20.1).

After being trained on 9 shading patterns, when it was called for shading pattern 3.20.10, it could estimate where the GFPP might be located. Its steps were not that random as in the first experiment, and as a result, it converged to the GFPP in only 8 steps, as shown in Fig. 5.19-5.24.

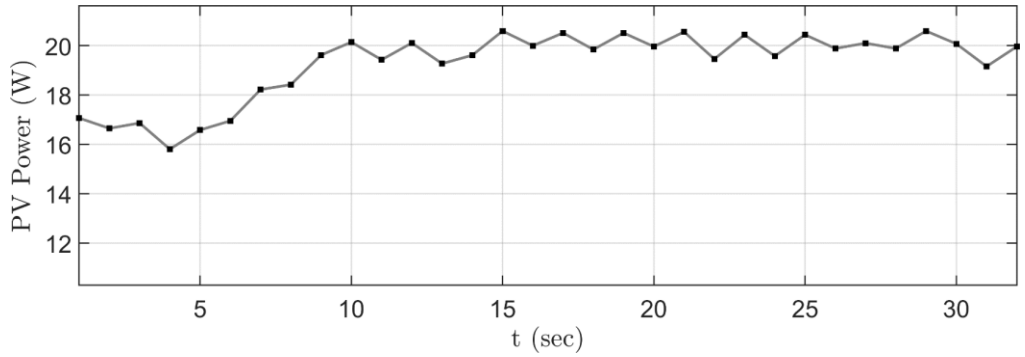

Figure 5.19: The PV array's power (shading pattern 3.20.10).

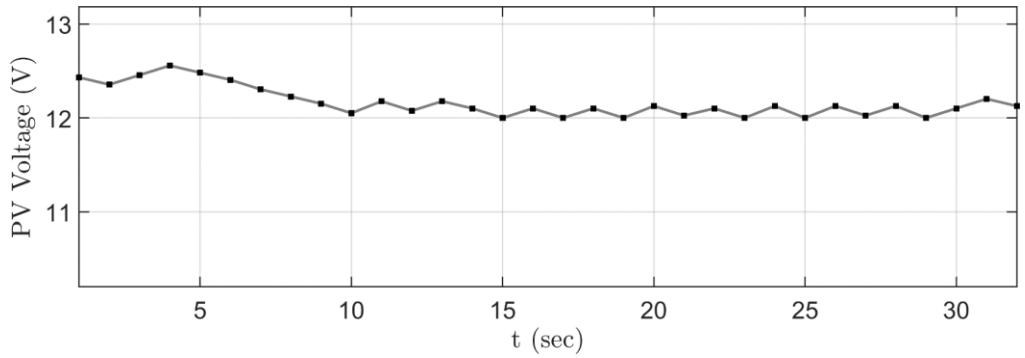

Figure 5.20: The PV array's voltage (shading pattern 3.20.10).

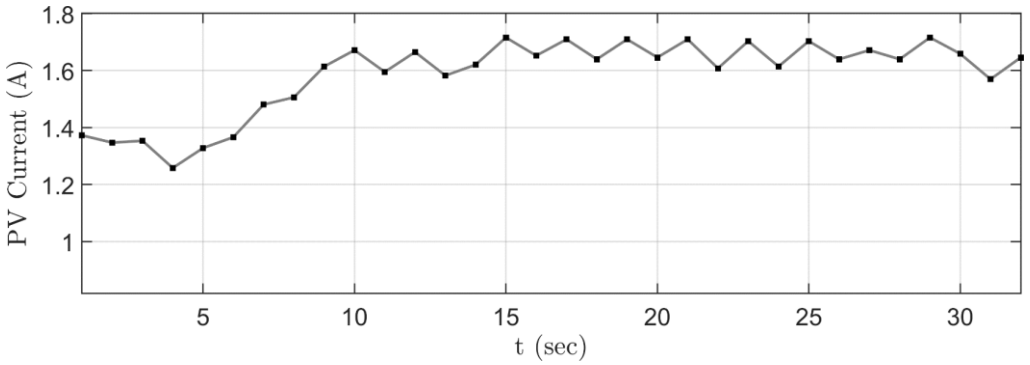

Figure 5.21: The PV array's current (shading pattern 3.20.10).

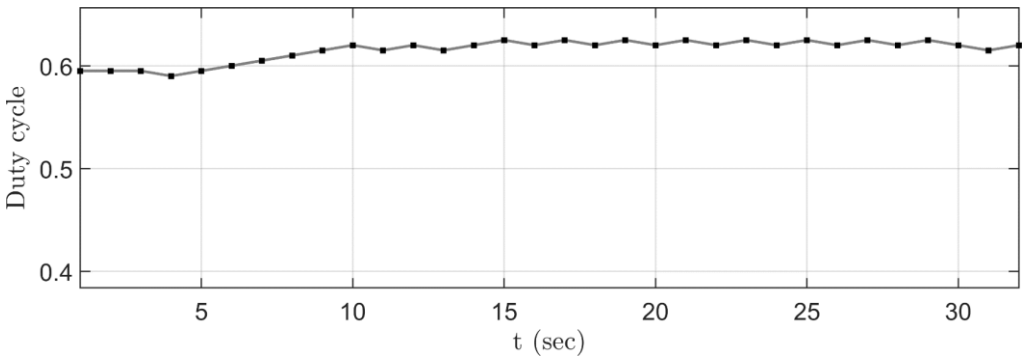

Figure 5.22: The DC/DC converter's duty cycle (shading pattern 3.20.10).

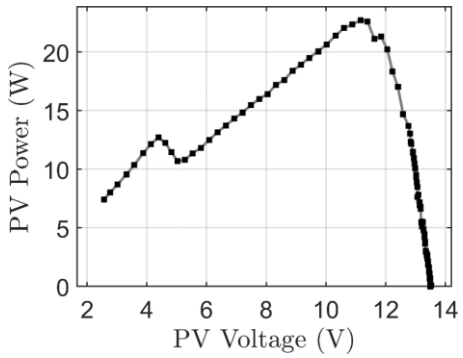

Figure 5.23: The PV array's P-V curve (shading pattern 3.20.10).

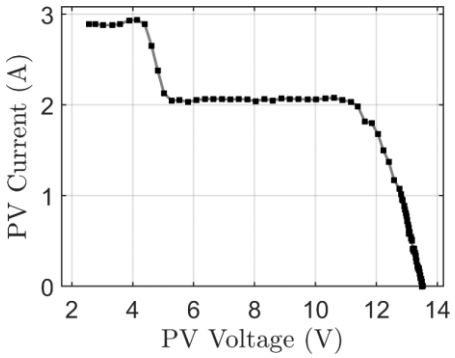

Figure 5.24: The PV array's I-V curve (shading pattern 3.20.10).

However, after the first 15 shading patterns (i.e. 3.20.1 to 3.20.15), the reference power changed from 20 W to 25 W, and consequently the quantization level on which it belongs. A different part of the state-space was going to be explored, for which the agent had not filled any cells of the Q-learning table. As a result, the agent is going to visit many states, in order to obtain knowledge for detecting the GFPP on the new reference power's quantized step. The waveforms of the agent's behavior follow in Fig. 5.25-5.30.

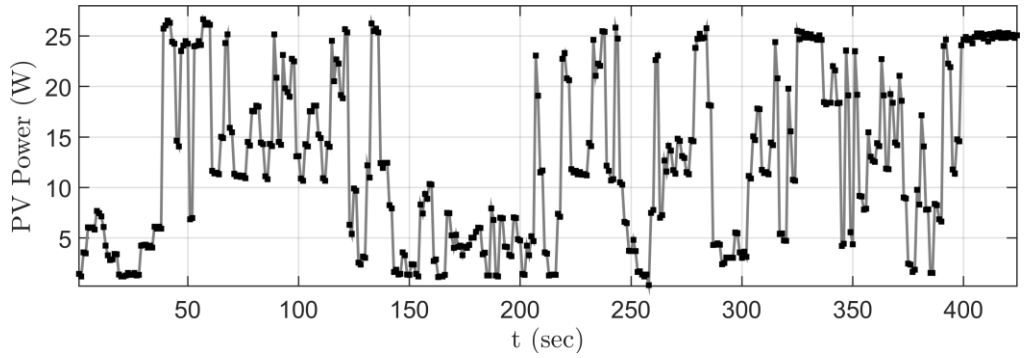

Figure 5.25: The PV array's power (shading pattern 3.25.1).

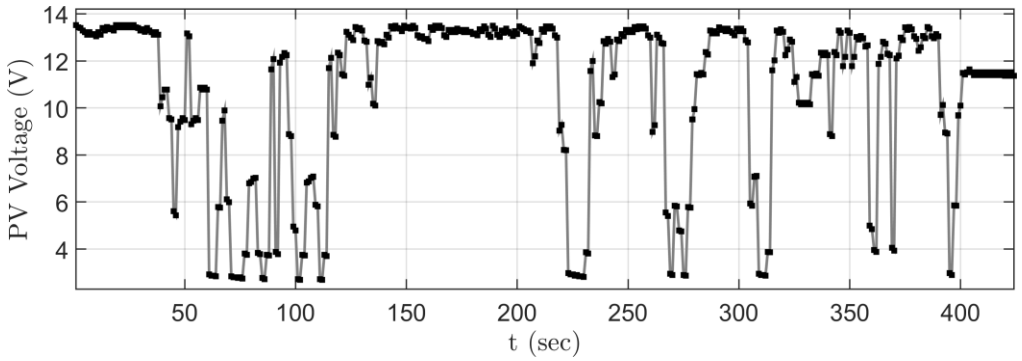

Figure 5.26: The PV array's voltage (shading pattern 3.25.1).

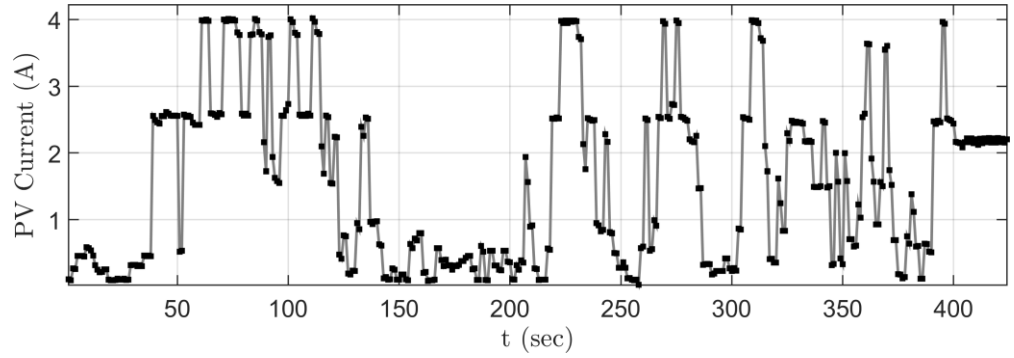

Figure 5.27: The PV array's current (shading pattern 3.25.1).

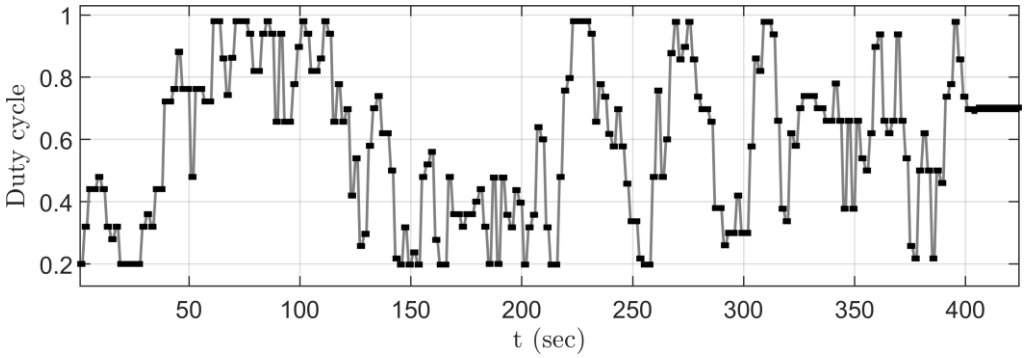

Figure 5.28: The DC/DC converter's duty cycle (shading pattern 3.25.1).

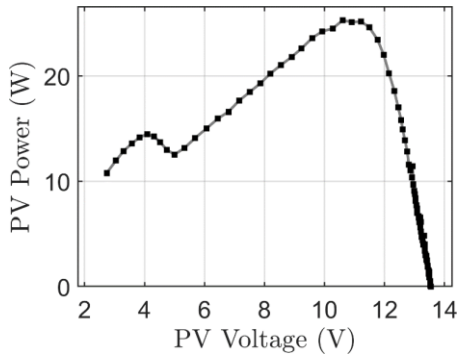

Figure 5.29: The PV array's P-V curve (shading pattern 3.25.1).

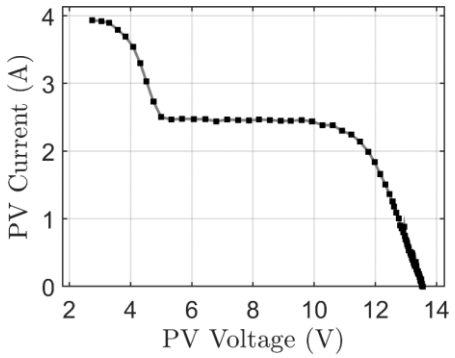

Figure 5.30: The PV array's I-V curve (shading pattern 3.25.1).

5.3. Experimental comparison of the three algorithms

On a real PV system, neither the environmental conditions, nor the reference power is going to remain stable forever. The solar irradiance can vary from minute to minute on a cloudy day and the temperature can change enough on gusts of wind. Moreover, on a grid-connected system, the reference power could be changed every minute by the grid operator, in order to satisfy the loads' power demand. Thus, in order to compare the performance of the three algorithms, four experimental tests cases were designed, on which the aforementioned quantities vary over time. The partial shading conditions were emulated by placing the two PV modules in different tilt angles, thus receiving different amounts of solar irradiation.

Test case 1: Step changes in power reference

During this experiment, the irradiance on the two PV panels is set to 1000 W/m^2 and 600 W/m^2, and the maximum available power is 26.99 W. The experimental waveforms are shown in Fig. 5.31 – 5.43. At the beginning of this test case, the three algorithms were called for a reference power of 20 W.

The first algorithm firstly measured the PV array's open-circuit voltage and then started tracking the reference power. Its mode was set to 1, in order to search for the first LMPP. After finding it, since it could not satisfy the reference power, it moved to mode 2, searching for the section divider point. Then, after moving again to mode 1, before finding the next LMPP, the reference power was tracked. The tracking needed 40 steps to be completed and the steady-state tracking error was 1.33%.

The second algorithm chose specific steps for the duty cycle that could help to reach the GMPP in a few steps. After converging to the GMPP, the PV array's output power was limited to the reference value. 28 steps were needed for the tracking process, and the steady-state tracking error was 2.51%.

The third algorithm chose specific steps for the duty cycle that could lead it exactly to the reference power point. Only 14 steps were needed for the tracking process, and the steady-state tracking error was 2.80%.

Table 5.9: Algorithms' performance overview (test case 1, t = 0 to t = 80 sec).

Test case 1 (t = 0 to t = 80 sec)			
Irradiance	1000 W/m^2 and 600 W/m^2		
Air temperature	25.6° C		
Reference power	20 W		
GMPP power	26.99 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	40	28	14
Steady-state tracking error	1.33 %	2.51 %	2.80 %

At *t* = 81 sec, the reference power changed to 30 W. The first algorithm continued to search for the second LMPP. Since it does not satisfy the reference power, it continued its search, aiming for the next SDP. When the PV array's voltage reached 90% of its open-circuit value, it jumped to mode 5 and concluded that there are no other LMPPs on the P-V curve. Thus, the operation point was set to the GMPP. A total of 38 steps were needed for this process, while the tracking error was 1.29%.

The second algorithm started fine-tuning the duty cycle, in order to track the reference power. After reaching the GMPP, it remained at that point, since it was impossible to match the reference power. 10 steps were needed for the fine-tuning, and the tracking error was 1.04%.

The third algorithm had to reset, since the new reference power value does not belong in the same quantization level with the previous value. After resetting, it tracked the reference power in just 2 steps, with a steady-state tracking error of 1.07%.

Table 5.10: Algorithms' performance overview (test case 1, t = 81 to t = 160 sec).

Test case 1 (t = 81 to t = 160 sec)			
Irradiance	1000 W/m^2 and 600 W/m^2		
Air temperature	25.6° C		
Reference power	30 W		
GMPP power	26.99 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	38	10	2
Steady-state tracking error	1.29 %	1.04 %	1.07 %

At *t* = 161 sec, the reference power dropped to 25 W. The first algorithm, since the PV array's output power is greater than the reference power, started limiting

it. After 10 steps, it converged to the desired power value, with a steady-state tracking error of 1.03%.

The second algorithm also started limiting the PV array's output power. It tracked the reference power after 7 steps, with a steady-state tracking error of 1.35%.

The third algorithm had to reset, since the new and the old reference power values did not belong to the same quantization level. Convergence was achieved after 6 steps with a steady-state tracking error of 1.60%.

Table 5.11: Algorithms' performance overview (test case 1, t = 161 to t = 240 sec).

Test case 1 (t = 161 to t = 240 sec)			
Irradiance	1000 W/m^2 and 600 W/m^2		
Air temperature	25.6° C		
Reference power	25 W		
GMPP power	26.99 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	10	7	6
Steady-state tracking error	1.03 %	1.35 %	1.60 %

This test case shows the third algorithm's advantage. By including the reference power on its state space, when it changes enough to jump to another quantization level, the algorithm's convergence time can be significantly low.

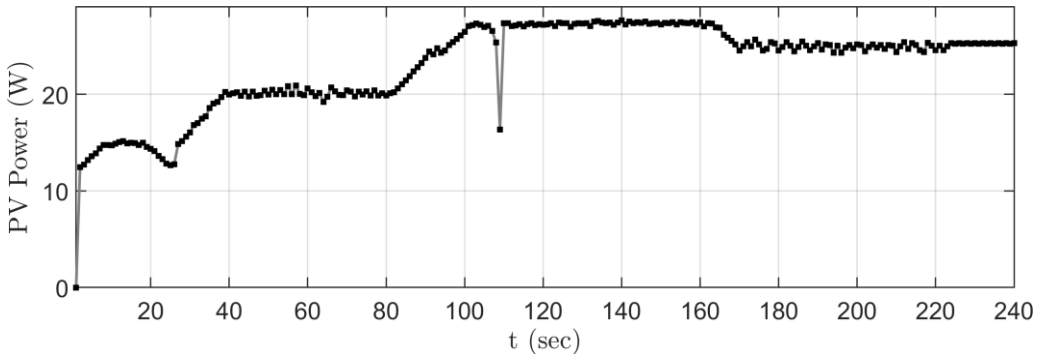

Figure 5.31: The PV array's power (test case 1, algorithm #1).

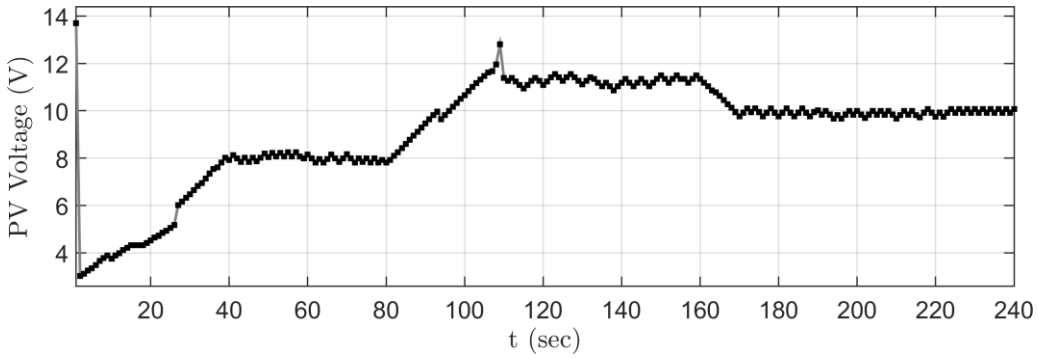

Figure 5.32: The PV array's voltage (test case 1, algorithm #1).

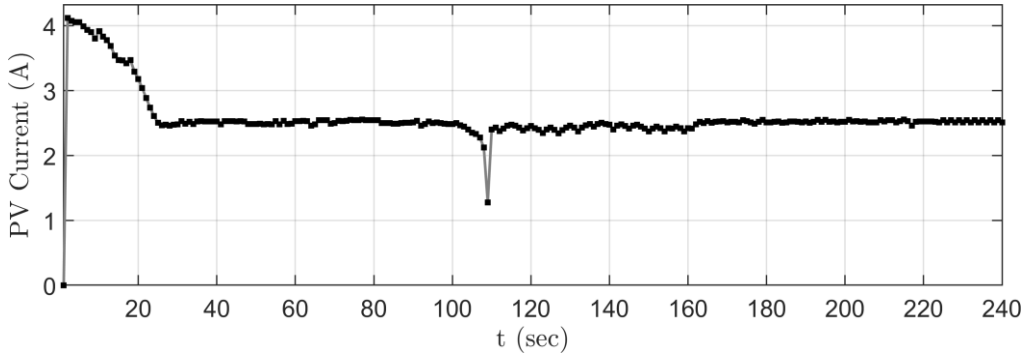

Figure 5.33: The PV array's current (test case 1, algorithm #1).

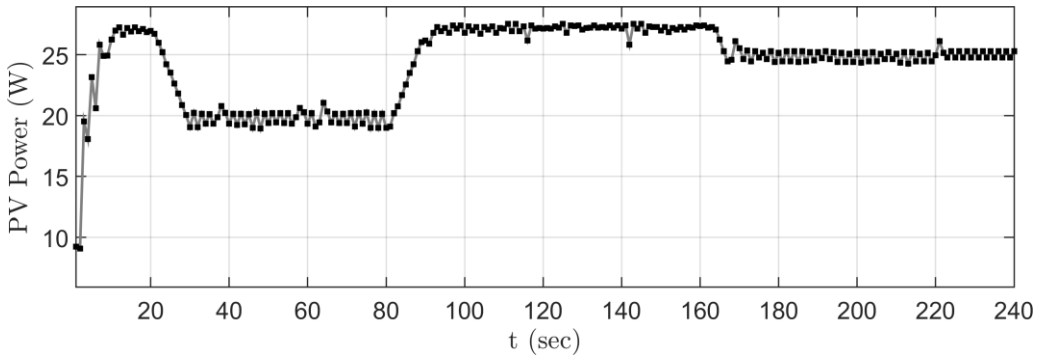

Figure 5.34: The PV array's power (test case 1, algorithm #2).

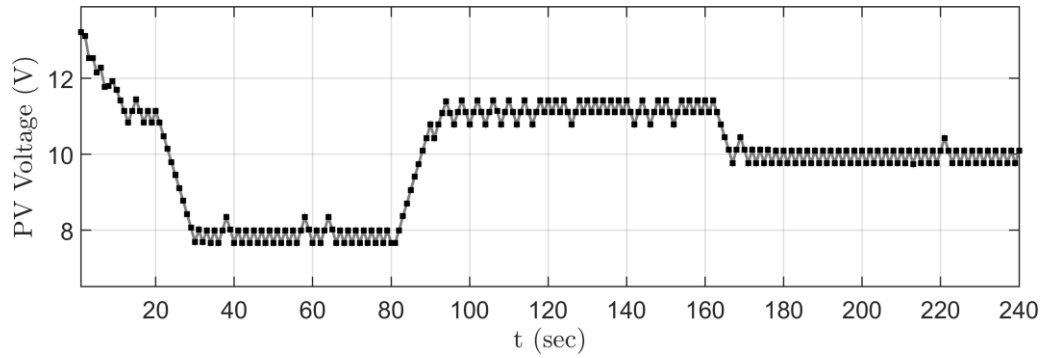

Figure 5.35: The PV array's voltage (test case 1, algorithm #2).

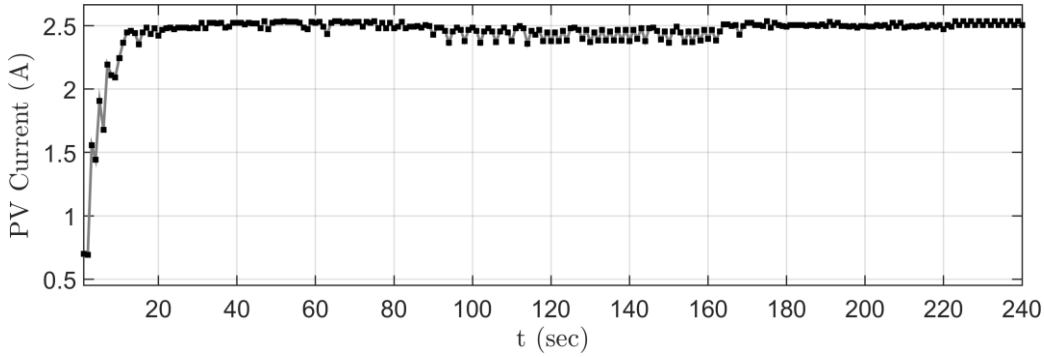

Figure 5.36: The PV array's current (test case 1, algorithm #2).

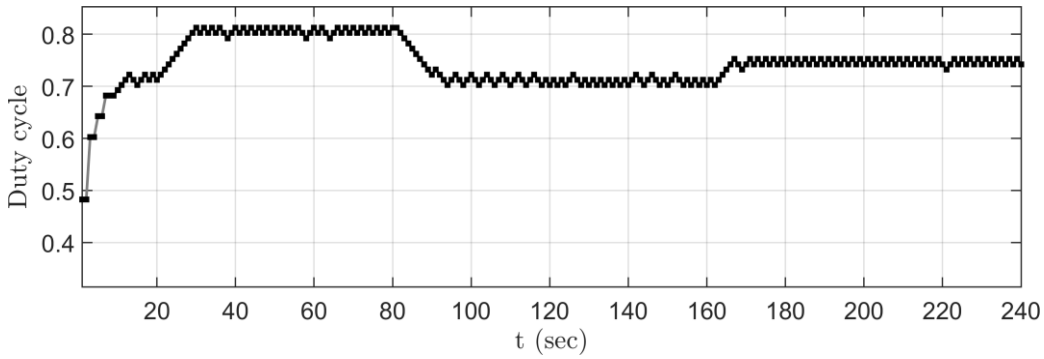

Figure 5.37: The DC/DC converter’s duty cycle (test case 1, algorithm #2).

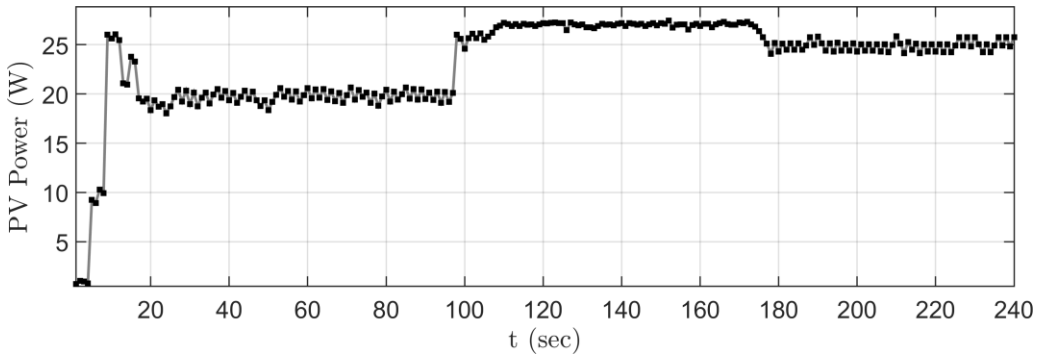

Figure 5.38: The PV array's power (test case 1, algorithm #3).

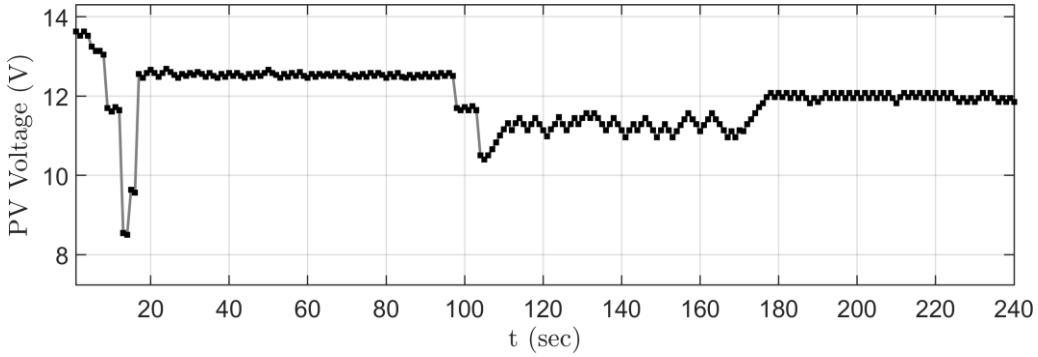

Figure 5.39: The PV array's voltage (test case 1, algorithm #3).

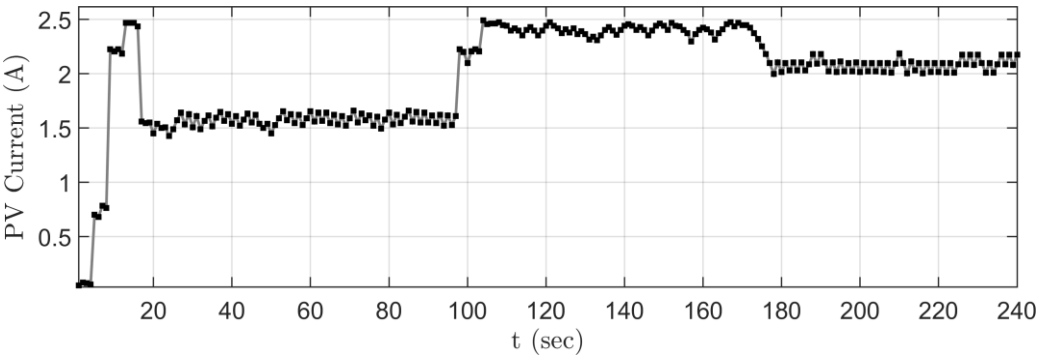

Figure 5.40: The PV array's current (test case 1, algorithm #3).

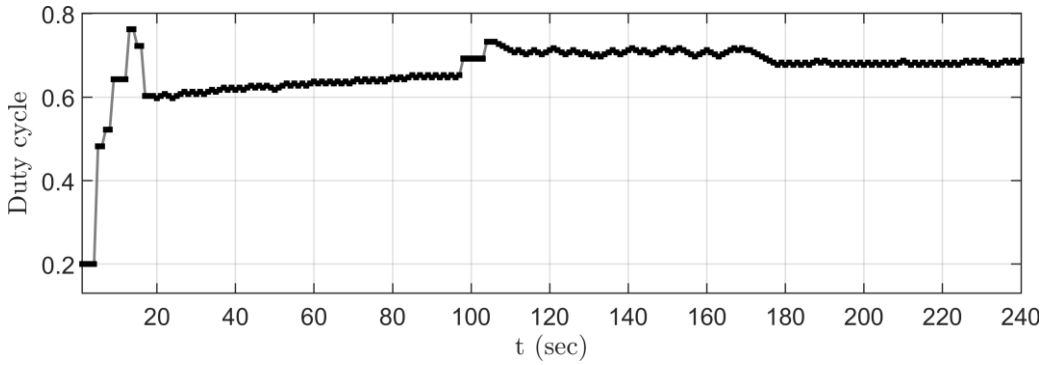

Figure 5.41: The DC/DC converter's duty cycle (test case 1, algorithm #3).

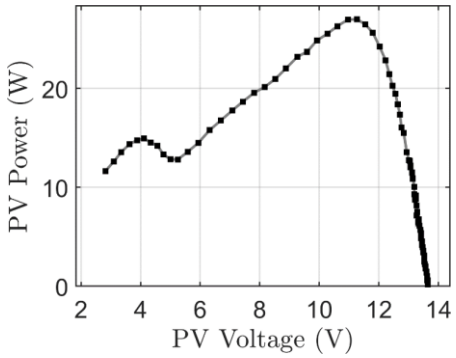

Figure 5.42: The PV array's P-V curve (test case 1).

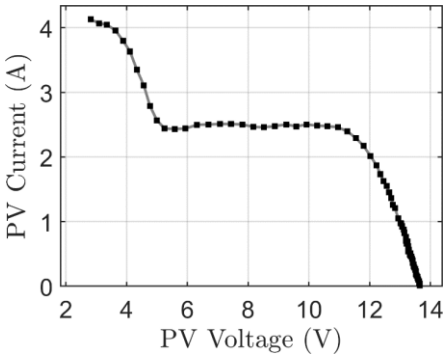

Figure 5.43: The PV array's I-V curve (test case 1).

Test case 2: Irradiance change, the new GMPP cannot satisfy the reference power

At the beginning of this experiment, the irradiance on the two PV panels is set to 900 W/m^2 and 600 W/m^2, respectively, with the maximum available power being 25.78 W. The reference power was set to 25 W for the whole test's duration. The experimental waveforms are shown in Fig. 5.44 – 5.58.

The first algorithm firstly measured the PV array's open-circuit voltage and then started tracking the reference power. Its mode was set to 1, in order to search for the first LMPP. After finding it, since it could not satisfy the reference power, it moved to mode 2, searching for the section divider point. Then, after moving again to mode 1, before finding the next LMPP, the reference power was tracked. The tracking needed 18 steps to be completed and the steady-state tracking error was 1.13%.

The second algorithm chose specific steps for the duty cycle, that could help in reaching the GMPP in a few steps. After converging to the GMPP, the PV array's output power was limited to the reference value. 28 steps were needed for the tracking process, and the steady-state tracking error was 1.53%.

The third algorithm chose specific steps for the duty cycle that could lead it exactly to the reference power point. 18 steps were needed for the tracking process, and the steady-state tracking error was 0.77%.

Table 5.12: Algorithms' performance overview (test case 2, t = 0 to t = 65 sec).

Test case 2 (t = 0 to t = 65 sec)			
Irradiance	900 W/m^2 and 600 W/m^2		
Air temperature	25.4° C		
Reference power	25 W		
GMPP power	25.78 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	18	28	18
Steady-state tracking error	1.13 %	1.53 %	0.77 %

At *t* = 66 sec, the irradiance on the two PV panels changed to 700 W/m^2 and 500 W/m^2, and the new maximum available power is 20.78 W. The first algorithm, to avoid setting the reference voltage to its minimum value and scan the whole curve, by setting its mode to 1, it searched for the next LMPP, whose power was 20.78 W. Since the reference power was not tracked, the reference voltage was inevitably set to its minimum value and the scanning of the P-V started, to find the GMPP. This process needed 43 steps to be completed and resulted to a steady-state tracking error of 1.21%.

The second algorithm after resetting tracked the GMPP in 4 steps, with a steady-state tracking error of 0.78%.

The third algorithm also had to reset, because of the environmental changes. Convergence to the GMPP was achieved after 10 steps with a steady-state tracking error of 2.65%.

Table 5.13: Algorithms' performance overview (test case 2, t = 66 to t = 130 sec).

Test case 2 (t = 66 to t = 130 sec)			
Irradiance	700 W/m^2 and 500 W/m^2		
Air temperature	25.4° C		
Reference power	25 W		
GMPP power	20.78 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	43	4	10
Steady-state tracking error	1.21 %	0.78 %	2.65 %

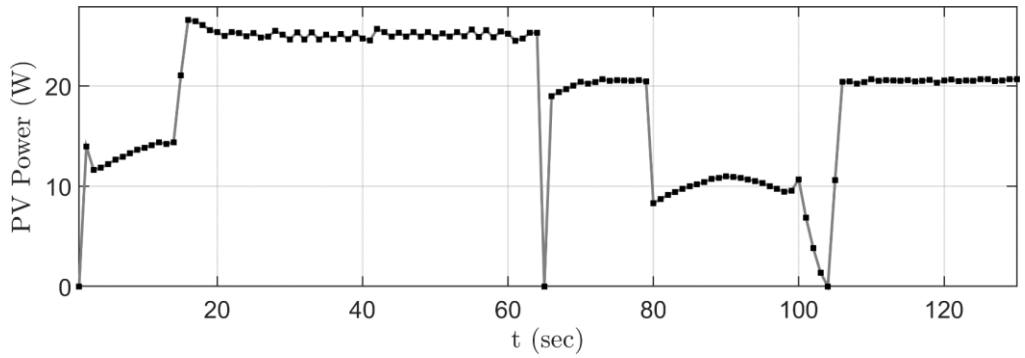

Figure 5.44: The PV array's power (test case 2, algorithm #1).

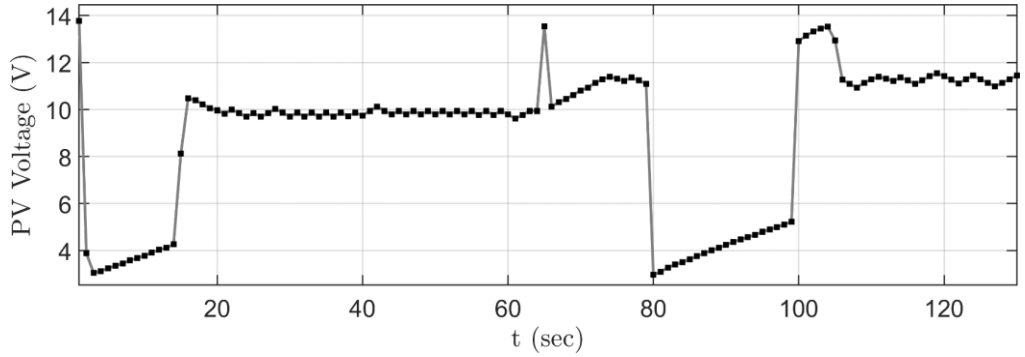

Figure 5.45: The PV array's voltage (test case 2, algorithm #1).

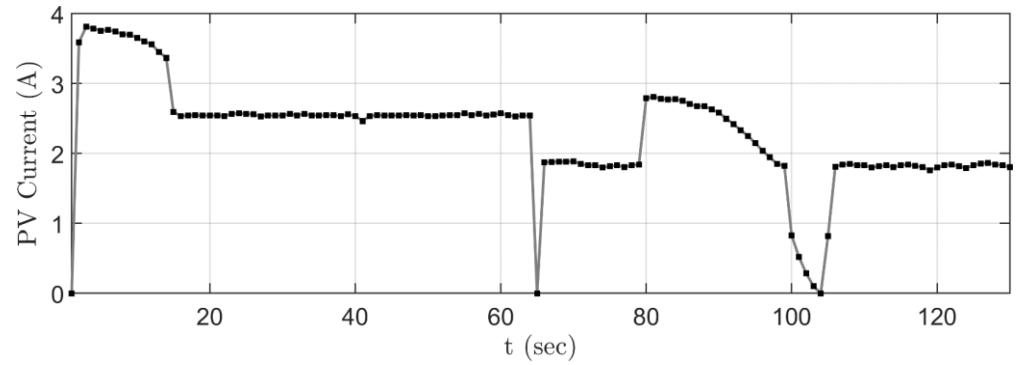

Figure 5.46: The PV array's current (test case 2, algorithm #1).

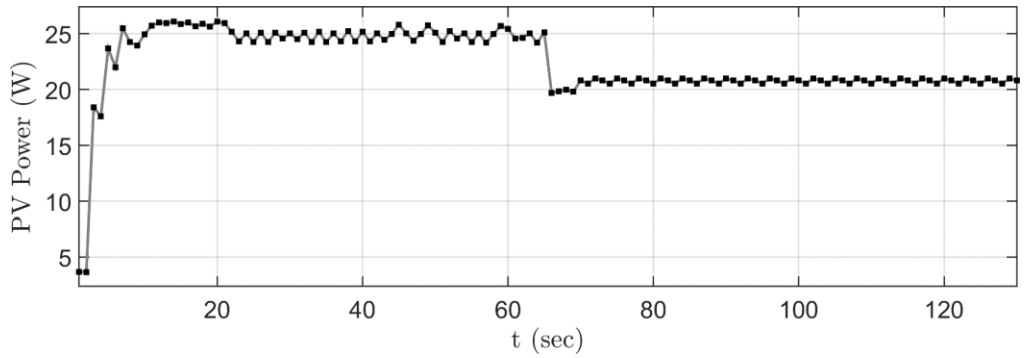

Figure 5.47: The PV array's power (test case 2, algorithm #2).

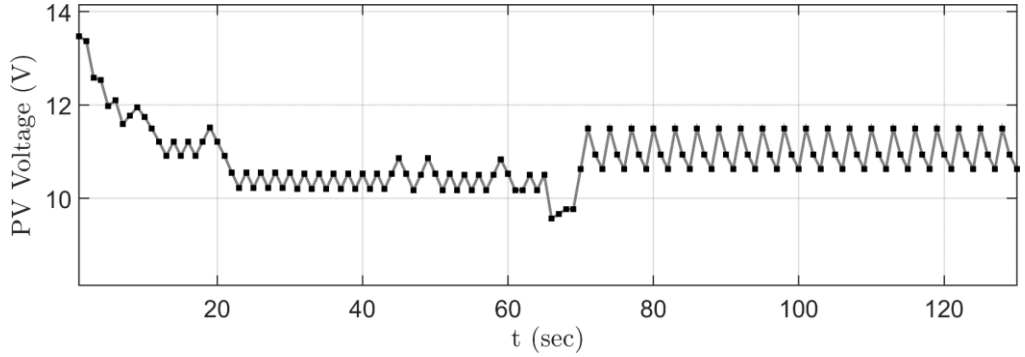

Figure 5.48: The PV array's voltage (test case 2, algorithm #2).

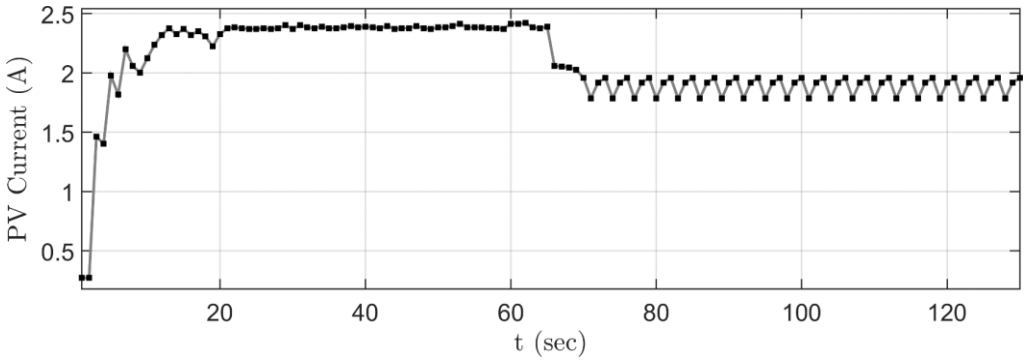

Figure 5.49: The PV array's current (test case 2, algorithm #2).

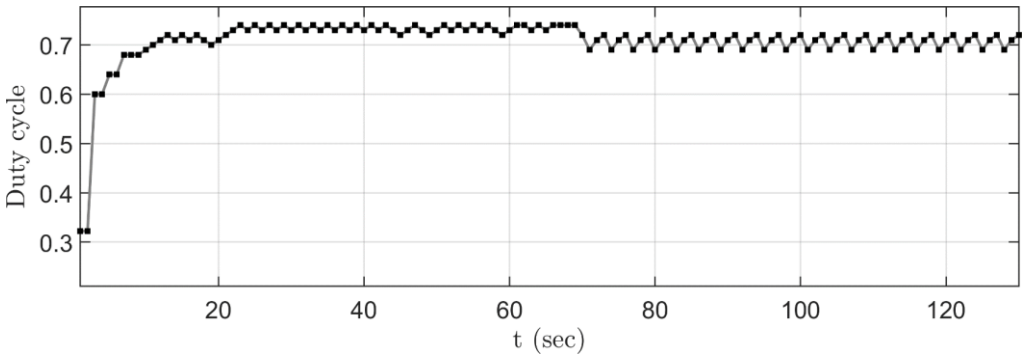

Figure 5.50: The DC/DC converter's duty cycle (test case 2, algorithm #2).

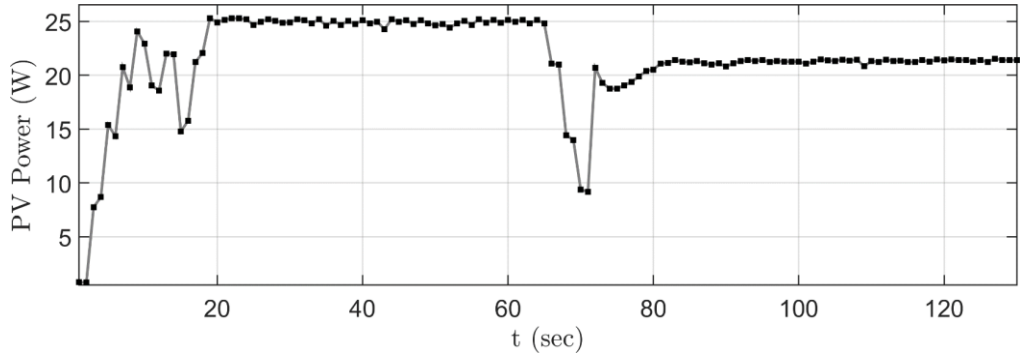

Figure 5.51: The PV array's power (test case 2, algorithm #3).

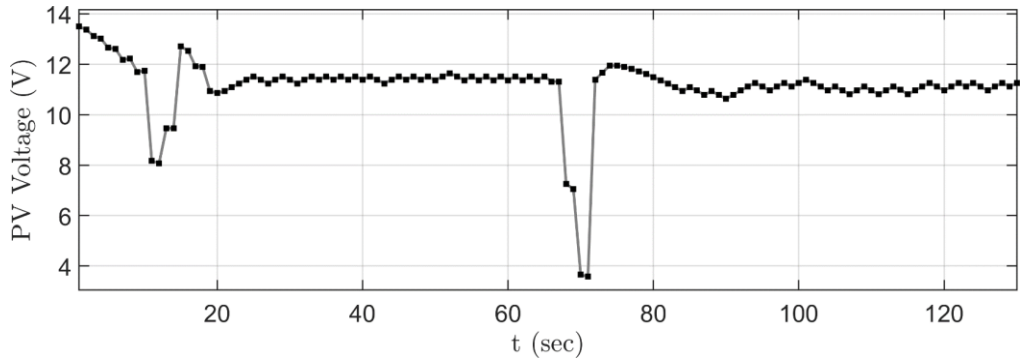

Figure 5.52: The PV array's voltage (test case 2, algorithm #3).

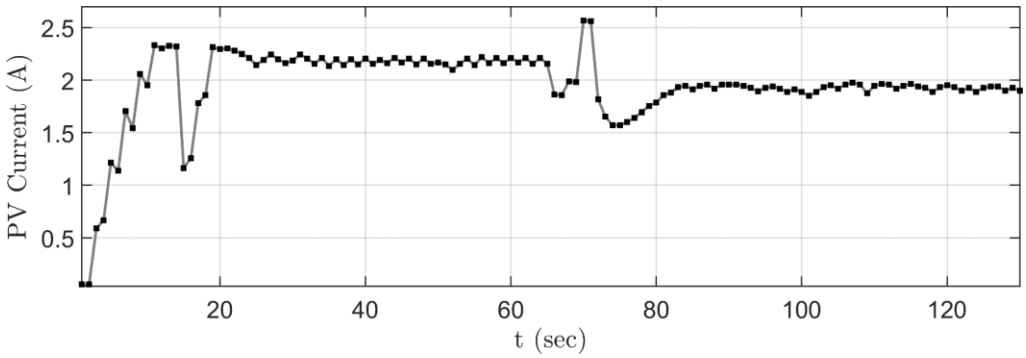

Figure 5.53: The PV array's current (test case 2, algorithm #3).

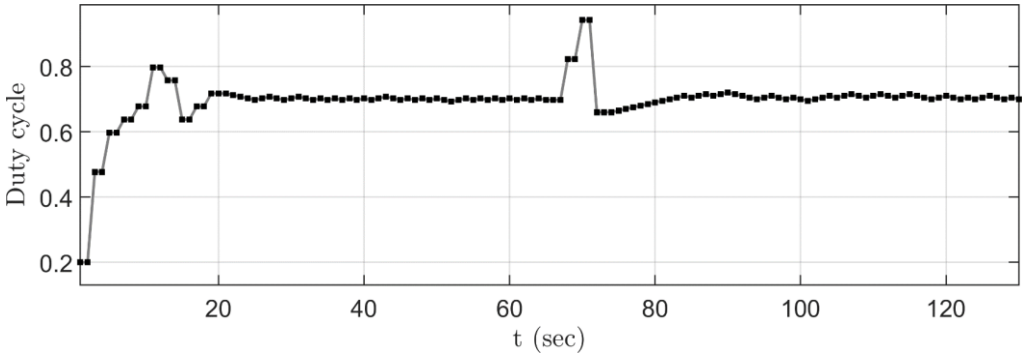

Figure 5.54: The DC/DC converter's duty cycle (test case 2, algorithm #3).

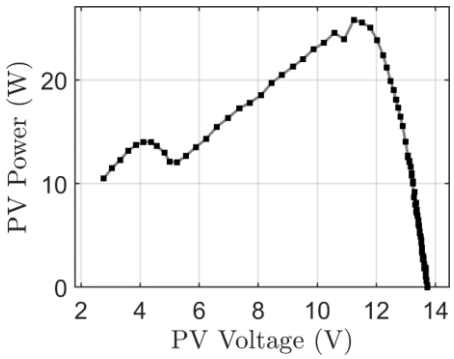

Figure 5.55: The PV array's P-V curve (test case 2, t = 0 to t = 65 sec).

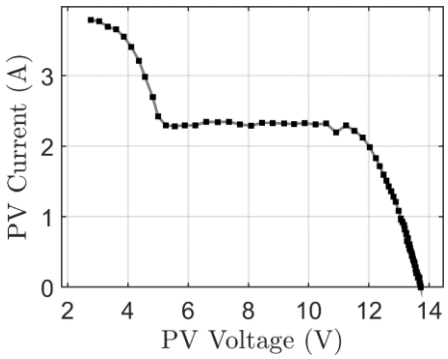

Figure 5.56: The PV array's I-V curve (test case 2, t = 0 to t = 65 sec).

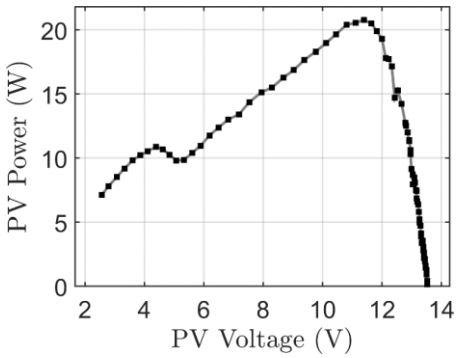

Figure 5.57: The PV array's P-V curve (test case 2, t = 66 to t = 130 sec).

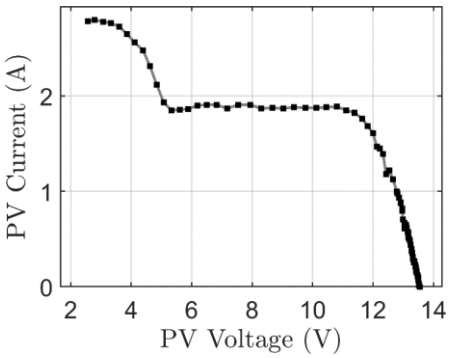

Figure 5.58: The PV array's I-V curve (test case 2, t = 66 to t = 130 sec).

Test case 3: Irradiance change, the new GMPP can satisfy the reference power

At the beginning of this experiment, the irradiance on the two PV panels is set to 900 W/m^2 and 400 W/m^2, respectively, with the maximum available power being 16.86 W. The reference power was set to 20 W for the whole test's duration. The experimental waveforms are shown in Fig. 5.59 – 5.73.

The first algorithm firstly measured the PV array's open-circuit voltage and then started tracking the reference power. Its mode was set to 1, in order to search for the first LMPP. After finding it, since it could not satisfy the reference power, it moved to mode 2, searching for the section divider point. Then, after moving again to mode 1 and finding the next LMPP, it jumped again to mode 2, in order to track the next SPD, since the LMPP did not satisfy the power reference. When the reference voltage reached 90% of the PV array's open-circuit value, it was concluded that the power reference could not be satisfied, thus the operation point was set to the GMPP. The tracking process needed 22 steps to be completed and the steady-state tracking error was 3.07%.

The second algorithm chose specific steps for the duty cycle, that could help him reach the GMPP in a few steps. After converging to the GMPP, it stayed there, since the power reference had not been satisfied. 10 steps were needed for the tracking process, and the steady-state tracking error was 1.57%.

The third algorithm needed only 8 steps to track the GMPP, with a steady-state tracking error of 1.30%.

Table 5.14: Algorithms' performance overview (test case 3, t = 0 to t = 65 sec).

Test case 3 (t = 0 to t = 65 sec)			
Irradiance	900 W/m^2 and 400 W/m^2		
Air temperature	25.4° C		
Reference power	20 W		
GMPP power	16.86 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	22	10	8
Steady-state tracking error	3.07 %	1.57 %	1.30 %

At t = 66 sec, the solar irradiance on the two PV panels changed to 850 W/m^2 and 550 W/m^2, respectively and the new maximum available power is 23.87 W. The first algorithm, to avoid setting the reference voltage to its minimum value and scan the whole curve, by setting its mode to 1, it searched for the next LMPP, whose power was 23.87 W. Since the reference power was tracked, the algorithm reduced significantly its convergence steps. This process needed 11 steps to be completed and resulted to a steady-state tracking error of 0.79%.

The second algorithm after resetting tracked the GMPP and limited the PV array's power to the reference value. This was achieved in 20 steps with a steady-state tracking error of 2.35%.

The third algorithm also had to reset, because of the environmental changes. Convergence to the GFPP was achieved after only 8 steps with a steady-state tracking error of 2.05%.

It can be derived from Table 5.15, that the last two algorithms showed a higher steady-state tracking error, compared to algorithm #1. Based on this thesis' requirements, these values are considered acceptable. However, they can be improved by decreasing the duty cycle step during the fine-tuning process. This could be implemented on a real system, where the grid operator would have defined an upper bound for the tracking error.

Table 5.15: Algorithms' performance overview (test case 3, t = 66 to t = 130 sec).

Test case 3 (t = 66 to t = 130 sec)			
Irradiance	850 W/m^2 and 550 W/m^2		
Air temperature	25.4° C		
Reference power	20 W		
GMPP power	23.87 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	11	20	8
Steady-state tracking error	0.79 %	2.35 %	2.05 %

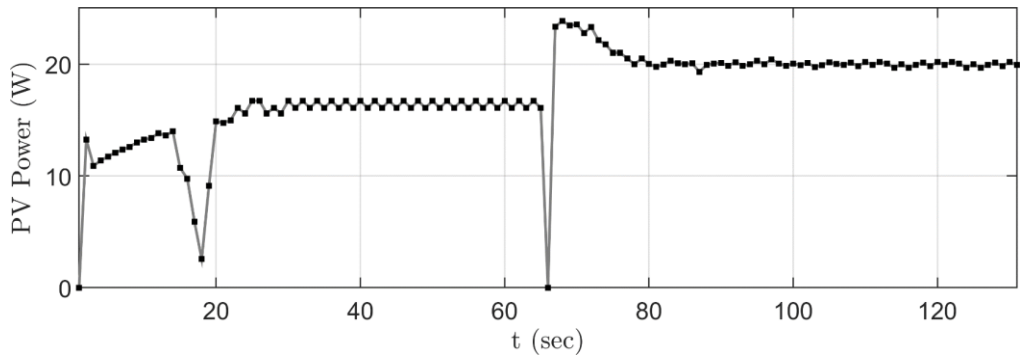

Figure 5.59: The PV array's power (test case 3, algorithm #1).

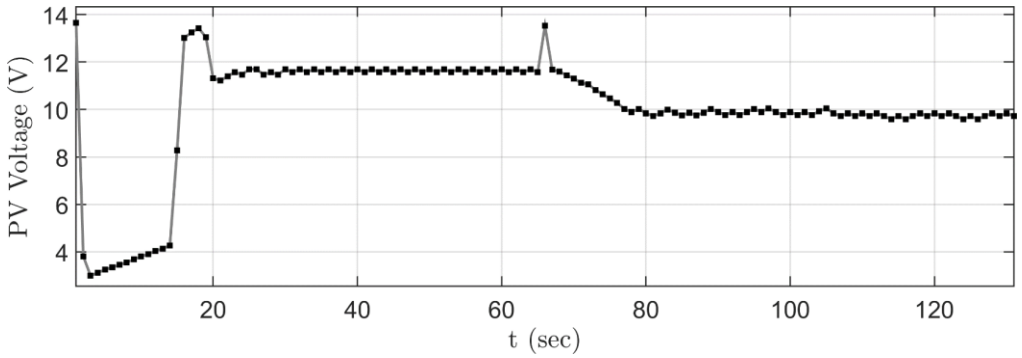

Figure 5.60: The PV array's voltage (test case 3, algorithm #1).

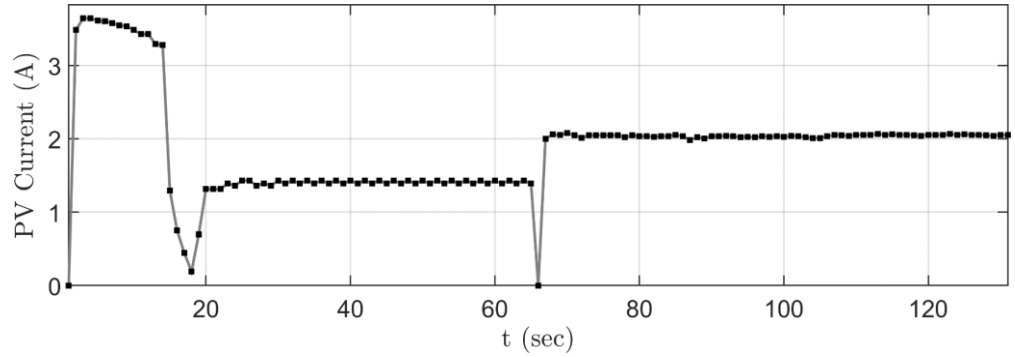

Figure 5.61: The PV array's current (test case 3, algorithm #1).

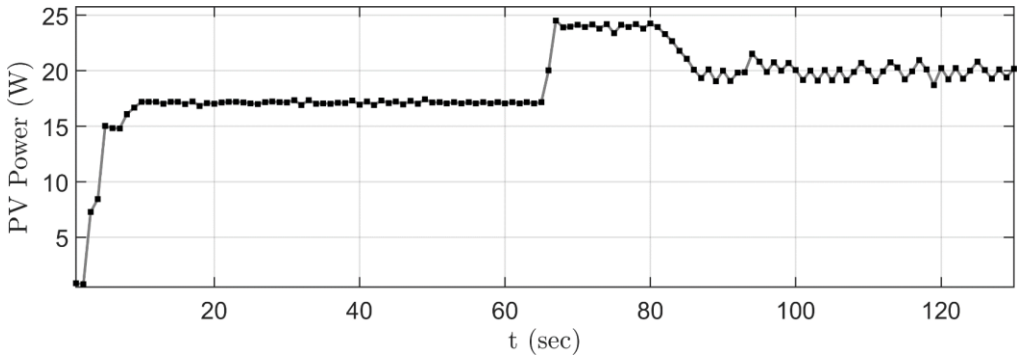

Figure 5.62: The PV array's power (test case 3, algorithm #2).

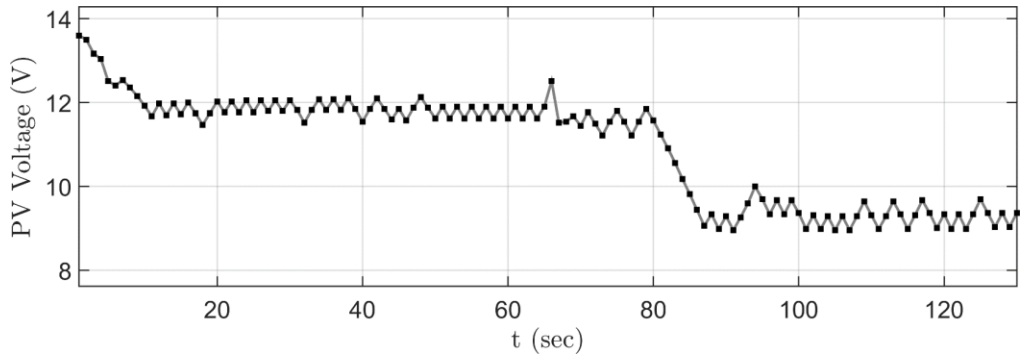

Figure 5.63: The PV array's voltage (test case 3, algorithm #2).

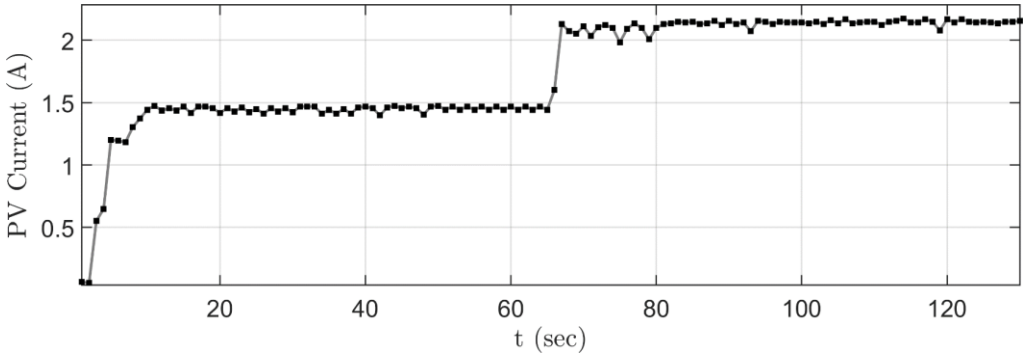

Figure 5.64: The PV array's current (test case 3, algorithm #2).

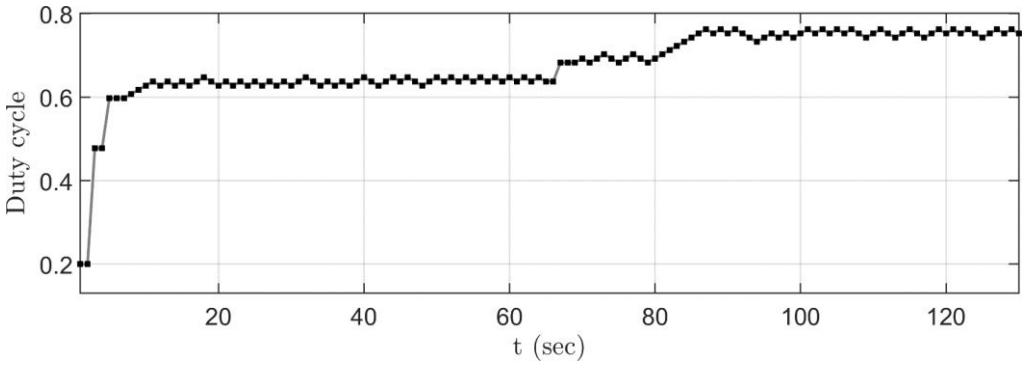

Figure 5.65: The DC/DC converter’s duty cycle (test case 3, algorithm #2).

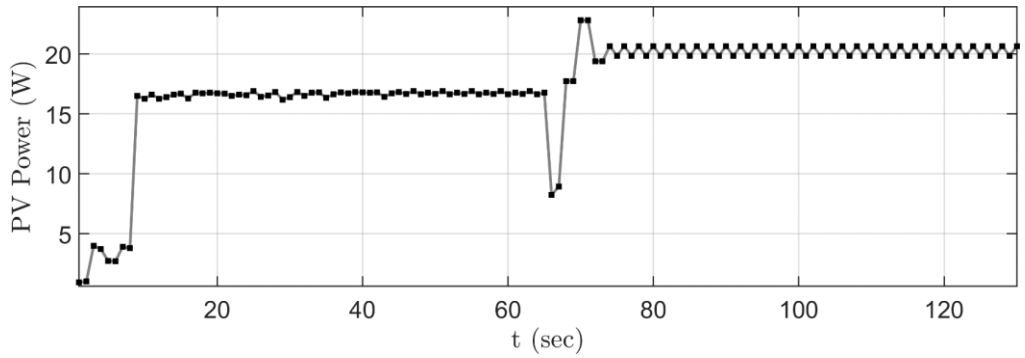

Figure 5.66: The PV array's power (test case 3, algorithm #3).

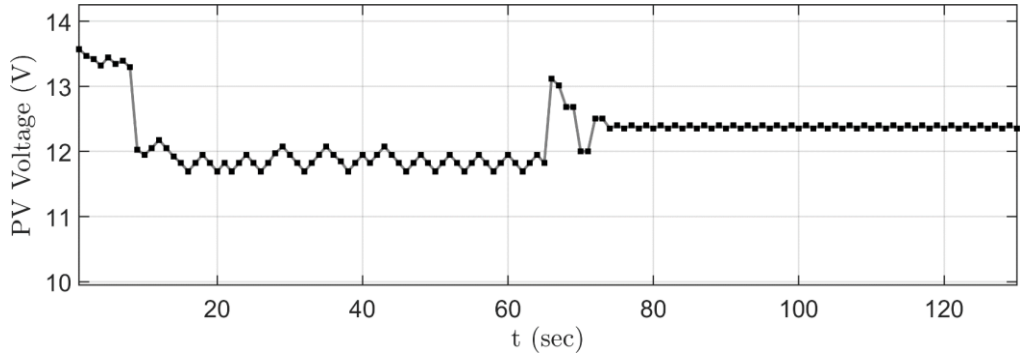

Figure 5.67: The PV array's voltage (test case 3, algorithm #3).

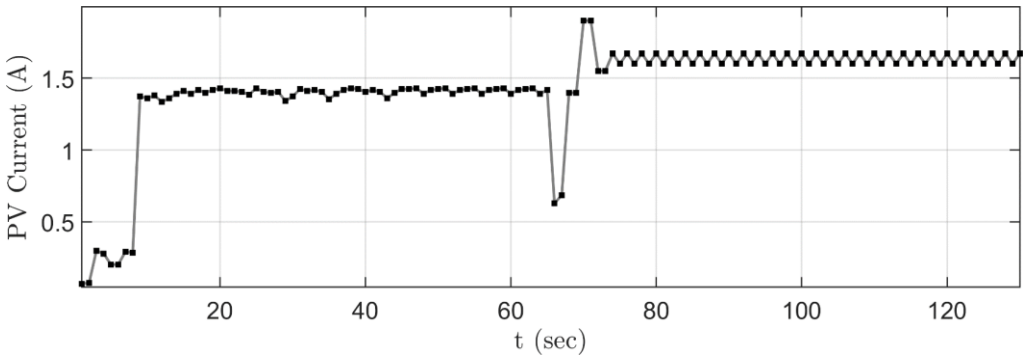

Figure 5.68: The PV array's current (test case 3, algorithm #3).

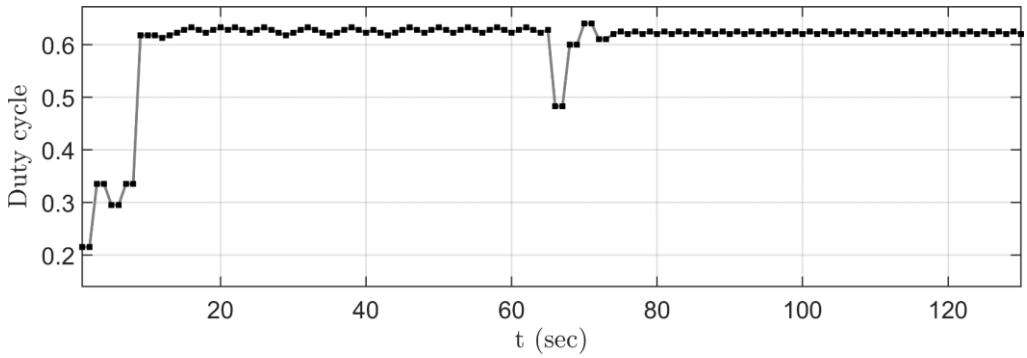

Figure 5.69: The DC/DC converter's duty cycle (test case 3, algorithm #3).

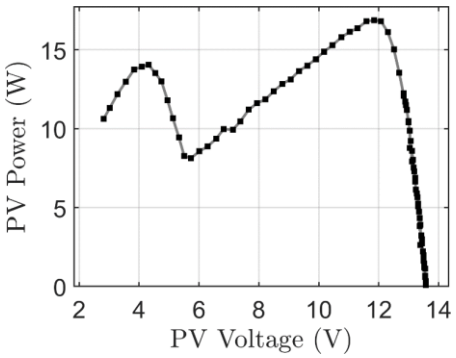

Figure 5.70: The PV array's P-V curve (test case 3, t = 0 to t = 65 sec).

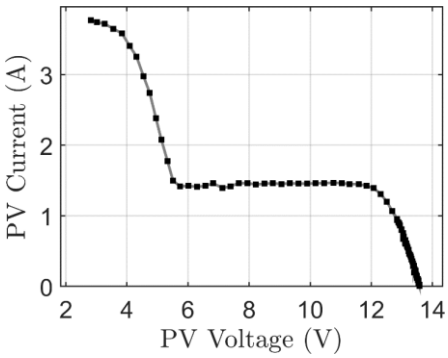

Figure 5.71: The PV array's I-V curve (test case 3, t = 0 to t = 65 sec).

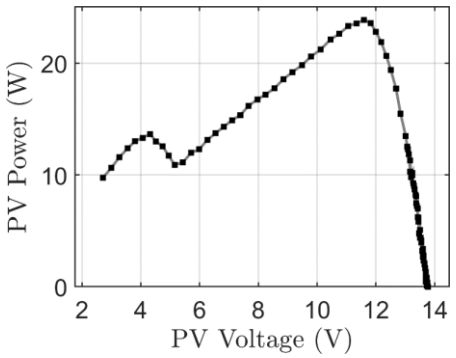

Figure 5.72: The PV array's P-V curve (test case 2, t = 66 to t = 130 sec).

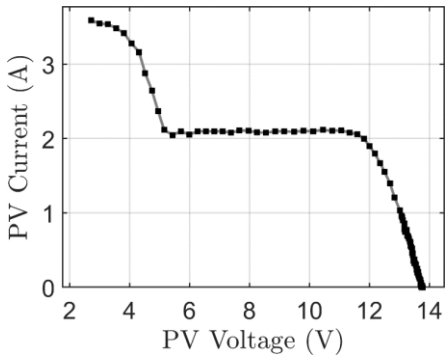

Figure 5.73: The PV array's I-V curve (test case 3, t = 65 to t = 130 sec).

Test case 4: Irradiance change, the GMPP moves from the left to the right region of the P-V curve

At the beginning of this experiment, the irradiance on the two PV panels is set to 250 W/m^2 and 900 W/m^2, with the maximum available power being 12.83 W. The reference power was set to 12 W for the whole test's duration. The experimental waveforms are shown in Fig 5.74 – 5.88.

The first algorithm firstly measured the PV array's open-circuit voltage and then started tracking the reference power. Its mode was set to 1, in order to search for the first LMPP. Before reaching it, the reference power was tracked in 5 steps, with a steady-state tracking error of 1.26%. The extremely low number of steps can be explained because the GFPP was found at the leftmost part of the P-V curve.

The second algorithm had to track the GMPP and then limited the output power to the reference value. This process required 38 steps to be completed, with a steady-state tracking error of 3.25%.

The third algorithm achieved tracking the reference power after 10 steps, with a steady-state tracking error of 1.82%.

Table 5.16: Algorithms' performance overview (test case 4, t = 0 to t = 50 sec).

Test case 4 (t = 0 to t = 50 sec)			
Irradiance	250 W/m^2 and 900 W/m^2		
Air temperature	27.5° C		
Reference power	12 W		
GMPP power	12.83 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	5	38	10
Steady-state tracking error	1.21 %	3.25 %	1.82 %

At *t* = 51 sec, the solar irradiance on the two PV panels changed to 500 W/m^2 and 700 W/m^2, respectively and the new maximum available power is 19.55 W. The first algorithm, to avoid setting the reference voltage to its minimum value and scan the whole curve, by setting its mode to 1, it searched for the next LMPP, whose power was 9.95 W. Since it did not satisfy the power reference, the reference voltage was set to its minimum value, to start scanning the PV array's P-V curve. After finding the first LMPP (on mode 1) and the SPD (on mode 2), it started searching for the next LMPP (on mode 1). Before finding it, the reference voltage was tracked, after a total of 38 steps, with a steady-state tracking error of 1.55%.

The second algorithm tracked the GMPP and then reduced the PV array's power to match the reference value. This process needed 32 steps to be completed and resulted to a steady-state tracking error of 2.50%.

The third algorithm tracked the reference power after only 12 steps, with a steady-state tracking error of 2.97%.

Table 5.17: Algorithms' performance overview (test case 4, t = 51 to t = 100 sec).

Test case 4 (t = 51 to t = 100 sec)			
Irradiance	500 W/m^2 and 700 W/m^2		
Air temperature	27.5° C		
Reference power	12 W		
GMPP power	19.55 W		
	Algorithm #1	**Algorithm #2**	**Algorithm #3**
Number of convergence steps	38	32	12
Steady-state tracking error	1.55 %	2.50 %	2.97 %

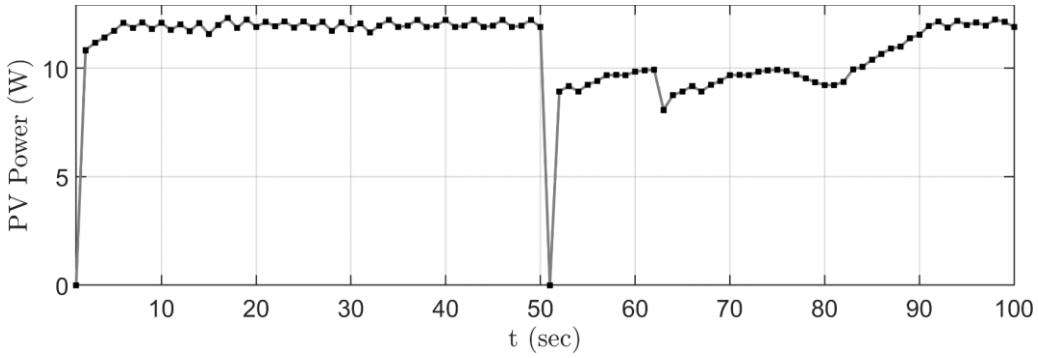

Figure 5.74: The PV array's power (test case 4, algorithm #1).

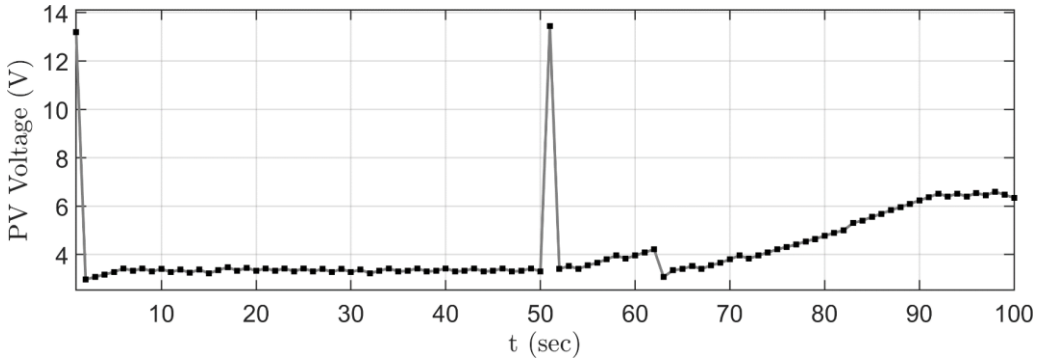

Figure 5.75: The PV array's voltage (test case 4, algorithm #1).

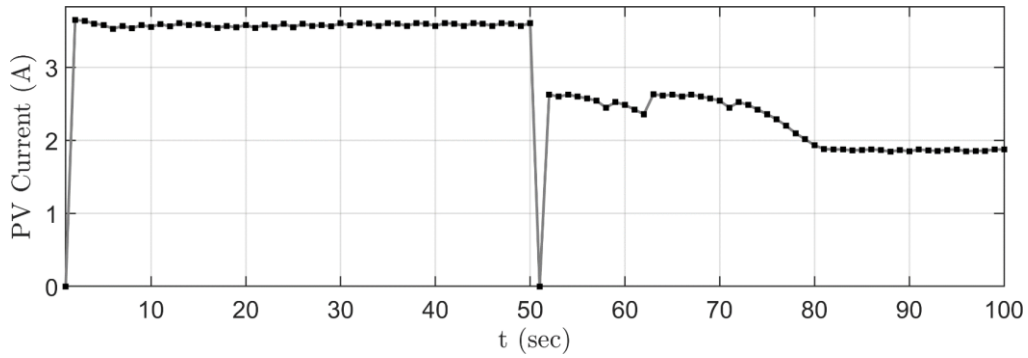

Figure 5.76: The PV array's current (test case 4, algorithm #1).

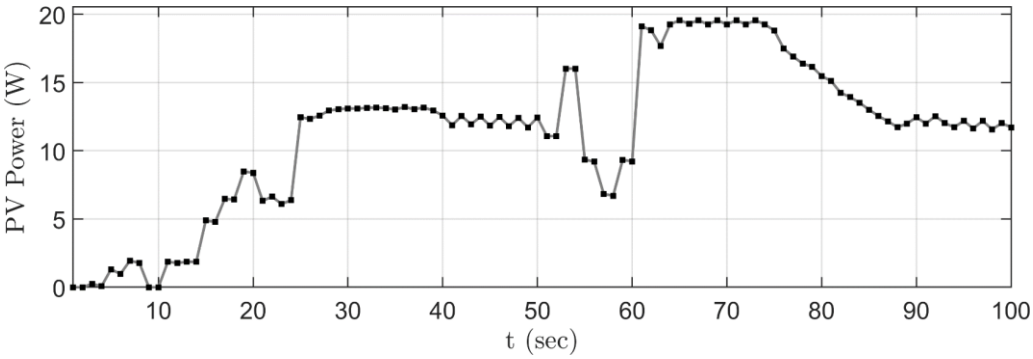

Figure 5.77: The PV array's power (test case 4, algorithm #2).

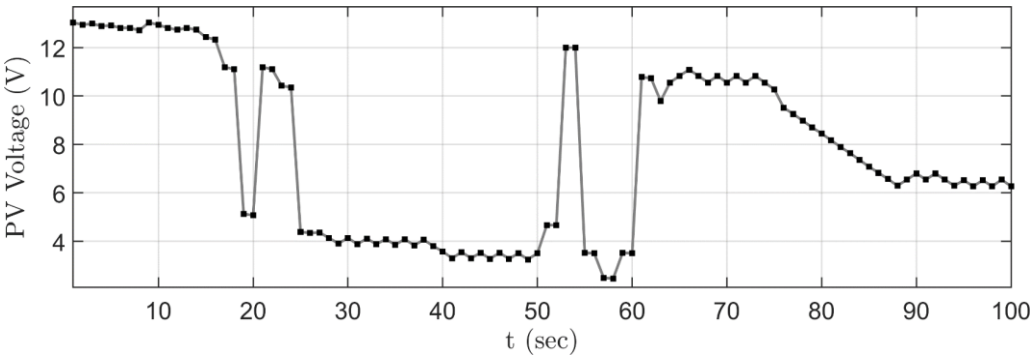

Figure 5.78: The PV array's voltage (test case 4, algorithm #2).

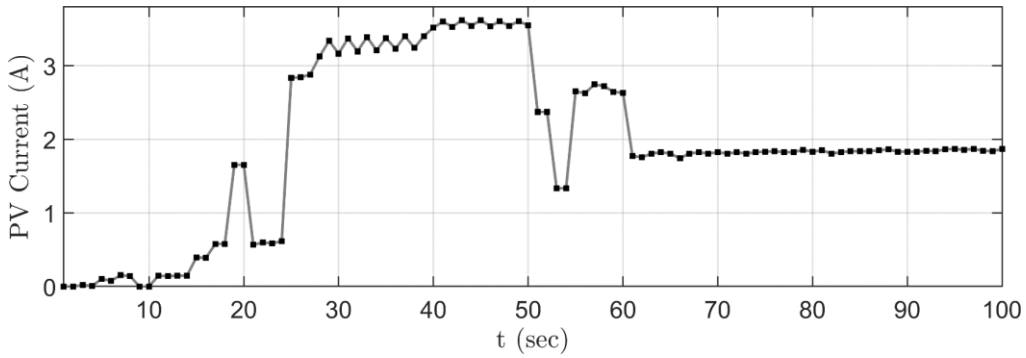

Figure 5.79: The PV array's current (test case 4, algorithm #2).

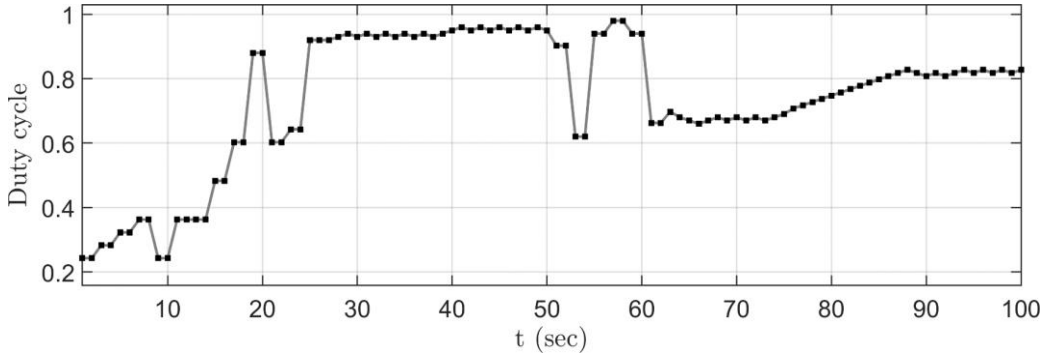

Figure 5.80: The DC/DC converter's duty cycle (test case 4, algorithm #2).

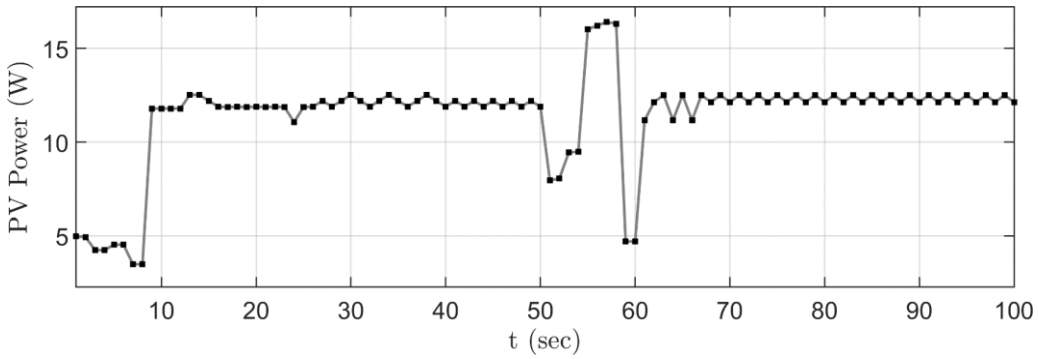

Figure 5.81: The PV array's power (test case 4, algorithm #3).

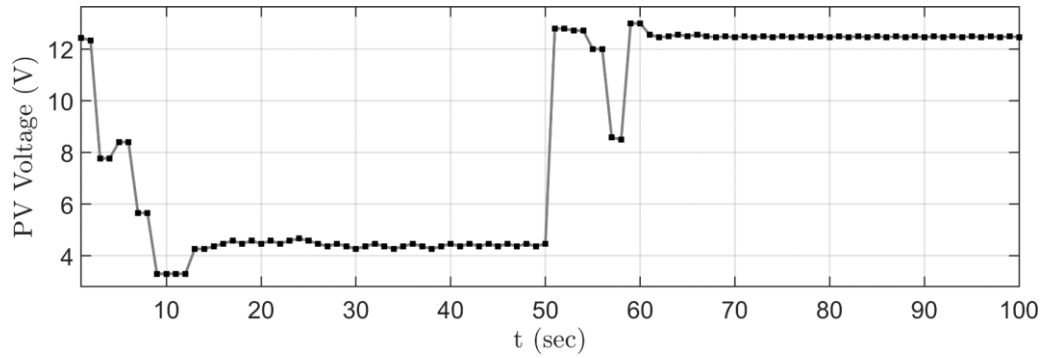

Figure 5.82: The PV array's voltage (test case 4, algorithm #3).

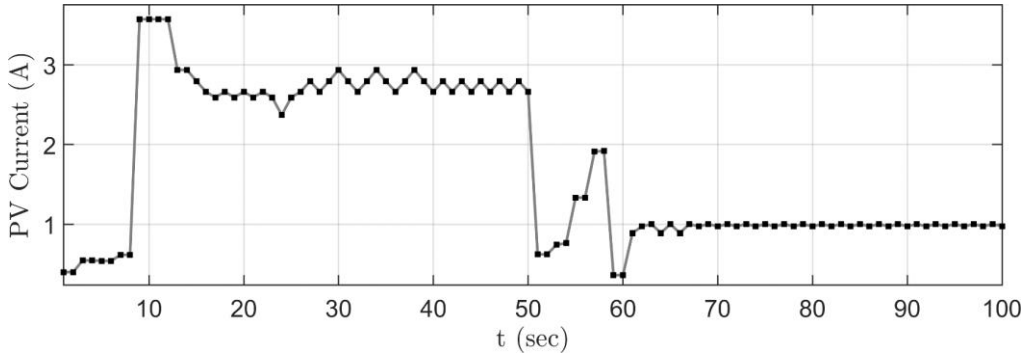

Figure 5.83: The PV array's current (test case 4, algorithm #3).

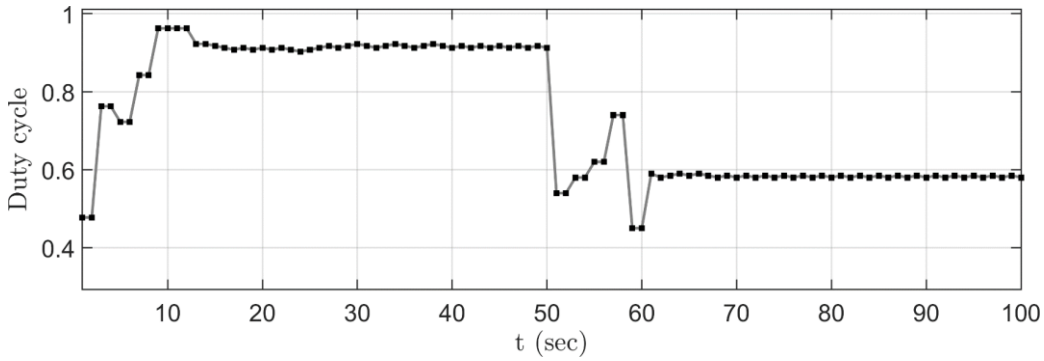

Figure 5.84: The DC/DC converter's duty cycle (test case 4, algorithm #3).

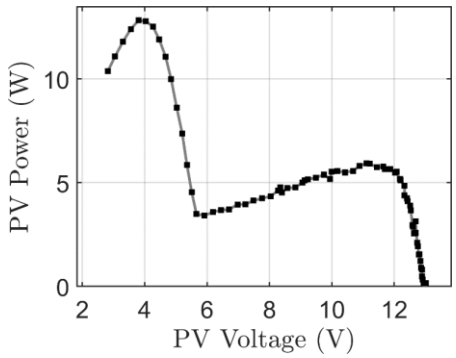

Figure 5.85: The PV array's P-V curve (test case 4, t = 0 to t = 50 sec).

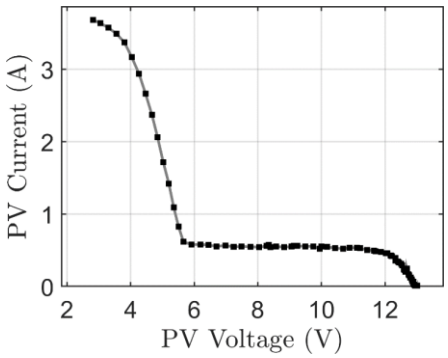

Figure 5.86: The PV array's I-V curve (test case 4, t = 0 to t = 50 sec).

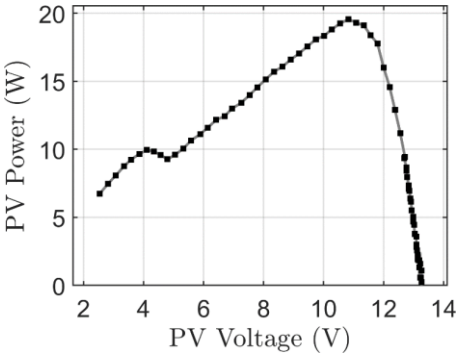

Figure 5.87: The PV array's P-V curve (test case 4, t = 51 to t = 100 sec).

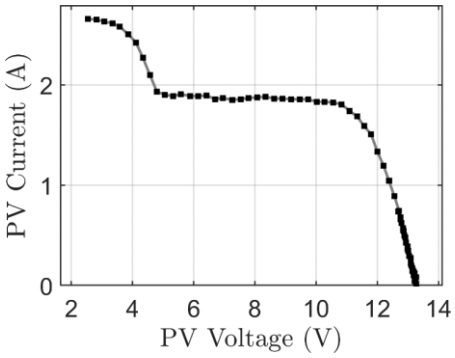

Figure 5.88: The PV array's I-V curve (test case 4, t = 51 to t = 100 sec).

Conclusions drawn from the experiments

After completing the experiments, the following conclusions can be drawn by observing their results:

1. After appropriate training, the two last algorithms can significantly reduce their convergence time. On the contrary, the first algorithm needs a specific amount of time, even if it has been called in the past for exactly the same shading pattern, with exactly the same reference power.
2. After a change of the reference power, the first and the second algorithm fine tune the PV array's output power. If this change was high, a lot of time may be needed for this process. On the other hand, the third algorithm will reset its operation and converge in a few steps.
3. After a change of the irradiance:
   - the first algorithm will converge in a few steps only if the reference power can be found either near the corresponding LMPP when it avoids resetting the reference voltage to its minimum value, or at the leftmost part of the

P-V curve when the reference voltage had to be reset to its minimum value.

- the second algorithm will converge in a few steps, only if the reference power is near the new GMPP value, otherwise it will spend a lot of time in fine-tuning.
- the third algorithm will converge after a small number of steps, for whichever value of reference power.

4. Since the third algorithm converges to the GFPP which has the highest voltage value, the PV array's current has a lower value, compared to the first two algorithms. Consequently, the power losses will be smaller.

6. Conclusions

In this thesis, two novel algorithms to control the output power of PV arrays under partial shading (a problem known as GFPPT) were presented. Both of them were based on a machine learning technique, called Q-learning, to achieve low convergence time. In order to evaluate their performance, an experimental system composed of two PV modules connected in series, a DC/DC boost converter, a battery bank and the required components, was deployed. Those two algorithms, and one more from the bibliography were tested on four test cases, in order to compare their performance.

According to the experimental results, it is established that the fact that the two new algorithms obtain knowledge about their actions over time significantly reduces their convergence time. In particular, the former, which tracks the GMPP and then fine-tunes the output power shows a great performance at tracking the GMPP and then spends some time to limit the output power to the desired level. On the contrary, the latter, which tracks the reference power, avoids the output power limiting process. Consequently, it exhibits an improved tracking performance. It was observed that the convergence time of the last algorithm can be even less than a quarter of the one's found in the bibliography. Additionally, its performance is not affected by the relative position of the GMPP and the GFPP on the power-voltage characteristic curve of the PV array, in contrast to the other two algorithms.

As a future extension of this thesis, the proposed algorithms' performance could be implemented on a grid-connected PV system, by adding a DC/AC inverter to the PV system, and a power management system which automatically calculates the reference power value, instead of entering it manually via a computer. Another interesting modification would be to develop a way of compressing the Q-learning tables' data. This would give the opportunity to execute the algorithms on microcontrollers with low internal memory size, without the need of an external memory module. Finally, the GFPPT problem could be approached by other types of algorithms, such as the Particle Swarm Optimization (PSO) algorithm, in order to determine whether the reference power can be tracked more efficiently or not.

References

[1] International Renewable Energy Agency (IRENA), Renewable Energy Capacity 2023, 2023.

[2] V.K. Papadias, K. Vournas and K. Ntelkis, Electric Power Generation – Power System Control and Stability, 1st ed., Symmetria Publications, 2011.

[3] Y. Yang, K.A.Kim, F. Blaabjerg and A. Sangwongwanich, Advances in Grid-Connected Photovoltaic Power Conversion Systems, 1st ed., Woodhead Publishing, 2018.

[4] H. D. Tafti, Q. Wang, C. D. Townsend, J. Pou and G. Konstantinou, "Global Flexible Power Point Tracking in Photovoltaic Systems Under Partial Shading Conditions", in *IEEE Transactions on Power Electronics*, vol. 37, no. 9, pp. 11332–11341, 2022.

[5] C. Kalogerakis, E. Koutroulis and M. G. Lagoudakis, "Global MPPT Based on Machine-Learning for PV Arrays Operating under Partial Shading Conditions", in *Applied Sciences*, vol. 10, no. 2, p. 700, Jan. 2020.

[6] K. Kagkarakis, Photovoltaic Technology, Symmetria Publications, 1992.

[7] I. Fragkiadakis, Photovoltaic Systems, Ziti Publications, 2006.

[8] S. Manias, Power Electronics, 4th ed., Symeon Publications, 2021.

[9] S. Manias, Industrial Electronics, 3rd ed., Symeon Publications, 2021.

[10] H. D. Tafti et al., "Extended Functionalities of Photovoltaic Systems With Flexible Power Point Tracking: Recent Advances", in *IEEE Transactions on Power Electronics*, vol. 35, no. 9, pp. 9342-9356, Sept. 2020.

[11] Infineon Technologies AG, Control integrated power system mini IPM technical description, AN2022-06, Rev. 1.0, Sept. 2022.

[12] Microchip, ATmega 640 / V-1280 / V-1281 / V-2560 / V-2561 / V Datasheet, Rev. DS40002211A, May 2020.

[13] "Arduino Mega 2560 Rev3 Schematic." (2019), [Online]. Available: https://docs.arduino.cc/hardware/mega-2560 (visited on May 22, 2023).

[14] Texas Instruments, LM393B, LM2903B, LM193, LM293, LM393 and LM2903 Dual Comparators, SLCS005AE, Revised Nov. 2020, Oct. 1979.

[15] Infineon Technologies AG, Control integrated power system IKCM30F60GD Datasheet, V. 2.5, Oct. 2019.

[16] E. Koutroulis and K. Kalaitzakis, Electric Measurements and Sensors - Operation Principles and Electronic Measurement Systems Design, Klidarithmos Publications, 2010.

[17] Texas Instruments, LMx58-N Low Power, Dual-Operational Amplifiers, SNOSBT3J, Revised March 2022, Jan. 2000.

[18] Allegro Microsystems, ACS712 Datasheet, MCO-0000197, Rev. 20, Feb. 2022.

[19] Texas Instruments, LM317 3-Terminal Adjustable Regulator, SLVS044Y, Revised Apr. 2020, Sept. 1997.

[20] Kimo Instruments, Solarimeter SL100 Technical Data Sheet, Feb. 2015.

[21] Kimo Instruments, Air Quality AQ110 Technical Data Sheet, Jul. 2015.

[22] Independent Power Transmission Operator (IPTO), Hellenic Electricity Transmission System (HETS) Grid Code, ver. 3.9, Aug. 2020.

[23] Z. Xie and Z. Wu, "A flexible power point tracking algorithm for photovoltaic system under partial shading conditions", in *Sustainable Energy Technologies and Assessments*, vol. 49, p. 101747, 2022.

[24] E. I. Batzelis, S. A. Papathanassiou and B. C. Pal, "PV System Control to Provide Active Power Reserves Under Partial Shading Conditions", in *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9163-9175, Nov. 2018.

[25] P. Verma and T. Kaur, "Power reserve control strategy of PV system for active power reserve under dynamic shading patterns", in *Array*, vol. 16, pp. 100150, Dec. 2022.

[26] Y. Wang, Y. Li and X. Ruan, "High-Accuracy and Fast-Speed MPPT Methods for PV String Under Partially Shaded Conditions", in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 235-245, Jan. 2016.

[27] E. Koutroulis and F. Blaabjerg, "Overview of Maximum Power Point Tracking Techniques for Photovoltaic Energy Production Systems", in *Renewable Energy Devices and Systems with Simulations in MATLAB and ANSYS*, 1st ed., F. Blaabjerg and D. M. Ionel, CRC Press – Taylor and Francis Group, 2017, ch. 5, pp. 91–130.

[28] R. S. Sutton and A. G. Barto, Reinforcement Learning – An Introduction, 2nd ed., MIT Press, 2018.

[29] S. Gupta, S. Bhambri, K. Dhingra, A. B. Buduru and P. Kumaraguru, "Multi-objective Reinforcement Learning based approach for User-Centric Power Optimization in Smart Home Environments", *2020 IEEE International Conference on Smart Data Services (SMDS)*, Beijing, China, 2020, pp. 89-96.