



**TECHNICAL UNIVERSITY OF CRETE**

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

**Diploma Thesis**

**Development of a System for Maximizing the Power  
Production of Photovoltaic Arrays Based on  
Reinforcement Learning**

**Theodoros Bountoukos**

**EXAMINATION COMMITTEE**

Assoc. Professor Eftychios Koutroulis (Supervisor)

Assoc. Professor Michail Lagoudakis

Professor Kostas Kalaitzakis

Chania, January 2021

## Table of Contents

<b>1. Introduction</b> .....	5
1.1 Subject of the thesis.....	5
1.2 Structure of the thesis.....	8
<b>2. Photovoltaic systems</b> .....	10
2.1 Photovoltaic systems and the manufacturing process.....	10
2.2 Solar cell mathematical model .....	11
2.3 I-V and P-V curves under uniform irradiation.....	12
2.4 I-V and P-V curves under non-uniform irradiation .....	14
2.5 DC/DC power converters .....	16
2.5.1 Basic DC/DC power converters .....	17
2.5.2 DC/DC Boost converter .....	18
2.6 Maximum power point tracking .....	23
2.6.1 Perturb & Observe (P&O) method.....	24
2.6.2 Particle Swarm Optimization method.....	25
2.7 The DC/DC Boost Converter Circuit .....	29
<b>3. Q-learning algorithm for MPP Tracking</b> .....	31
3.1 Reinforcement learning .....	31
3.2 Q-Learning for MPP tracking.....	36
3.2.1 Action selection policy .....	37
3.2.2 State-Space .....	39
3.2.3 Action-space .....	40
3.2.4 Reward .....	42
3.2.5 Discount Factor and Learning Rate .....	43
3.3 The overall Q-learning MPPT algorithm.....	43

<b>4. Impementation of the GMPPT system</b> .....	47
4.1 Introduction .....	47
4.2 The microcontroller-based control unit.....	48
4.3 Programming of the Q-learning algorithm .....	50
4.4 I-V and P-V curve tracer .....	58
4.5 Components of the DC/DC Boost converter circuit .....	59
<b>5. Experimental Results</b> .....	65
5.1 Introduction .....	65
5.2 Analysis of the results for shading pattern 1 .....	68
5.3 Analysis of the results for shading pattern 2 .....	93
5.4 Analysis of the results for shading pattern 3 .....	118
5.5 Analysis of the results for shading pattern 4 .....	142
5.5 Summarized results.....	151
<b>6. Conclusions</b> .....	153
<b>7. References</b> .....	155

## **Abstract**

The global demand for energy is rising rapidly and conventional methods of producing energy contribute to global warming. Renewable energy sources are able to solve these problems. Solar energy produced by photovoltaic arrays is considered to be amongst the major renewable energy sources, abundantly available. The photovoltaic arrays however yield very low efficiency under non-uniform incident solar irradiance operating conditions. The subject of this thesis is the development of an electronic system for maximizing the power production of photovoltaic arrays. For that reason, a reinforcement learning-based global Maximum Power Point Tracking algorithm was developed. The PV system developed in this thesis consists of an MPPT control unit, a DC/DC Boost converter and two batteries as load. For the implementation of the MPPT control unit, a Q-learning algorithm, as well as a Particle Swarm Optimization (PSO) algorithm were developed. The Q-learning algorithm under study was used in multiple experiments for alternative shading patterns of the PV array and its performance was compared to that of the PSO algorithm. The experimental results demonstrated that the Q-learning MPPT algorithm converged faster and more accurately to the Global MPP than the PSO MPPT algorithm when an appropriate learning process was applied.

# 1. INTRODUCTION

---

## 1.1 Subject of the thesis

Human actions overload earth's atmosphere with carbon dioxide and other pollutants that add to global warming. Such gases behave like a blanket, which means that heat is trapped. From intense storms to drought, ice caps melting and extinction, the outcome is severe and has extremely harmful impacts. Around 30 percent of global warming emissions in the US come from the energy industry. Fuels like coal as well as natural gas account for much of those pollutants. The most common greenhouse gas is carbon dioxide; however there is an abundance of air contaminants that can also cause global warming, such as methane. However, various sources of energy create different quantities of these contaminants. For the majority of developing countries, reducing greenhouse gas emissions is a major priority. To date, relative to the 1990s, the European Union has managed to reduce greenhouse emissions by 20 percent. EU countries plan to use renewable energy sources to meet at the minimum 32 percent of their demands by 2030.

Renewable energy originates from inexhaustible sources (unlike fossil fuels). Wind, solar, biomass, geothermal and hydro, all of which occur naturally, constitute renewable energy sources. Renewable energy is non-polluting. In the energy production process, many types do not release any greenhouse gasses or radioactive waste. It is a sustainable source of energy that is reliable, cost-effective and efficient for the long term.

Solar power is clean green energy obtained from sunlight or in some instances, from sun-generated heat. Building solar power systems in a residential area involves setting up a solar

photovoltaic (PV) or solar thermal system on a roof. The photovoltaic definition comes from: photo = light (energy from the sun's particles) and voltaic = voltage output. Unlike fossil fuels that are finite, solar energy is a renewable and free energy source that is sustainable and completely inexhaustible. It is also a non-polluting energy source and when generating electricity, it does not produce any greenhouse gases.

Gases that contribute to the greenhouse effect are only released during the manufacturing of the components of the PV array. When their installation is complete, without the emission of greenhouse gases, electricity production by solar irradiation is possible. During their lifetime, PV modules can generate significant amounts of energy, more than the energy needed during their production. The main benefit of PV panels is that they can be built in remote areas where an electricity network is not available. In practice, most of the time, the electricity produced by renewable sources does not originate from off-grid installations, but from grid-connected installations where the electricity is supplied directly to the electric grid. Therefore, it is acknowledged that a way for EU countries to develop economically is through renewable energy production. It is also very useful for domestic use to generate RES power, since by supplying electric energy to the grid it can be a way of financial revenue growth.

PV power generation has several drawbacks. Owing to the high price of its components (DC/AC inverters, wiring, batteries, photovoltaic panels and their installation), it is more costly than other resources. Cost reduction can be accomplished by increasing the efficiency of renewable energy sources, resulting in higher revenues and lower costs of energy produced by RES. The energy produced by the PV system is also low in days when the sunlight level is low, including cloudy or rainy days. For that reason, on sunny days, batteries are used to store energy. In addition, in order to be installed, a PV power generation installation requires a lot of space and sometimes roofs do not have sufficient space to accommodate the amount of PV panels needed to meet the necessary energy demand. Lastly, as mentioned, during the manufacturing process of photovoltaic panels and after their disposal, they emit dangerous chemicals some of which are lead, silicon tetrachloride and hexafluorathane.

To generate electricity, PV systems take advantage of the photovoltaic effect. The first observation of that effect was performed in 1839 by the engineer Antoine Becquerel. Only an extremely small percentage of our energy demands is met today by the use of solar energy. The cost of photovoltaic installations has gradually been decreasing since 2009. It is of significant importance that technology based on solar energy is less costly than fuel or nuclear power systems. In combination with the gradual improvement in efficiency of PV systems, the aforementioned observations will make systems based on solar energy the prevailing energy source in the decades to come.

Upon exposure to light, voltage and current can be developed in the illuminated material. This phenomenon called the photovoltaic effect is both physical and chemical. This implies that the PV modules are directly influenced by the incident irradiance. A major problem therefore occurs, and that issue is the behavior of the PV modules under non-uniform irradiance and partial shading induced by neighboring artifacts such as houses, trees, clouds, soil, etc. The power that a PV system outputs can be greatly influenced by that effect, since in its power-voltage curve a photovoltaic array that operates under non-uniform irradiance produces several peaks. This is the whole premise of this thesis, attempting to solve the problems created by non-uniform irradiance. The PV arrays must be controlled in order to take care of the problem and minimize it so that they produce the maximum possible power level. Maximum power point tracking (MPPT) methods are employed to achieve that. These methods seek to maximize the extraction of power from the PV system under any environmental condition so that they improve the efficiency of the PV plant. Trying to improve the hardware parts such as the power converters is difficult and quite costly in already existing PV plants. What can be done though is to optimize the MPPT algorithms performance. It can be done in already existing and operating PV plants hence there is no need for more expenses.

There is a wide variety of MPPT implementation techniques. For instance such algorithms are the Perturb & Observe and the Particle Swarm Optimization method. As claimed by

experimental results, the PSO MPP tracking method is considered to be the most powerful MPPT algorithm since it is able to find and converge to the MPP in any environmental condition, no matter its position on the PV curve of the photovoltaic array.

In this thesis, a system for maximizing the power production of a PV array was designed. Specifically, to maximize the power output of photovoltaic arrays operating under non-uniform irradiance, a Q-learning MPP tracking algorithm was developed. The system developed for that purpose consists of:

- Two photovoltaic panels, connected in series,
- A DC/DC Boost converter and
- An Arduino-based board programmed with the Q-learning algorithm which controls the DC/DC converter PWM signal.

The control unit senses both the PV array output voltage and current values. These two values are dependent on the PWM signal value, the light that hits the photovoltaic panels as well as their temperature. Changing the aforementioned parameters can result in changes of the voltage and current sensed.

## 1.2 Structure of the thesis

The structure of this thesis is the following:

- **Chapter 2:** Information and functionality of photovoltaic systems, analysis of current-voltage and power-voltage curves, as well as information and specifications on hardware components design, such as the DC-DC Boost converter.
- **Chapter 3:** Description of the Q-learning algorithm.

- **Chapter 4:** Description of the algorithm used to observe the I-V curve of the array and details on the source code developed for the MPPT process, as well as specifications of the hardware components used, such as the DC-DC Boost converter.
- **Chapter 5:** Experiments for different shading patterns, and their presentation.
- **Chapter 6:** Conclusion analysis and proposals for future extensions.

# 2. PHOTOVOLTAIC SYSTEMS

---

## 2.1 Photovoltaic systems and the manufacturing process

It is important to know the manufacturing process of a cell because later on in this thesis it has to be explained why certain measures have to be taken and what these measures mean for this implementation. One example that is going to be discussed later on is the existence of bypass diodes, why they are needed and what problems can result from their use.

Photovoltaic cells are disks made out of silicon that produce electricity purely from sunlight. Those disks serve for a wide range of applications, as energy sources. Research into solar-produced electricity followed incredibly quickly when research into electricity began. Antoine Becquerel exposed a chemical battery to sun light only to observe voltage being produced by it. The efficiency of this first sunlight-to-electricity conversion was just one percent. Charles Fritts created the first solar cell using gold coated selenium, with just one percent efficiency, in the 1880s.

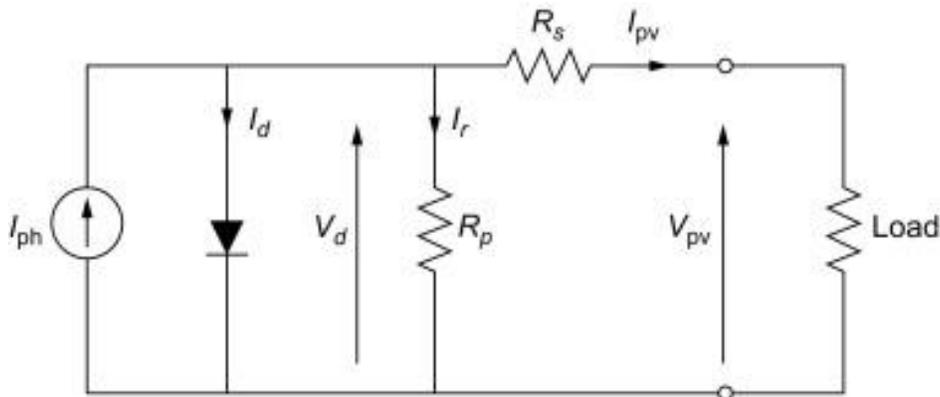
According to the explanation of the photoelectric effect by Albert Einstein in 1905, metal absorbs and retains energy from light until a large amount of light hits it. However, little progress was made until diode and transistor research gave Bell scientists Gordon Pearson, Darryl Chapin, and Cal Fuller the necessary knowledge to produce a four percent efficiency silicon solar cell in 1954.

A solar cell's fundamental component is pure silicon. Silicon is not pure naturally but derived from silicon dioxide. To create an excess of electrons and a deficit of electrons, respectively, the

pure silicon is doped with boron and phosphorous in order to make a semiconductor capable of conducting electricity. An anti-reflective coating such as titanium dioxide is also needed for silicon disks.

## 2.2 Solar cell mathematical model

A photovoltaic cell is modelled as a p-n junction with nonlinear characteristics. As shown in Figure 1, a diode connected in parallel to a current source is the simplest equivalent circuit. The current that is produced is proportional to the light that hits the cell. The photovoltaic cell operates as a diode in reverse mode during the night. However it generates a diode current if the photovoltaic cell is illuminated. The I-V features of the cell are determined by the properties of the diode. Each photovoltaic cell has a resistance which can be represented as a series resistance,  $R_s$ , and a shunt resistance  $R_p$ , which, however, can be neglected as an approximation, because it exhibits a very high value.



**Figure 1.** Photovoltaic cell equivalent circuit.

Therefore, the current of the photovoltaic cell is the difference between the current produced by the sunlight,  $I_{ph}$  and the diode current  $I_d$ :

$$I_{PV} = I_{ph} - I_0 \left( e^{\frac{q(V_{PV} + I_{PV}R_s)}{nkT}} - 1 \right) \quad (2.1)$$

where:

$q$  is the electron charge,

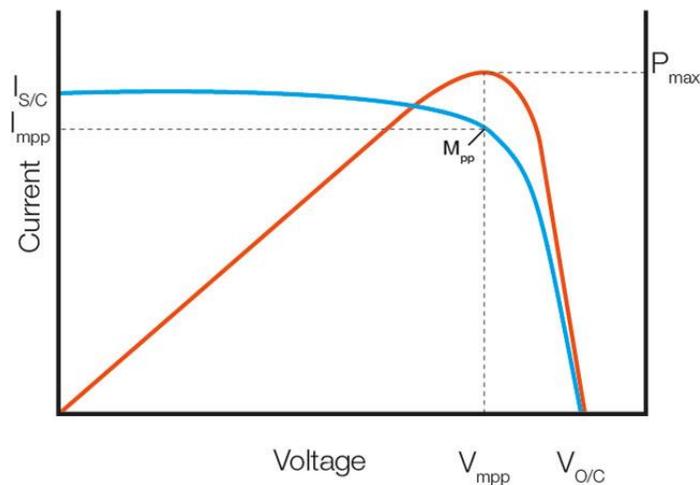
$n$  is the diode's ideality factor.

$k$  is the Boltzmann constant,

$T$  is the temperature.

### 2.3 I-V and P-V curves under uniform irradiation

The current-voltage (I-V) characteristics of either a photovoltaic cell or module or array provide a detailed overview of its ability and efficiency to convert solar energy. Knowing the I-V characteristics of photovoltaic arrays is crucial in evaluating the performance as well as the efficiency of the PV system. The amount of light that reaches the cell regulates the current, and as the temperature of the photovoltaic panel increases, its voltage decreases.



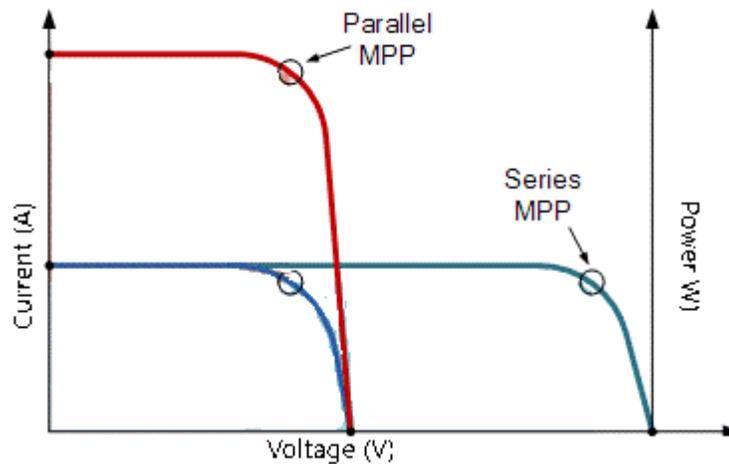
**Figure 2.** The I-V and P-V curves of a PV array.

The graph shown in Figure 2 shows the characteristic of the current-voltage of a photovoltaic array under uniform irradiance. The power that a photovoltaic array delivers is the product of the current and voltage. The power-voltage (P-V) curve above is obtained through the multiplication of all current-voltage couples ranging from short-circuit conditions to open-circuit conditions for a specific radiation. When there is no load connected to the positive and negative leads of the photovoltaic panel the voltage is at its maximum value and is called open-circuit voltage ( $V_{oc}$ ) while the current has a zero value. This condition is called open circuit. On the other extreme, in the short circuit condition when the leads of the photovoltaic are connected to each other, the current maximum value is obtained called short-circuit current ( $I_{sc}$ ) while the voltage has a zero value.

There is however a point where the photovoltaic array produces the maximum possible power for the specific environmental conditions, seen at the top right of the dotted rectangle in the above figure. At this point of maximum power, the voltage and current produced are indicated as  $V_{MP}$  and  $I_{MP}$ , respectively. This maximum power point (MPP) is the ideal point of operation of a photovoltaic array.

In the I-V characteristic, the MPP of a photovoltaic array is located near the ‘knee’ of the curve. From the values of  $V_{oc}$  and  $I_{sc}$ , the corresponding values of  $V_{mp}$  and  $I_{mp}$  can be estimated:  $V_{mp}$  is equal to 0.8–0.9 of  $V_{oc}$  and  $I_{mp}$  is equal to 0.85–0.95 of  $I_{sc}$ . Because both the voltage and current of the photovoltaic array depend on temperature, the actual output power generated by the cell will vary with temperature.

In order to increase the PV array voltage and current, multiple PV panels may be connected in either series or parallel, or a combination of both. The voltage capacity rises when photovoltaic modules are connected in series with the current being the same for all modules within a string, however the current increases when connected in parallel with the voltage being the same for all strings (as shown in Figure 3).



**Figure 3.** The I-V and P-V curves for PV modules/cells connected in series and parallel.

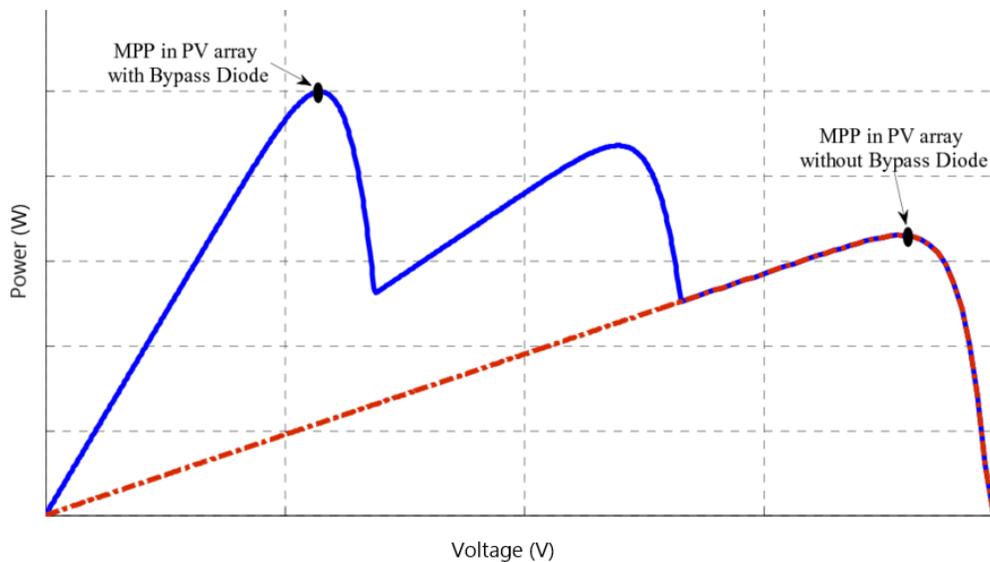
## 2.4 I-V and P-V curves under non-uniform irradiation

A typical photovoltaic module is made up of a few photovoltaic cells connected in series. The series connection forces the photovoltaic cells of the module to produce the same amount of current. During partial shading, however, the cells affected produce lower current. Even if the unshaded cells are capable of producing high current, the shaded cells will force them to produce lower than that. This drastically decreases the photovoltaic module's performance. Moreover, in the photovoltaic module, the lower irradiated solar cells dissipate power generated by unshaded or more irradiated photovoltaic cells due to them becoming reverse biased. As a consequence, on the photovoltaic module, the photovoltaic cells that dissipate power become overheated and hot spots may appear. If the dissipated power exceeds the maximum power that can be sustained by the cell, it will be damaged [1].

In order to avoid the destructive effects of partial or full shading, a bypass diode is connected in reverse parallel to each of the cells that are connected in series. That is one very simple yet effective way of protecting the photovoltaic cells or modules from non-uniform irradiance effects. This provides an alternate path for the current produced to flow through, because when shaded the flow through the cell cannot happen. When the diodes become biased they consume part of the power that the unshaded cells produce and therefore a voltage drop is introduced.

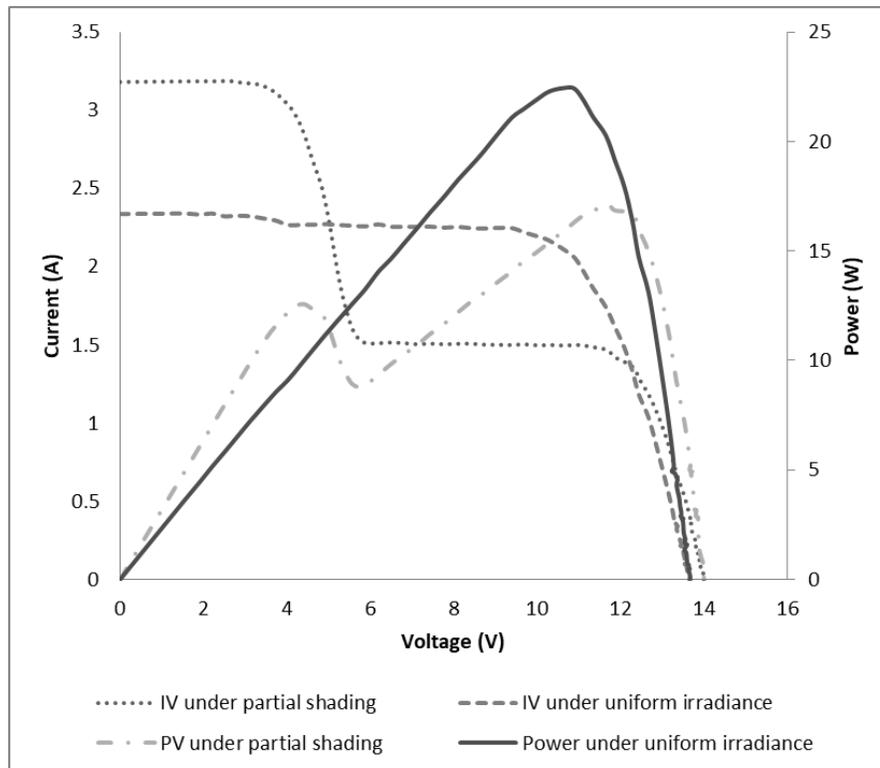
Subsequently the power that is delivered by the photovoltaic modules is affected. Thus, the impact of the bypass diode on the maximum power of the photovoltaic module must be investigated [2].

All cells and all modules of a photovoltaic array produce the same current under uniform operating conditions and the diodes do not conduct since they are reverse biased. Therefore, they do not dissipate power and do not cause multiple peaks. Next, a partially shaded photovoltaic module is considered which is composed of two unshaded submodules and one shaded. The bypass diodes do not conduct as long as the photovoltaic module's current is lower than the  $I_{SC}$  of the shaded submodule. In this case, the modules total voltage is the sum of each submodule's voltages. However the bypass diode of the shaded submodule conducts when the module's current is higher than the  $I_{SC}$  of the shaded submodule. In this case, due to the bypass diode, the P-V characteristic curves of the photovoltaic module could exhibit multiple peaks. In the figure below (Figure 4) the P-V curve during partial shading is presented for a photovoltaic array with and without the use of bypass diodes.



**Figure 4.** P-V characteristics during partial shading for a PV array with and without the use of bypass diodes.

In Figure 5 the I-V as well as the P-V curves are presented for both uniform and non-uniform irradiation so that the effects that partial shading and bypass diodes have on a photovoltaic array characteristics can be observed.



**Figure 5.** I-V and P-V curves of PV array under uniform and non-uniform irradiation.

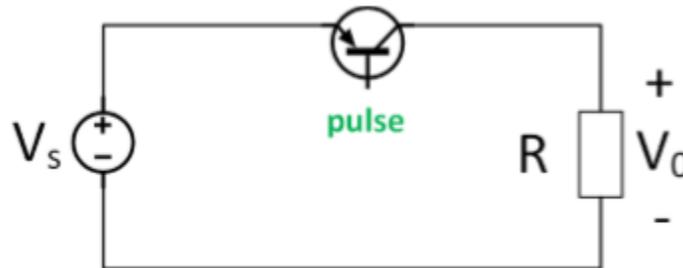
## 2.5 DC/DC power converters

A DC/DC power converter is an electronic circuit capable of converting the voltage applied to its input into a higher or lower voltage. There are 3 different types of DC/DC power converters: the DC/DC Boost converter, the DC/DC Buck converter and the DC/DC Buck-Boost converter. The

DC/DC Boost converter is only capable of boosting, as its name implies, the voltage that is applied to its input. The DC/DC Buck converter is only capable of decreasing the voltage that is applied to its input. However, a DC/DC Buck-Boost converter is capable of both boosting and decreasing the voltage based on the needs of the circuit it is connected to.

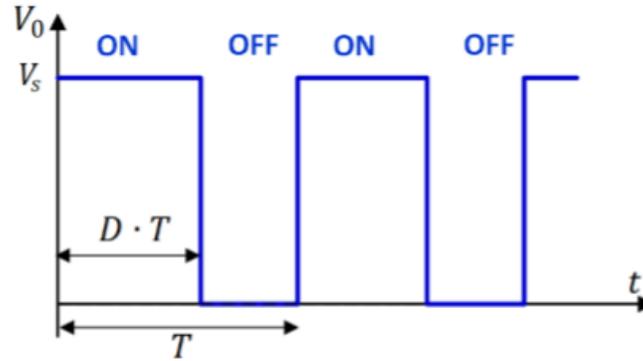
### 2.5.1 Basic DC/DC power converters

A basic form of DC/DC power converter consists of a transistor, the source of voltage and the load connected in series with the transistor, as shown in Figure 6.



**Figure 6.** Basic DC/DC power converter (DC chopper).

The transistor operates as an electrically controlled switch. Assuming that the transistor is ideal, which means that it does not dissipate any power, the output voltage is the same with the input voltage only when the switch is turned ON (ON state) and 0 otherwise (OFF state). Such a circuit is known as DC-chopper. The output voltage versus time is shown in Fig. 7.



**Figure 7.** Output voltage for ON and OFF states.

The output voltage is calculated as follows:

$$\bar{V}_0 = \frac{1}{T} \int_0^T V_0(t) dt = \frac{1}{T} \int_0^{DT} V_s dt = V_s * D \quad (2.2)$$

$$D = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T} = t_{on} * f \quad (2.3)$$

where:

$V_0$  is the output voltage of the DC chopper,

$T$  is the period of the PWM signal,

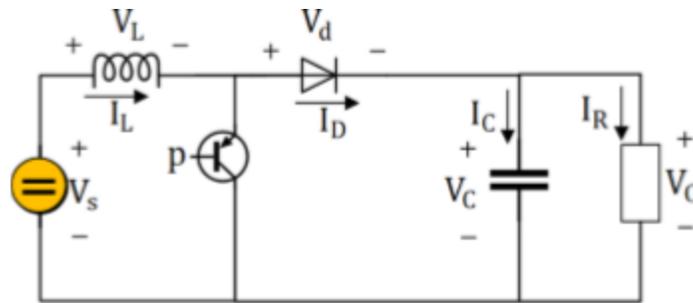
$D$  is the duty cycle of the PWM signal,

$t_{on}$  is the time interval that the transistor or switch is ON,

$t_{off}$  is the time interval that the transistor or switch is OFF.

## 2.5.2 DC/DC Boost converter

In DC/DC Boost converters, the input voltage is lower than the output voltage. If  $T_{on}$  which is the time in which the switch is turned on, is divided by the period duration  $T$ , a value called Duty Cycle is calculated. The circuit of a DC/DC Boost converter is presented in Figure 8.



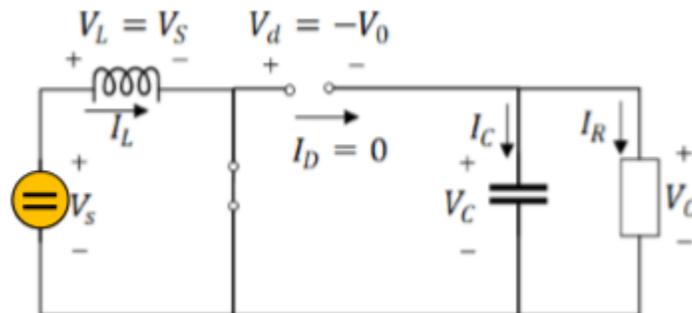
**Figure 8.** DC/DC Boost converter.

For the DC/DC Boost converter there are two operating modes (phases). The first phase is when the switch is turned ON (ON state) and the second phase is when the switch is turned OFF (OFF state). The following have to be taken into consideration:

1. The current passing through the coil is continuous and always positive.
2. The capacitor is big enough so the output voltage is steady.
3. The switching period is  $T$ . The switch is closed (ON state) for  $D \cdot T$  and open for  $(1-D) \cdot T$ .
4. The circuit operates at the steady state.

As mentioned before there are two “phases” that have to analyzed separately.

**Phase 1 (P1)**



**Figure 9.**DC/DC Boost converter during ON state.

When the switch is turned ON (ON state) then the diode becomes reverse-biased (as shown in Figure 9). That means that there is no current passing through the diode thus no power is dissipated. The voltage across the inductor is:

$$V_L = V_S = L \frac{dI_L}{dt} \Rightarrow$$

$$\frac{dI_L}{dt} = \frac{V_S}{L} \Rightarrow$$

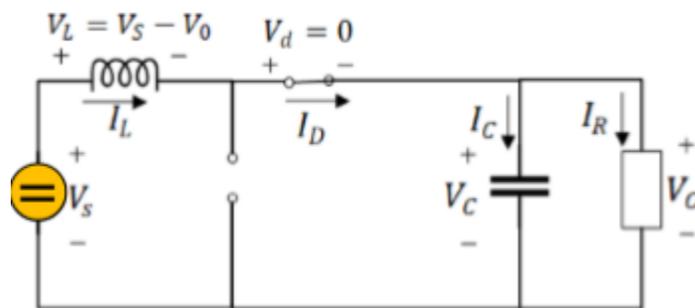
$$I_L = \frac{V_S}{L} t + c_1, \quad 0 \leq t \leq D * T \quad (2.4)$$

The current is increased linearly. The current change during the first operating mode (P1) can be calculated with the following equation:

$$\Delta I_{L(\phi_1)} = I_L(D * T) - I_L(0) \Rightarrow$$

$$\Delta I_{L(\phi_1)} = \frac{V_S}{L} D * T \quad (2.5)$$

### Phase 2 (P2)



**Figure 10.** DC/DC Boost converter during OFF state.

When the switch is turned OFF (OFF state) then the diode becomes forward biased (as shown in Figure 10). That means that now current is passing through the diode thus power is dissipated. Since the switch is now OFF this means that in order for the inductor to resist to the changes in current passing through it, the diode has to become biased so that current can continue to flow through the inductor. That means that the current still flows towards the same direction as it did in P1. The voltage across the coil is:

$$\begin{aligned}V_L &= V_s - V_0 = L \frac{dI_L}{dt} \Rightarrow \\ \frac{dI_L}{dt} &= \frac{V_s - V_0}{L} \Rightarrow \\ I_L &= \frac{V_s - V_0}{L} t + c_2, \quad D * T \leq t \leq T\end{aligned}\tag{2.6}$$

The current change during the second operating mode (P2) can be calculated with the following equation:

$$\begin{aligned}\Delta I_{L(\phi_2)} &= I_L(T) - I_L(D * T) \Rightarrow \\ \Delta I_{L(\phi_2)} &= \frac{V_s - V_0}{L} (1 - D) * T\end{aligned}\tag{2.7}$$

The operation of the circuit under steady-state requires that the current that passes through the inductor at the end of the period T is the same as it was in the beginning, so that the overall change in current passing through the inductor is equal to 0.

This means that:

$$\Delta I_{L(\phi_1)} + \Delta I_{L(\phi_2)} = 0 \Rightarrow$$

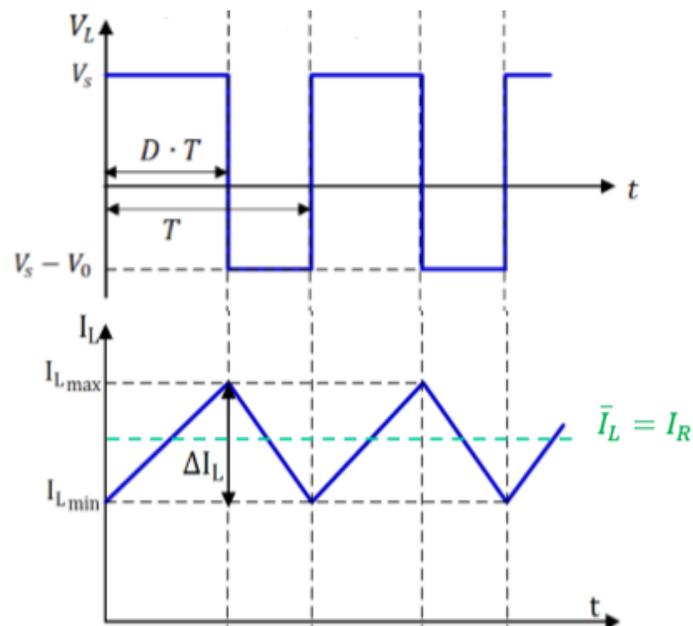
$$\frac{V_S}{L} D * T + \frac{V_S - V_0}{L} (1 - D) * T = 0 \Rightarrow V_0 = \frac{V_S}{1 - D} \quad (2.8)$$

Thus the converter produces a voltage that is equal to or higher than the voltage applied at the input since  $D < 1$ .

For this analysis to be valid, a continuous flow of current through the inductor is necessary. Since  $I_{\min} = 0$  is the limit between continuous inductor current flow and discontinuous inductor current flow the above equations can be used to determine the values of  $L$  and  $f$  that will allow the circuit to operate with a continuous inductor current flow. Since the switching frequency is fixed, the minimum value for  $L$  can be determined:

$$L_{\min} = \frac{(1 - D)^2}{2 * f} R * D \quad (2.9)$$

In Figure 11 the voltage and current can be seen versus time for both operating phases.

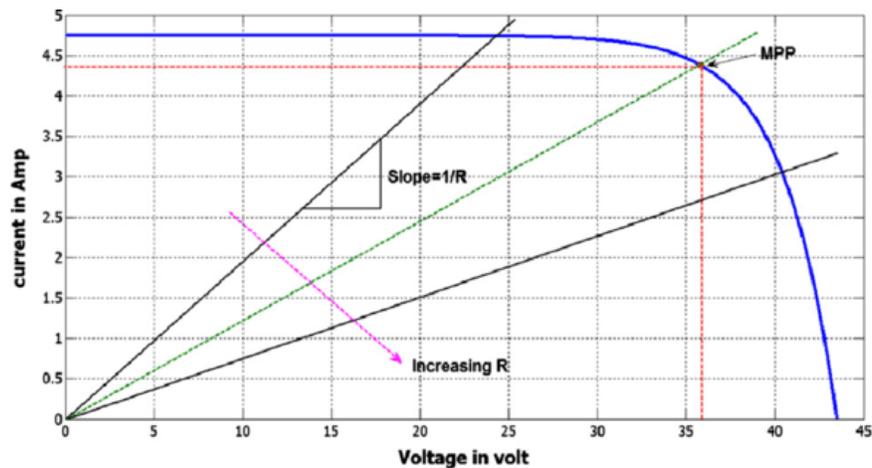


**Figure 11.** Current and voltage in P1 and P2.

## 2.6 Maximum power point tracking

If a load is connected to a photovoltaic panel directly, the panel operating point would most probably vary from the MPP. The panel's operating point is determined by the impedance that is seen by it. Therefore by changing the panel-seen impedance the operating power point can be moved so that it becomes equal to the maximum power point. Changing the DC/DC Boost converter duty cycle, causes the panel-seen impedance to change. At a specific impedance the operating and maximum power point will converge. With variations in environmental conditions such as temperature changes or irregular irradiance, the I-V curve of the photovoltaic array can vary vastly. Thus, with such dynamically changing operating conditions, fixing the duty cycle to a value is not viable [3].

Algorithms that constantly sample the photovoltaic panel voltage and current are used by MPPT implementations, and then adjust the duty cycle accordingly. To implement such algorithms, microcontrollers are employed. The impedance of the load can be seen in Figure 12, which is shown as  $1/R$ . It can be easily seen that the operating power point of the panel can be moved until it reaches the MPP by decreasing the impedance or in other words, increasing the load resistance, the load being the DC/DC Boost converter.



**Figure 12.** The I-V curve and the impedance seen by the photovoltaic panel.

Many methods for the implementation of the MPPT process have been developed. The Perturb & Observe (P&O) and Particle Swarm Optimization algorithms, which will be introduced below, are among them [4].

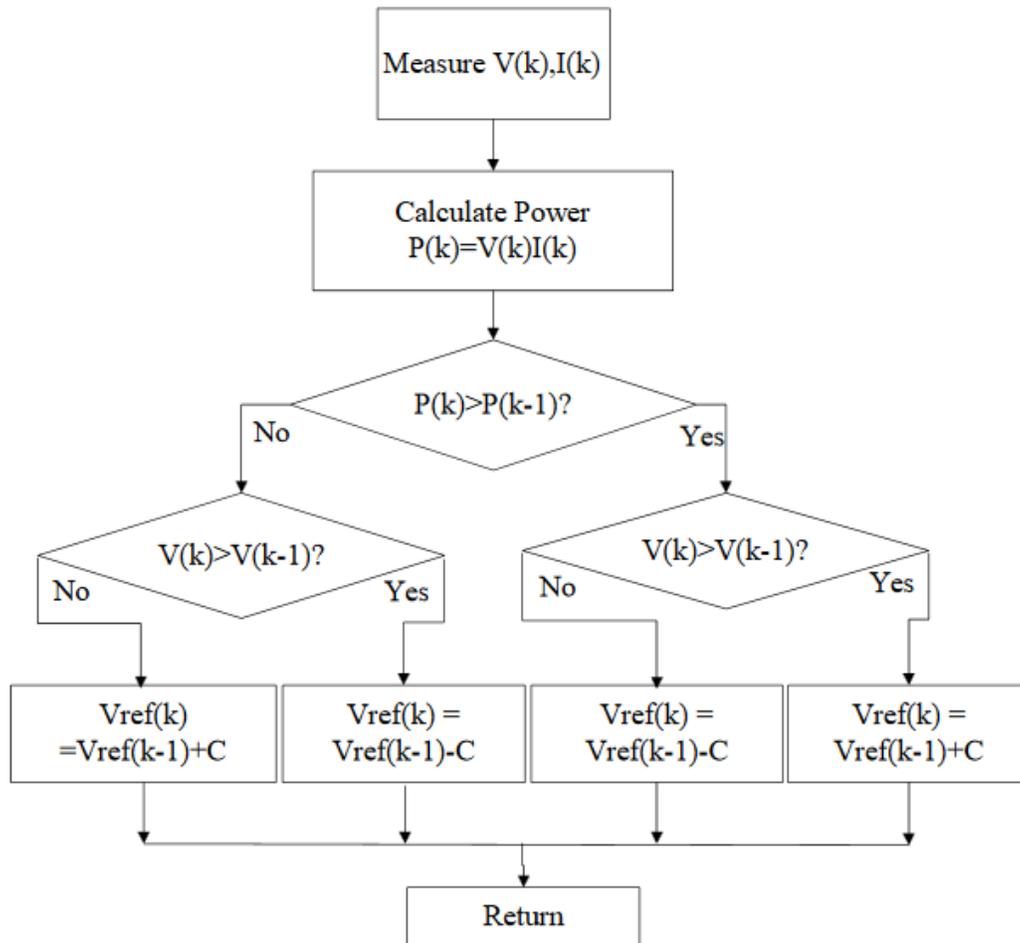
### **2.6.1 Perturb & Observe (P&O) method**

In commercial PV products, the P&O method is the most frequently used MPP tracking algorithm. This is basically a form of a "trial and error" method. The power converter control unit increases by a small amount the reference for the converter output power, and if the latter has been increased as well, it can increase and keep increasing until it begins to decrease, by which point the reference is decreased by the control unit to prevent the PV output power from collapsing.

While the P&O algorithm implementation is simple, it exhibits the following disadvantages:

- 1) Even with a stable sunlight condition, the PV system can still run in an oscillating mode, causing the power converter output power to fluctuate, and
- 2) During sudden changes of incident sunlight, the operation of the photovoltaic system can fail to detect the MPP.

The P&O method is used for its low complexity, high reliability, and tracking efficiency, mainly in practical PV systems. Figure 13 demonstrates the P&O algorithm flowchart. The present state power is determined using the present state voltage and present state current values and is then compared to the power of the previous state. The next change in voltage is kept in the previous change direction if the power rises. Otherwise, the change direction is reversed [5], [6].



**Figure 13.** The Perturb-and-Observe MPPT method flowchart.

## 2.6.2 Particle Swarm Optimization method

The PSO algorithm operates based on the behavior of birds. A swarm of individuals (called particles) is maintained by the PSO algorithm, where a possible solution is represented by each particle. The former follow a basic behavior: the progress of other, close particles and their own accomplishments are emulated. Therefore the location of each particle is influenced by both the best solution found by the particle itself and the best solution found by every particle in the

whole population. In recent years, the Particle Swarm Optimization algorithm has become a better-developed optimization algorithm [7]. Via continuous iteration, it looks for the optimal solution and eventually uses the size of the value of the objective function or the function to be optimized to determine the efficiency of the solution. The Particle Swarm Optimization MPP tracking algorithm follows the below steps of execution [8]:

- The power converter's duty cycle corresponds to the location of the particles. The evaluation function (fitness value) is the photovoltaic array generated power. The more particles used, the more precise the convergence to the global MPP (GMPP). However more particles results in a longer execution time.
- These particles are positioned at a particular location during initialization. Having some previous knowledge of where the GMPP is located should result in the particles being positioned around it. It should also be noted that the particles have to be located in the range between the maximum and minimum duty cycle values, called  $D_{\max}$  and  $D_{\min}$  respectively.
- Each duty cycle corresponds to a voltage-current couple. To measure the power produced by the photovoltaic array (fitness value) this pair is used. This is done for every particle. It should be noted that the settling time of the power converter has to be lower than the interval time between the evaluation of successive particles in order to obtain accurate samples.
- A criterion should be established for the evaluation of how accurate a solution is compared to the optimal solution. In the MPPT problem, that criterion is the generated photovoltaic output power (fitness function).
- The fitness value of the highest value for each particle is called  $P_{\text{BEST},i}$ . If a specific particle discovers a higher fitness value in the future, then the  $P_{\text{BEST},i}$  is updated to that value. After that,  $G_{\text{BEST}}$  is selected as the particle with the highest  $P_{\text{BEST},i}$  between the particles.
- Then, the position as well as the velocity of every particle are modified after the assessment of every particle. Each particle's position and velocity are defined by:

$$U_i(k + 1) = wU_i(k) + c_1r_1(P_{\text{best},i} - x_i(k)) + c_2r_2(G_{\text{best}} - x_i(k)) \quad (2.10)$$

$$x_i(k + 1) = x_i(k) + u_i(k + 1) \quad (2.11)$$

where:

$x_i$  is the i-th particle position

$u_i$  is the i-th particle velocity

$k$  is the number of iterations

$w$  is the inertia weight

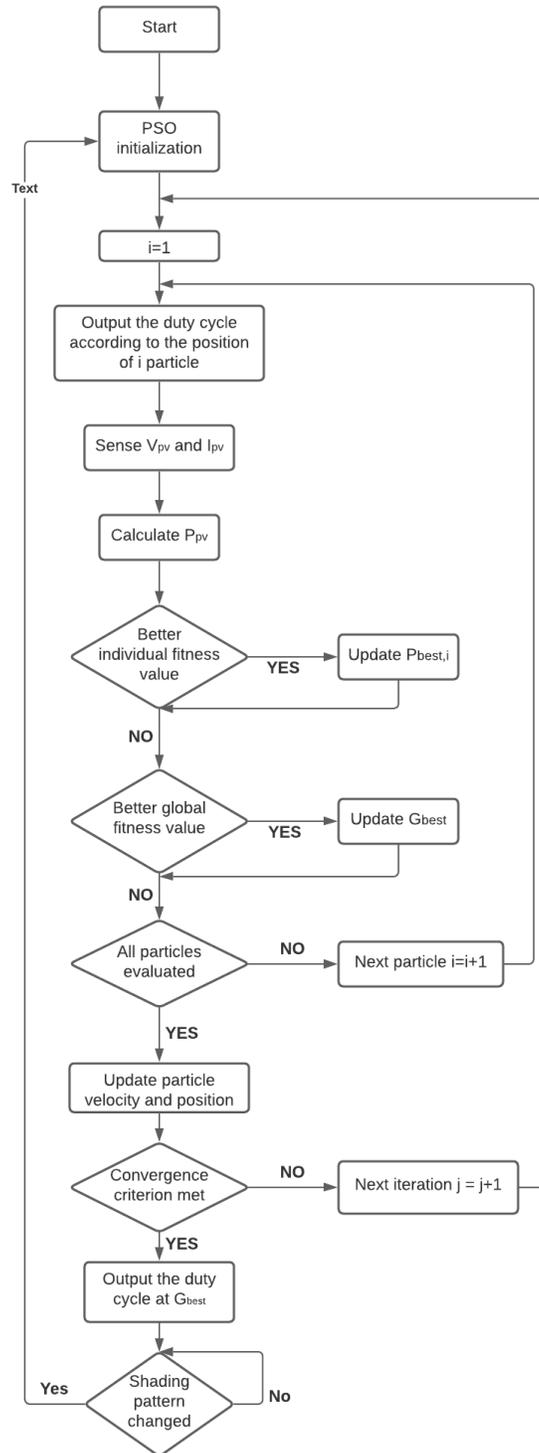
$r1, r2$  are uniformly distributed random values between 0 - 1 and

$c1, c2$  are the cognitive and social coefficient, respectively.

The MPP tracking algorithm will have converged once the value of the iterations reaches its maximum, or once all the particle velocities drop lower than a predefined minimum [9]. The algorithm is reinitialized once the equation that follows becomes true:

$$\frac{|P_{PV,new} - P_{pv,last}|}{P_{PV,last}} \geq \Delta P(\%) \quad (2.12)$$

The flowchart of the Particle Swarm Optimization MPPT method is presented in Figure 15.



**Figure 15.** Particle Swarm Optimization method flowchart.

## 2.7 The DC/DC Boost Converter Circuit

As mentioned before, a photovoltaic array should not be directly connected to a load because the operating point of the photovoltaic array will mostly differ from the maximum power point due to the impedance seen by it. That is why the DC/DC Boost converter is connected between the photovoltaic array and the load. This means that now the load seen by the photovoltaic array is the DC/DC converter. In this thesis, the load consists of two 12-volt batteries connected in series. The purpose of the MPPT algorithm is to change the duty cycle of the transistor in the DC/DC Boost converter so that the desired voltage at the DC/DC converter input is attained. In paragraph 2.5.3 it was shown that the equation that gives the voltage that the DC/DC Boost converter outputs is the following:

$$V_0 = \frac{V_S}{1 - D} \quad (2.13)$$

Since, in case of a battery load, the output voltage is fixed, the previous equation is transformed into:

$$V_S = V_0 * (1 - D) \quad (2.14)$$

The voltage at the DC/DC converter input is the voltage provided by the photovoltaic array, so essentially by shifting the duty cycle of the DC/DC converter the operating point in the I-V curve of the photovoltaic array is shifted until the maximum power point is attained. The component that is responsible for the modulation of the DC/DC converter duty cycle is a microcontroller.

The microcontroller in this thesis is an Arduino Uno. On that, the MPPT algorithm is executed. The microcontroller senses the input voltage and current of the DC/DC converter (i.e. current and voltage provided by the photovoltaic array) and produces the PWM signal that controls the transistor of the DC/DC Boost converter.

In this thesis the transistor that acts as a switch for the DC/DC converter is a MOSFET. The electrical conductivity of this device is determined by the voltage of the gate; the capability to

adjust conductivity depending on the voltage that is applied can be used to amplify or switch electronic signals. With increasing miniaturization, it is easy to scale MOSFETs down to smaller dimensions. Their power consumption is considerably less than bipolar transistors, and also allow for much higher density. Thus, BJTs are about twenty times larger than MOSFETs, which also have higher switching speeds.

Between the MOSFET's gate and the other terminals, there is a large stray capacitance, which has to be charged each time the MOSFET turns on and discharged each time it turns off. The gate capacitor has to be charged to a specific voltage and above for the transistor to turn on, as a transistor requires a specific gate voltage. Similarly, the gate must be discharged to turn-off the MOSFET.

The transition from a conducting to a non-conducting state and vice versa does not happen immediately when switching the transistor. The switching time must therefore be kept short and for that reason a gate driver is required. A gate driver is an amplifier that receives the low power signal which is produced by a microcontroller and produces a sufficient amount of current to charge the gate of a MOSFET. The signal coming from the microcontroller is the PWM signal generated by the microcontroller MMPT algorithm.

An initial burst of current must be pumped into the gate to turn on FETs and then a holding voltage is required to keep the transistor on. To discharge the gate capacitance, a similar burst of opposite polarity current is required for turn-off, and then a '0' or negative holding voltage is required to keep the transistor off. It is also critical that a low impedance to the switch gate is present in the gate-drive circuit that provides this supply.

# 3. Q-LEARNING ALGORITHM FOR MPP TRACKING

---

## 3.1 Reinforcement learning

The tutoring of Machine Learning (ML) models to undertake a series of decisions is called reinforcement learning. A computational agent is known to be able to explore a discrete, finite environment. In an unpredictable, potentially complex environment, the agent learns to attain a goal. Artificial intelligence faces a scenario similar to a game in reinforcement learning. In order to find a solution to the problem at hand, the machine uses a trial and error method. For the actions it performs the artificial intelligence earns either a positive or a negative (as punishment) reward in order to get the system to do what the programmer desires [10]. Its objective is to maximize the overall reward. While the programmer is the one that sets the reward policy, the model is given zero direction or suggestions/hints on how to solve the problem. It is up to the model to figure out how to maximize the reward by executing the task, beginning with completely random trials and ending with sophisticated tactics. The agent has to choose from a finite set of actions during exploration [11]. The agent is considered as a controller of a controlled Markov process. A collection of state-action can lead to a reward  $r_n$  obtained by the agent, as described before, a reward that can be the same or different for every set of state-action. In accordance with the following equation, each state probabilistically changes to  $y_n$ :

$$Prob[y_n \equiv y | x_n, a_n] = P_{x,y[a]} \quad (3.1)$$

where:

- $P_{xy}[\alpha]$  is the probability that under action  $a$  the agent moves from state  $x$  to  $y$
- $\alpha$  is the action
- $x$  is the present state
- $y$  is the state the agent will visit provided that it takes action  $a$  from state  $x$ .

In order to assist the agent to maximize the overall expected discounted reward, the algorithm has to evaluate the most optimum policy. The discounted reward is the reward that can be earned in  $s$  number of steps. It is used to achieve balance between future and immediate rewards, so that rather than future ones, immediate rewards are worth more [12]. The higher value of the immediate reward is based on one value which is the discount factor  $\gamma$ . In  $s$  number of steps this discount factor  $\gamma^s$  is multiplied by the future reward.

The value of state  $x$ , under a policy  $\pi$  is calculated by the following equation:

$$V^\pi(x) = R_x(\pi(x)) + \gamma \sum_y P_{xy}[\pi(x)] V^\pi(y) \quad (3.2)$$

where:

- $P_{xy}[\pi(x)]$  is the probability that under action  $\pi(x)$  the agent moves from state  $x$  to  $y$
- $R_x$  is the immediate reward
- $\gamma$  is the discount factor
- $x$  is the present state
- $y$  is the state the agent will visit provided that it takes action  $a$  from state  $x$
- $\pi^*$  is the action selection policy and
- $V$  is the value of a given state.

The actions that the agent chooses to perform are recommended by the policy  $\pi$ . It receives an immediate reward by choosing this action and goes to a particular state whose value and probability are equal to  $V_\pi(y)$  and  $P_{xy}[\pi(x)]$  respectively. Dynamic Programming (DP) theory guarantees that one, at least, ideal stationary policy  $\pi$  exists that is provided by:

$$V^*(x) = V^{\pi^*}(x) = \max_a \left\{ R_x(a) + \gamma \sum_y P_{xy}[a] V^{\pi^*}(y) \right\} \quad (3.3)$$

where:

$R_x$  is the immediate reward

$P_{xy}[a]$  is the probability that under action  $a$  the agent goes from state  $x$  to  $y$

$a$  is the action

$\pi^*$  is the action selection policy and

$V$  is the value of a given state.

The reinforcement learning algorithm used in this study is called Q-learning and it is an algorithm that is model-free (MF). The transition probability distribution is not required for a MF algorithm to solve a problem. Thus, an MF algorithm is called an "explicit" algorithm for trial and error.

The update function is defined by the following equation:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha * [R_t + \gamma * \max Q(S_{t-1}, a_{t-1}) - Q(S_t, a_t)] \quad (3.4)$$

where:

$\alpha$  is the learning rate

$R_t$  is the immediate reward

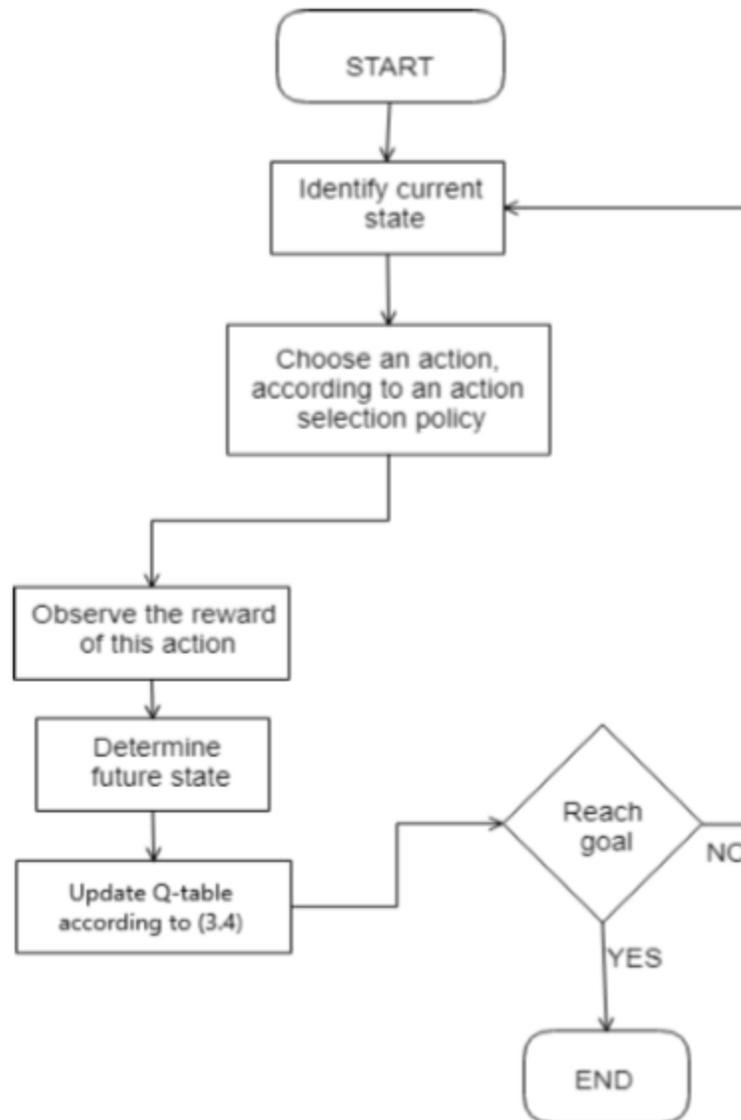
$\gamma$  is the discount factor

$Q(S_{t+1}, \alpha_{t+1})$  is the next state Q value and

$Q(S_t, \alpha_t)$  is the current state Q value.

The learning rate defines the degree to which newly acquired data override old data. A 0 value prevents the agent from learning anything (using prior knowledge exclusively), while a 1 makes the agent just consider the latest data. A learning rate of  $\alpha_t=1$  is optimal in fully deterministic environments. If the problem is stochastic, the algorithm converges on the learning rate under some technical conditions that need it to become zero. A constant learning rate is sometimes used, in practice, for example all  $t$  have  $\alpha_t = 0$ .

The importance of future rewards is determined by the discount factor. By just considering current rewards, the agent will become short-sighted with a value of 0, while a value approaching 1 will make it aim for a high long-term reward. For  $\gamma=1$ , all environmental histories become indefinitely long without a terminal state. Even with a discount factor just marginally lower than 1, Q-function learning results in the propagation of errors and instabilities when the value function is approximated with an artificial neural network. In that case, learning is sped up by beginning with a discount factor with a lower value which then increases towards a final value. The Q-learning algorithm flowchart is presented in Figure 16.



**Figure 16.** The Q-learning algorithm process.

The following basic steps should explain all of the above:

**Step 1:** The state that it is visited is identified by the agent. To help the agent distinguish various states and not confuse them as identical, a discretized state-space is required.

**Step 2:** Depending on the action selection policy, the agent chooses the appropriate action. The actions are the parameter that enables the agent to explore the state-space.

**Step 3:** According to the outcome of the previous step, the reward is earned.

**Step 4:** The Q-table is updated once the future action is determined.

**Step 5:** Finally the agent examines if the objective has been attained.

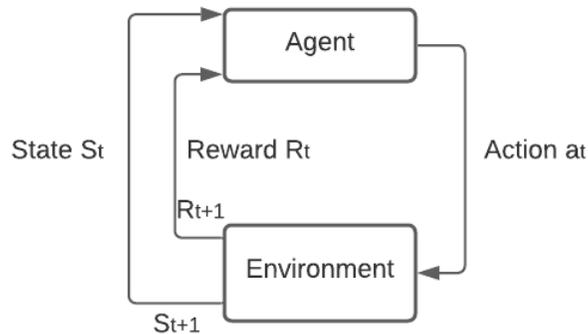
## 3.2 Q-Learning for MPP tracking

In real life, sequential decision-making is a common issue, such as a child attempting to walk by extending a leg forward and turning his body. The child will have a chance to achieve his target by taking a series of acts (keep moving forward). A systematic framework for describing this sequential problem of decision-making is given by the Markov decision process (MDP). Learning by engaging with the world is the innate learning ability of a person. The individual will receive feedback after interacting with an unfamiliar system according to their own understanding of the system. A criterion for determining how "good or bad" an action is under a particular situation is given by this feedback signal. Actions with better results will have a greater chance of being picked, whereas actions with bad results will have a smaller chance of being carried out in the future. A discrete time stochastic control process is what the PV MPPT problem is. When results are both partly random and under the influence of a decision-maker, it provides a mathematical basis for modeling decision making. Due to the above, it can be regarded as a Markov Decision Process (MDP). To model the interaction between the agent and the controlled environment, an MDP is used. It is comprised of:

1. The state-space,
2. The set of all actions and
3. The reward obtained by selecting the action  $a$  while in state  $s$  which causes the agent to go to state  $s'$ .

For inspecting optimization issues, like the MPP tracking problem, MDPs are useful. Using a temporal difference Q-learning algorithm is an effective way of solving problems like that. The aim of Q-learning is to figure out an action selection policy that will maximize the total received discounted rewards over the future.

The basic principle of the Q-learning algorithm is presented in Figure 17:



**Figure 17.** Simplest form of Q-learning.

It was stated before that from a collection of observations, states, actions and rewards, does a Q-learning algorithm learn. The agent observes its current state during each time-step. After that, depending on its  $\pi$  action selection policy, it selects its action. The future state ( $S_{t+1}$ ) is observed after the selection of an action. It then gets rewarded with an  $R_t$  reward and chooses the highest Q-value between the future values over actions,  $\max_a Q(S_{t+1} | a)$ . Learning rate  $\alpha$  determines, based on the current Q-value, the effect of the estimate over the last Q-value in the last Q-update [13].

### 3.2.1 Action selection policy

Each pair of state-action must be estimated so that every Q-value after its calculation is the optimal one. All possible actions should be chosen for every state visited by the agent. This can be done by the selection of the most appropriate action policy. Boltzmann exploration has been used in this study. For sequential decision-makers under uncertainty, Boltzmann exploration is a classic technique and is one of the most common instruments in Reinforcement Learning. This strategy requires an action with weighted probabilities instead of either choosing the best action or taking a random action. By the following equation, is each probability calculated:

$$p(S, a_i) = \frac{e^{\frac{Q(S, a_i)}{\tau}}}{\sum_{a_i} e^{\frac{Q(S, a_i)}{\tau}}} \quad (3.5)$$

where:

$\tau$  is the Boltzmann exploration parameter

$a_i$  is the taken action and

$Q(S, a_i)$  is the Q-table cell current value.

The  $\tau$  element is known as temperature. It determines the randomness of the action selection. If temperature  $\tau$  is high, every probability value is close to each other. This concludes with a selection of an arbitrary action, which relies on the variable  $N_r$ , which is an arbitrary number between 0 and 1. Provided that the  $N$  number of visits is increased for a given state, the temperature is reduced. The temperature  $\tau$  is equal to the minimum value when a specific number of visits  $N$  have taken place. This implies that the exploration has reached its end and the agent greedily chooses the highest Q-value action. Below is the presentation of the temperature function:

$$\tau = \begin{cases} \tau_{min} + \left(1 - \frac{N}{N_{max}}\right) * (\tau_{max} - \tau_{min}), & (N \leq N_{max}) \\ \tau_{min}, & (N \leq N_{max}) \end{cases} \quad (3.6)$$

where:

$\tau_{min}$  is the minimum value for the temperature

$\tau_{max}$  is the maximum value for the temperature

$N$  is the number of a specific state's visits and

$N_{max}$  is the maximum number of visits for a state.

### 3.2.2 State-Space

In this way of the state-space definition, the current state depends not only on the present duty cycle and power generated by the photovoltaic modules but also from the previous time-step duty cycle. Better accuracy is achieved this way. This however takes a longer learning time. There is also a discretized state-space. It is described by the following formula:

$$state = \{S | S_{i,j,p} = (D_i, P_j, DO_k), i \in [1,2,3 \dots n], j \in [1,2,3 \dots m], k \in [1,2,3 \dots p]\} \quad (3.8)$$

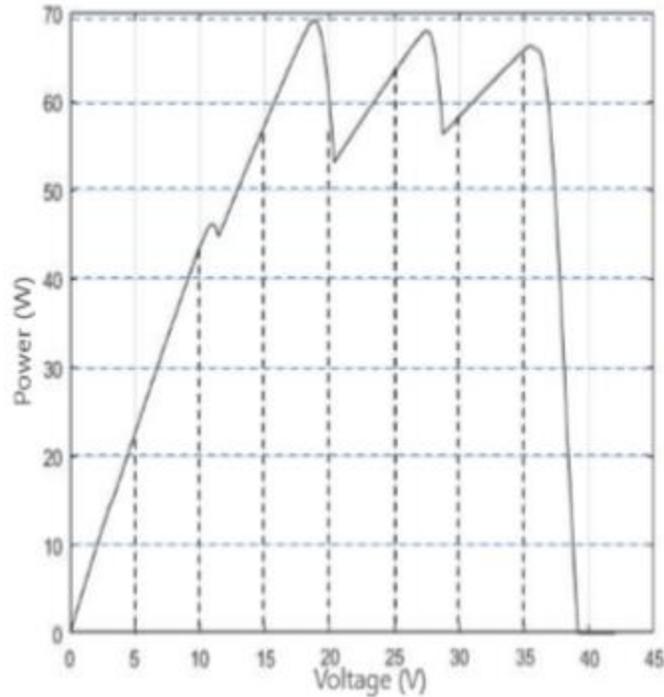
where:

$n$  is the number of steps the duty cycle (D) is equally quantized in

$m$  is the number of steps the PV power (P) is equally quantized in and

$k$  is the number of steps of the of duty cycle of a previous time-step (DO) is equally quantized in.

The theoretical values that the duty cycle can take is between 0 and 1, but due to operational limitations of the DC/DC power converter, practically it cannot take values close to the upper and lower limits. For that reason it was bound by the values of 0.19 and 0.93. A better approach to the MPP would be concluded with large numbers of  $n$ ,  $m$ , however this entails more time to learn. On the contrary, in a shorter learning time, small numbers of equally quantized steps will result to worse MPP. A trade-off between precision and learning time is applied. The discrete state-space is defined using the power and duty cycle steps which are the horizontal and vertical dotted lines, respectively, shown in Figure 18. Here, the duty cycle is quantized in 8 equal steps and the photovoltaic array output power is quantized in 7 steps. The state is jointly determined by both the steps in which the power and duty cycle belong to. It should be noted that each duty cycle value corresponds to a particular voltage of the photovoltaic array. If a duty cycle for instance corresponds to a power of 53 Watts and a voltage of 18 Volts, the state of the agent is presumed to (6.4), since 53 Watts belong to the 6th step of the photovoltaic output power and 18 Volts belongs to the 4th step of the voltage.



**Figure 18.** Discrete state-space using duty cycle and power steps.

### 3.2.3 Action-space

The action-space set of the system's controller is represented below:

$$A = \{a\} = \{+0.04, -0.04, +0.12, -0.12, +0.28, -0.28, 0\}$$

These values represent the changes over the current duty cycle, and each corresponds to actions 1 through 7 accordingly.

To ensure that the majority of duty cycle values are picked, a large range of different actions is chosen. This is important so that the likelihood of convergence at an incorrect peak of the photovoltaic array power-voltage curve is minimized. The selection of each action is based on the Boltzmann exploration (analyzed previously). The highest-Q-value action is chosen for that specific state when the exploration is over. The global MPP of the photovoltaic array is considered to be found when the aforementioned is action 7, which is the no-increment action. In

this case, since actions 1 through 6 result in lower power their Q-values are lower than the one for action 7. The point at which the agent is at if action 7 is chosen is considered to be the global MPP. The update equation of the duty cycle is given below:

$$D_{t+1} = D_t + a \quad (3.9)$$

where:

$D_{t+1}$  is the next step duty cycle

$D_t$  is the current step duty cycle and

$a$  is the increment in duty cycle value whether that is positive, negative or zero.

If the current duty cycle is equal to 0.50 and action 3 is chosen by the agent, the next duty cycle value will be  $0.50+0.12=0.62$ . A maximum value (0.93) and a minimum value (0.19) for the duty cycle have been established. It should be noted that if the agent chooses action 7 in a state it will stay in that state for the next step as well.

Actions 1, 3 and 5 are regarded as an increment, and actions 2, 4 and 6 are regarded as a decrement. If the duty cycle meets its minimum value and an even action is chosen, then the value of the duty cycle won't change and the same applies if an odd action besides action 7 is chosen when the duty cycle is equal to its upper bound. All actions need to be chosen in any state that the agent has visited to enable it to learn the optimal route for that specific MPP. This is the reason that the selections for the increase and decrease are graded. This happens because there are several peaks on the photovoltaic array power-voltage curve that must be explored under partial shading. There is a chance of two peaks with no noticeable difference between their values being present. If these two peaks are relatively close it is necessary to make a small change in the duty cycle so that the agent understands which peak is greater. However, if the peaks are widely apart in order to move to the other peak with precision, the agent will need an action with a higher increment or decrement. The agent might falsely consider that the peak it is at is the highest if this action does not exist, because it has not managed to visit a peak higher than that.

### 3.2.4 Reward

The reward is the deciding factor for the most appropriate action. Each value of the duty cycle corresponds to a couple of current-voltage values and a particular power value is produced for that pair. The power values determine the reward according to the following equation that will evaluate the action selection:

$$R_t = \begin{cases} +2, & \text{if } P(t+1) - P(t) > 1 \\ 0, & \text{if } 0 \leq P(t+1) - P(t) \leq 1 \\ -2, & \text{if } P(t+1) - P(t) < 0 \end{cases} \quad (3.10)$$

where:

$R_t$  is the immediate reward

$P(t)$  is the current step's power value and

$P(t+1)$  is the next step's power value.

Every time the difference between the power value in the next step and the power value in the current step is higher than 1 then the agent will be awarded a positive reward for that particular state-action couple to encourage it to hold this map selection. If the aforementioned difference is lower than 1 or equal to zero, it is assumed the produced power hasn't changed, and there would be no reward. Almost all of the time, this reward is referred to action 7. There is a small possibility, however, of the presence of points that have the same power but different duty cycles. Finally, if the difference in the produced power is less than -1 for these two time-steps, then the agent will obtain a negative reward. This happens because when there is an output power reduction, a very small negative Q-value is required. In this way, the probability of the agent reselecting the behavior that caused that power reduction the next time it visits that state is low.

### 3.2.5 Discount Factor and Learning Rate

The importance of future rewards is determined by the discount factor as mentioned before. The agent is seeking, instead of the immediate reward, to increase the total future rewards and find the optimal route.

It must be guaranteed that the MPPT method based on Q-learning will converge. A visit frequency criterion was considered in order to accomplish this. The agent should be urged to consider that states with lower number of visits have data that is more valuable than the data acquired by states with a larger number of visits. Hence the former states will have a higher learning rate than the latter. If a state's number of visits is higher than or equal to 15, the experience gained is considered to be mature enough to exploit. If a state's number of visits is less than 15, it is then assumed that the agent has not yet figured out the action that will give the greatest reward and lead to the GMPP. There has to be a period of time where the agent learns to cope with real environmental conditions that affect the photovoltaic system. The agent is considered trained after that span, and it is presumed that this technique will have a high MPP tracking efficiency in a limited number of time steps.

The value of the learning rate is calculated as follows:

$$\alpha = \frac{k_1}{k_2 + k_3 * N} \quad (3.11)$$

where:

$k_1, k_2, k_3$  are factors that determine the learning rate for a state with 0 visits

$N$  is the number of visits of a specific state and

$\alpha$  is the learning rate.

### 3.3 The overall Q-learning MPPT algorithm

The execution steps of the Q-learning MPPT algorithm are the following:

**Step 1:** Initialization of the Q-table. A Q-table is generated using four dimensions, according to what was stated earlier. Owing to no previous knowledge, this table is loaded with zeros. Furthermore each previously described parameter is set to 0 or to a default value. Finally, a table that includes the number of visits from each state is specified and its values are also initialized to 0. The Q-table size depends on the step quantization number of all the states, which are the current duty cycle, the power produced and finally the previous time-step duty cycle.

**Step 2:** The PV array generated power is measured once the photovoltaic array's voltage and current enter the steady-state. Then, the state is decided.

**Step 3:** The learning rate is calculated.

**Step 4:** The temperature  $\tau$  is calculated as well as each action's probability for that specific state.

**Step 5:** The number of visits is assessed for the specific state. The P&O algorithm is executed if the maximum number of visits has been achieved and the highest Q-value is the one for action 7. In other words, the P&O algorithm is executed to try and find a better approach to fine-tune that point if the state that is attempted to be found is the GMPP.

**Step 6:** An arbitrary number between 0 and 1 is chosen if the above decision is not true. Then the probabilities that were calculated in step 4 are compared to that number. A specific change in the duty cycle is encoded by each probability:

•  $0 < N_{\text{RANDOM}} \leq \text{Prob1}$ , action 1

•  $\text{Prob1} < N_{\text{RANDOM}} \leq (\text{Prob1} + \text{Prob2})$ , action 2

•  $(\text{Prob1} + \text{Prob2}) < N_{\text{RANDOM}} \leq (\text{Prob1} + \text{Prob2} + \text{Prob3})$ , action 3

•  $(\text{Prob1} + \text{Prob2} + \text{Prob3}) < N_{\text{RANDOM}} \leq (\text{Prob1} + \text{Prob2} + \text{Prob3} + \text{Prob4})$ , action 4

•  $(\text{Prob1} + \text{Prob2} + \text{Prob3} + \text{Prob4}) < N_{\text{RANDOM}} \leq (\text{Prob1} + \text{Prob2} + \text{Prob3} + \text{Prob4} + \text{Prob5})$ , action 5

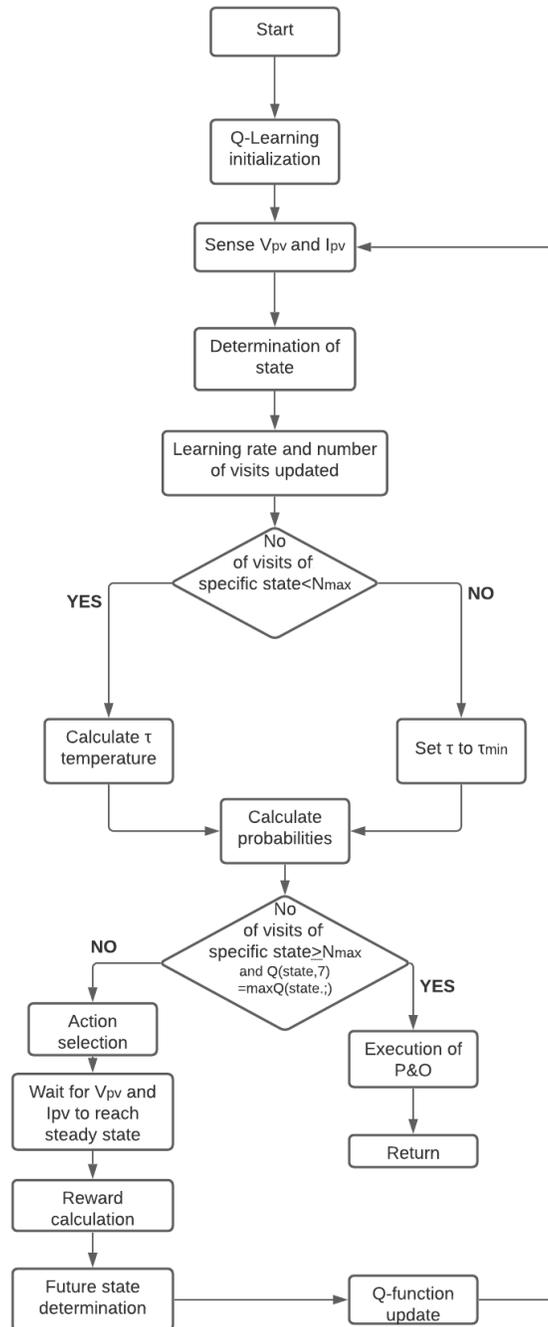
•  $(\text{Prob1} + \text{Prob2} + \text{Prob3} + \text{Prob4} + \text{Prob5}) < N_{\text{RANDOM}} \leq (\text{Prob1} + \text{Prob2} + \text{Prob3} + \text{Prob4} + \text{Prob5} + \text{Prob6})$ , action 6

- *else, action 7.*

**Step 7:** The algorithm before sensing, waits for the voltage and current signals of the photovoltaic array to reach the steady-state, after changing the action. The difference between the previous and the current time-step power is then calculated. Finally the reward is determined.

**Step 8:** the algorithm returns to **step 2** once the state of the next time-step is decided, and the Q-function is updated.

The flowchart of the MPPT algorithm based on Q-learning is presented in Figure 19:



**Figure 19.** The flowchart of the Q-Learning algorithm for MPP tracking.

# 4. IMPLEMENTATION OF THE GMPPT SYSTEM

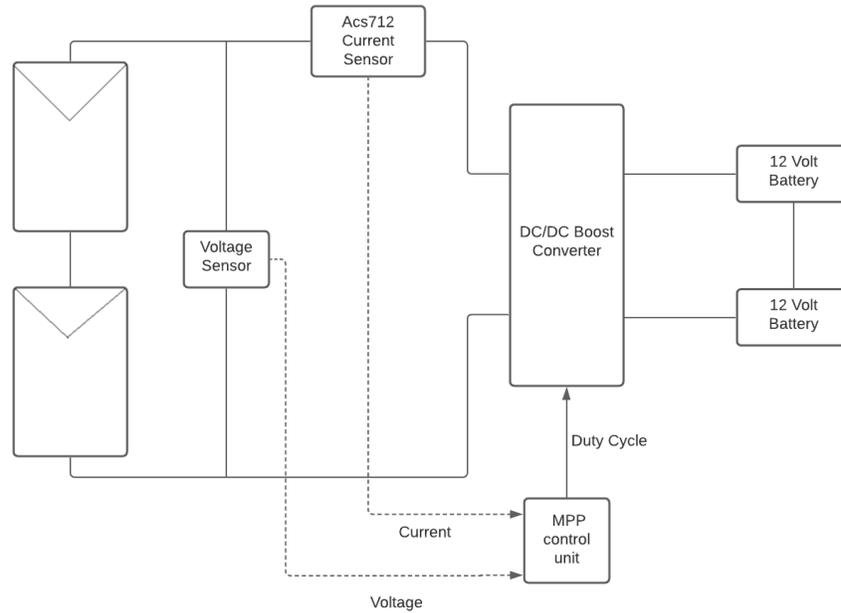
---

## 4.1 Introduction

A block diagram of the overall Global MPPT system, which was developed in this thesis, is presented in Fig. 20. It consists of :

1. A photovoltaic array consisting of two photovoltaic modules connected in series,
2. An MPPT control unit,
3. A DC/DC Boost converter and
4. Two 12V batteries connected in series.

The design and operation of the individual components of the GMPPT system, are described in the following paragraphs.



**Figure 20.** Block diagram of the overall Global MPPT system

## 4.2 The microcontroller-based control unit

As mentioned before, the reason that the Q-learning algorithm is needed is to adjust the duty cycle of a DC/DC Boost converter so that the maximum power that a photovoltaic array is able to provide under specific irradiance is extracted. In other words, a microcontroller programmed with the Q-learning algorithm is needed that senses the current and the voltage provided by the photovoltaic array and produces the PWM signal that controls the DC/DC Boost converter's MOSFET switch. The microcontroller used in this thesis is an Arduino Uno microcontroller.

The Arduino Uno is a board based on the ATmega328P microcontroller and is open source. It has 14 digital and 6 analog I/O pins, while 6 of the digital pins are capable of producing a PWM signal, and is programmable through a type B USB cable. Both a USB cable and an external +9V battery can power it, although voltages ranging from +7 to +20 volts are accepted. The Arduino Uno has 32 KB of flash memory, 2 KB of SRAM and 1 KB of EEPROM.

As mentioned before the Arduino Uno has 6 digital I/O pins that are capable of producing a PWM output. The problem is that the switching frequency of every pin is too low for this

implementation's needs. Pins 3, 9, 10, 11 can produce PWM signals at 490 Hz, whereas pins 5 and 6 can produce PWM signals at 980 Hz. This method does not use the `analogWrite()` function that would normally be used when a PWM signal needs to be created. The `analogWrite()` function gives the hardware PWM a simple interface but does not provide any frequency control.

There are three PWM timers on the ATmega168P/328P chip, controlling 6 PWM outputs. More control over frequency can be obtained than `analogWrite()`, by directly manipulating those timer registers.

The Atmega 328 has 3 counters and normally runs at 16 MHz. A PWM signal is generated by a counter through clock-tick counting, a comparator and a register. The purpose of the counter is to generate the duty cycle resolution. One full counter cycle is one period of the PWM signal. The size of the counter (or in other words the resolution of the duty cycle) is 8 bits, corresponding to 256 values. So the 16MHz clock is divided by 256 and can give a maximum frequency of 62.5kHz. The clock can be divided by factor to obtain lower frequencies. This is called pre-scaling. For setting and creating the duty cycle output, the register and the comparator are used. The register sets for how long the output is high during each cycle. One would for example, set the register to  $256 * 0.25 - 1 = 63$  for a 25 percent duty cycle. Comparing the value of the register with the current counter value is what the comparator is used for. The comparator produces a logic high value in the output if the register value is equal to or lower than the current counter value. Otherwise the comparator produces a logic low value.

### **Timer Registers**

To manage each timer, a few registers have to be used. The key control bits for the timer are held by the Timer/Counter Control Registers TCCRnA and TCCRnB. There are many groups of bits in these registers:

- Waveform Generation Mode bits (WGM): these bits control the overall mode of the timer. (these bits are split between TCCRnA and TCCRnB).
- Clock Select bits (CS): these control the clock prescaler.
- Compare Match Output A Mode bits (COMnA): these enable/disable/invert output A.

- Compare Match Output B Mode bits (COMnB): these enable/disable/invert output B.

The levels at which the A and B outputs will be affected are set by the OCRnA and OCRnB Output Compare Registers. The corresponding output will be changed, as defined by the mode when the timer value matches the register value.

In this case the maximum frequency is chosen with an 8 bit resolution. From what was explained above, a PWM signal at 62.5 KHz is obtained.

### 4.3 Programming of the Q-learning algorithm

The Q-Learning algorithm that was described in Chapter 3 was developed for a theoretical model, a simulation. Simulink was used to simulate the PV arrays, the microcontroller and the DC-DC Boost converter [14]. The advantage of simulating a model like this case's model is that one can use the resources of the computer, which includes processing power, memory etc. The problem that occurs when trying to bring this model to life with real microcontrollers is that the resources provided by the computer are no longer available. As mentioned above, the Arduino Uno has 32 kb of flash memory and only 2 kb of SRAM.

In Chapter 3 it was mentioned that  $Q(S_t, a_t)$  values and the number of visits for every state-space  $N$  are stored into separate tables which in coding terms corresponds to arrays. It was also mentioned that in this implementation, the  $Q$  values depend on current duty-cycle, power provided by the PV array, previous time-step duty-cycle, and the action the agent has to take. That means that the array that is used to store  $Q$  values in has to have 4 dimensions: 1) number of quantized steps of the duty cycle, 2) number of quantized steps of power, 3) number of quantized steps of the duty cycle of previous time steps, 4) 7 actions. If the right number of quantized steps is chosen so that high accuracy with low convergence time is obtained, a  $Q$  array that is too large to be stored into 2 kb of SRAM is formed. So, the question that arises is how large  $Q$  and  $N$  arrays are stored without having to use a computer. Previously the Arduino was discussed and the fact that the microcontrollers can use shields was mentioned. The answer to the above question came in the form of a shield. More specifically an SD card reader. What that shield can

do is allow the communication between the Arduino microcontroller and an SD card. Thus, an SRAM is not needed to store these two large arrays.

However, storing the arrays in that way is quite a challenge. The only way the values of the array in an SD card can be stored, is by writing them in a file, whether that is a csv or as in this case a simple text file. Starting in the setup the files are created and initialized with 0. The challenging part is retrieving the data written in the file or updating a value. Since each value of the array is stored in a single line, a way had to be found to turn the array dimensions into in-file lines. What that means is that if, for instance, the value of  $Q[0][0][0][0]$  needs to be retrieved from the file the first line would be returned. Since that method of turning array dimensions into in-file lines does not exist, a simple algorithm had to be created through experimentation. The method followed to create the equation that makes this conversion possible is presented below.

As mentioned above the Q array has 4 dimensions which are  $D1_{\max} = 8$ ,  $D2_{\max} = 30$ ,  $D3_{\max} = 10$  and  $D4_{\max} = 7$  in its early stage. Those values were later tackled so that a faster convergence is accomplished. Thus, the array is  $Q[D1][D2][D3][D4]$ . If every instance of the array is expanded, every instance can be assigned to an in-file line, however the problem is that this implementation's array has  $18 \cdot 30 \cdot 10 \cdot 7 = 37.800$  elements and each instance cannot be assigned to a line manually. So what can be done is to take small numbers of instances and observe the pattern. The following board will give a better understanding to the above procedure.

The way array values are stored into the file is by exhausting every dimension first which means that when starting from  $[0][0][0][0]$  D1 is exhausted first. That method can be seen in Table 1.

**Table 1.** Array to in-file line for 1 dimension.

<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>IN-FILE LINE</b>
0	0	0	0	0
1	0	0	0	1
2	0	0	0	2
3	0	0	0	3
.	.	.	.	.
.	.	.	.	.
18	0	0	0	18

As seen while every other dimension is 0, the in-file line is the same as D1. So it is now easy to understand that from  $[0][0][0][0] \rightarrow [18][0][0][0]$  the equation is:  $ID\_line = D1$ .

**Table 2.** Array to in-file line for 2 dimensions.

<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>IN-FILE LINE</b>
0	1	0	0	19 or 18+1+0
1	1	0	0	20 or 18+1+1
2	1	0	0	21 or 18+2+1
3	1	0	0	22 or 18 +3+1
.	.	.	.	.

.	.	.	.	.
18	1	0	0	37 or 18+18+1
0	2	0	0	38 or 36+2+0
1	2	0	0	39 or 36 +2+1
2	2	0	0	40 or 36+2+2
3	2	0	0	41 or 36+2+3
.	.	.	.	.
.	.	.	.	.
18	2	0	0	56 or 36+2+18

While D3 and D4 are 0 the in-file line can be written as a combination of D1 and D2. Taking a closer look (Table 2) it can be seen that  $19=1*18+1+0$ ,  $20=1*18+1+1$ ,  $21=1*18+2+1$ , and  $22=1*18+3+1$ . From the above observation the equation can be transformed from  $ID\_line=D1$  to  $2D\_line= D2*D1_{max}+D2+D1$ ,

where:

$2D\_line$  is the in-file line if the Q-array had 2 dimensions

$D2$  is the current value of the second dimension of the Q-array

$D1_{max}$  is the maximum value of the first dimension of the Q-array

$D1$  is the current value of the first dimension of the Q-array.

It can also be checked if that equation is acceptable for the instances of the first Table(Table 1). D2 can be easily set to 0 and it can be confirmed that this equation is valid for the first Table (Table 1). Now the equation when a third dimension is greater than 0 will be created.

**Table 3.** Array to in-file line for 3 dimensions.

<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>IN-FILE LINE</b>
<b>18</b>	<b>30</b>	<b>0</b>	<b>0</b>	<b>588</b>
0	0	1	0	589 or $1*588+1+0$
1	0	1	0	590
2	0	1	0	591
3	0	1	0	592
.	.	.	.	.
.	.	.	.	.
18	0	1	0	607
0	1	1	0	608
1	1	1	0	609
2	1	1	0	610
3	1	1	0	611
.	.	.	.	.
.	.	.	.	.
18	30	1	0	626

It is observed in Table 3 that before line 589, the equation of line 588 has every dimension of  $2D\_line$  in its maximum value. This means that for line 588 the equation becomes  $2D\_line_{max} = D2_{max} * D1_{max} + D2_{max} + D1_{max} = 30 * 18 + 30 + 18 = 588$ ,

where:

$D1_{max}$  is the maximum value of the first dimension of the Q-array

$D2_{max}$  is the maximum value of the second dimension of the Q-array.

Now if line 589 is observed (Table 3), it is seen that  $3D\_line = D3 * 2D\_line_{max} + D3 + 2D\_line = D3 * [D2_{max} * D1_{max} + D2_{max} + D1_{max}] + D3 + [D2 * D1_{max} + D2 + D1]$ ,

where:

$2D\_line$  is the in-file line if the Q-array had 2 dimensions

$3D\_line$  is the in-file line if the Q-array had 3 dimensions

$D2$  is the current value of the second dimension of the Q-array

$D1_{max}$  is the maximum value of the first dimension of the Q-array

$D2_{max}$  is the maximum value of the second dimension of the Q-array

$D1$  is the current value of the first dimension of the Q-array

$D3$  is the current value of the third dimension of the Q-array.

At this point, 3 equations for converting array dimensions (of 1, 2 and 3 dimensions) into in-file lines have been created. Although the same procedure could be followed to create the fourth and final equation for converting the Q array's dimensions into in-file lines, there is a much faster method of implementing that process. For that purpose, the three equations developed should be considered:

$$1D\_line = D1 \quad (3.12)$$

$$2D\_line = D2 * D1_{max} + D2 + D1 \quad (3.13)$$

$$3D\_line = D3 * [D2_{max} * D1_{max} + D2_{max} + D1_{max}] + D3 + [D2 * D1_{max} + D2 + D1] \quad (3.14)$$

It can be seen from the three equations (3.12)-(3.14) that a recurring pattern appears. To get  $2D\_line$ ,  $1D\_line$  is assigned its maximum value, resulting in  $1D\_line_{max}$ , and it is multiplied by the current  $D2$  value. Then the current  $D2$  value and the  $1D\_line$  are added. In other words 3.13 can be written as  $2D\_line = D2 * 1D\_line_{max} + D2 + 1D\_line_{max}$ . Now if  $3D\_line$  is examined it can be seen that the exact same pattern appears. To get  $3D\_line$ ,  $2D\_line$  is assigned its maximum value, resulting in  $2D\_line_{max}$  and it is multiplied by the current  $D3$  value. Then, the current  $D3$  value and the  $2D\_line$  were added. In other words (3.14) can be written as  $3D\_line = D3 * 2D\_line_{max} + D3 + 2D\_line$ . That way the fourth and final equation can be easily obtained, as follows:

$$4D\_line = D4 * 3D\_line_{max} + D4 + 3D\_line = D4 * [D3_{max} * [D2_{max} * D1_{max} + D2_{max} + D1_{max}] + D3_{max} + [D2_{max} * D1_{max} + D2_{max} + D1_{max}]] + D4 + [D3 * [D2_{max} * D1_{max} + D2_{max} + D1_{max}] + D3 + [D2 * D1_{max} + D2 + D1]] \quad (3.15)$$

where:.

$4D\_line$  is the in-file line if the Q-array had 4 dimensions

$3D\_line$  is the in-file line if the Q-array had 3 dimensions

$2D\_line$  is the in-file line if the Q-array had 2 dimensions

$D2$  is the current value of the second dimension of the Q-array

$D1_{max}$  is the maximum value of the first dimension of the Q-array

$D2_{max}$  is the maximum value of the second dimension of the Q-array

$D1$  is the current value of the first dimension of the Q-array

$D_3$  is the current value of the third dimension of the Q-array

$D_{3_{max}}$  is the maximum value of the third dimension of the Q-array.

So the only thing that was left to do was to create a function that takes the four dimensions of the Q array and using the above equation converts them into in-file lines. Later this function was called inside two other functions. These two functions were created for retrieving data from lines of the file or updating values stored in these lines.

As mentioned above, a way of converting arrays into in-file lines had to be created. The same equation that was created for the previous step will now be used for retrieving or updating an already existing value stored in the file. There are 2 ways someone can approach that task. The first is to read every character of a line up to the new-line character, store the entire string that is read and compare the line that it is currently at with the line that was asked to be read. If the current line is not the line that was asked to be read then it “dumps” the string that it has read up to this point and then proceeds to read the next line. Essentially, this is a sequential search and at large files (such as those formed in the system developed in this thesis) the time it takes to find a value might be quite large since it is reading everything until it finds what is required. The second way and also the way that was used here, is an Arduino function called `MyFile.seek()`. `MyFile` is the file that has to be searched and `seek()` points to the position that is used as its parameter. However, the in-file line cannot be used as a parameter in `seek()` and it cannot be expected for the line to be found. The problem here is that `seek()` does not operate with in-file lines but with bytes. What this means is that if for instance line 100 needs to be visited and what is in it needs to be read, `seek(100)` cannot be used. The type of the value that is stored in every line has to be known and every line has to have a fixed length. So, if in every line a float is stored, in order to go to line 100 and read what’s in it, `seek(100*4)` has to be used, 4 of course being the size of the float. Once that step was taken all that had been left was to write the two functions:

- 1. Retrieve function:** Convert array dimension that was asked into an in-file line through the equation mentioned in the previous chapter, then use `file.seek()` to go to the line, and finally copy the entire line into a string which is later returned.
- 2. Update function:** Convert array dimension that was asked into an in-file line through the same equation, then use `file.seek()` to go to the line, and finally copy the value that was passed as a parameter to that function into the specific line, deleting any value stored in that line previously.

Previously it was discussed that arrays can't be used to store values. The entire algorithm is written in the Arduino IDE, which means that the language used to code the entire Q-learning algorithm was a version of C. In C, when a value has to be stored in an array it is simply written as: `array[x]=value`, so if this code was able to use arrays, then storing a Q value into the Q-array would be as simple as writing: `Q-array[x]=value`. Also, retrieving the value from an instance would be as simple as writing: `RetrievedValue = Q-array[x]`. Since files are used instead of arrays, the equations described earlier were used to take the place of the simple assignment equations that would have been used otherwise:

`Q-array[x]=value -> UpdateFunction(value, inFileLine(x))`

`RetrievedValue = Q-array[x] -> RetrievedValue =RetrieveFunction(inFileLine(x))`

So, when the entire algorithm was translated from theory to actual code, whenever and wherever an equation between arrays would be used, the above functions were used instead. This was the only part that this code differed from the one that would have been developed if arrays could be used. The rest of the code was developed by translating the algorithm steps, shown in the algorithm flowchart (i.e. Fig. 19) into actual code.

## 4.4 I-V and P-V curve tracer

Finally, the last part of the code was developed for measuring the current-voltage (I-V) curve of a PV array. The idea behind this method is to start with a 0% duty cycle, then sense the current I

and voltage  $V$  provided by the PV array, and then increase the duty cycle by a small step repeating the sensing of  $I$  and  $V$  up until a 100% duty cycle is reached. At that point every possible current and voltage will have been sensed and the I-V curve can be created by plotting the values that were sensed.

The circuit used for the Q-learning algorithm, is used for that process as well. This means that if the MOSFET has to be controlled through the Arduino a fast PWM signal has to be used in the same way that a fast PWM signal is used to control the MOSFET in the Q-learning algorithm part [15].

For the I-V curve tracer two methods of control were developed. The first was an automatic sweeper. What it did was to start with a 0% duty cycle, sense the voltage and current and then increase the duty cycle by 1 step repeating the above process till it reached 100%. Since a fast PWM signal of 62.5 kHz is used, 256 steps are obtained. Therefore 0-255 corresponds to 0-100% duty cycle. Then, the values sensed were sent to a software program called MegunoLink in order to be plotted.

The second method of control was a manual sweeper, where the one controlling the tracer is responsible for setting its parameters. In the manual sweeper, step presets and frequency presets are used. Therefore, a step for increasing the duty cycle and also the frequency of the PWM can be chosen [16].

## **4.5 Components of the DC/DC Boost converter circuit**

### **ACS712:**

In manufacturing, commercial, and communications networks, the ACS712 offers affordable and accurate solutions for sensing either AC or DC current. The unit consists of an accurate, low-offset linear Hall sensor circuit located near the surface of the die with a copper conduction path. The applied current that flows through this path produces a magnetic field which is sensed and converted into a proportional voltage by the integrated Hall IC. The precision of the system is optimized by the magnetic signal's close proximity to the Hall transducer. The low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed after packaging for precision,

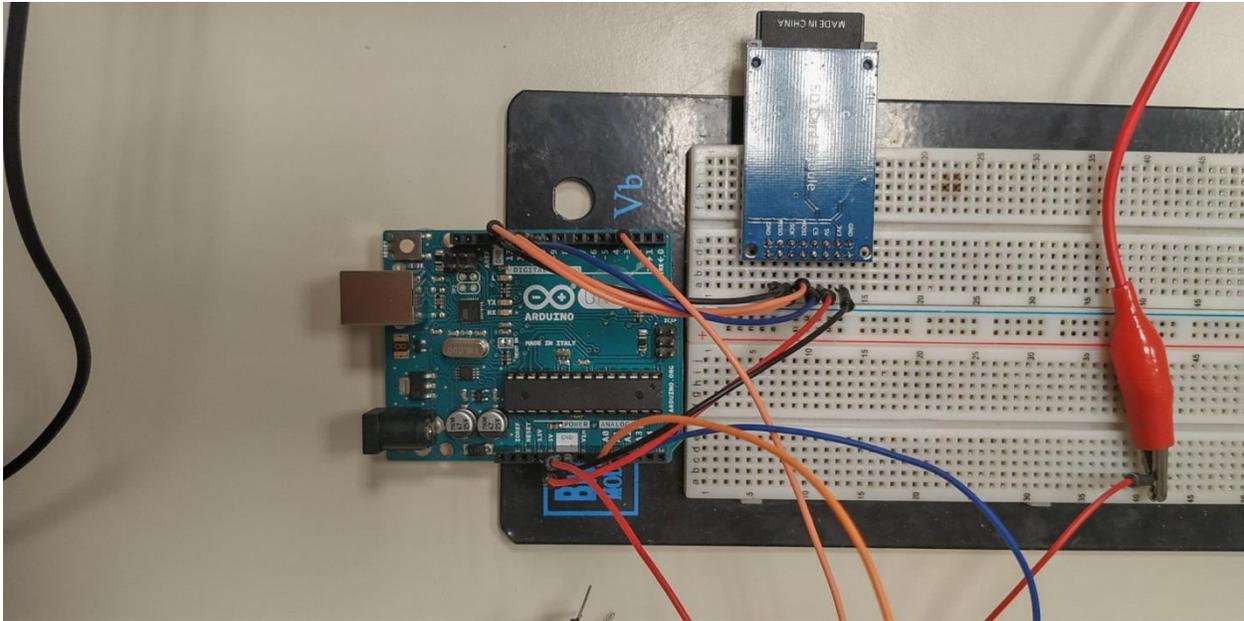
provides an accurate, proportional voltage. If the current flow through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4) is increasing, the output of the device has a positive slope, which is the path used for current sensing. This conductive path's typical internal resistance is 1.2 m $\Omega$ , providing low power losses. The thickness of the copper conductor enables the system to operate under up to 5 times overcurrent conditions. The conductive path terminals are electrically isolated from the sensor leads (pins 5 through 8). That enables the ACS712 current sensor to be used without the use of opto-isolators, or other expensive isolation techniques in applications that require electrical isolation.

### **LM358:**

Low power drain, a common mode input voltage range extending to ground/VEE, and single supply or split supply operation are featured by these dual operational amplifiers using circuit designs designed for Quad Operational Amplifiers. In single supply applications, these amplifiers have many distinct advantages over common types of operational amplifiers. They can operate at supply voltages as low as 3 V or as high as 32 V, with quiescent currents around one-fifth of those associated with the MC1741. The necessity for external biasing components is eliminated by including the negative supply to the common mode input voltage range. The negative power supply voltage is also included in the output voltage range. This op-amp was used to amplify the current and voltage sensed from the photovoltaic array.

### **DC- DC Boost converter, Arduino microcontroller and PV's**

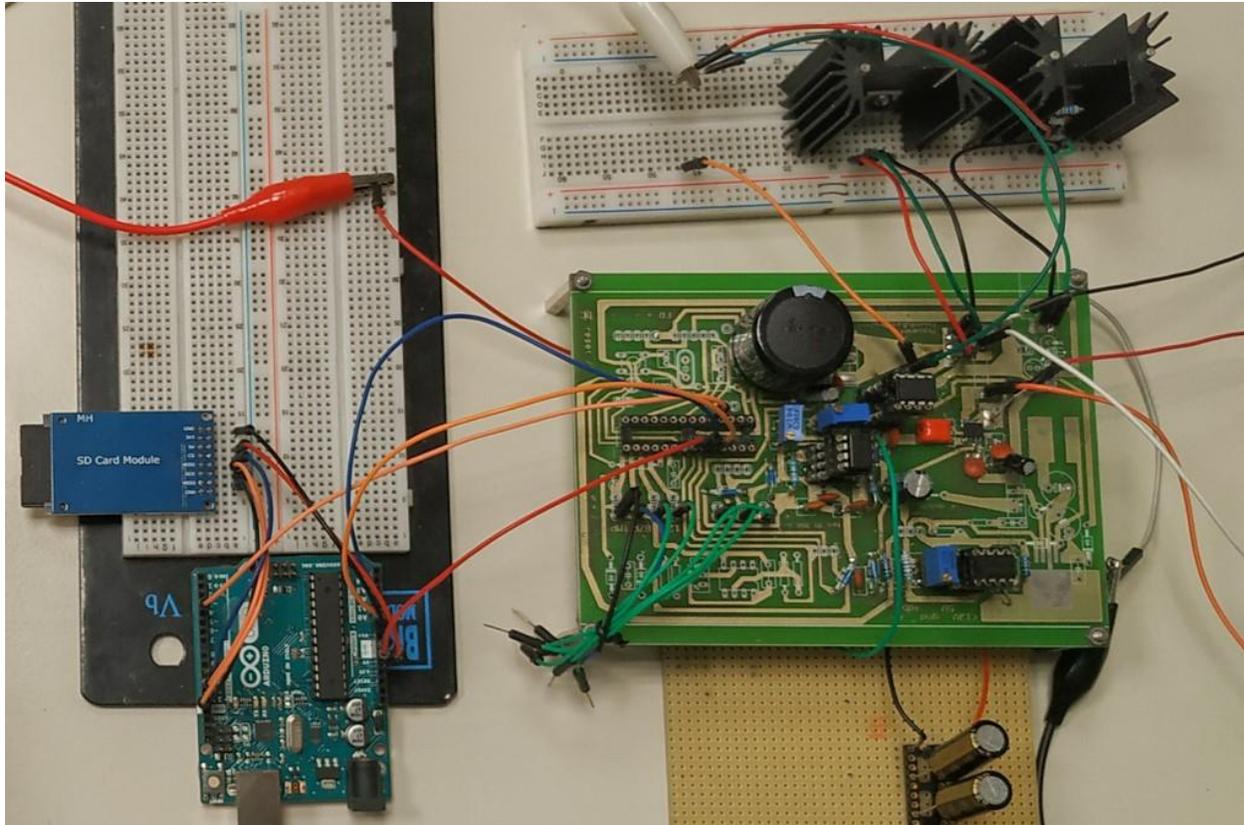
In Figure 21 the microcontroller, which is an Arduino Uno, can be seen, as well as the SD card shield and the connection between the two.



**Figure 21.** The Arduino Uno and SD card shield of the experimental prototype GMPPT system.

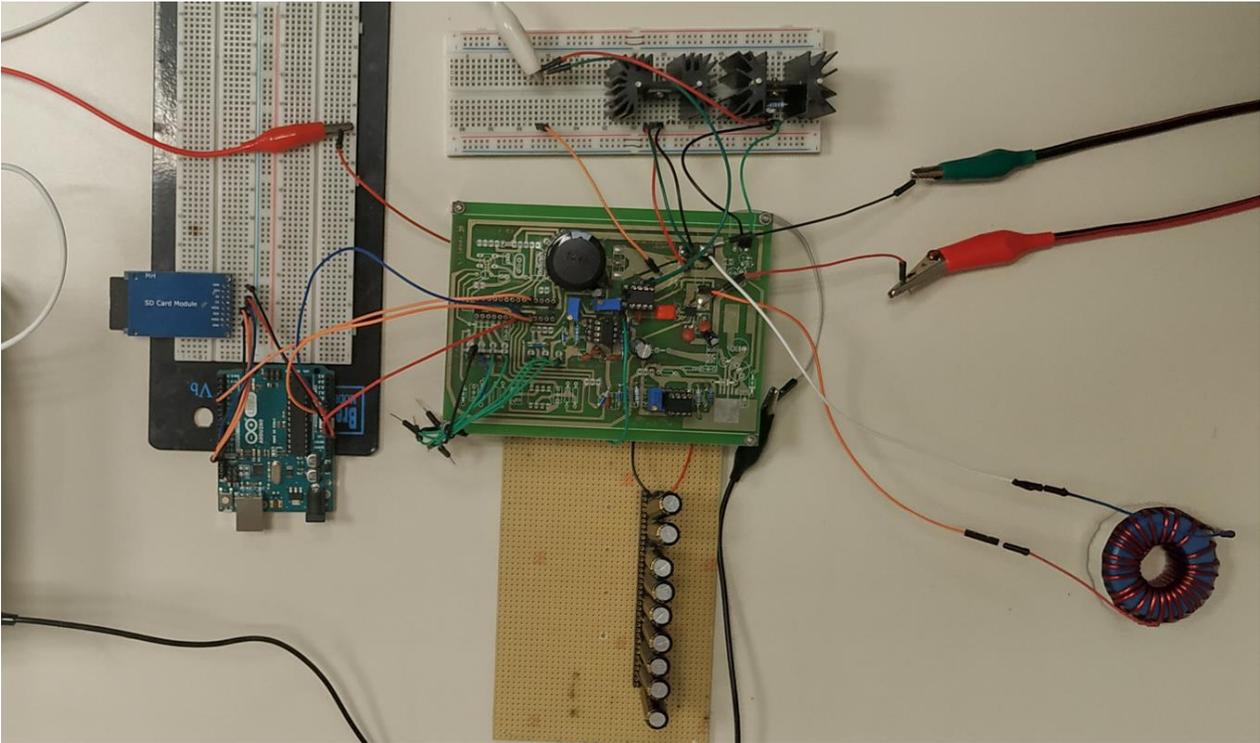
The connection between the Arduino Uno and the DC/DC Boost converter is shown in Figure 22. Four cables can be seen connecting to the Boost converter circuit and those are:

- 1) PWM signal, ending to the MOSFET driver
- 2) Voltage sensor output, connecting the Arduino and an LM358 chip
- 3) Current sensor output, connecting the Arduino and an LM358 chip
- 4) Ground, which has to be common with the DC/DC Boost converter ground.



**Figure 22.** Connection between Arduino Uno and the DC/DC Boost converter in the experimental prototype GMPPT system.

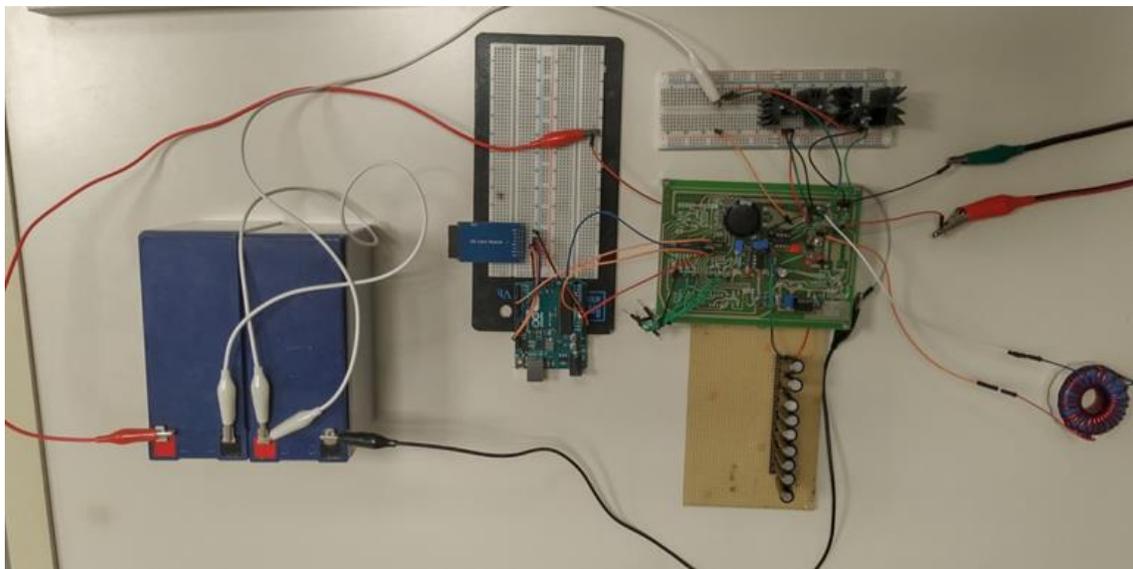
A photo of the DC/DC Boost converter and its components is presented in Figure 23. On the bottom right, the inductor used in series with the positive lead of the DC/DC Boost converter input can be seen. Below the DC/DC Boost converter, an array of 9 capacitors connected in parallel can be also seen. Lastly, a small breadboard can be seen on top that contains the DC/DC Boost converter MOSFET and a voltage regulator that converts the 12 Volts from one battery to 5 Volts, providing power to the control unit circuits. In Figures 24(a) and 24(b) the entire experimental prototype system that was developed in this thesis is presented, which consists of the Arduino microcontroller, the DC/DC Boost converter, the batteries and the photovoltaic modules.



**Figure 23.** The DC/DC Boost converter circuit.



(a)



(b)

**Figure 24.** (a) The PV array and (b) The overall GMPPT system which was developed in this thesis.

# 5. EXPERIMENTAL RESULTS

---

## 5.1 Introduction

Through simulations it was observed that the best results are obtained when the Q-learning algorithm has knowledge of the previous state duty cycle. Therefore, this method was used to evaluate the efficiency of the Q-learning algorithm under real operating conditions. The Q-learning algorithm had to be evaluated from two aspects. The first being efficiency and the second being convergence time. The algorithm's efficiency is easy to measure and can always be adjusted by changing values the algorithm uses like the discount factor, maximum number of visits, initial duty cycle etc. However, the convergence time is harder to evaluate since something pre-existing is needed to compare it to. That's why a second algorithm was also developed and that algorithm is the PSO algorithm.

In total, 35 experiments were conducted. Each experiment consisted of three steps:

1. In the beginning of each experiment the I-V and P-V curves of the arrangement of the photovoltaic modules had to be extracted. The way this was accomplished is by setting the duty cycle to 100% and continuously decreasing it up to 0%, measuring the voltage and current under each decrement of the duty cycle. Since the power is a product of the voltage and current values both the I-V and P-V curves were able to be drawn. Since the Arduino IDE is not capable of drawing a X-Y plot, another program had to be used which would process the values sensed by the Arduino and plot them. That program is called Megunolink and it

features two very significant advantages. The first is that the curves can be observed in real time which means that all the local MPP's and global MPP locations can be seen. That knowledge is very important for the next two steps because the position of the global MPP is now known and it can easily be determined whether the Q-learning and PSO algorithms have converged successfully. If that option was not available, then the data would have to be saved in a .csv file after each step and then the curves would be constructed from the saved data to check whether they were successful or not. The second advantage is that once the curves were obtained all the data could be saved in .csv files and that way the plots needed could be reconstructed which will be covered in the next two steps.

2. In the second step, the Q-learning algorithm had to be executed. Once the algorithm had converged the knowledge acquired by the first step could be used to determine whether it had converged correctly. In order for that to happen both measurements had to be in the same format. This means that since the P-V curve from the first step was extracted, the power-voltage couple had to be plotted the same way. Plotting the power-voltage that way was only convenient for that comparison but what was needed were the following plots: Current-number of steps, Voltage-number of steps, Power-number of steps and duty cycle-number of steps. This is the part where saving the data in .csv files is advantageous. The I-V and P-V curves could be plotted the way needed in order to observe in real time whether the algorithm had converged correctly and the data could be later used to construct the plots mentioned above.
3. The third and final step is executing the PSO algorithm. The I-V and P-V curves had to be extracted the same way as in the second step in order to determine whether the PSO algorithm had also converged correctly but the following plots needed to be extracted: Current-number of steps, Voltage-number of steps, Power-number of steps and duty cycle-number of steps for every particle and every generation with an extra addition which is the position of each particle. So once again the I-V and P-V curves were plotted in order to compare the corresponding results with the first step and the data were saved for production of the required plots.

Before each experiment the temperature and irradiance of each photovoltaic module was measured. The partial shading conditions were emulated by adjusting the tilt angle of each PV module of the PV array. Once the Q-learning and PSO converge, the P&O algorithm takes over for fine-tuning the duty cycle to the GMPP during the short-term changes of solar irradiation and ambient temperature. The final values used in the Q-learning algorithm, which were chosen based on many trials, are shown in Table 4. Dimensions 1-4 are the dimensions of the Q array, corresponding to the number of equally quantized steps for the duty cycle, power, previous state duty cycle and action, respectively.

**Table 4.** The operating parameters of the Q-learning algorithm.

$N_{\max}$	15	$t_{\min}$	0.08	Dimension3	5
k1	20	$t_{\max}$	0.8	Dimension4	7
k2	25	Dimension1	7		
k3	0.6	Dimension2	10		

The final values used in the PSO algorithm, which were chosen based on many trials, are shown in Table 5. A variable inertia weight parameter was chosen starting at  $w_{\max}$  and decreasing linearly to  $w_{\min}$  over time. C1 and C2 are the acceleration factors and variable values were chosen for these two factors as well since they yielded best results. C1 decreased linearly over time from  $c1_{\max}$  to  $c1_{\min}$  and c2 increased linearly over time from  $c2_{\min}$  to  $c2_{\max}$  according to [17]-[19].

**Table 5.** The operating parameters of the PSO algorithm.

particles	5	$c1_{\min}$	0.5
iterations	10	$c1_{\max}$	2
$w_{\max}$	0.7	$c2_{\min}$	0.5

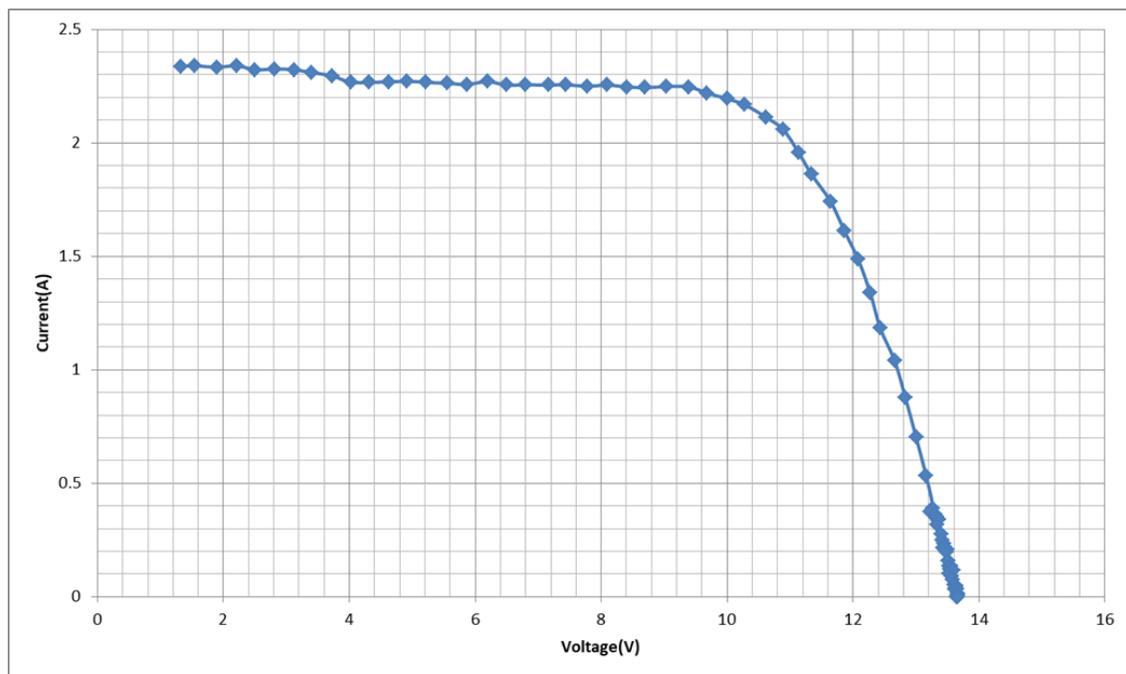
$w_{\min}$	0.3	$c2_{\max}$	2
------------	-----	-------------	---

## 5.2 Analysis of the results for shading pattern 1

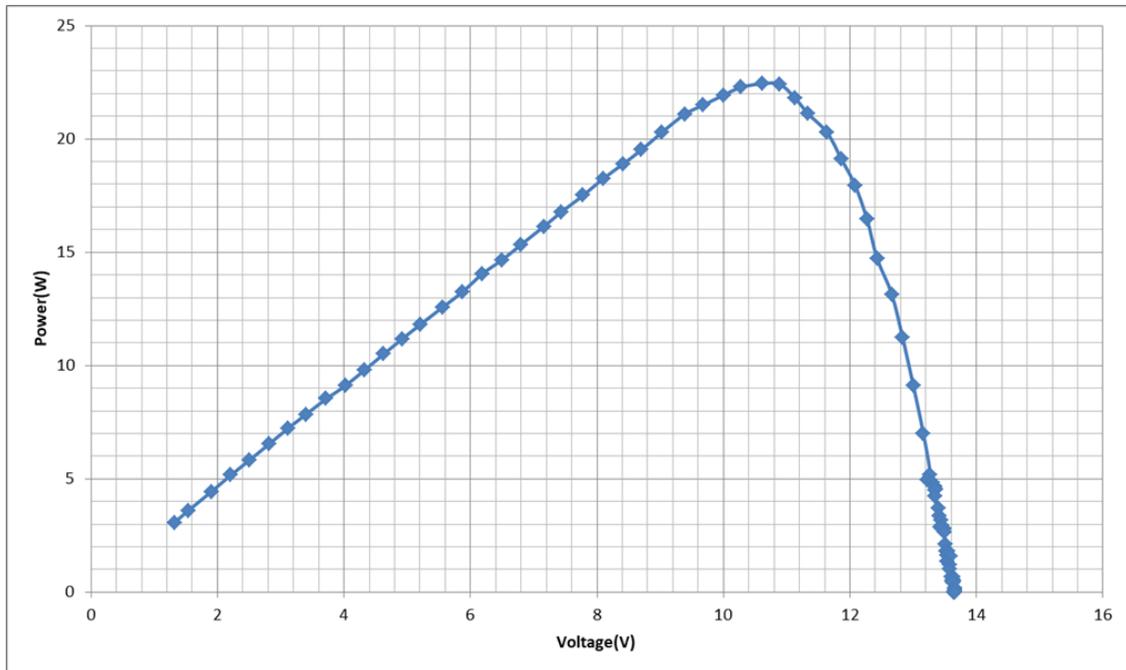
### Experiment No. 1:

Solar irradiance for the two photovoltaic modules PV1:  $600 \text{ W/m}^2$ , PV2:  $600 \text{ W/m}^2$  and Ambient temperature:  $25^\circ\text{C}$ .

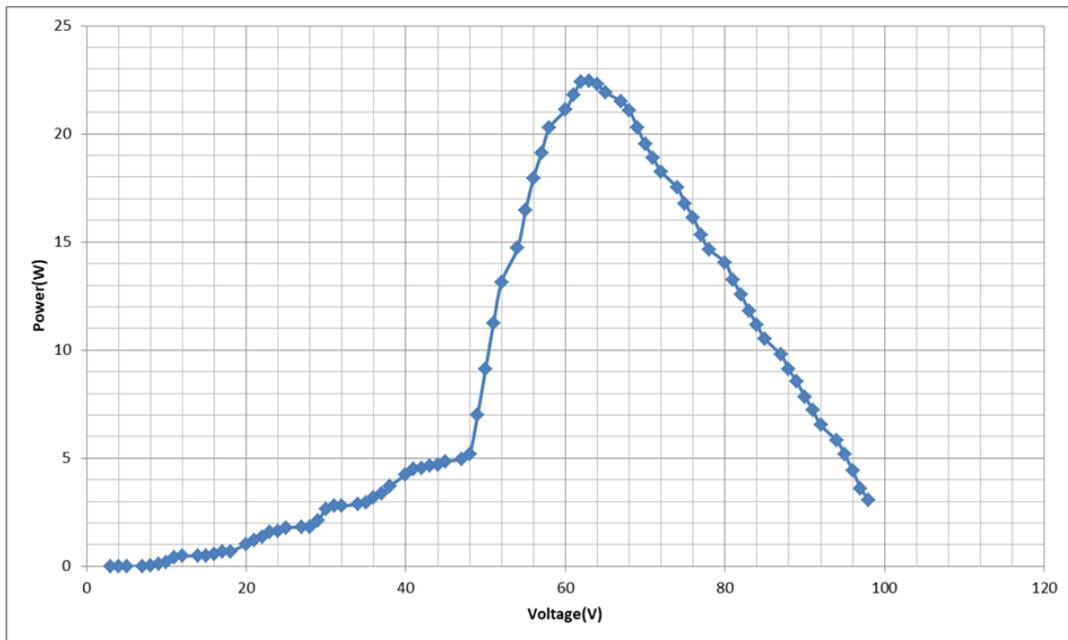
In Fig. 25 the PV array output current-voltage curve is presented and in Fig. 26 the PV array output power-voltage curve is presented. The PV modules operate under uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 27, which helps to detect in which duty cycle value the MPP is located.



**Figure 25.** The PV array output current-voltage curve.



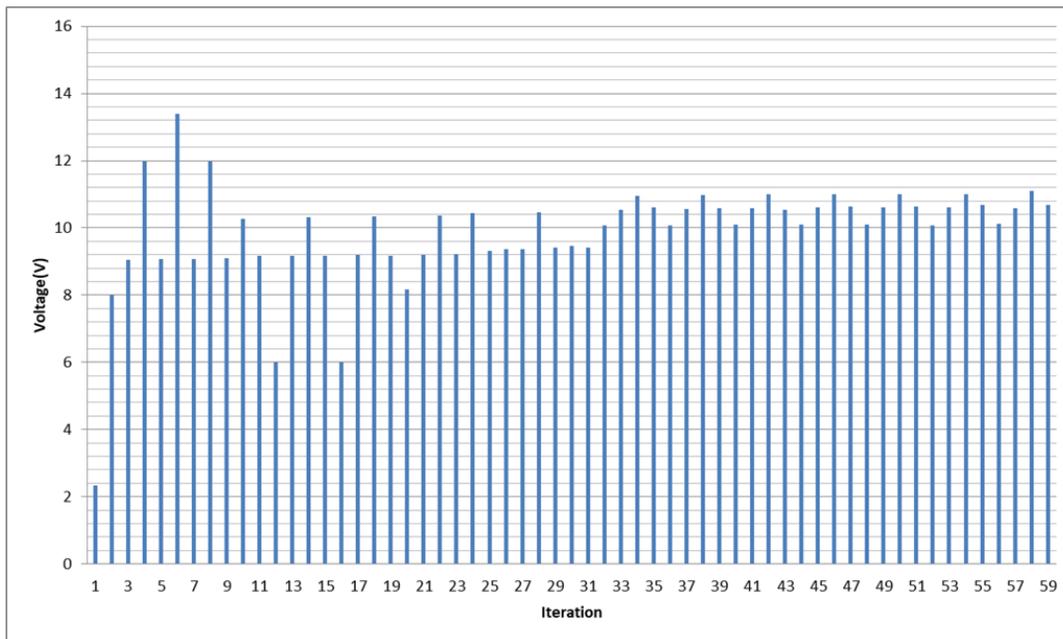
**Figure 26.** The PV array output power-voltage curve.



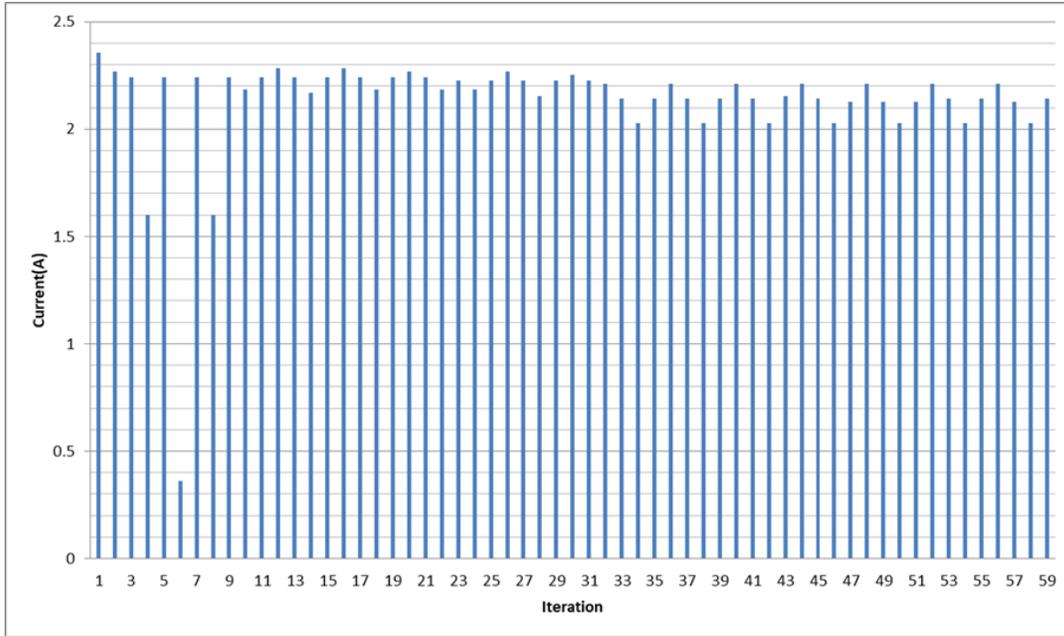
**Figure 27.** The PV array output power versus the duty cycle.

**Results of the Q-learning GMPPT algorithm:**

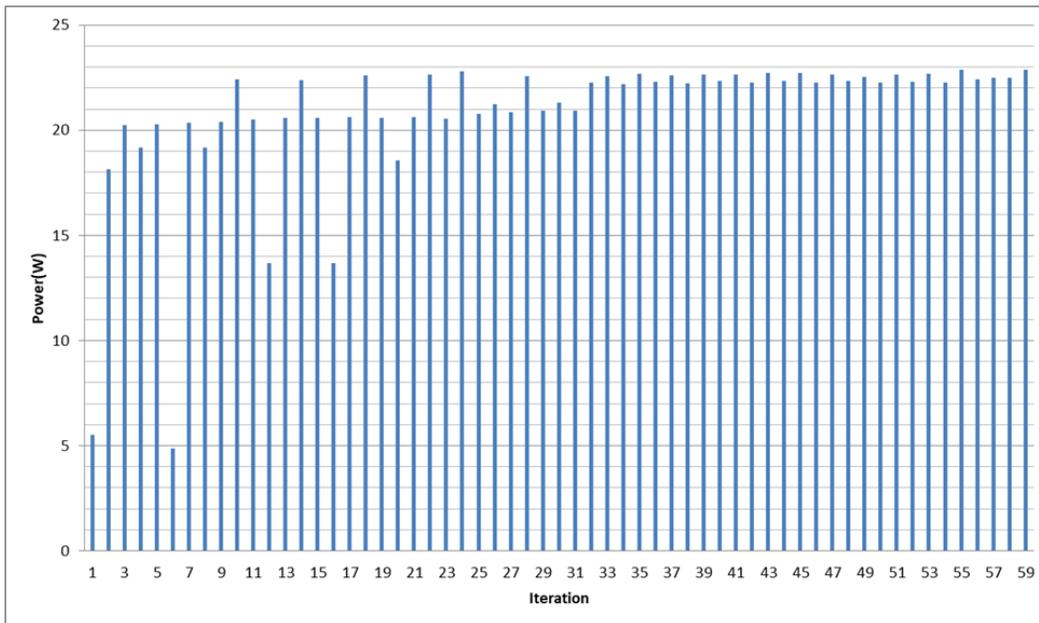
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 28-31 present the results of the above plots for shading pattern 1.



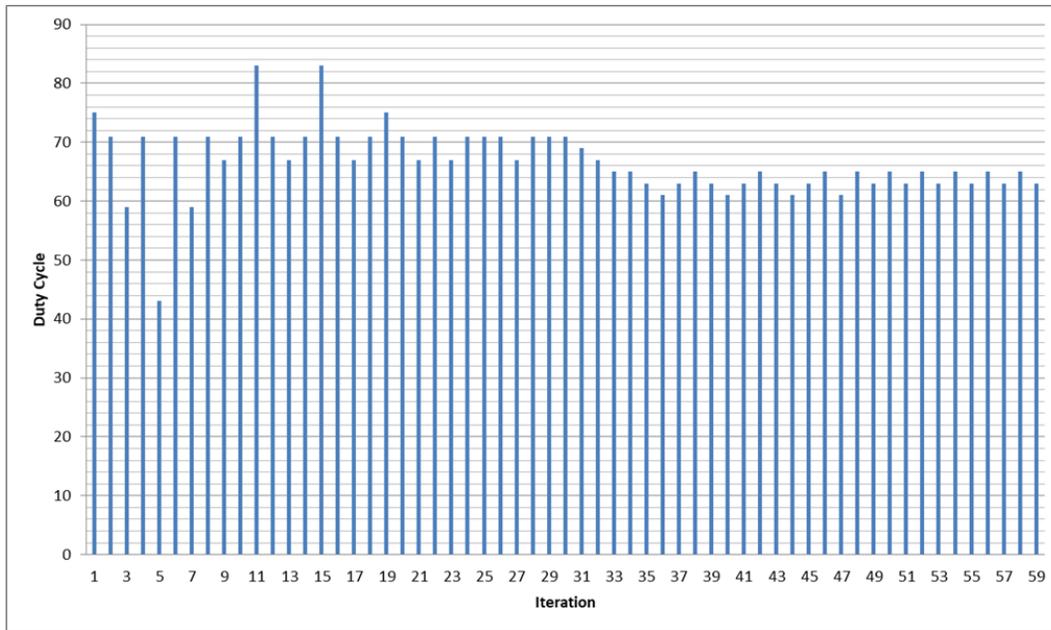
**Figure 28.** The PV array output voltage versus the number of search step.



**Figure 29.** The PV array output current versus the number of search step.



**Figure 30.** The PV array output power versus the number of search step.



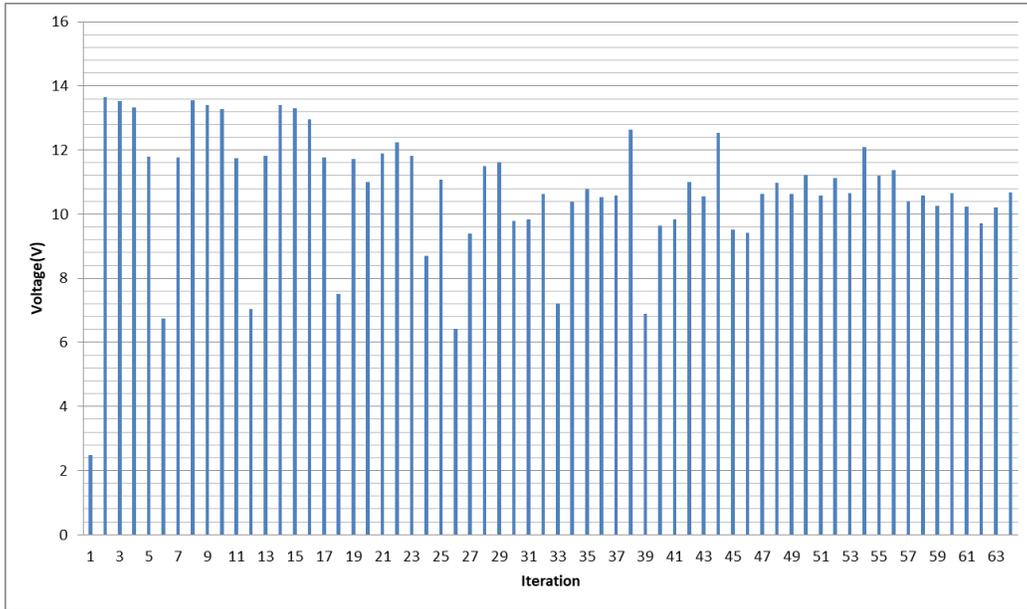
**Figure 31.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. This is due to the fact that the agent has prior knowledge of where the GMPP may be located. Thus, it does not have to visit many states until it converges to a specific duty cycle value. It is considered that the exploration is over, when the oscillations start. These oscillations are due to the P&O method, which is applied at that time with a perturbation step of 2%. The oscillations start occurring at approximately the 35<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate. The oscillations can be seen on all four graphs.

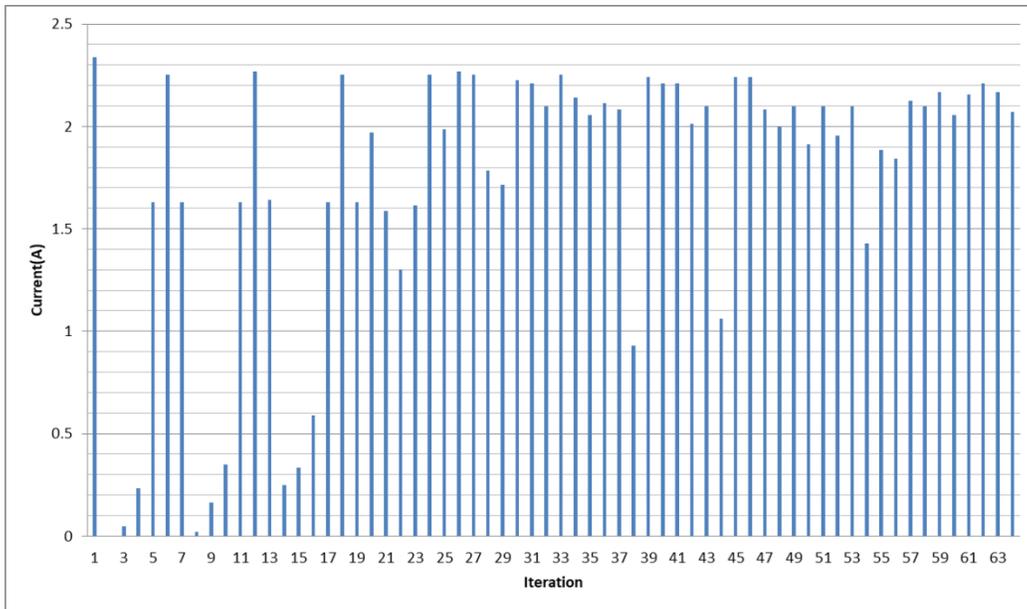
### **Results of the PSO GMPPT algorithm:**

Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the

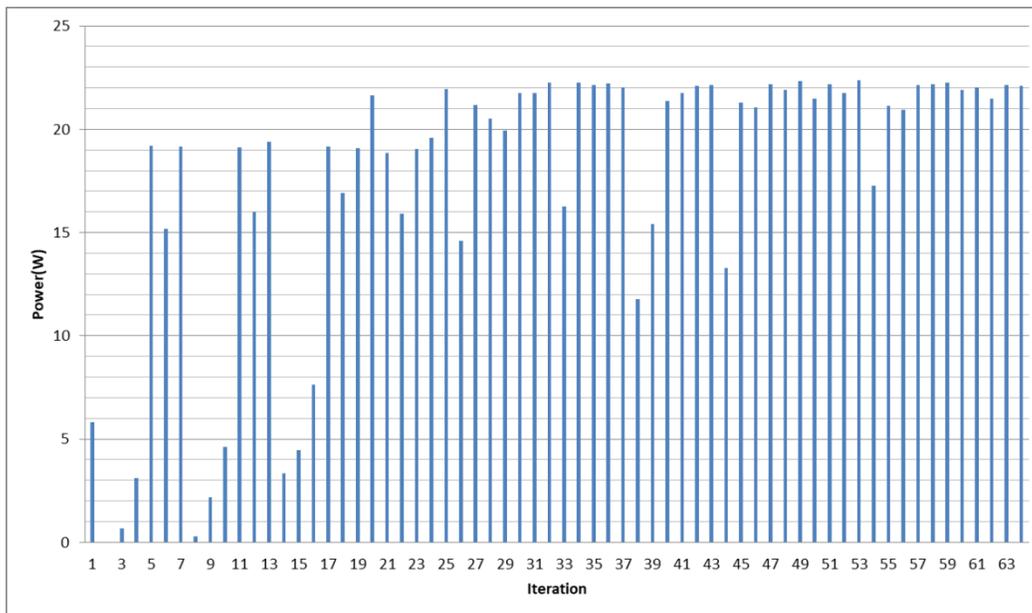
number of steps until convergence plot. Figure 32 through 36 present the results of the above plots for shading pattern 1.



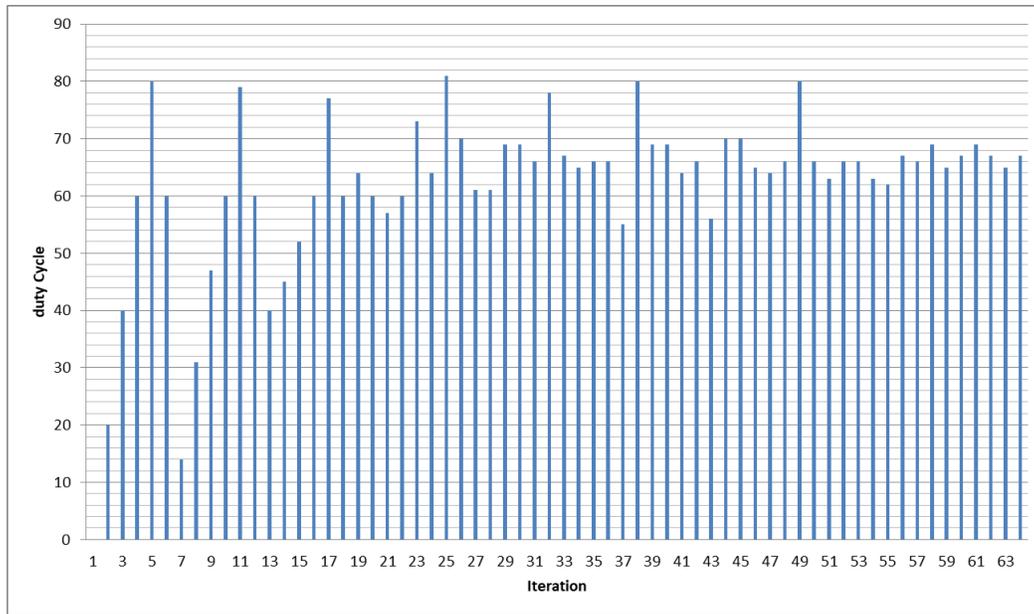
**Figure 32.** The PV array output voltage versus the number of search step.



**Figure 33.** The PV array output current versus the number of search step.

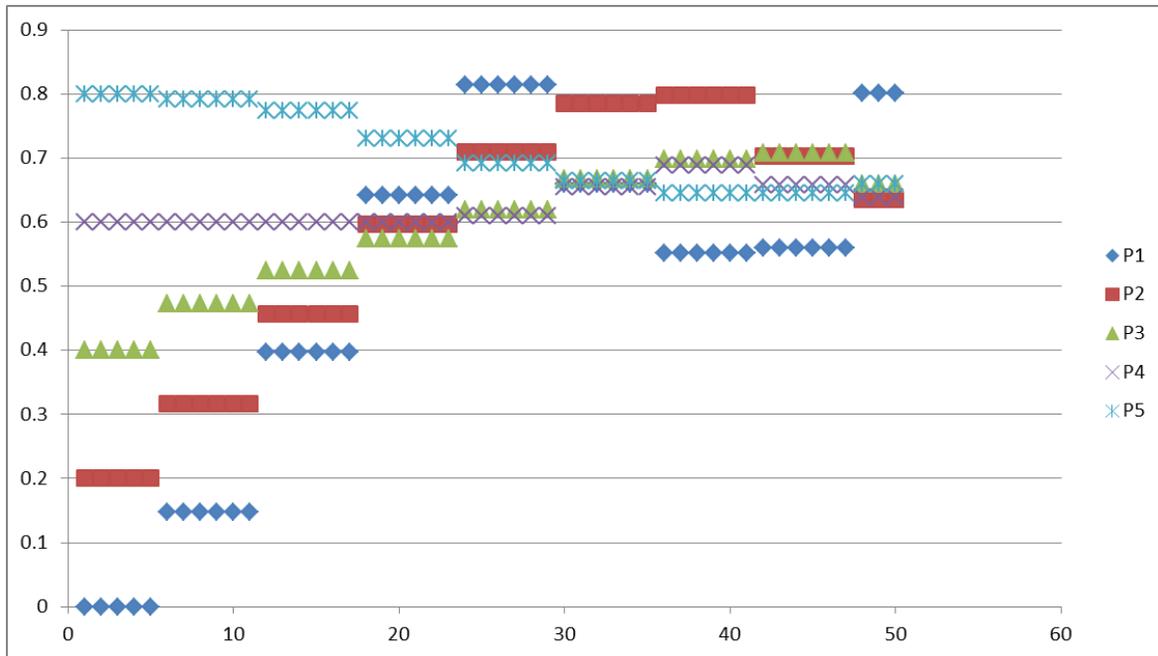


**Figure 34.** The PV array output power versus the number of search step.



**Figure 35.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is also when the oscillations start. These oscillations are due to the P&O method, which is used at that time with a perturbation step of 2%. The oscillations start at the 50<sup>th</sup> iteration when the PSO converges and the P&O algorithm starts to operate. The oscillations can be seen on all four graphs.



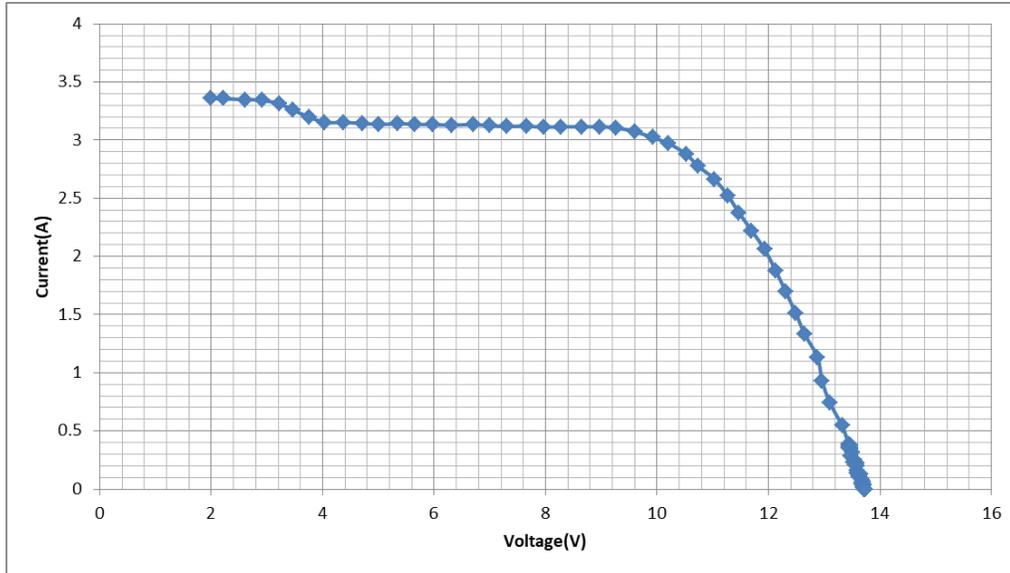
**Figure 36.** The particles position versus the number of search step.

As shown in Figure 31, the particles 2, 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine-tune the duty cycle.

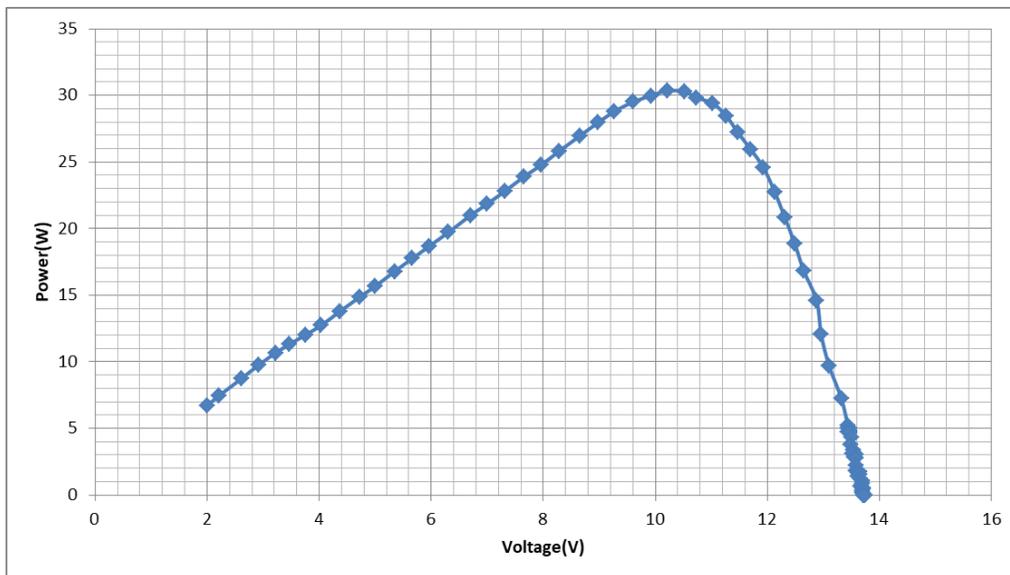
### **Experiment No 2:**

**Solar irradiance for the two photovoltaic modules PV1:  $820 \text{ W/m}^2$ , PV2:  $850 \text{ W/m}^2$  and Ambient temperature:  $24^\circ\text{C}$ .**

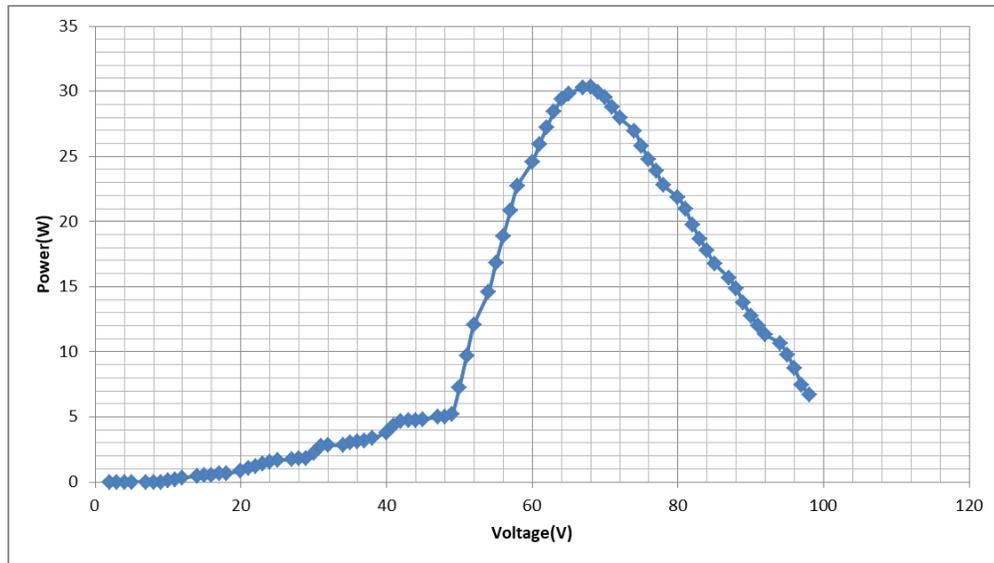
In Fig. 37 the PV array output current-voltage curve is presented and in Fig. 38 the PV array output power-voltage curve is presented. The PV modules operate under uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithm the power-duty cycle curve is depicted in Fig. 39, which helps to detect in which duty cycle value the MPP is located.



**Figure 37.** The PV array output current-voltage curve.



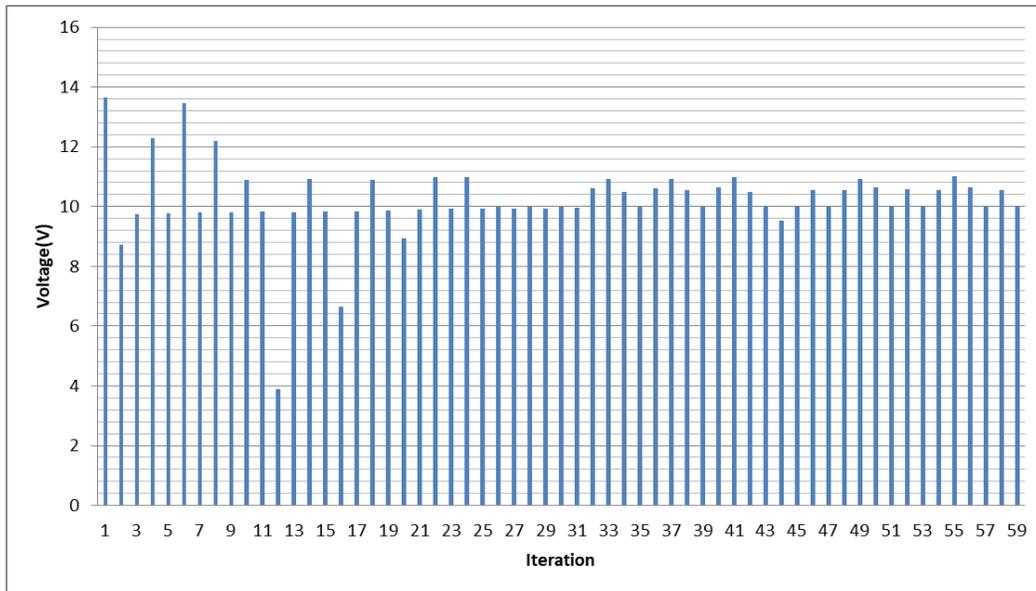
**Figure 38.** The PV array output power-voltage curve.



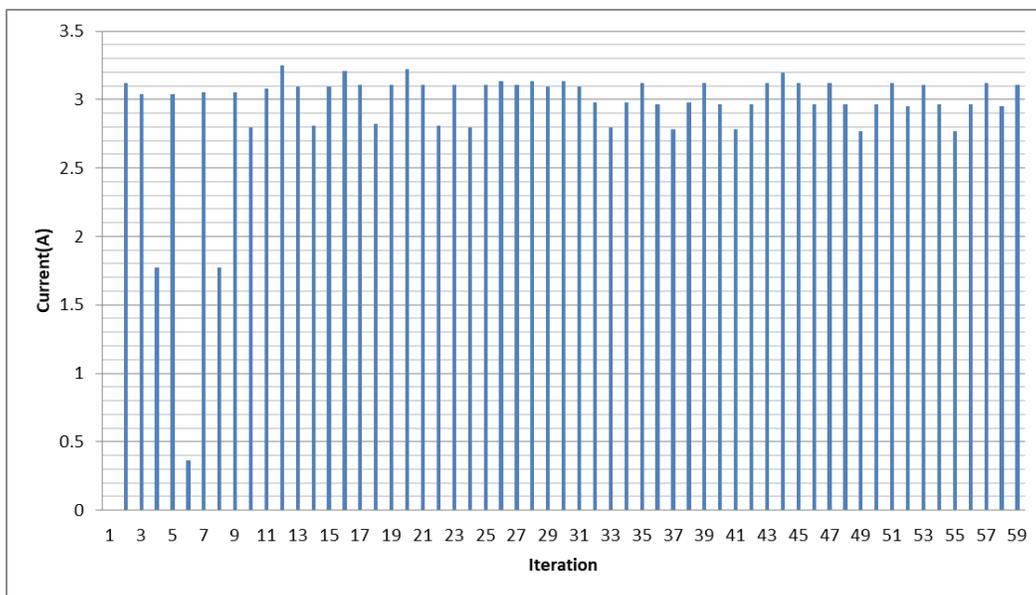
**Figure 39.**The PV array output power-duty cycle curve.

#### **Results of the Q-learning GMPPT algorithm:**

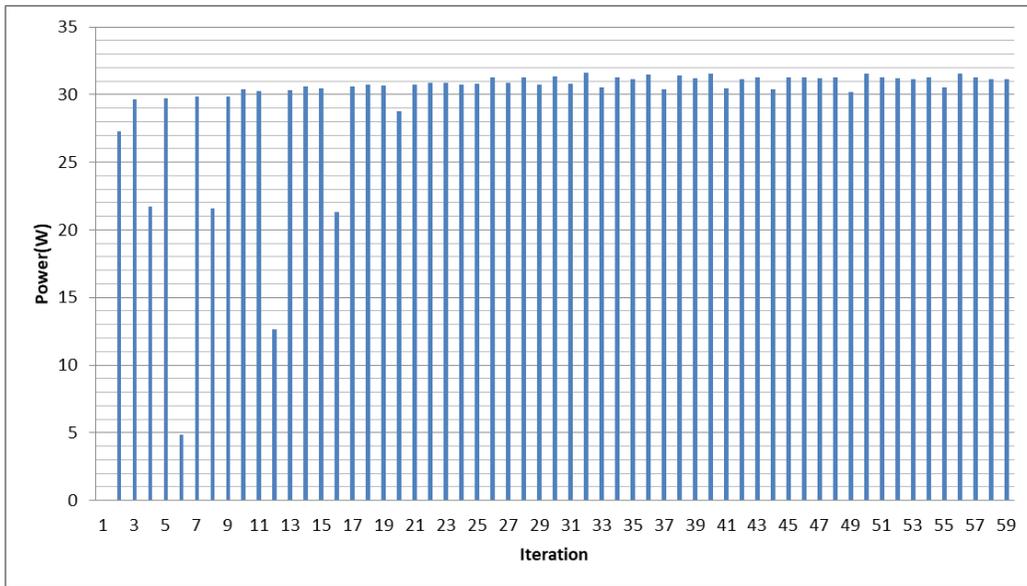
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 40-43 present the results of the above plots for shading pattern 1.



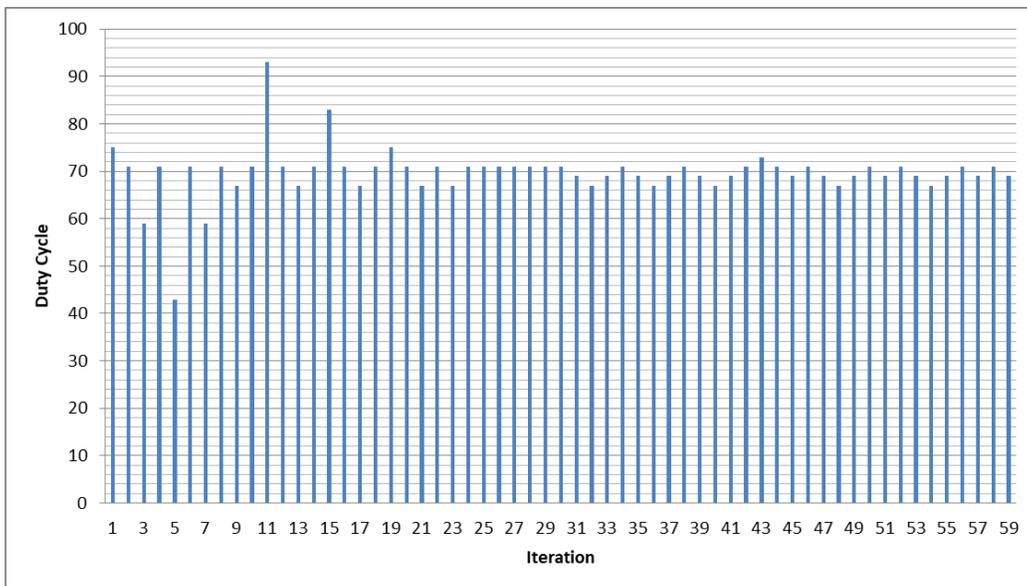
**Figure 40.** The PV array output voltage versus the number of search step.



**Figure 41.** The PV array output current versus the number of search step.



**Figure 42.** The PV array output power versus the number of search step.



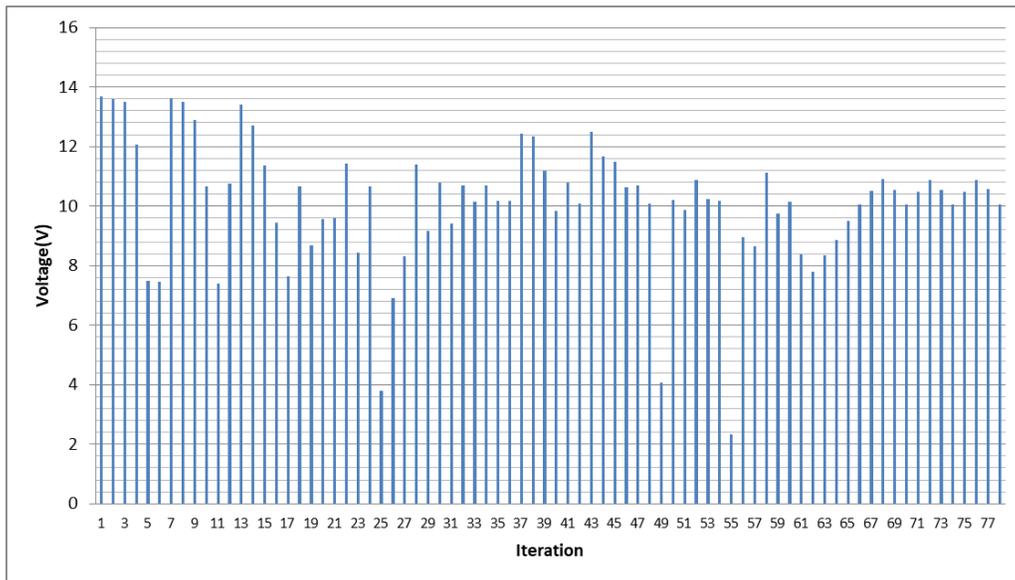
**Figure 43.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. Thus, it does not have to visit many states until it converges to a specific duty cycle value. It is considered that the exploration is over, when the oscillations start. These oscillations

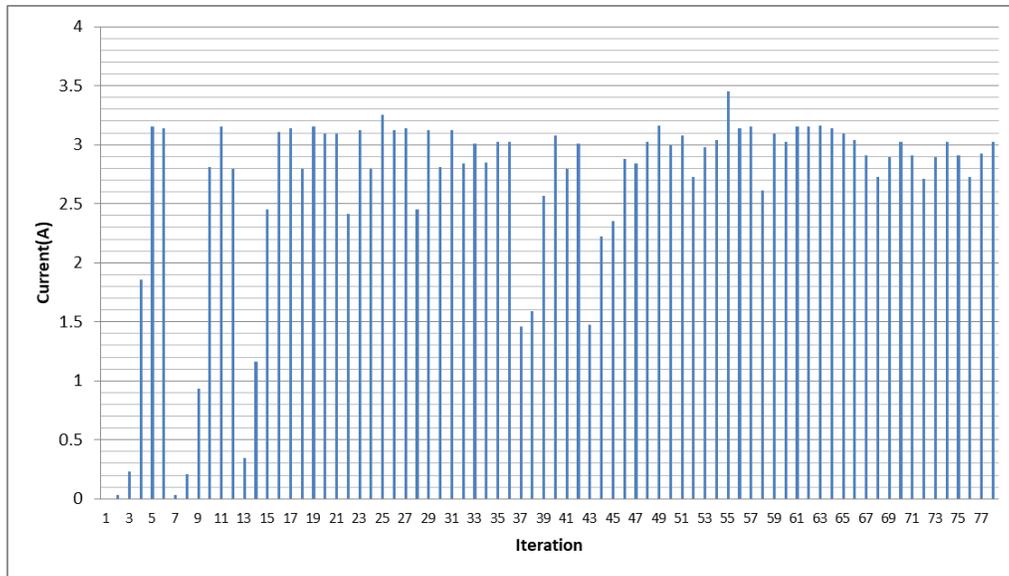
are due to the P&O method, which is applied at that time with a perturbation step of 2%. The oscillations start occurring at approximately the 32<sup>nd</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate. The oscillations can be seen on all four graphs.

### Results of the PSO GMPPT algorithm:

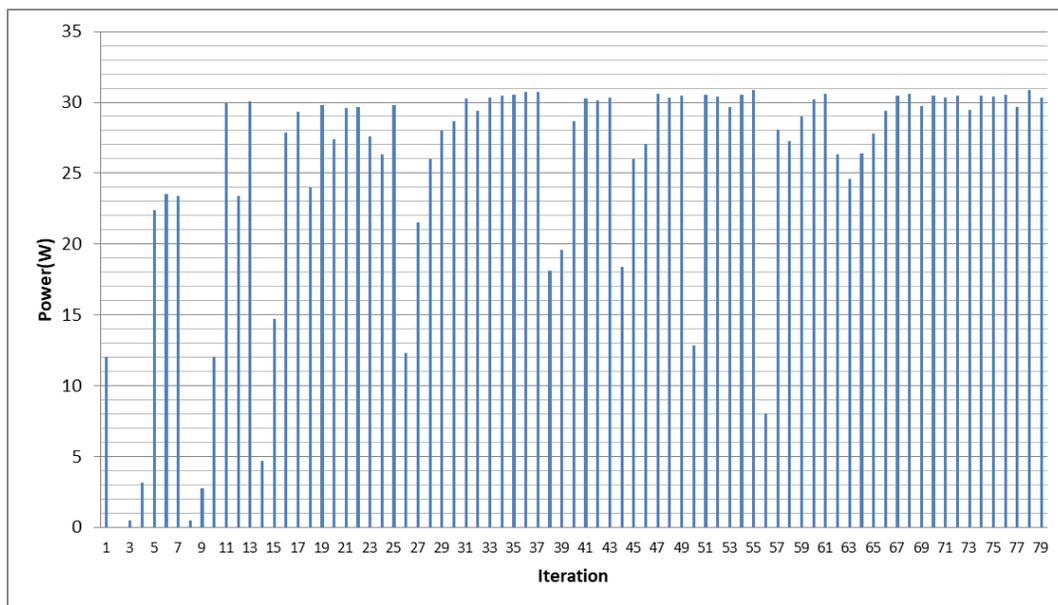
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 44-48 present the results of the above plots for shading pattern 1.



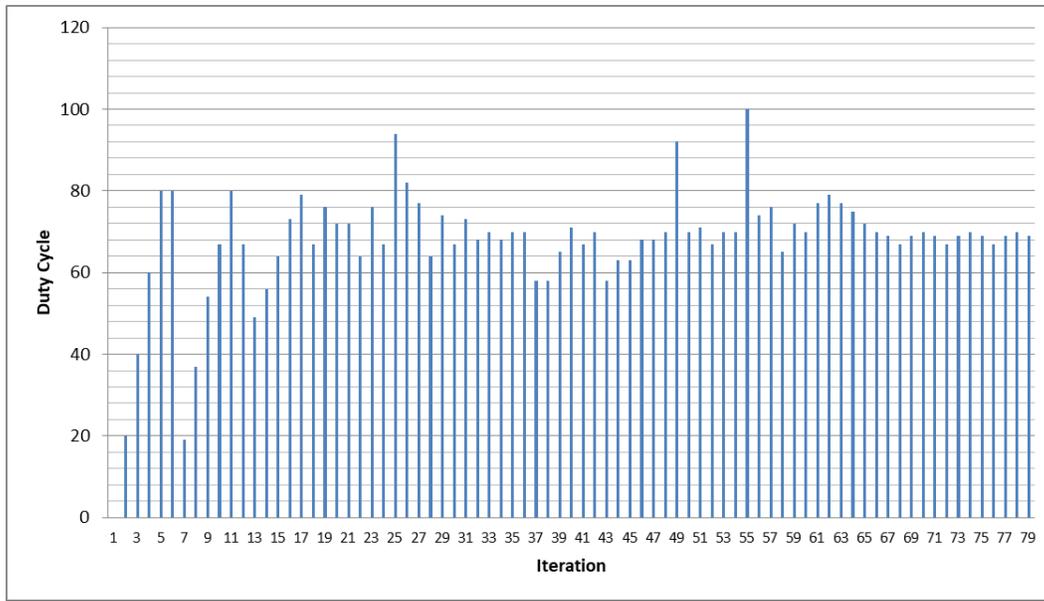
**Figure 44.** The PV array output voltage versus the number of search step.



**Figure 45.** The PV array output current versus the number of search step.

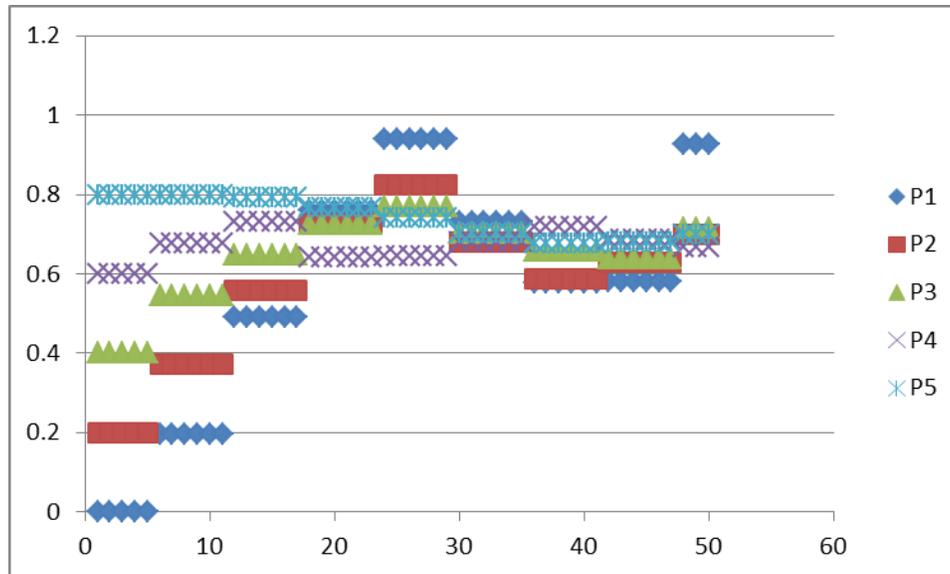


**Figure 46.** The PV array output power versus the number of search step.



**Figure 47.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is also when the oscillations start. These oscillations are due to the P&O method, which is used at that time with a perturbation step of 2%. The oscillations do not start at approximately the 50<sup>th</sup> iteration when the PSO converges and the P&O starts to operate because it takes the P&O algorithm a few more steps to fine tune the duty cycle. The oscillations can be seen on all four graphs starting at approximately iteration 68.



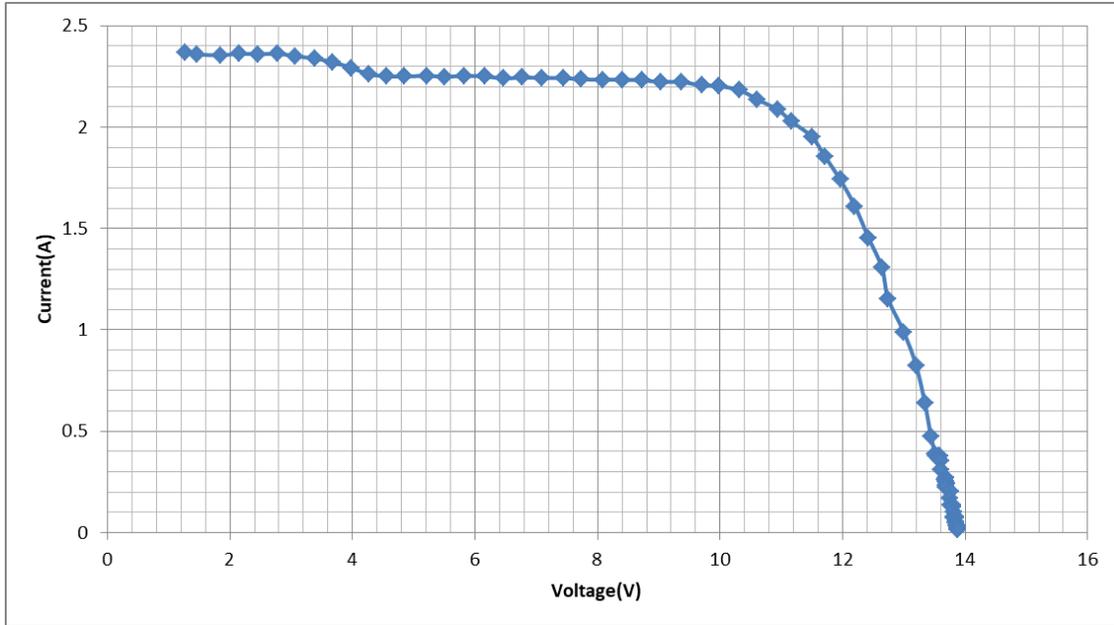
**Figure 48.** The particles position versus the number of search step.

As shown in Figure 48, the particles 2, 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

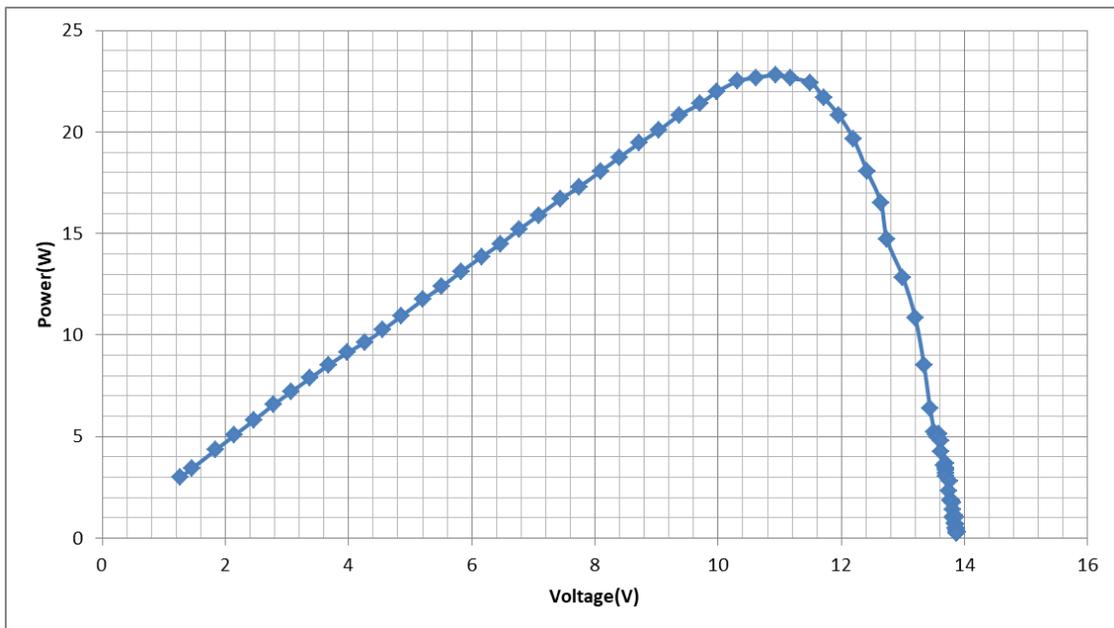
### **Experiment No 3:**

**Solar irradiance for the two photovoltaic modules PV1: 575 W/m<sup>2</sup>, PV2:600 W/m<sup>2</sup> and Ambient temperature: 21°C.**

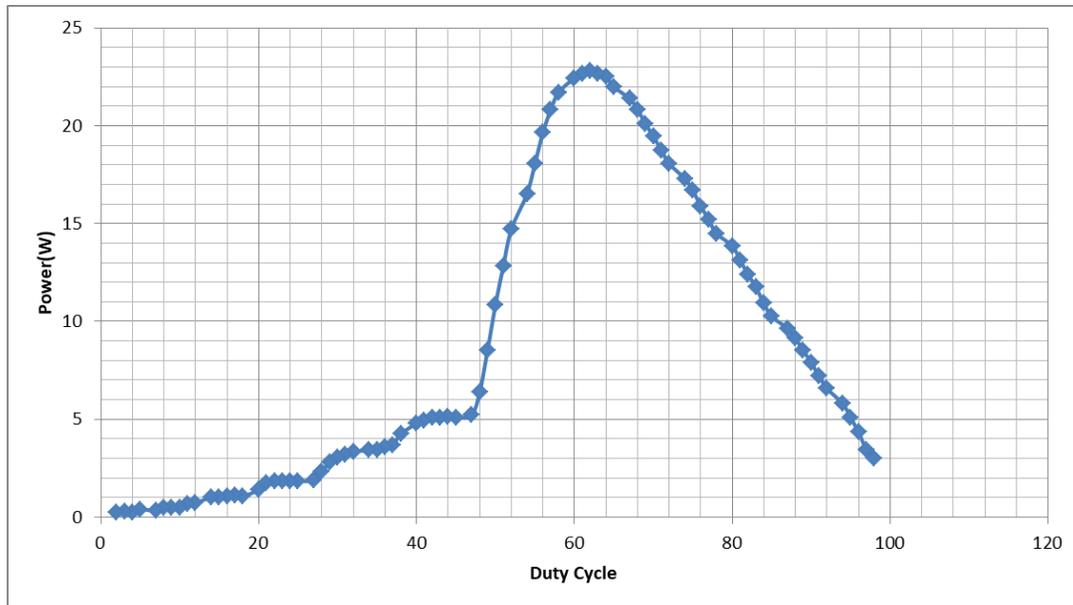
In Fig. 49 the PV array output current-voltage curve is presented and in Fig. 50 the PV array output power-voltage curve is presented. The PV modules operate under a uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 51, which helps to detect in which duty cycle value the MPP is located.



**Figure 49.** The PV array output current-voltage curve.



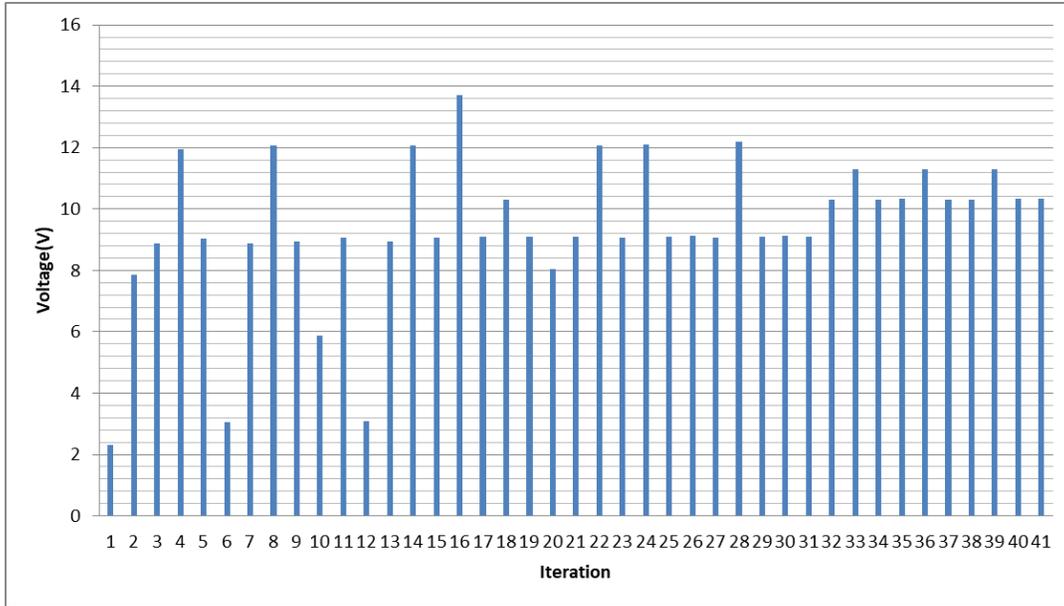
**Figure 50.** The PV array output power-voltage curve.



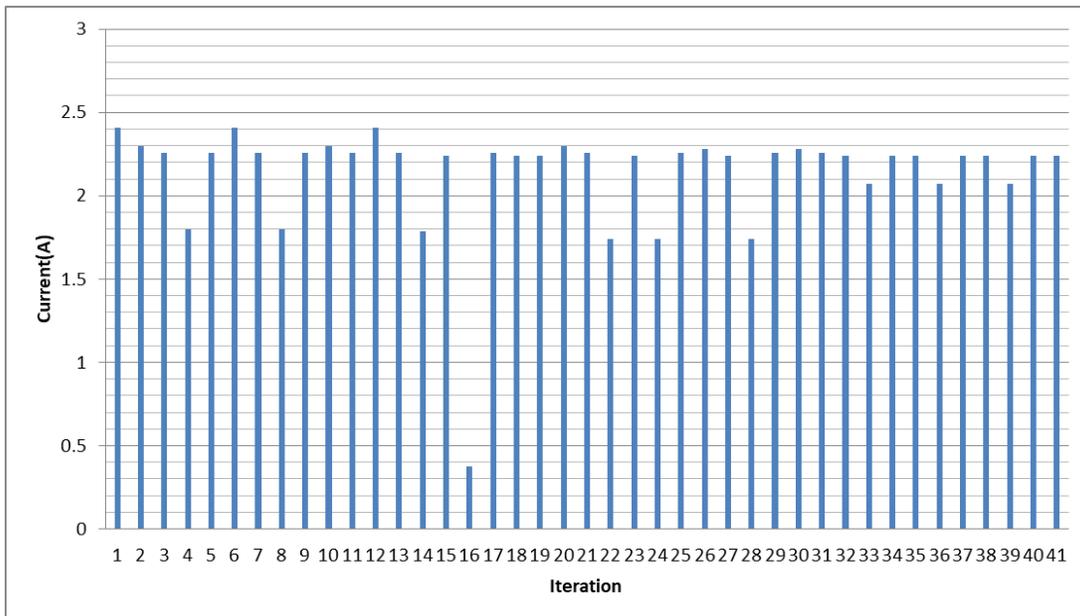
**Figure 51.** The PV array output power-duty cycle curve.

#### **Results of the Q-learning GMPPT algorithm:**

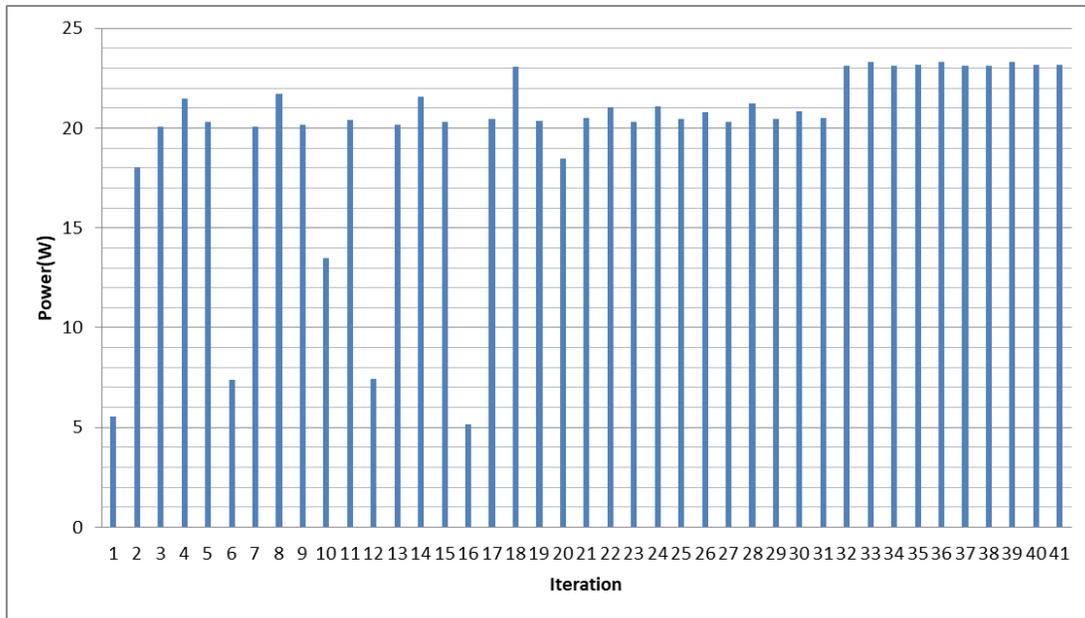
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 52-55 present the results of the above plots for shading pattern 1.



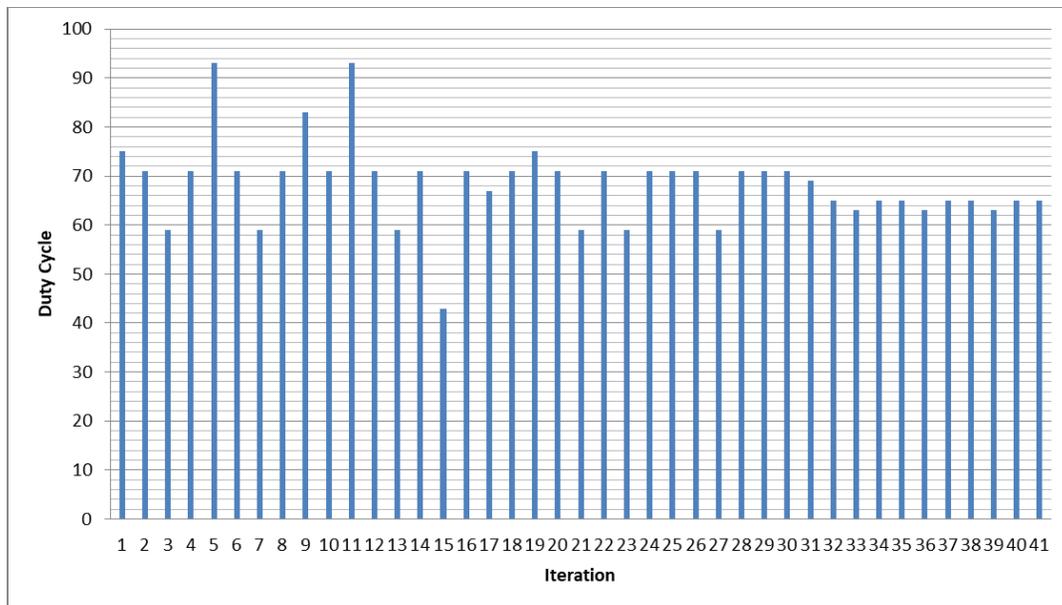
**Figure 52.** The PV array output voltage versus the number of search step.



**Figure 53.** The PV array output current versus the number of search step.



**Figure 54.** The PV array output power versus the number of search step.



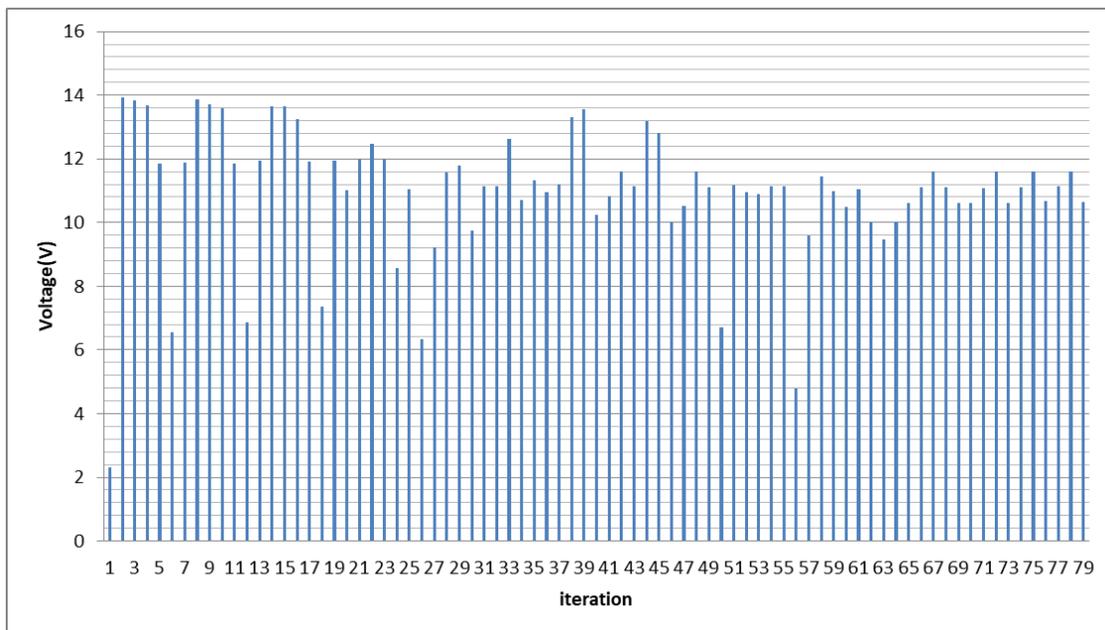
**Figure 55.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. Thus, it does not have to visit many states until it converges to a specific duty cycle value. It is considered that the exploration is over, when the oscillations start. These oscillations

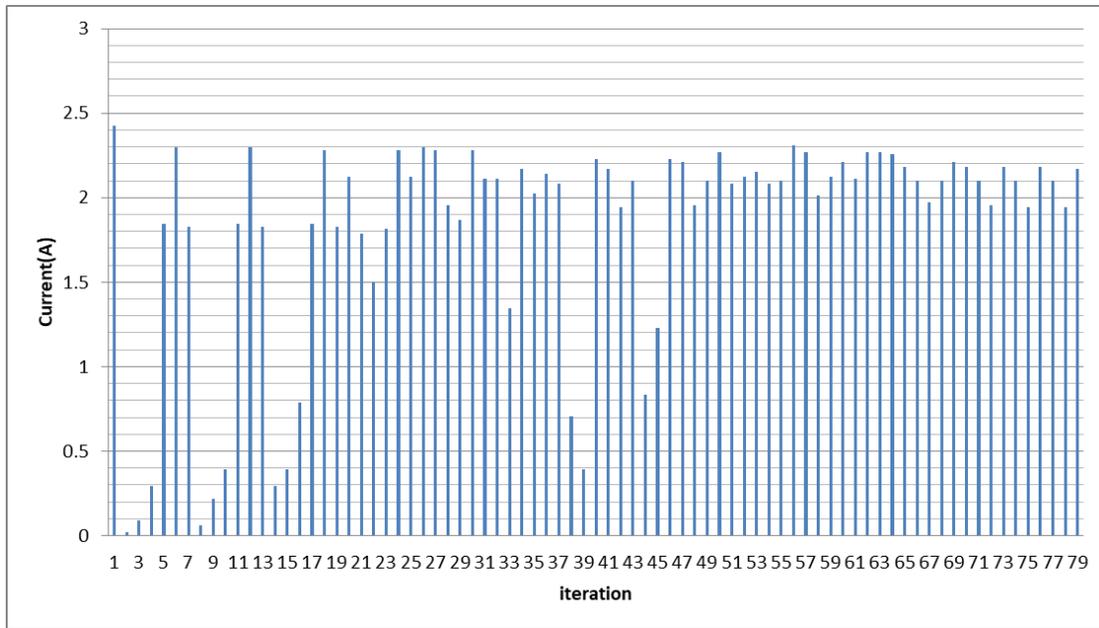
are due to the P&O method, which is applied at that time with a perturbation step of 2%. The oscillations start at approximately the 32<sup>nd</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate. The oscillations can be seen on all four graphs.

### Results of the PSO GMPPT algorithm:

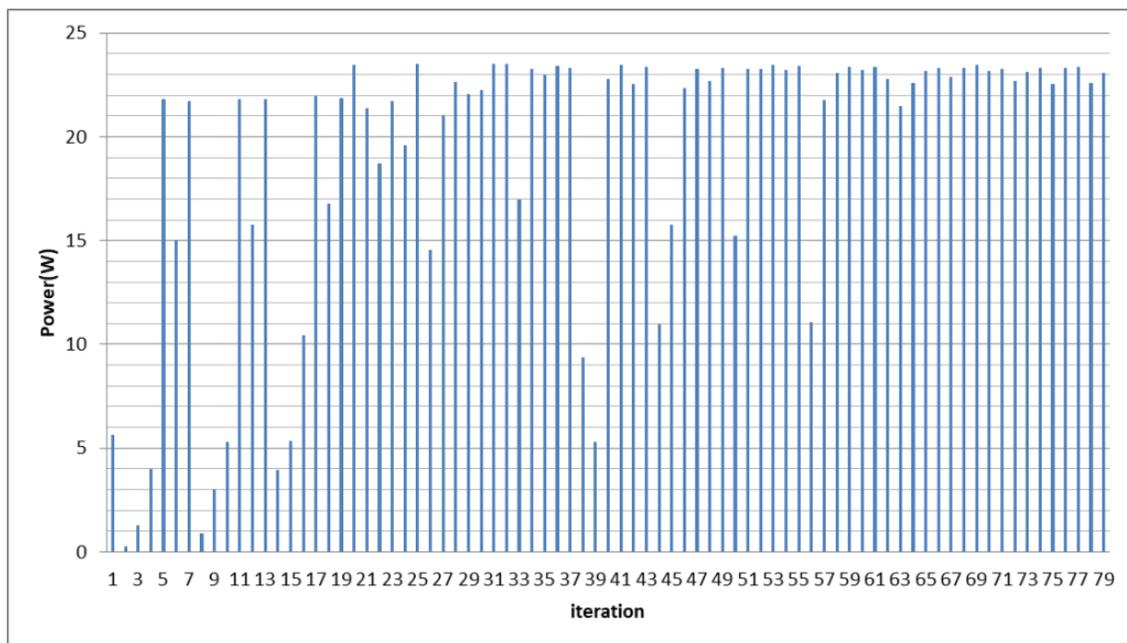
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs 56-60 present the results of the above plots for shading pattern 1.



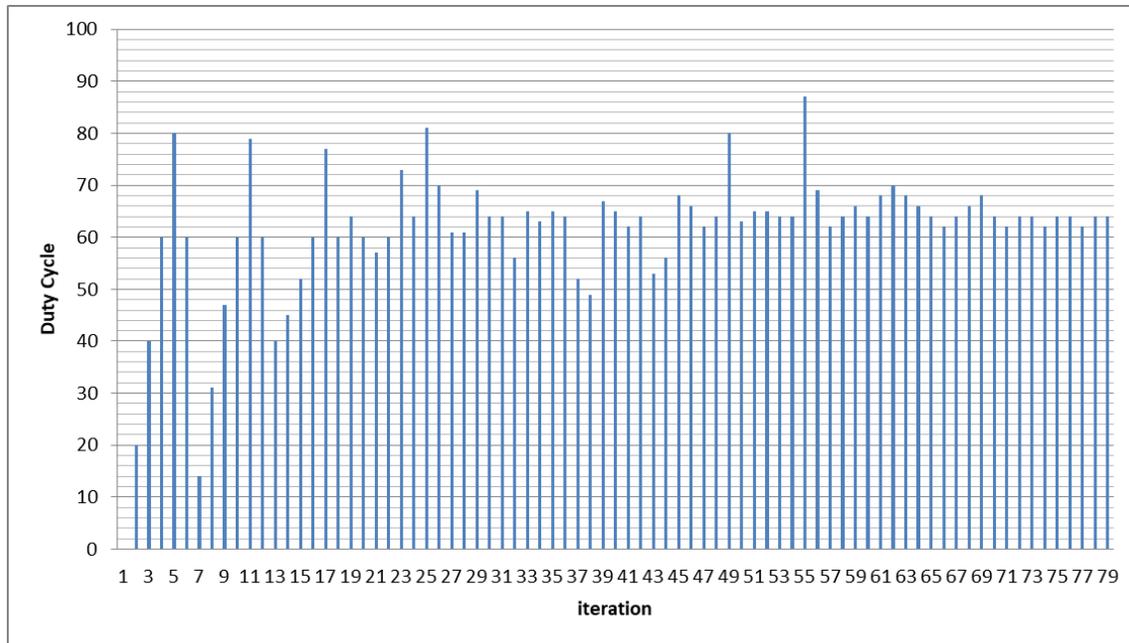
**Figure 56.** The PV array output voltage versus the number of search step.



**Figure 57.** The PV array output current versus the number of search step.

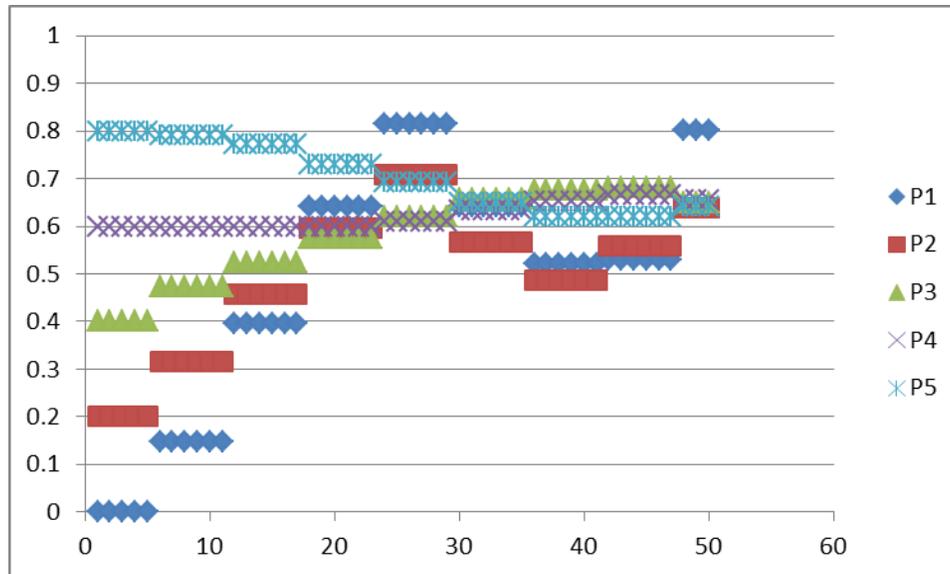


**Figure 58.** The PV array output power versus the number of search step.



**Figure 59.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is also when the oscillations should start. These oscillations are due to the P&O method, which is used at that time with a perturbation step of 2%. The oscillations do not start occurring at approximately the 50<sup>th</sup> iteration when the PSO algorithm converges and the P&O algorithm starts to operate because it takes the P&O algorithm a few more steps to fine tune the duty cycle. The oscillations can be seen on all four graphs starting at approximately iteration 68 and the effect of a slight lower temperature can be seen again in the stability of the power oscillation once the P&O fine tunes the duty cycle.



**Figure 60.** The particles position versus the number of search step.

As shown in Figure 60, the particles 2, 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

In this pattern, the Q-learning algorithm was tested and compared to the PSO algorithm based on a very small irradiance difference between the two photovoltaic modules accomplished by 0 or close to 0 angle difference between them. Taking a closer look at the power curves it can be seen how the angle between the photovoltaic modules and the sun affects the value of the maximum power that can be obtained but not as much the duty cycle under which operation at the MPP is achieved. It was expected by both algorithms to converge correctly as they did, with the Q-learning algorithm being faster every time and slightly more precise. Even though the P&O algorithm was applied in both cases, it was able to converge more accurately in the Q-learning method since more duty cycle values were able to be explored due to the variety of actions that could be applied. For instance if the MPP was located under 60% duty cycle, the Q-learning would be able to sense the power at that level because the actions allowed the agent to explore the 60% duty cycle, but the PSO can explore based on the velocity of the particles meaning that

if the velocity is not favorable enough the 60% duty cycle might not be explored at all and instead it can converge to either 63% or 59% where the P&O, having a 0.2 step, could fine-tune it to 61% or 57% which is clearly a duty cycle with a slight deviation.

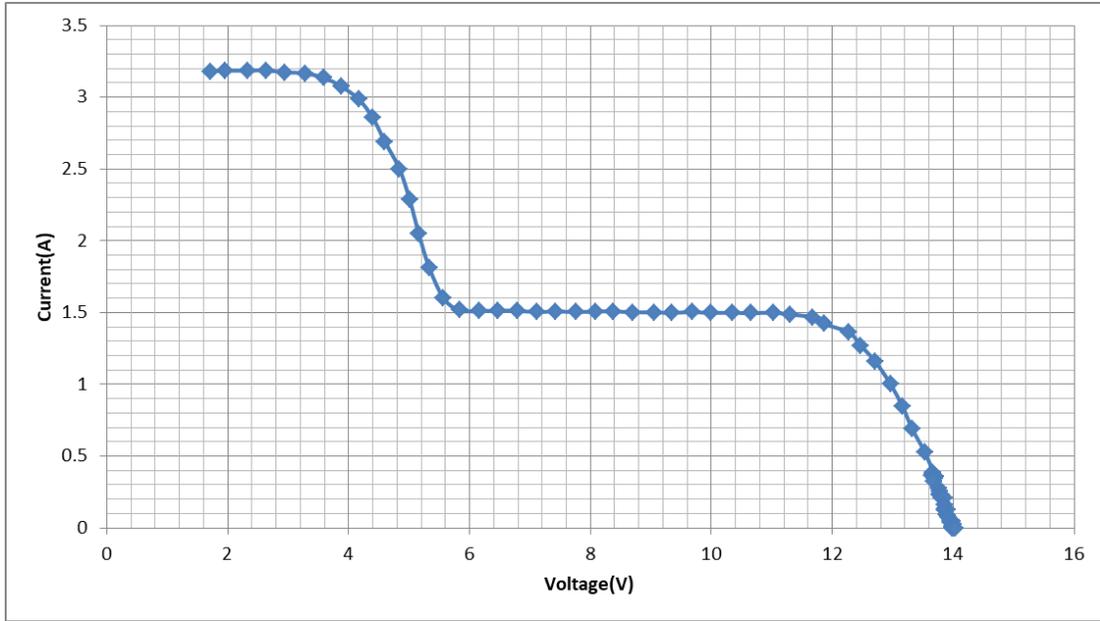
### **5.3 Analysis of the results for shading pattern 2**

In this shading pattern, it will be observed how a 20-45 degree angle between the photovoltaic modules can affect the I-V and P-V curves of the photovoltaic modules, the speed and accuracy of the Q-learning as well as the PSO algorithm operation.

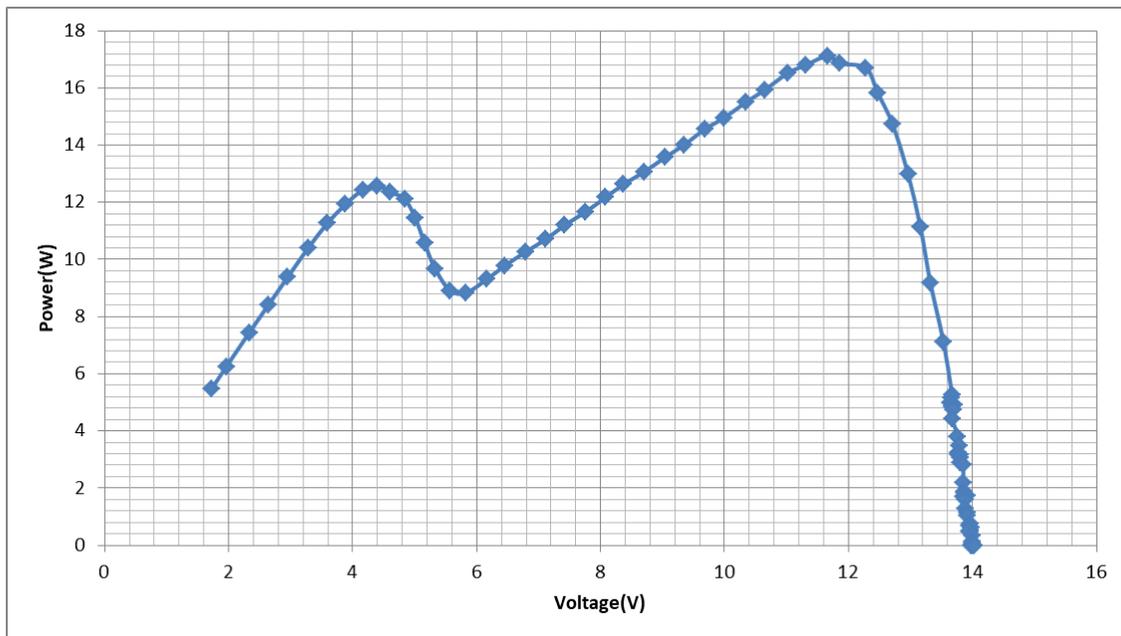
#### **Experiment No 1:**

**Solar irradiance for the two photovoltaic modules PV1: 480 W/m<sup>2</sup>, PV2:810 W/m<sup>2</sup> and Ambient Temperature: 24°C.**

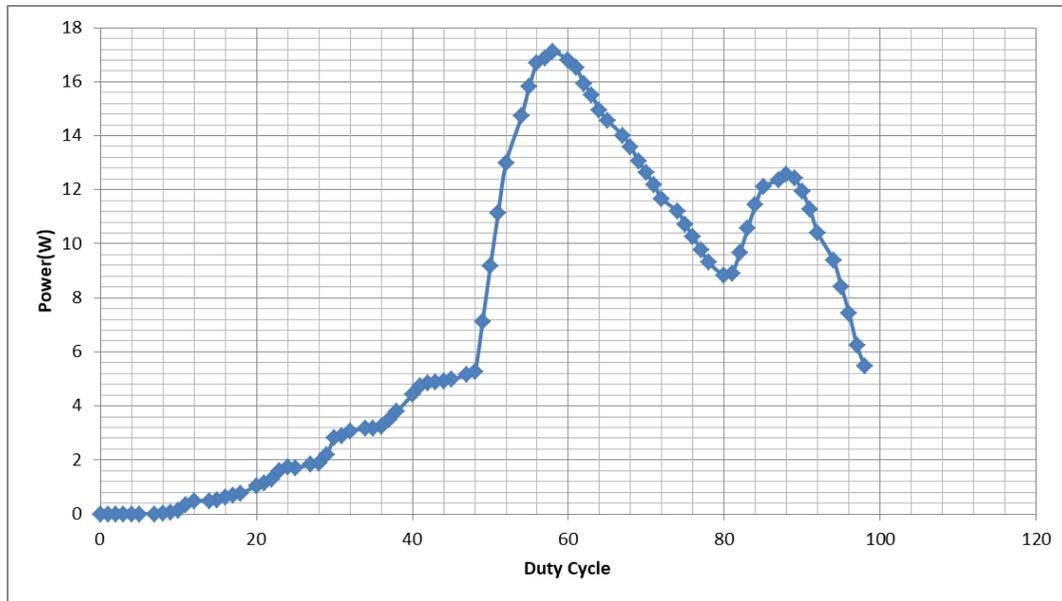
In Fig. 61 the PV array output current-voltage curve is presented and in Fig. 62 the PV array output power-voltage curve is presented. The PV modules operate under a non-uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig 63 which helps to detect in which duty cycle value the MPP is located.



**Figure 61.** The PV array output current-voltage curve.



**Figure 62.** The PV array output power-voltage curve.

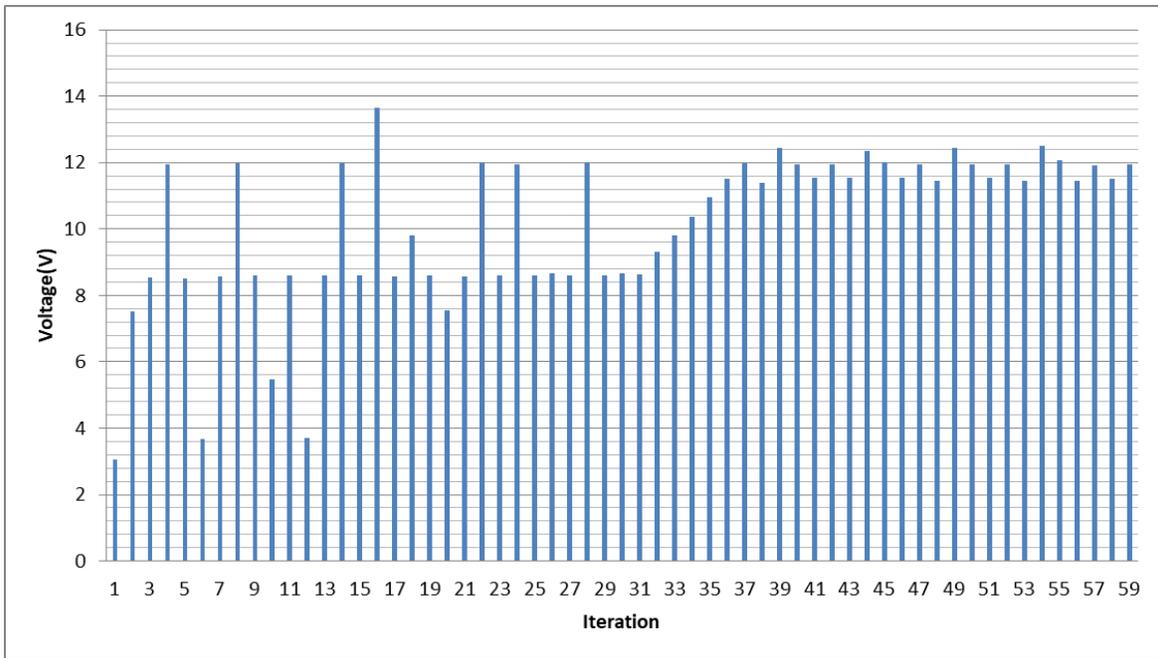


**Figure 63.** The PV array output power-duty cycle curve.

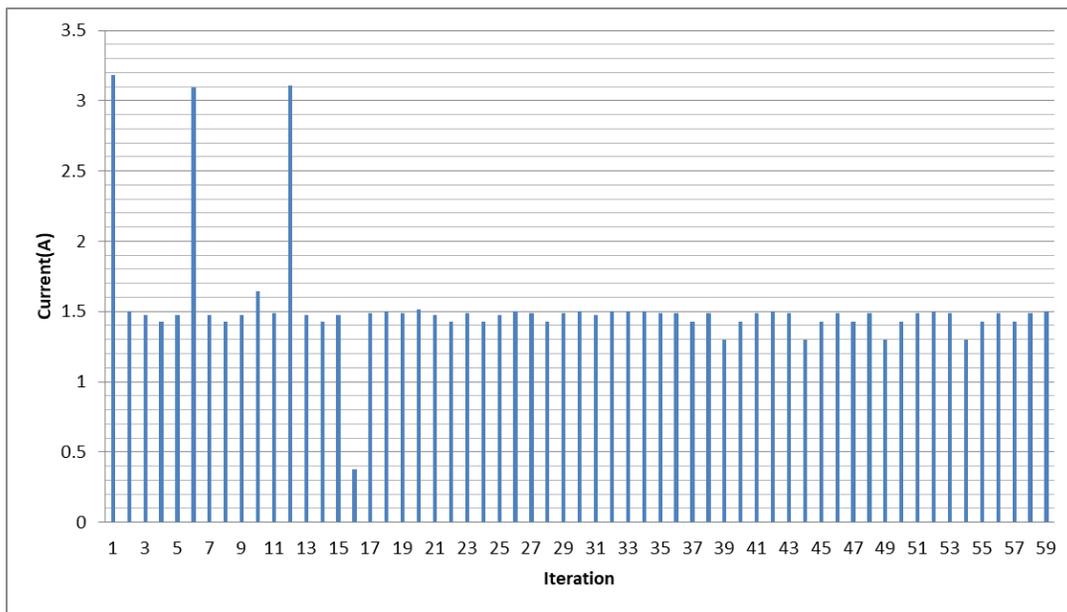
For these irradiance values and the angle between the photovoltaic modules and the angle between the photovoltaic modules and the sun, the global MPP is located at approximately 58-59% duty cycle but since there is an angle between the two photovoltaic modules a second small peak appears at approximately 88-89% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic (as shown in Figure 63). The presence of a second peak can be observed in all three graphs.

#### **Results of the Q-learning GMPPT algorithm:**

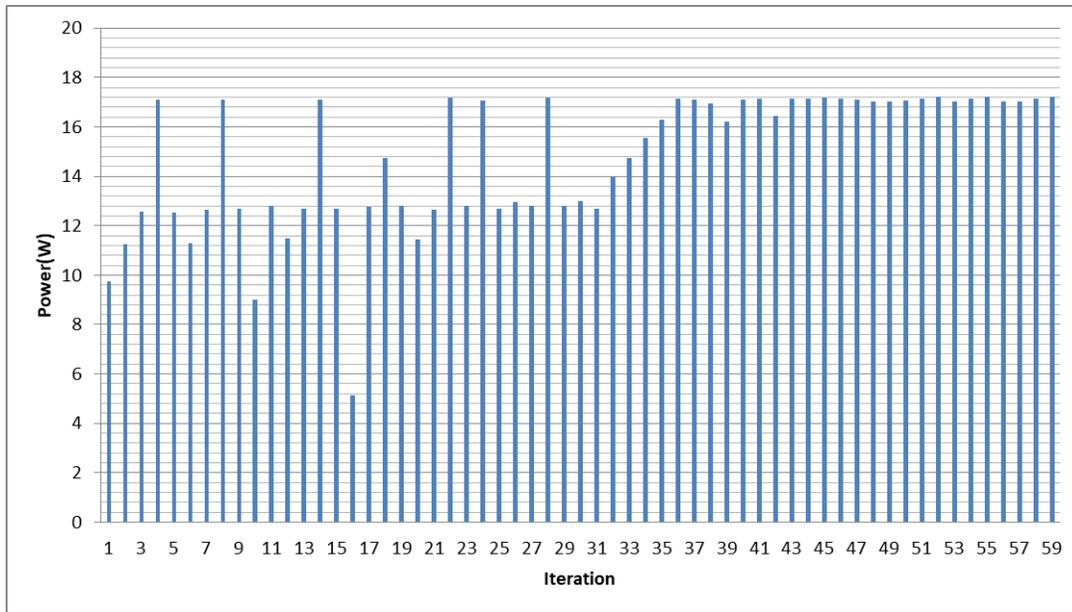
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 64-67 present the results of the above plots for shading pattern 1.



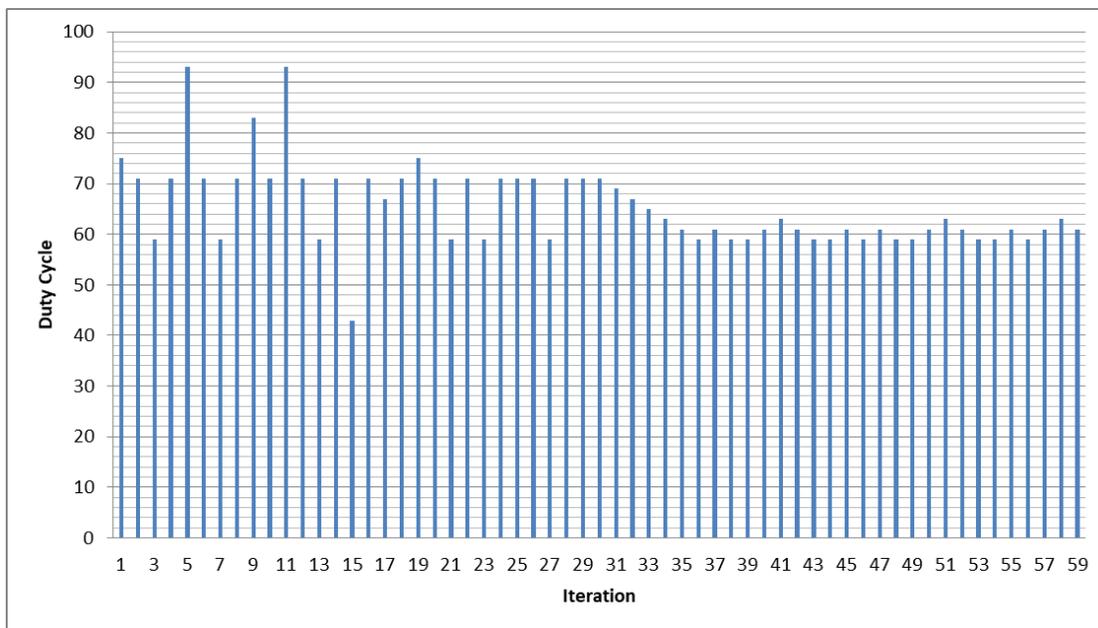
**Figure 64.** The PV array output voltage versus the number of search step.



**Figure 65.** The PV array output current versus the number of search step.



**Figure 66.** The PV array output power versus the number of search step.



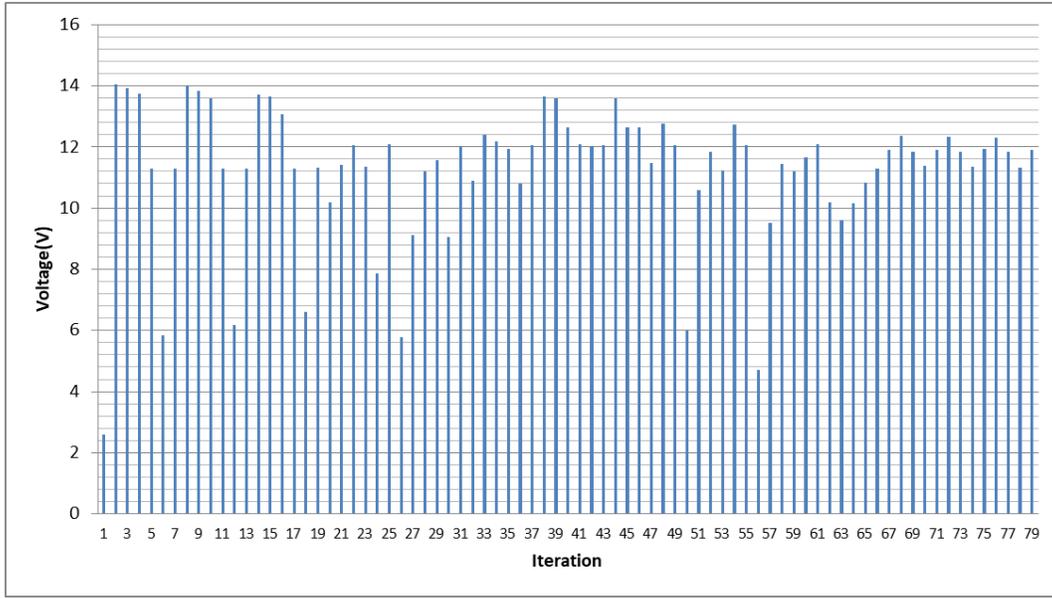
**Figure 67.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge; however that convergence time is longer than it was for shading pattern 1. It is considered that the exploration is over, when the oscillations start. These oscillations are due to

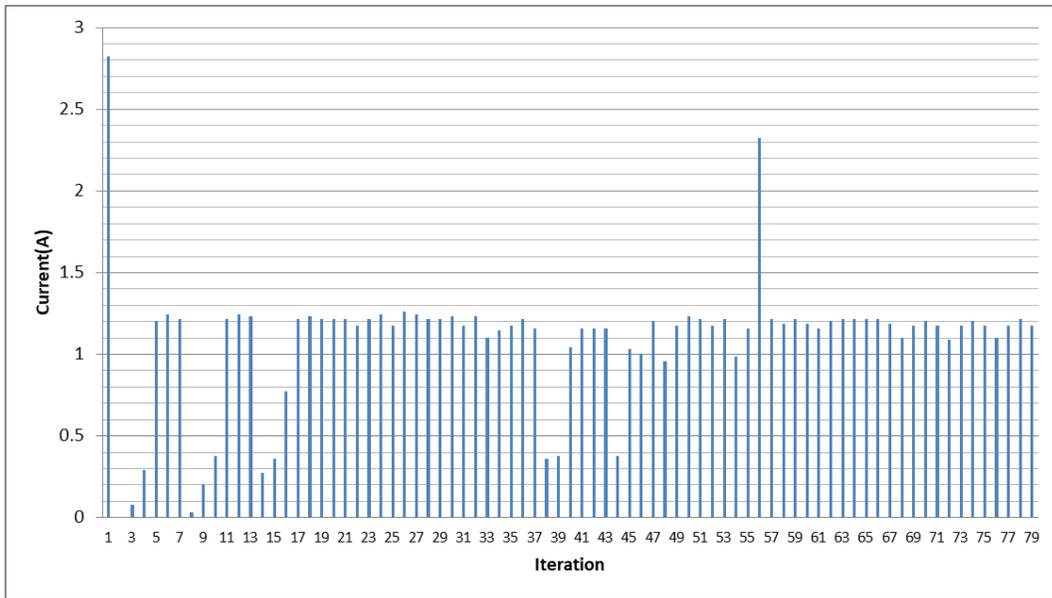
the P&O method, which is applied at that time with a perturbation step of 2%. The oscillations start occurring at approximately the 37<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate for fine-tuning. The oscillations can be seen on all four graphs and it can also be seen that it took 5 extra iterations for the Q-learning to converge compared to shading pattern 1 in which it converged at approximately the 32<sup>nd</sup> iteration. All the oscillations appear small except for the voltage-iteration graph, this time. The current now is more stable than it was on shading pattern 1 because the global MPP appears at a lower duty cycle which means closer to the  $V_{OC}$  and at a much lower current level than if it were on shading pattern 1, also meaning that changes in duty cycle around the MPP do not affect the current as much as they affect the voltage.

**Results of the PSO MPPT algorithm:**

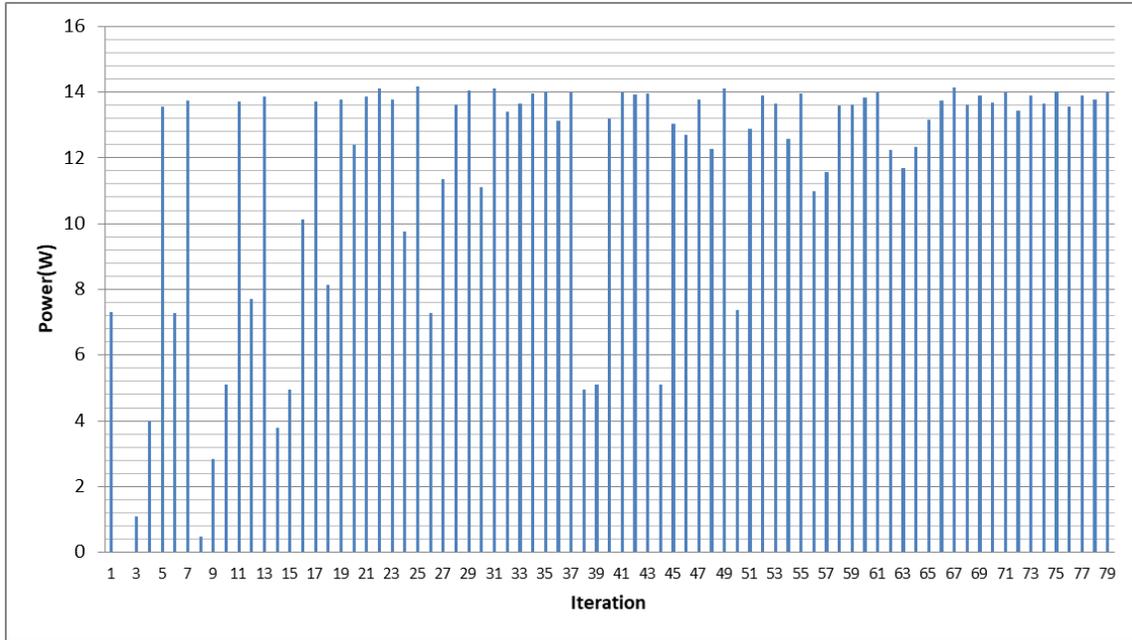
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 68-72 present the results of the above plots for shading pattern 2.



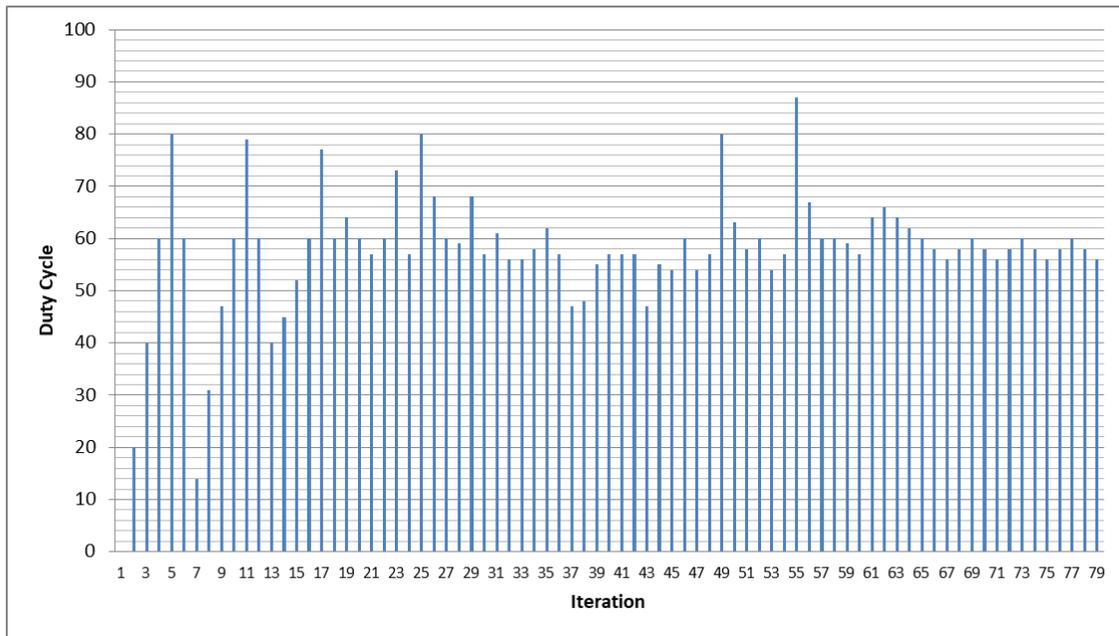
**Figure 68.** The PV array output voltage versus the number of search step.



**Figure 69.** The PV array output current versus the number of search step.

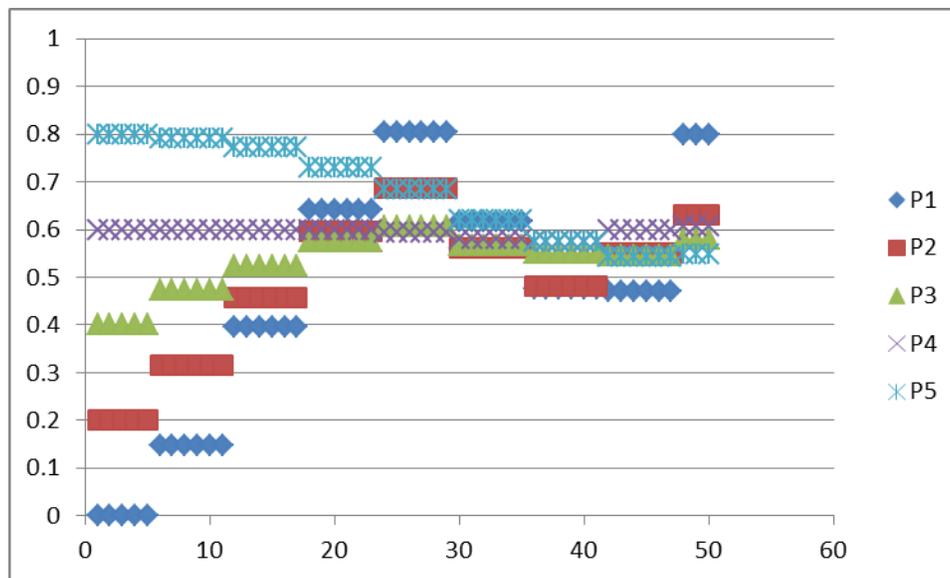


**Figure 70.** The PV array output power versus the number of search step.



**Figure 71.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm and longer than it would under shading pattern 1. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is when the P&O starts to operate for fine-tuning. The perturbation step for the P&O is 0.2 or 2%. The oscillations start at approximately the 67<sup>th</sup> iteration when the P&O converges. The oscillations can be seen on all four graphs. The same phenomenon is observed with the voltage stability as was in the Q-learning algorithm. The current now is more stable than it was on shading pattern 1 because the global MPP appears at a lower duty cycle which means closer to the  $V_{OC}$  and at a much lower current level than if it were on shading pattern 1, also meaning that changes in duty cycle around the MPP do not affect the current as much as they affect the voltage.



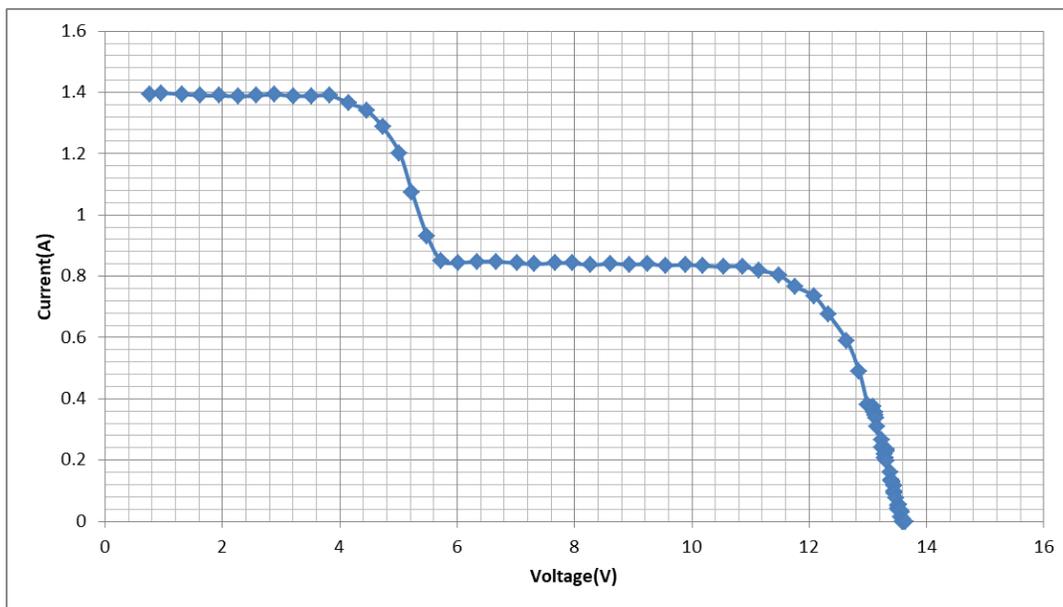
**Figure 72.** The particles position versus the number of search step.

As shown in Figure 72, the particles 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

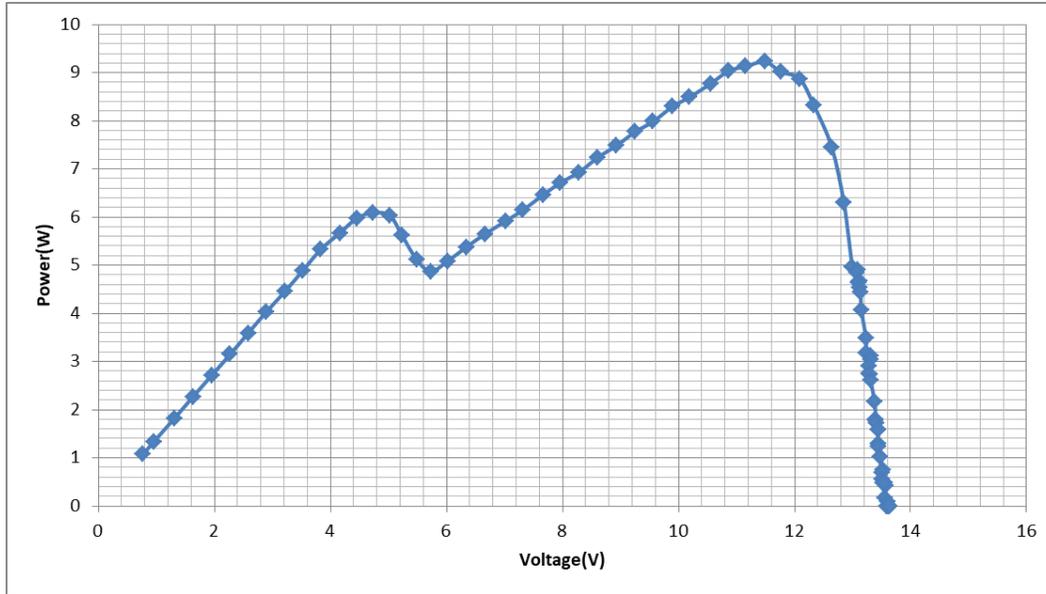
**Experiment No 2:**

**Solar irradiance for the two photovoltaic modules PV1: 265 W/m<sup>2</sup>, PV2:385 W/m<sup>2</sup> and Ambient temperature: 18°C.**

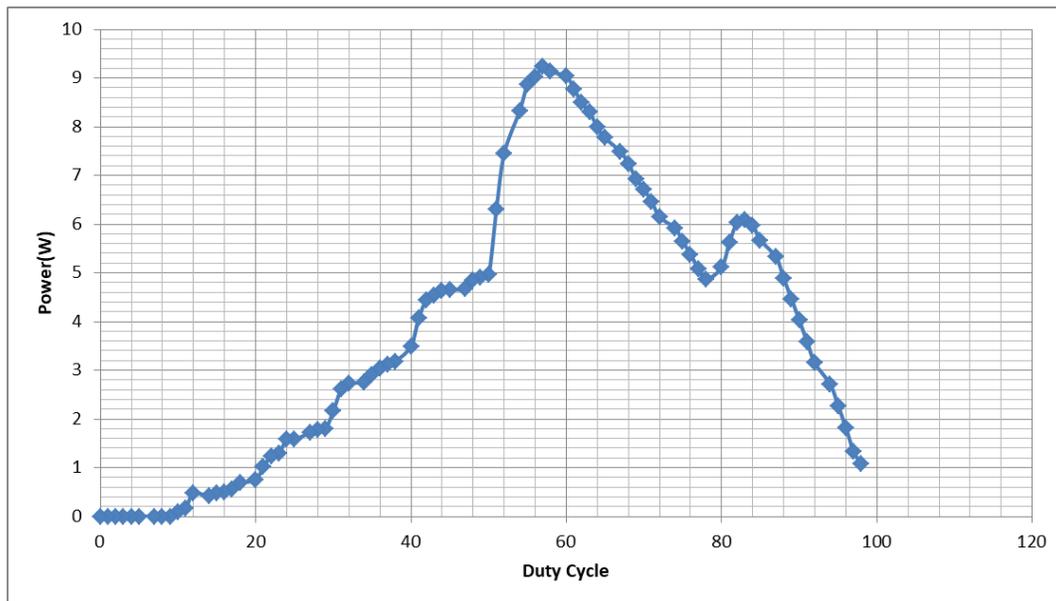
In Fig. 73 the PV array output current-voltage curve is presented and in Fig. 74 the PV array output power-voltage curve is presented. The PV operates under a non-uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig.75 which helps to detect in which duty cycle value the MPP is located.



**Figure 73.** The PV array output current-voltage curve.



**Figure 74.** The PV array output power-voltage curve.



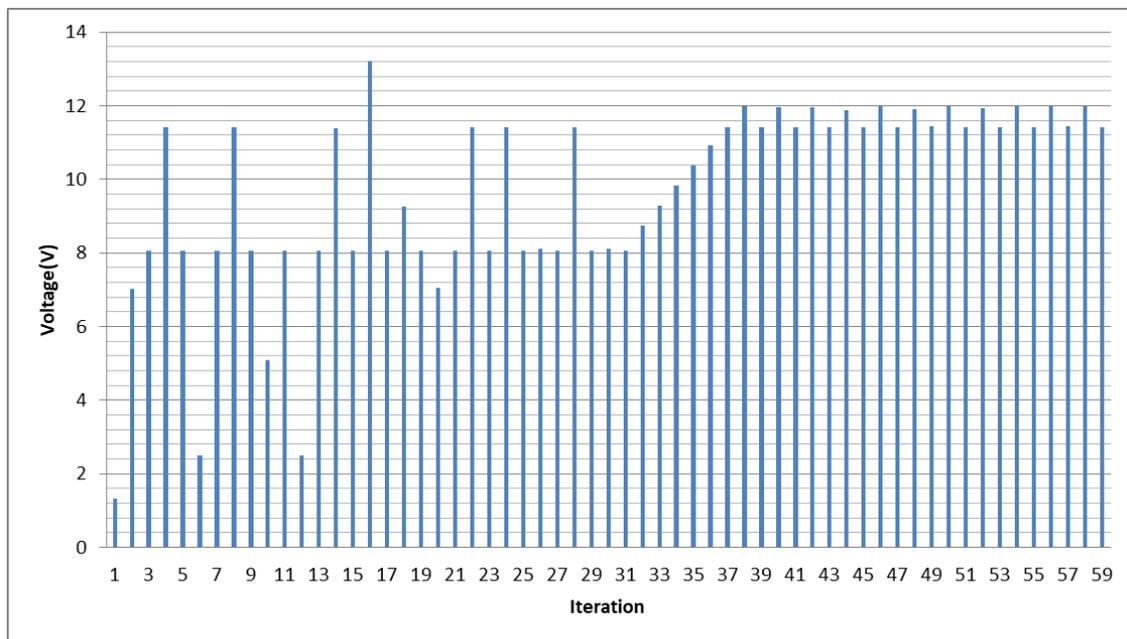
**Figure 75.** The PV array output power-duty cycle curve.

For those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun, the global MPP is located at approximately 57-58% duty

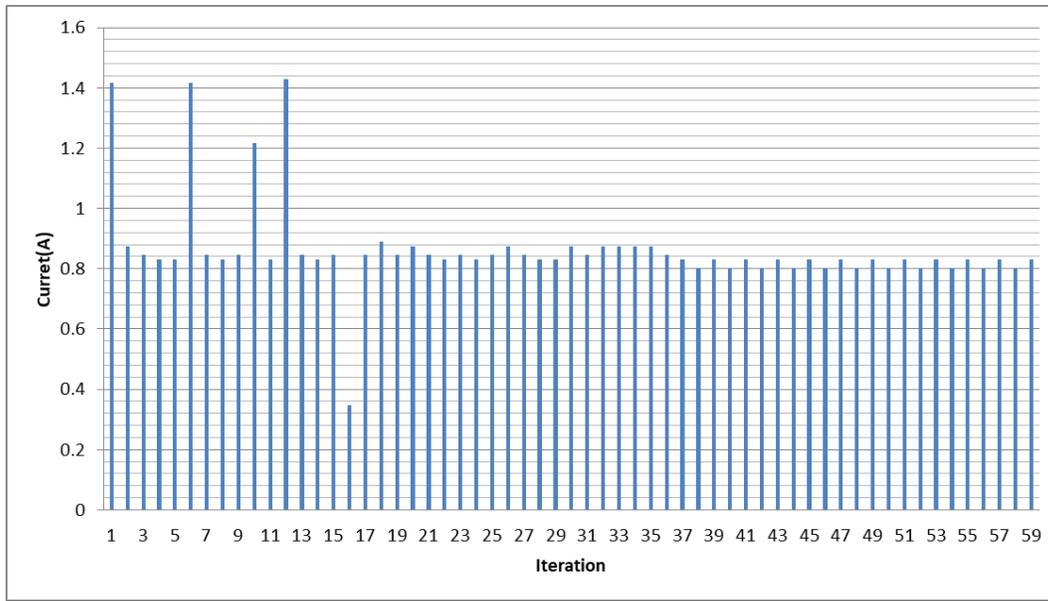
cycle but since there is an angle between the two photovoltaic modules a second small peak appears at approximately 88-89% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module as shown in Figure 75. The presence of a second peak can be observed in all three graphs.

### Results of the Q-learning MPPT algorithm:

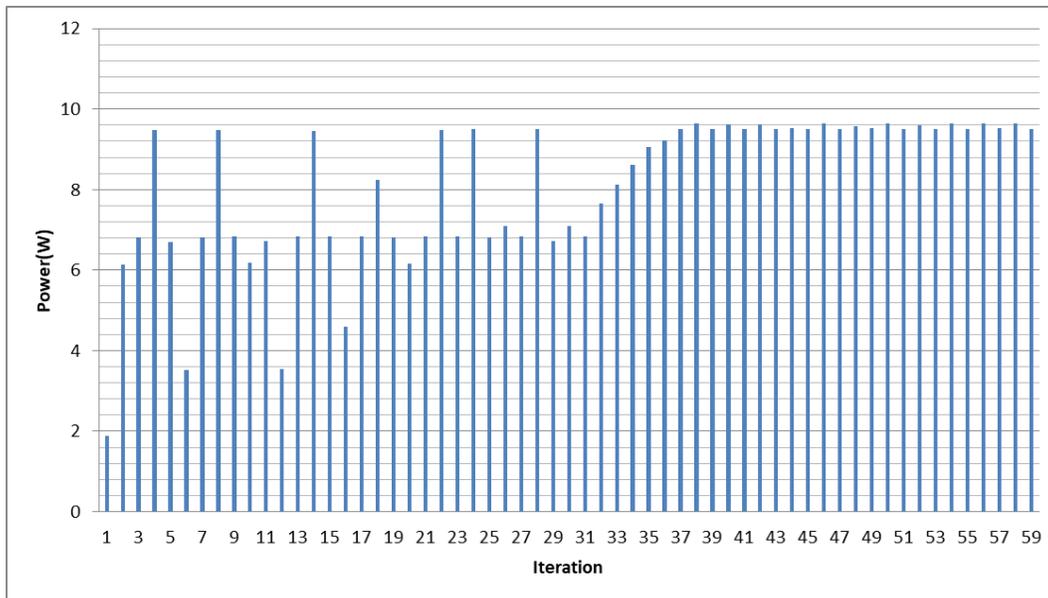
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 76-79 present the results of the above plots for shading pattern 1.



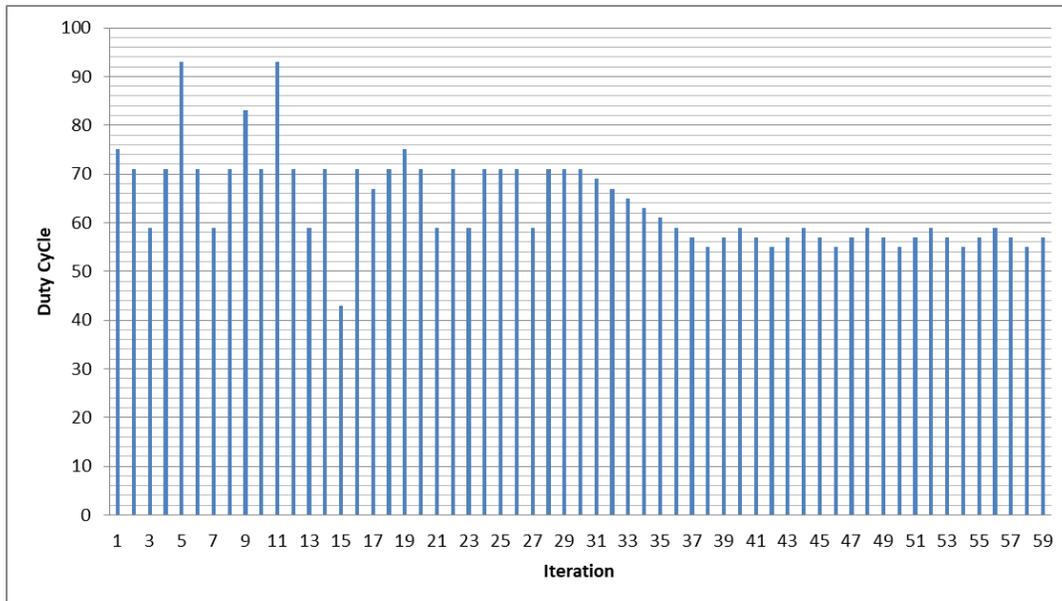
**Figure 76.** The PV array output voltage versus the number of search step.



**Figure 77.** The PV array output current versus the number of search step.



**Figure 78.** The PV array output power versus the number of search step.



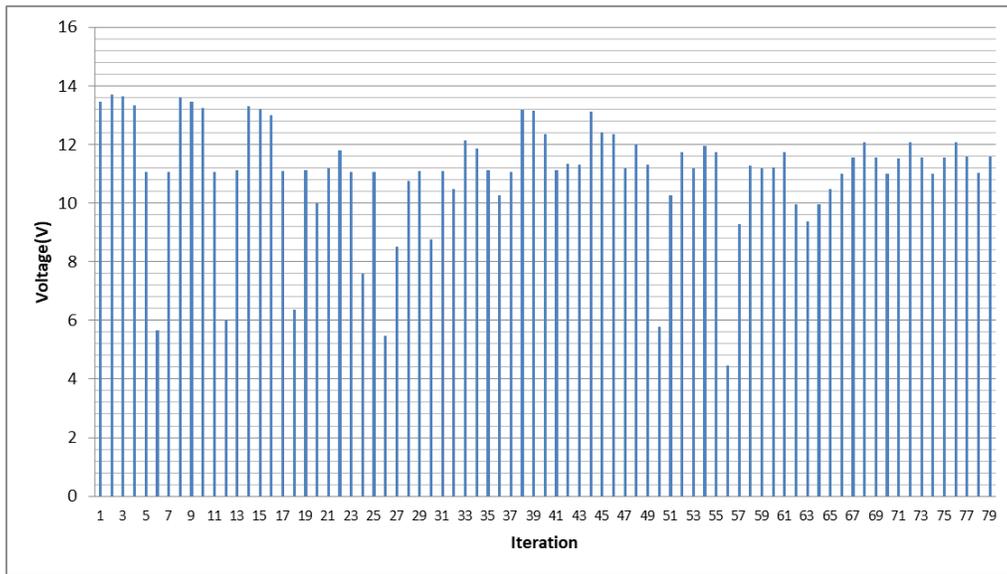
**Figure 79.**The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge; however that convergence time is longer than it was for shading pattern 1. It is considered that the exploration is over, when the oscillations start. The oscillations start occurring at the 40<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm takes over for fine-tuning. The oscillations can be seen on all four graphs and it can also be seen that it took 6 extra iterations for the Q-learning to converge compared to shading pattern 1 in which it converged at the 32<sup>nd</sup> iteration. All the oscillations appear small. The current now is more stable around the MPP than it was on shading pattern 1 because the global MPP appears at a lower duty cycle which means closer to the  $V_{OC}$  and at a much lower current level than for shading pattern 1, since the irradiance is much lower, also meaning that changes in duty cycle around the MPP do not affect the current as much as they affect the voltage.

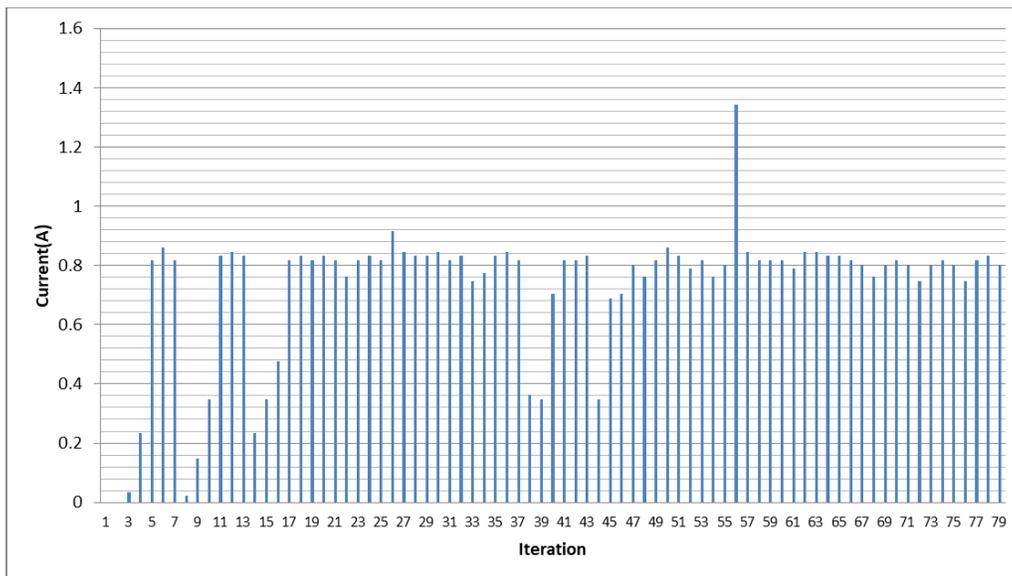
### **Results of the PSO GMPPT algorithm:**

Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot,

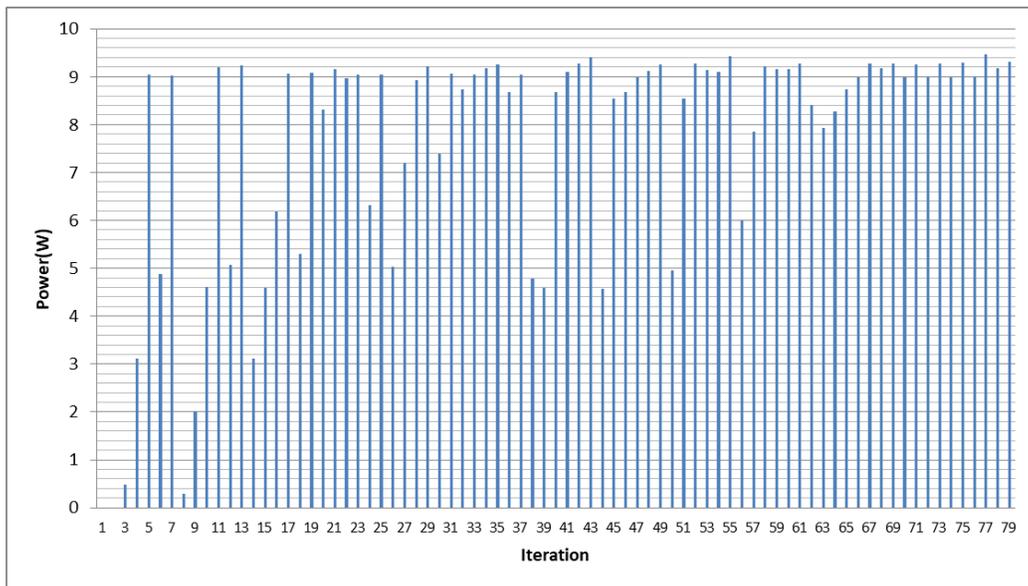
the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 80-84 present the results of the above plots for shading pattern 2.



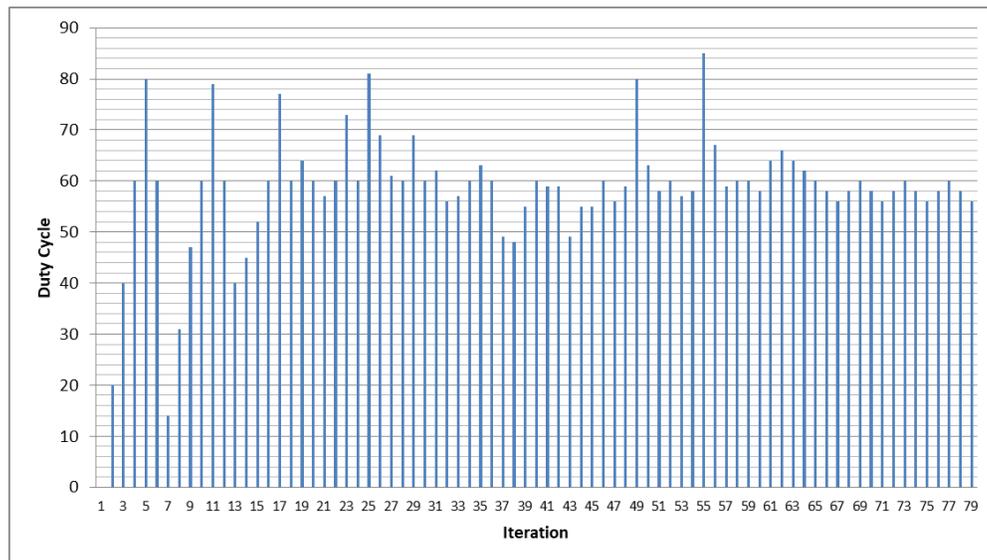
**Figure 80.** The PV array output voltage versus the number of search step.



**Figure 81.** The PV array output current versus the number of search step.



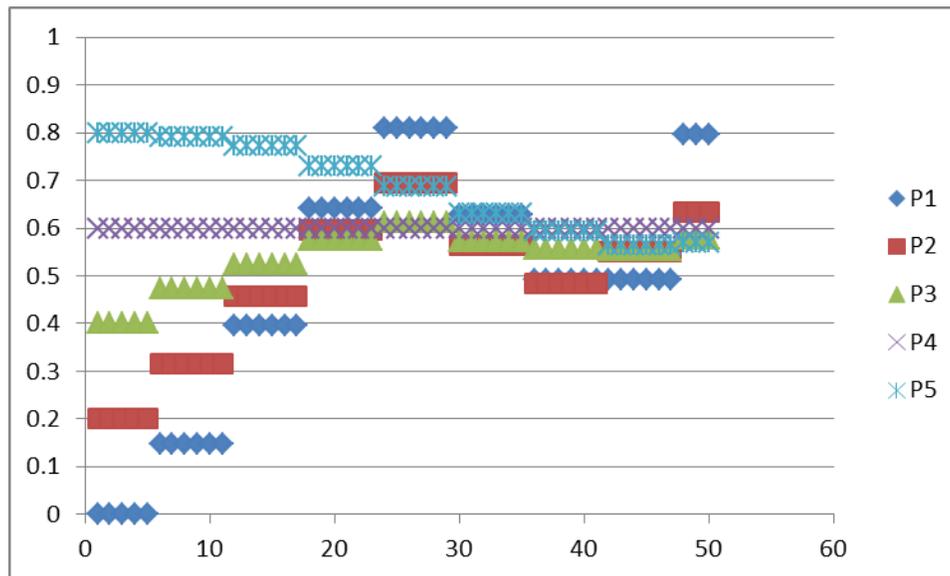
**Figure 82.** The PV array output power versus the number of search step.



**Figure 83.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm and longer than it would under shading pattern 1. It is considered that the exploration is over when the number of iterations

reach 50 (5 particles \* 10 generations) which is when the P&O algorithm starts to operate for fine-tuning. The oscillations start at the 67<sup>th</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs. The current now is more stable than for shading pattern 1 because the global MPP appears at a lower duty cycle which means closer to the  $V_{OC}$  and at a much lower current level since the irradiance is much lower than in shading pattern 1. Once again the PSO algorithm converges accurately.



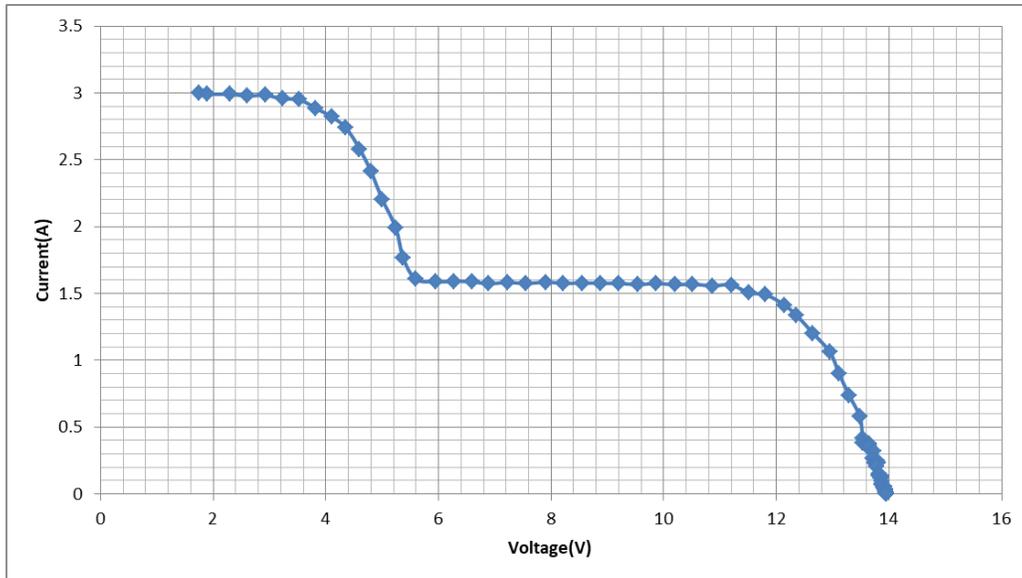
**Figure 84.** The particles position-iterations or No of steps.

As shown in Figure 84, the particles 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

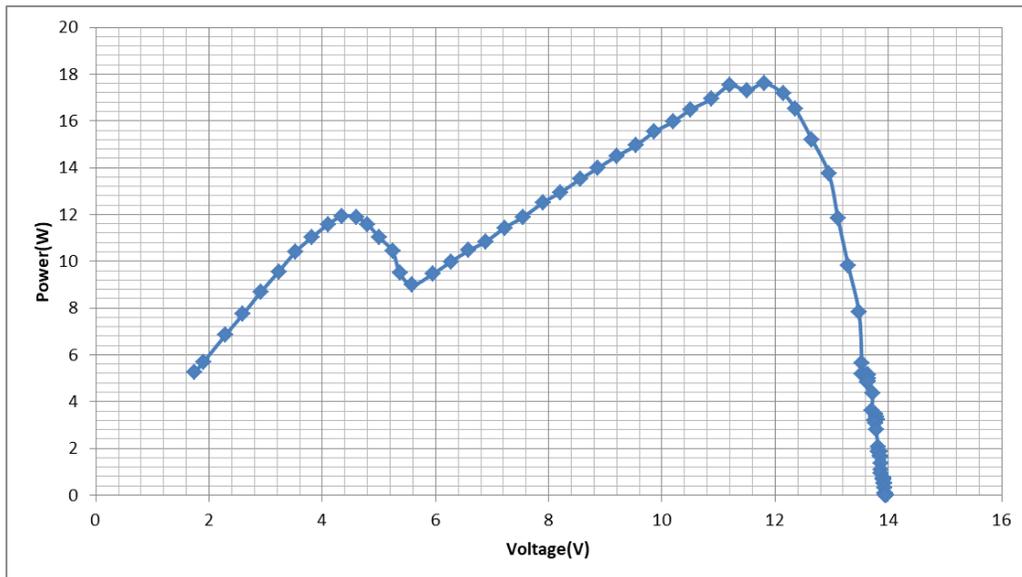
### **Experiment No 3:**

**Solar irradiance for the two photovoltaic modules PV1: 810 W/m<sup>2</sup>, PV2: 425 W/m<sup>2</sup> and Ambient temperature:23°C.**

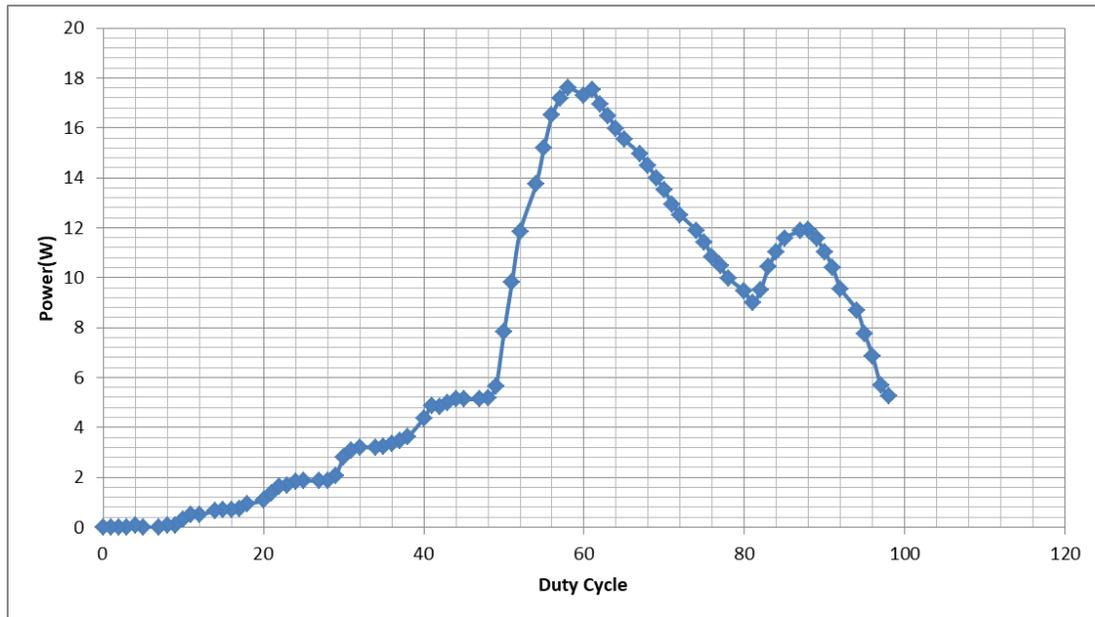
In Fig. 85 the PV array output current-voltage curve is presented and in Fig.86 the PV array output power-voltage curve is presented. The PV operates under a non-uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 87 which helps to detect in which duty cycle value the MPP is located.



**Figure 85.** The PV array output current-voltage curve.



**Figure 86.** The PV array output power-voltage curve.

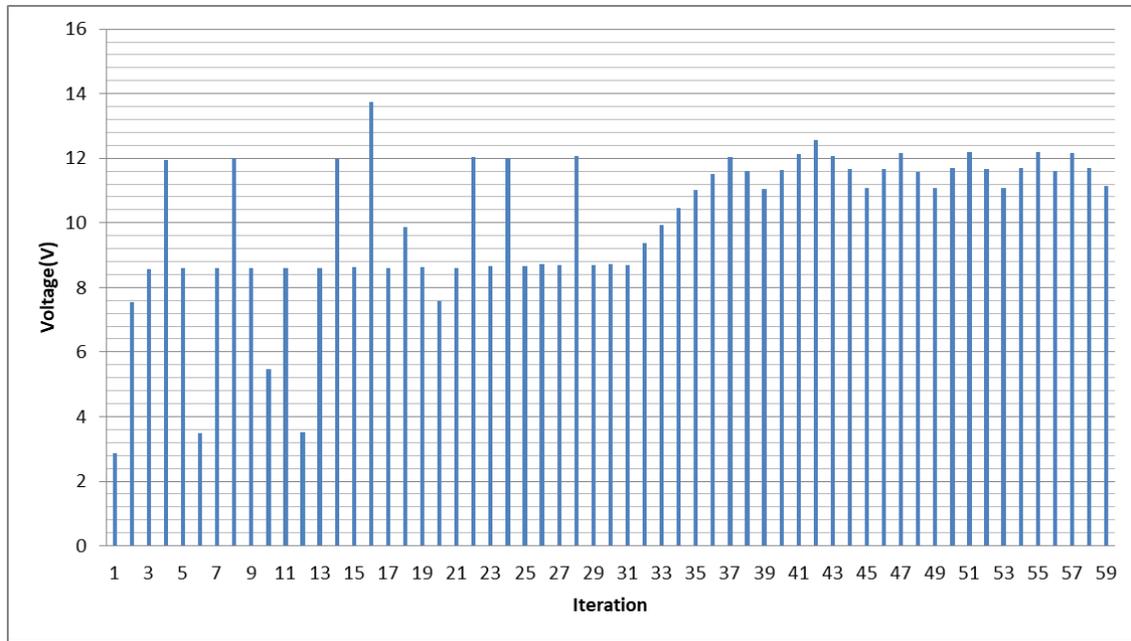


**Figure 87.** The PV array output power-duty cycle curve.

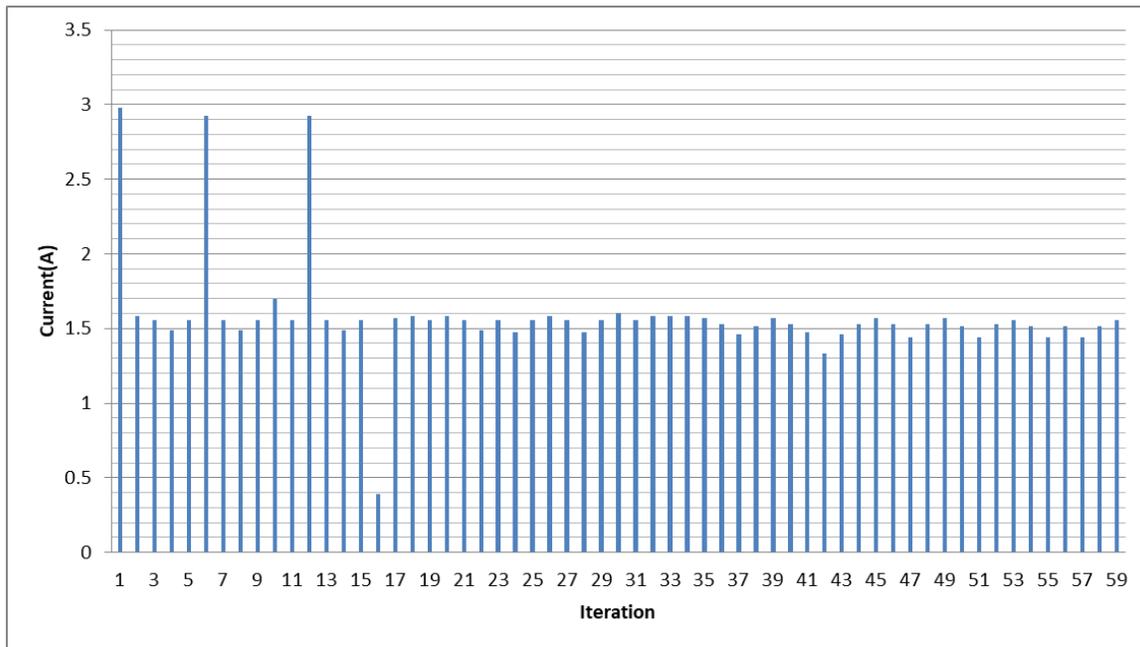
For those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun the global MPP is located at approximately 59% duty cycle but since there is an angle between the two photovoltaic modules a second small peak appears at approximately 88-89% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module as shown in Figure 87. The presence of a second peak can be observed in all three graphs.

### **Results of the Q-learning GMPPT algorithm:**

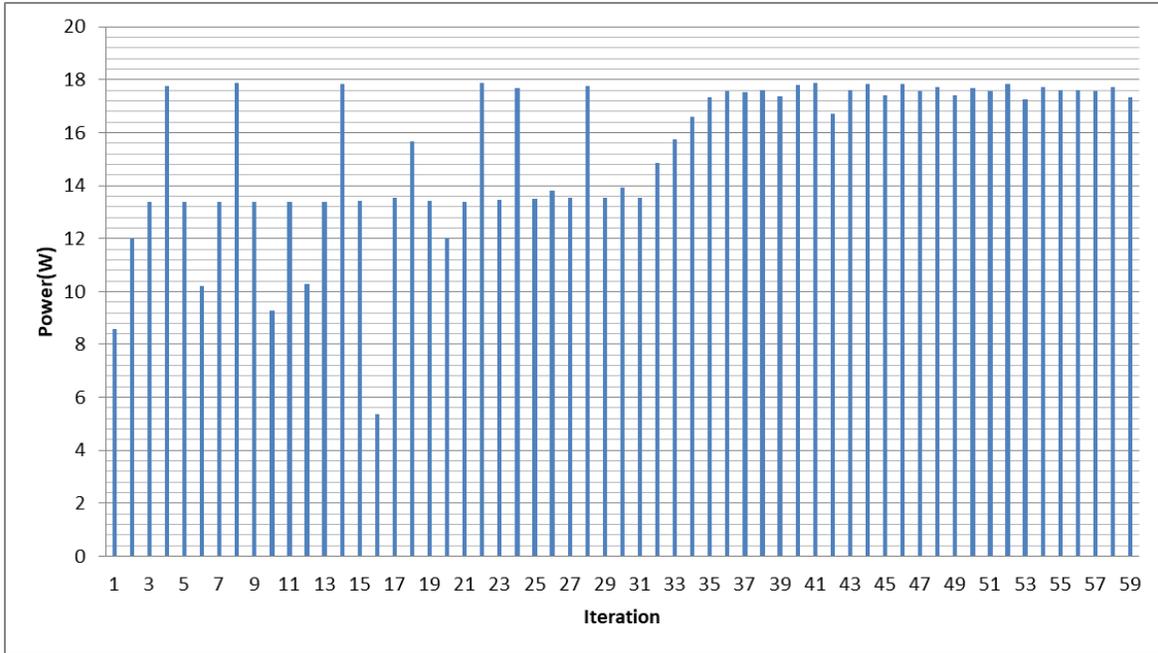
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 88-91 present the results of the above plots for shading pattern 1.



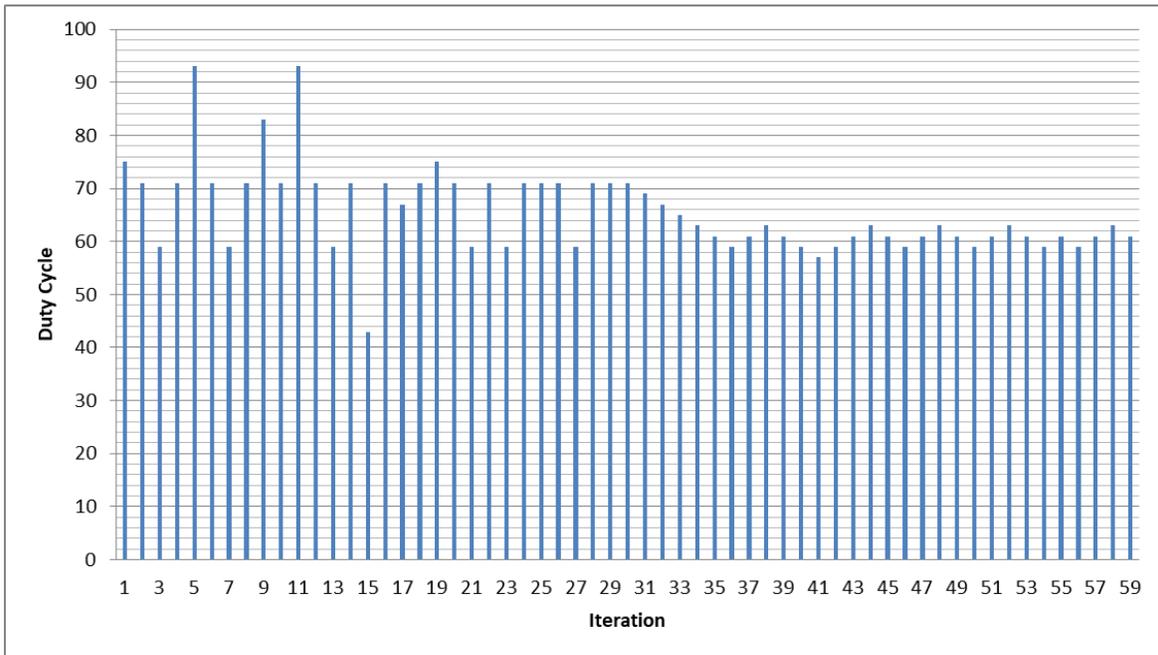
**Figure 88.** The PV array output voltage versus the number of search step.



**Figure 89.** The PV array output current versus the number of search step.



**Figure 90.** The PV array output power versus the number of search step.

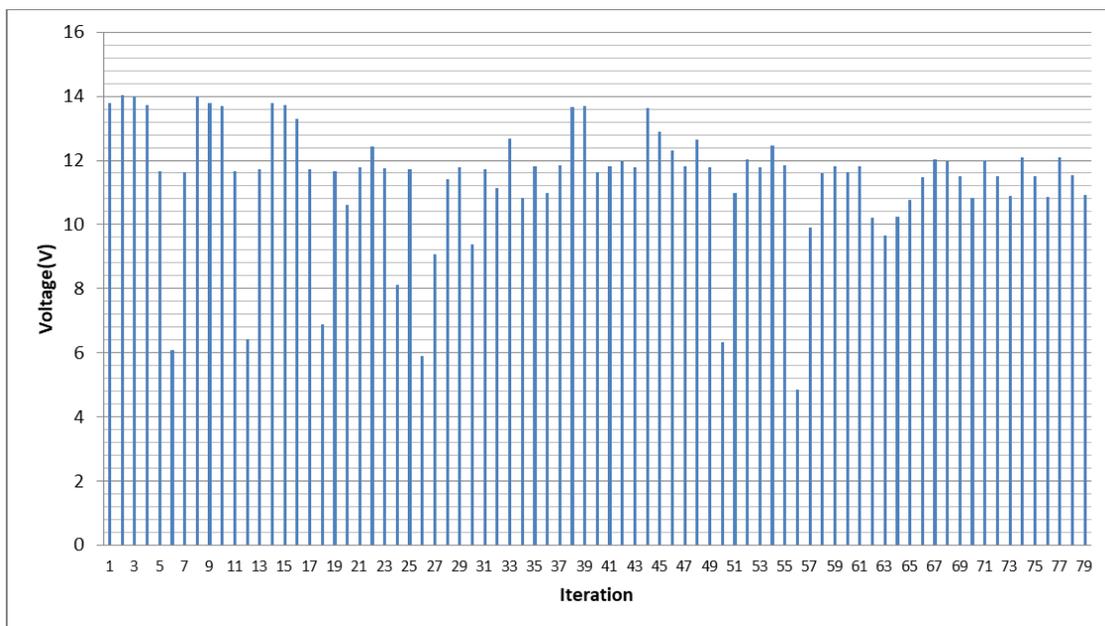


**Figure 91.** The PV array output duty cycle versus the number of search step.

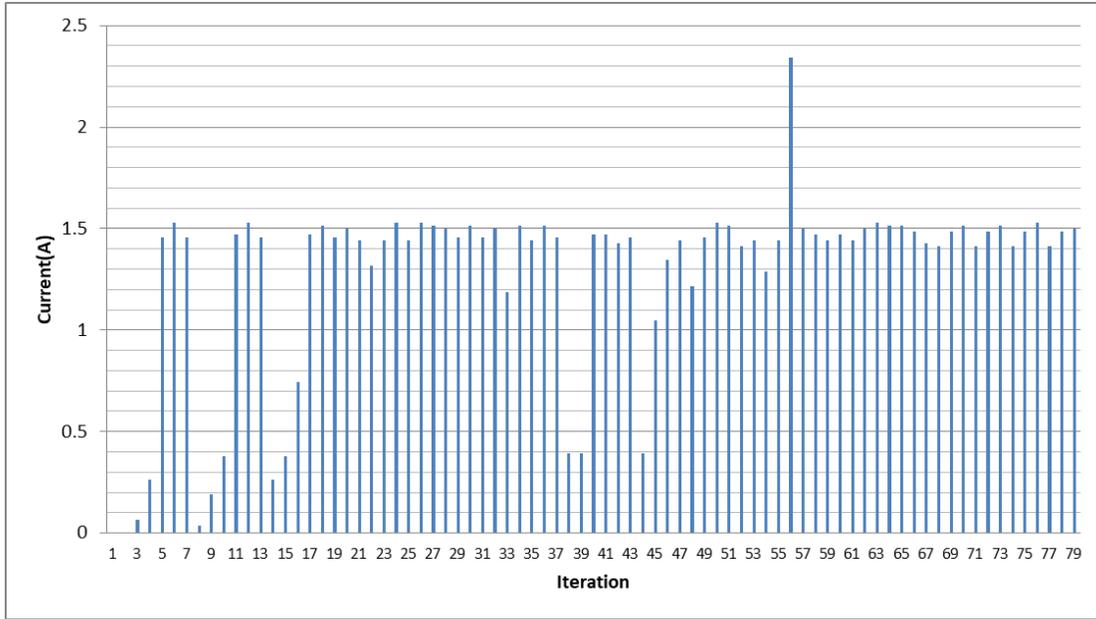
It is observed that the algorithm does not need a relatively long amount of time in order to converge; however that convergence time is longer than it was for shading pattern 1 once again. It is considered that the exploration is over, when the oscillations start. The oscillations start at the 36<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate for fine-tuning. The oscillations can be seen on all four graphs.

### Results of the PSO algorithm:

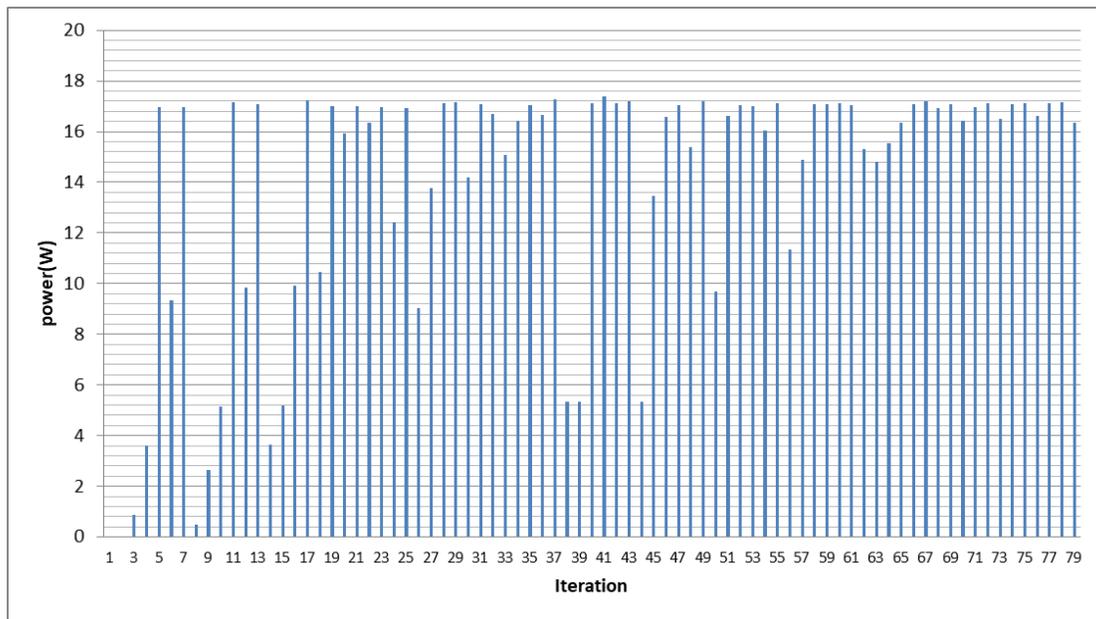
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 92-96 present the results of the above plots for shading pattern 2.



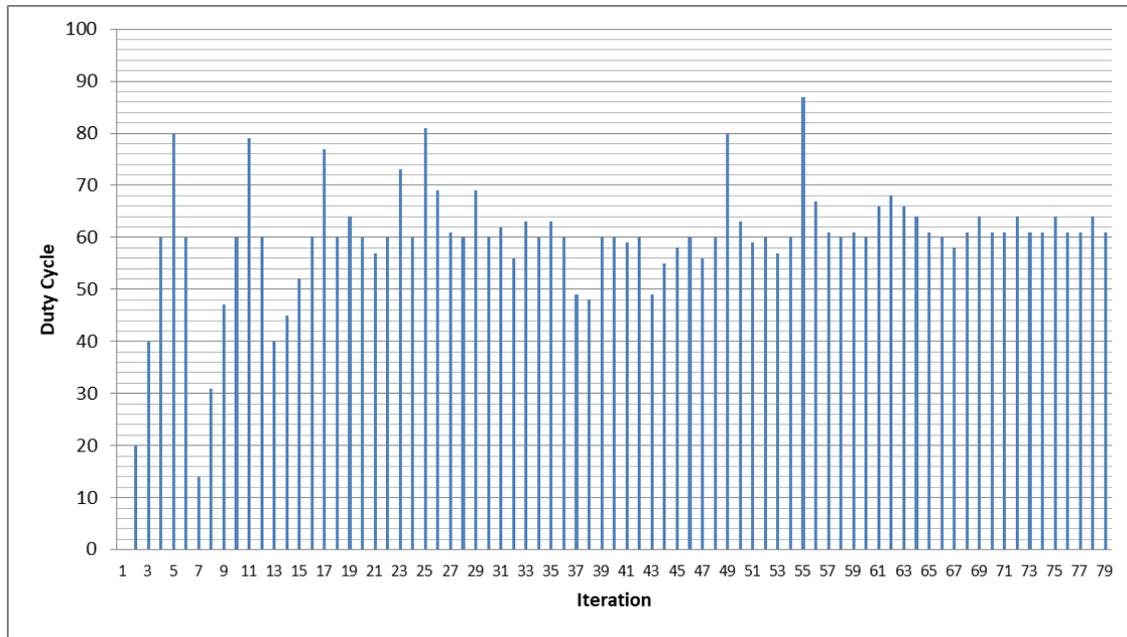
**Figure 92.** The PV array output voltage versus the number of search step.



**Figure 93.** The PV array output current versus the number of search step.

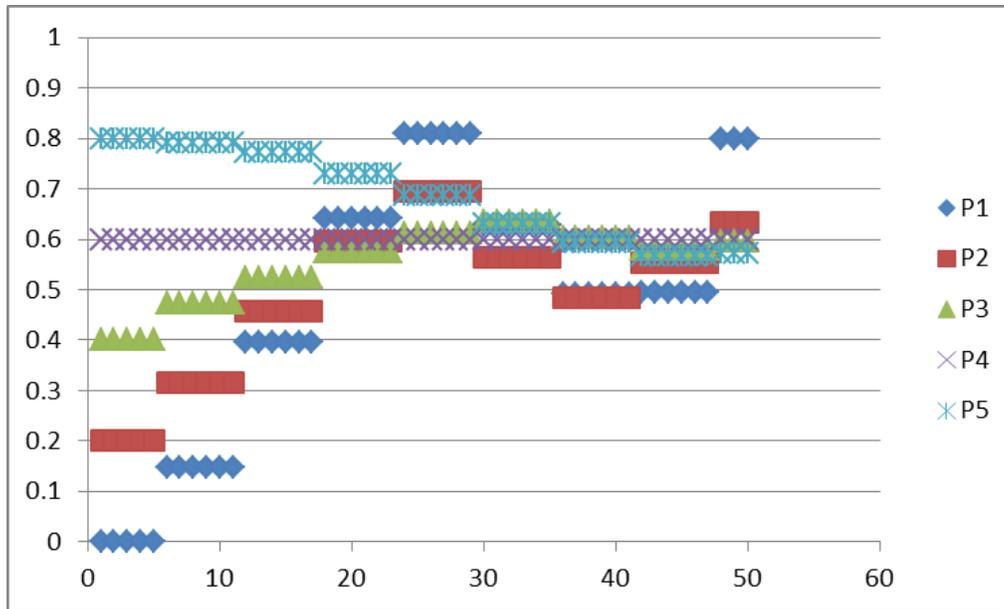


**Figure 94.** The PV array output power versus the number of search step.



**Figure 95.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm and longer than it would under shading pattern 1. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is when the P&O starts to operate for fine-tuning. The oscillations start at the 68<sup>st</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs. The current now is more stable than it was on shading pattern 1 because the global MPP appears at a lower duty cycle which means closer to the  $V_{OC}$  and at a much lower current level since the irradiance is much lower than in shading pattern 1. Once again the PSO converges accurately.



**Figure 96.** The particles position versus the number of search step.

As shown in Figure 96, the particles 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

### **Comments on the second pattern:**

In this pattern the Q-learning algorithm was tested and compared to the PSO algorithm based on a higher than in the first shading pattern irradiance difference between the two photovoltaic modules accomplished by 20-45 degree angle difference between them. Taking a closer look at the power curves it can be seen how the angle between the photovoltaic modules and the sun affects the value of the maximum power that can be obtained but not so much the duty cycle under which the MPP appears. In the second measurement it can be seen how lower the maximum power is compared to the other two. It was expected by both algorithms to converge correctly as they did, with the Q-learning algorithm being faster every time and slightly more precise. Even though the P&O algorithm was applied in both cases, it was capable to converge more accurately in the Q-learning algorithm since more duty cycle values could be explored due to the variety of actions that could be applied. For instance, if the MPP was located at a 60%

duty cycle, the Q-learning algorithm would be able to sense the power at that level because the actions allowed the agent to explore the 60% duty cycle, but the PSO can explore the search-space based on the velocity of the particles meaning that if the velocity is not favorable enough, then the 60% duty cycle might not be explored at all and instead it can converge to either 63% or 59% where the P&O, having a 0.2 step, could fine-tune it to 61% or 57% which is clearly a duty cycle with a slight deviation. Lastly, it was observed how a second smaller peak affected the convergence speed (i.e. the number of iterations at which the algorithms converged).

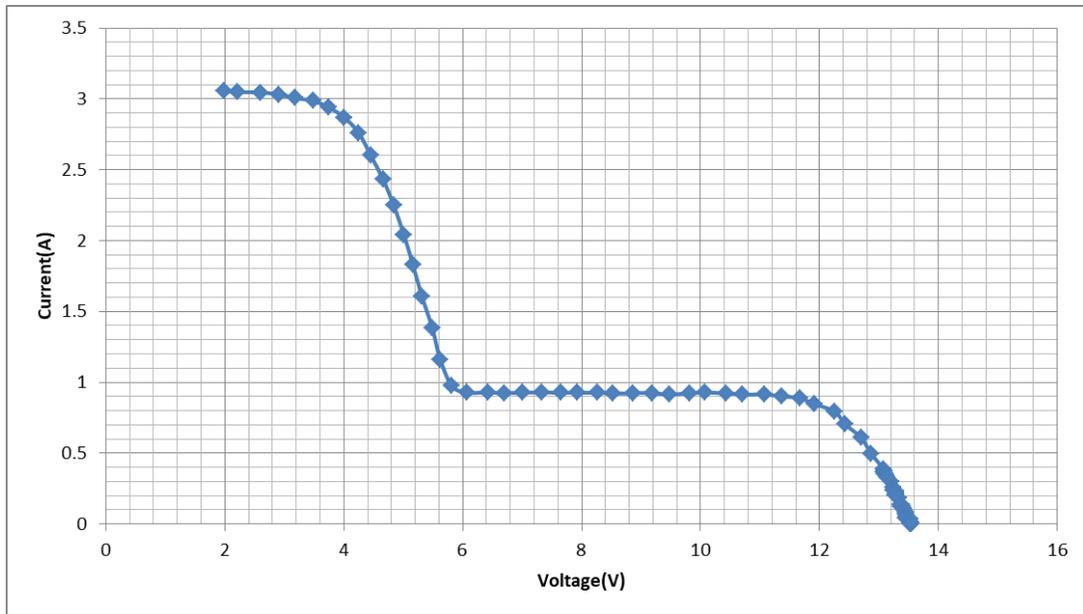
## **5.4 Analysis of the results for shading pattern 3**

In this shading pattern, it will be observed how a 45-90 degree angle between the photovoltaic modules can affect the I-V and P-V curves of the photovoltaic modules, the speed and accuracy of the Q-learning algorithm, as well as the PSO algorithm.

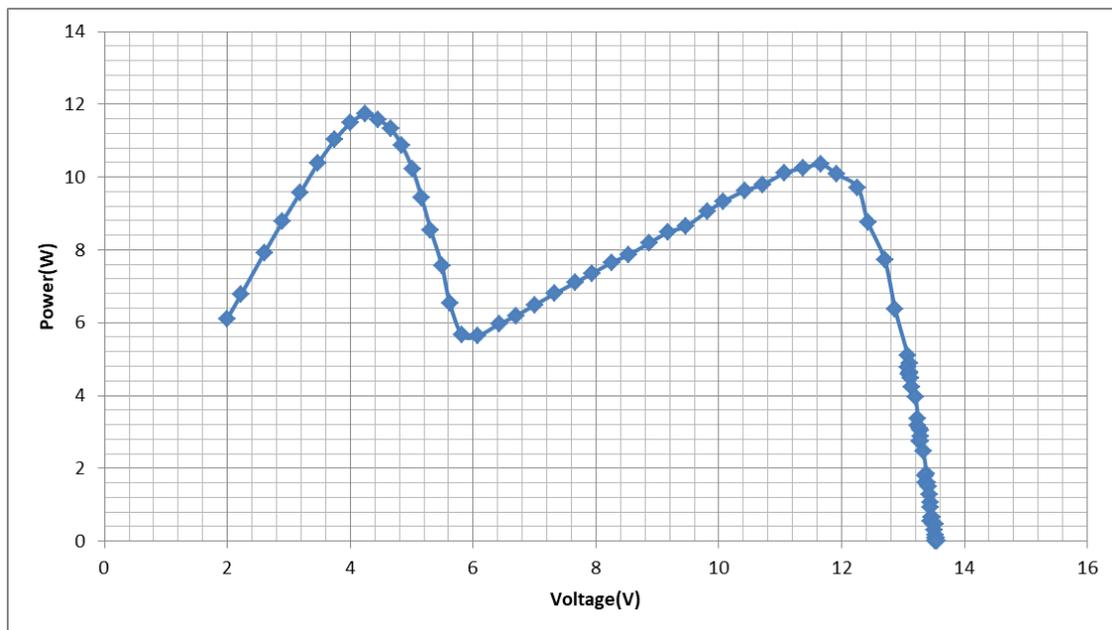
### **Experiment No 1:**

**Solar irradiance for the two photovoltaic modules PV1: 770 W/m<sup>2</sup>, PV2: 305 W/m<sup>2</sup> and Ambient temperature: 23°C.**

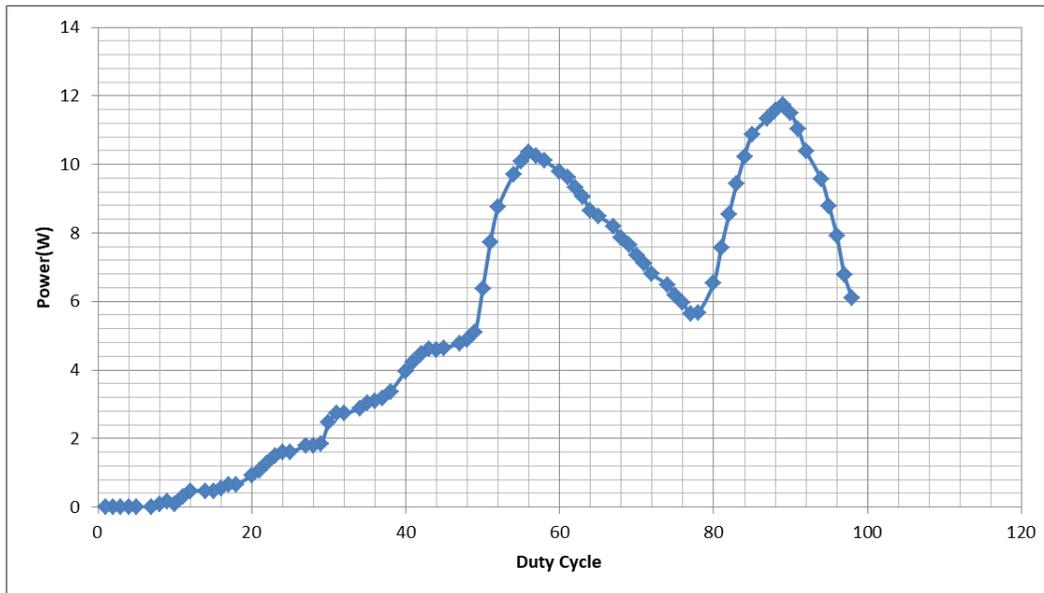
In Fig. 97 the PV array output current-voltage curve is presented and in Fig. 98 the PV array output power-voltage curve is presented. The PV array operates under a non-uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 99 which helps to detect in which duty cycle value the MPP is located.



**Figure 97.** The PV array output current-voltage curve.



**Figure 98.** The PV array output power-voltage curve.

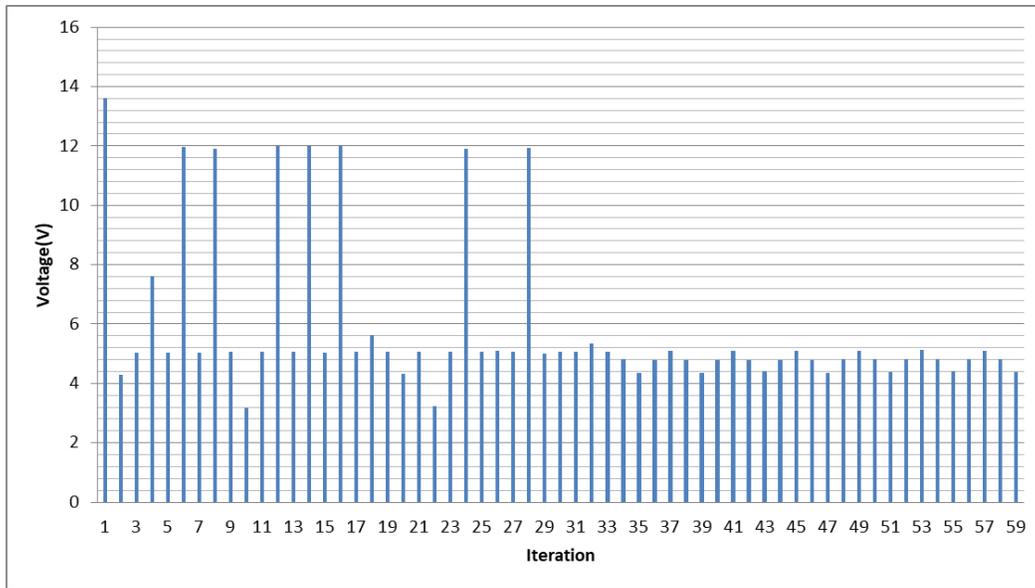


**Figure 99.** The PV array output power-duty cycle curve.

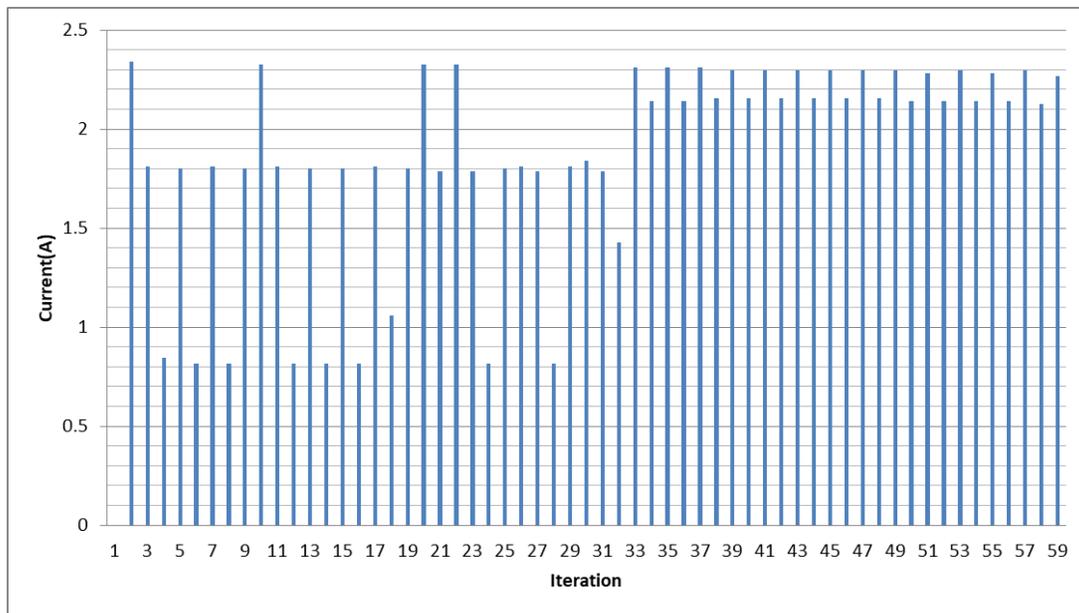
For those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun, the global MPP is located at approximately 88-89% duty cycle but since there is an angle between the two photovoltaic modules, a second peak appears with a power value at approximately 56% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module, as shown in Figure 99. The presence of a second peak can be observed in all three graphs.

#### **Results of the Q-learning GMPPT algorithm:**

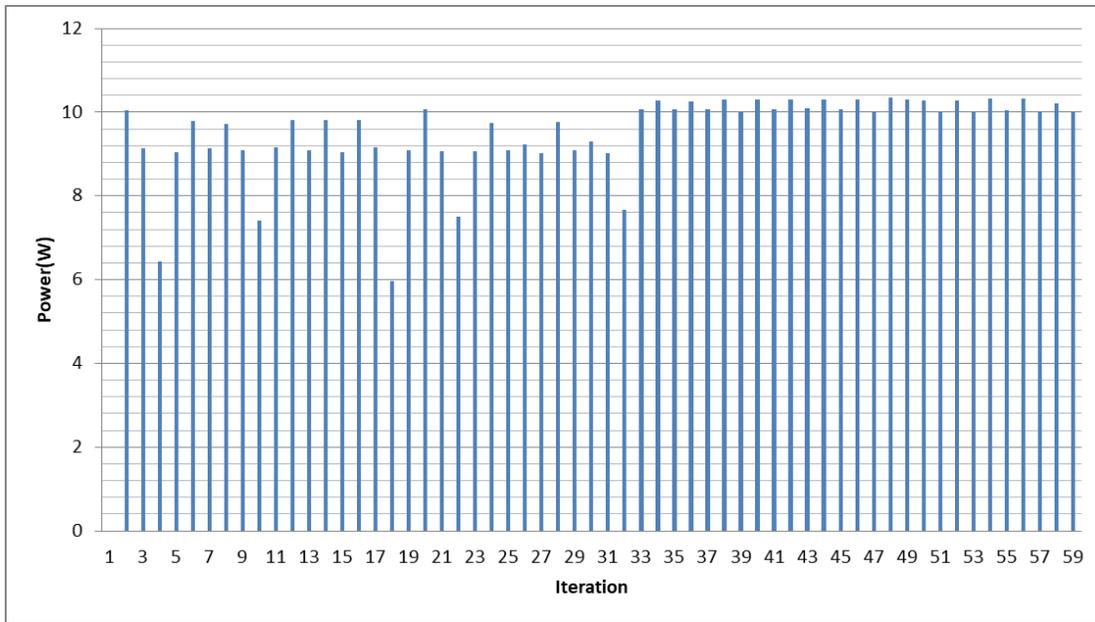
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 100-103 the results of the above plots for shading pattern 1.



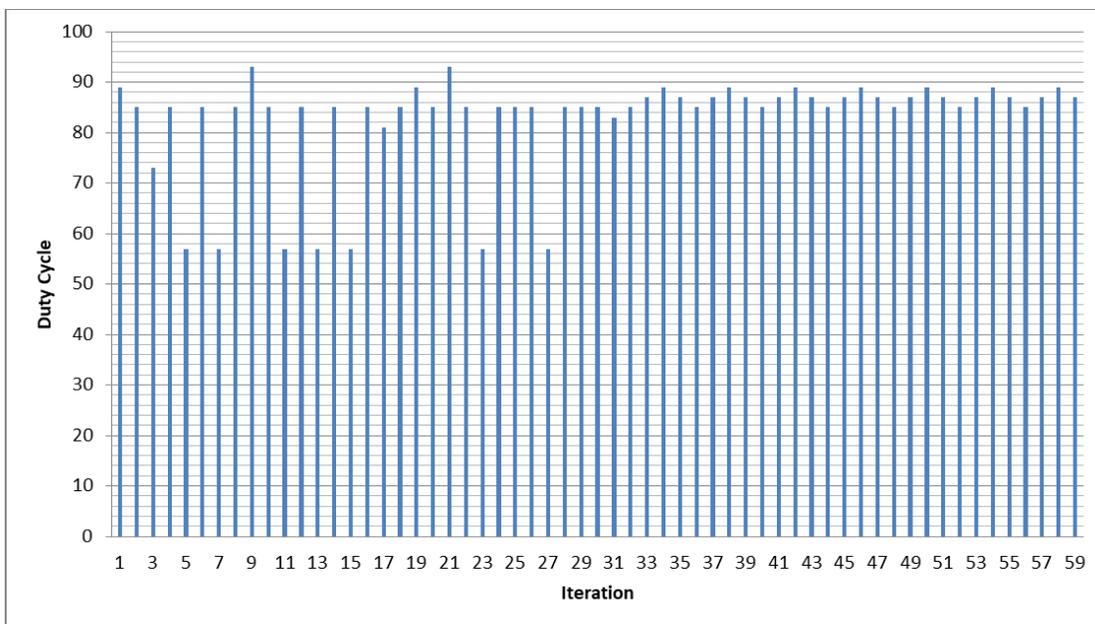
**Figure 100.** The PV array output voltage versus the number of search step.



**Figure 101.** The PV array output current versus the number of search step.



**Figure 102.** The PV array output power versus the number of search step.



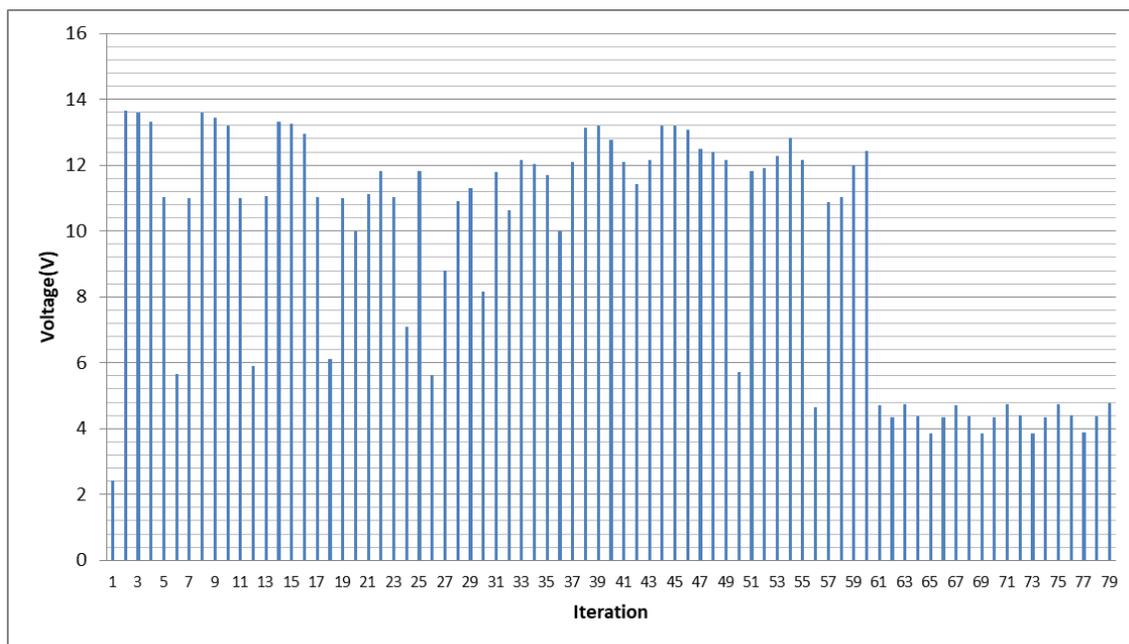
**Figure 103.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. It is considered that the exploration is over, when the oscillations start. The oscillations start at the 33<sup>rd</sup> iteration when the Q-learning algorithm converges to the right MPP

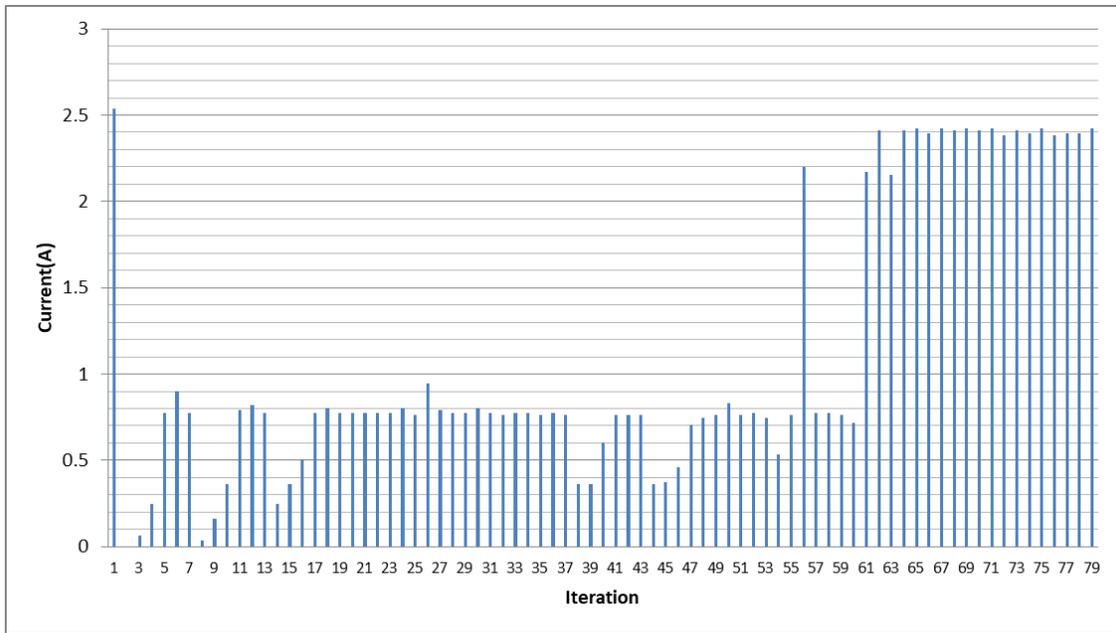
and the P&O algorithm starts to operate for fine-tuning. The oscillations can be seen on all four graphs. The algorithm explored both peaks and converged correctly at the global MPP.

### Results of the PSO GMPPT algorithm:

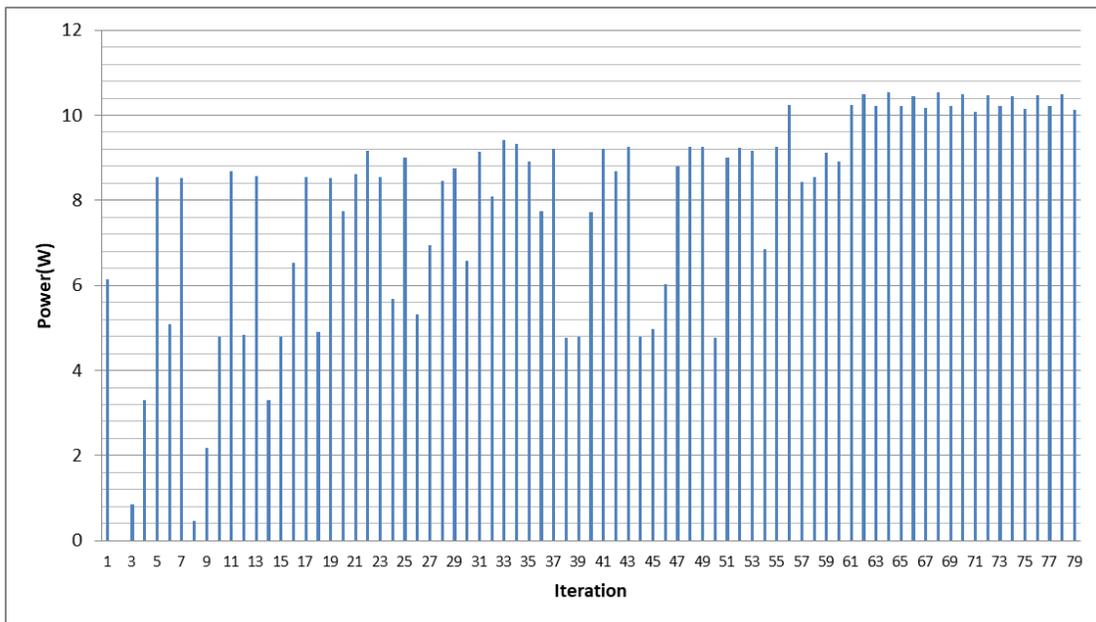
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs.104-108 present the results of the above plots for shading pattern 2.



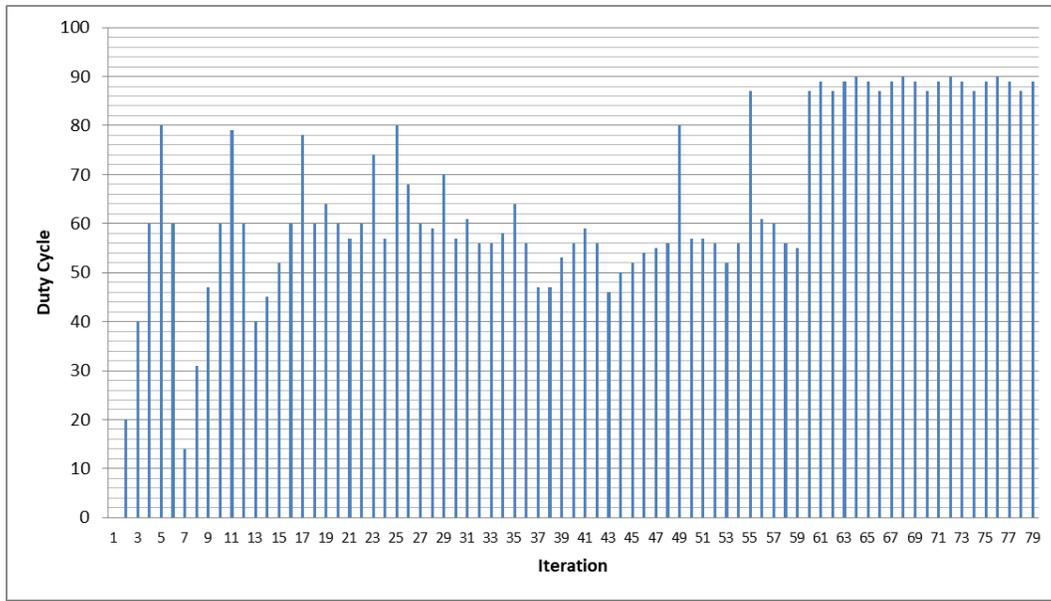
**Figure 104.** The PV array output voltage versus the number of search step.



**Figure 105.** The PV array output current versus the number of search step.

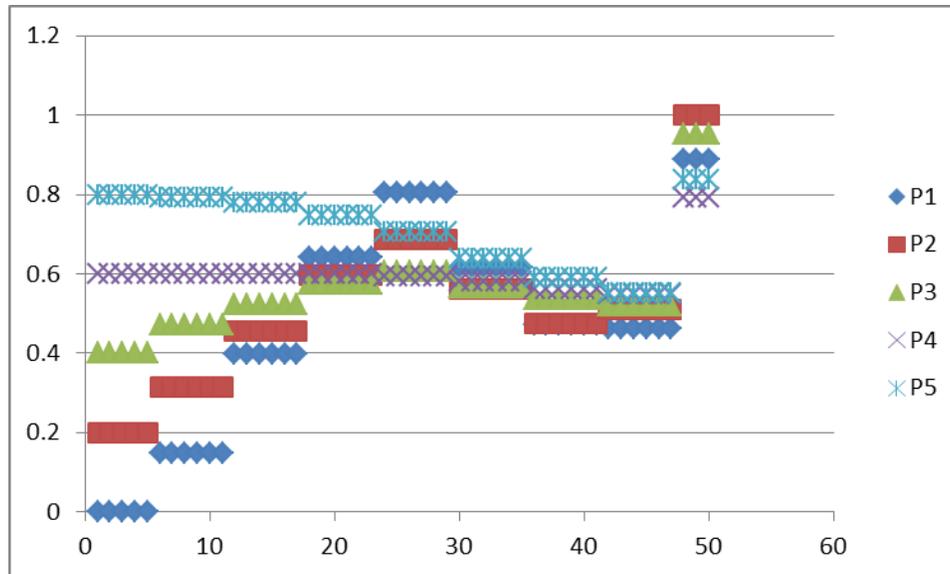


**Figure 106.** The PV array output power versus the number of search step.



**Figure 107.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is when the P&O algorithm starts to operate for fine-tuning. The perturbation step for the P&O algorithm is 2%. The oscillations start at the 61<sup>st</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs.



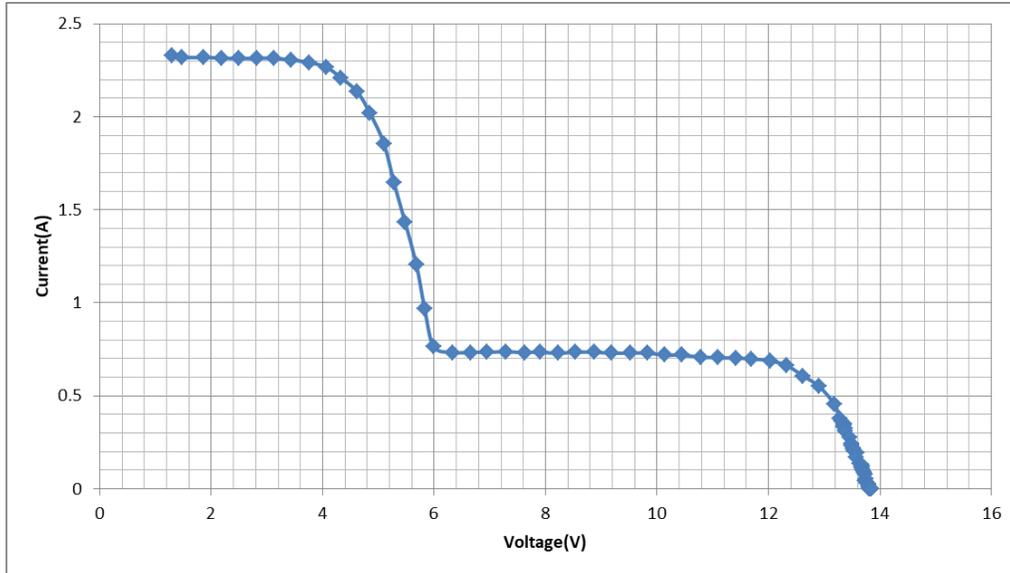
**Figure 108.** The particles position versus the number of search step.

As shown in Figure 108, the particles 1, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

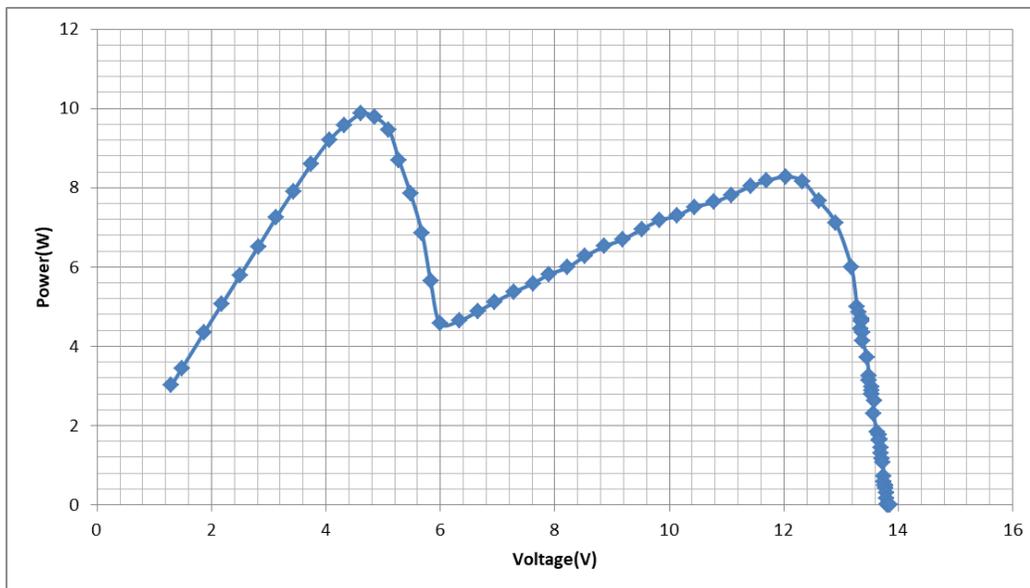
### **Experiment No 2:**

**Solar irradiance for the two photovoltaic modules PV1: 275 W/m<sup>2</sup>, PV2:580 W/m<sup>2</sup> and Ambient temperature: 21°C.**

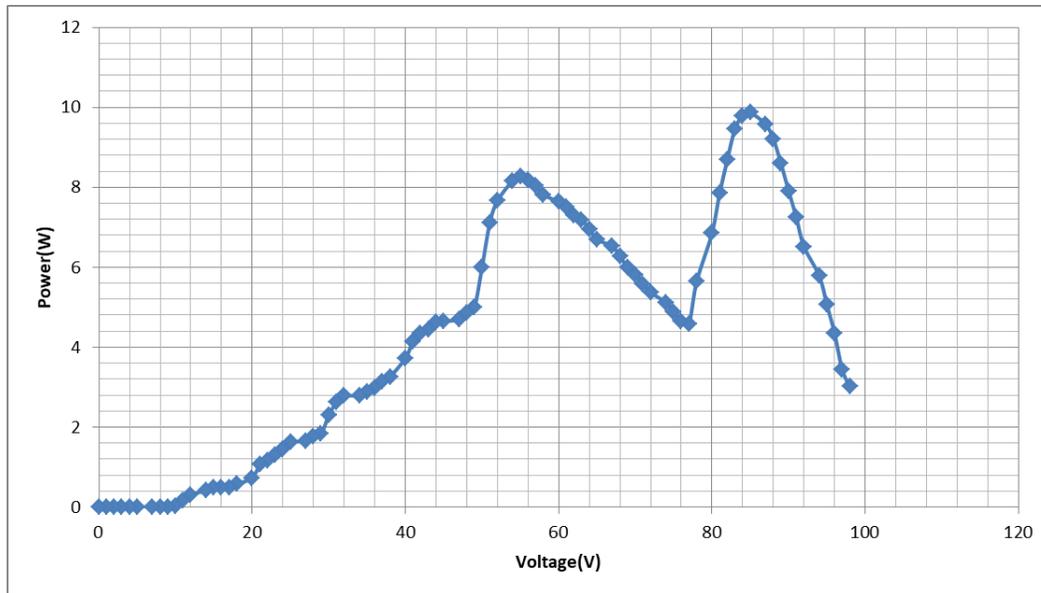
In Fig. 109 the PV array output current-voltage curve is presented and in Fig. 110 the PV array output power-voltage curve is presented. The PV array operates under a non-uniform incident solar irradiance. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 111 which helps to detect in which duty cycle value the MPP is located.



**Figure 109.** The PV array output current-voltage curve.



**Figure 110.** The PV array output power-voltage curve.

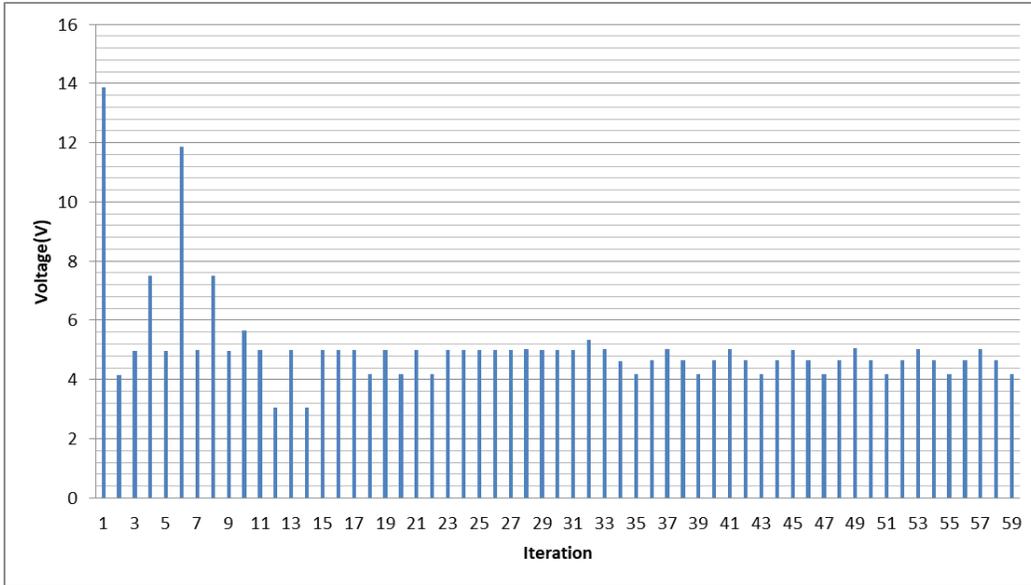


**Figure 111.** The PV array output power-duty cycle curve.

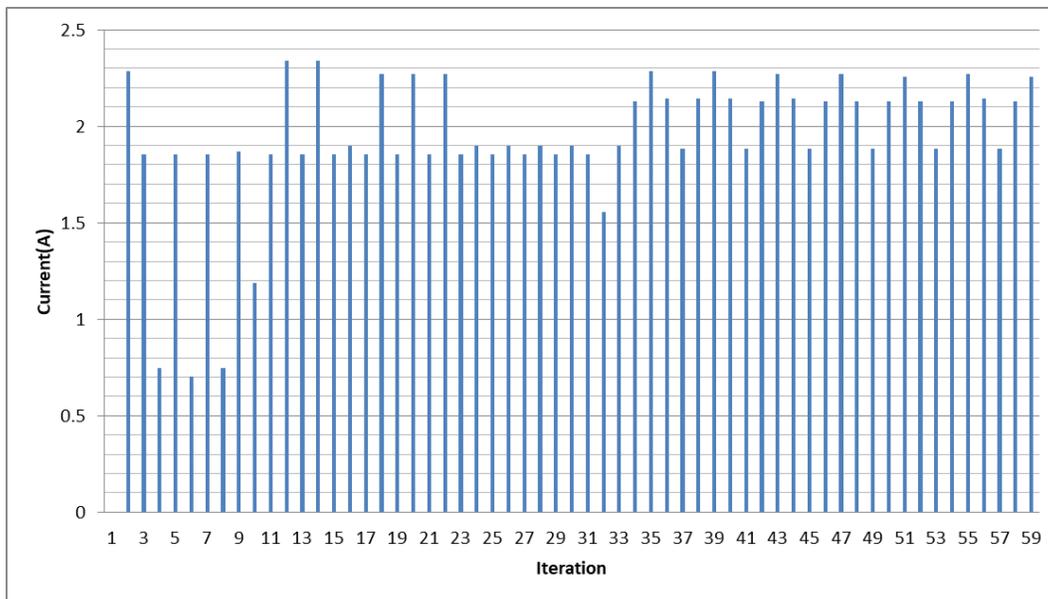
For those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun, the global MPP is located at approximately 85-86% duty cycle but since there is an angle between the two photovoltaic modules, a second peak appears with a power value at approximately 55% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module, as shown in Figure 111. The presence of a second peak can be observed in all three graphs.

### **Results of the Q-learning GMPPT algorithm:**

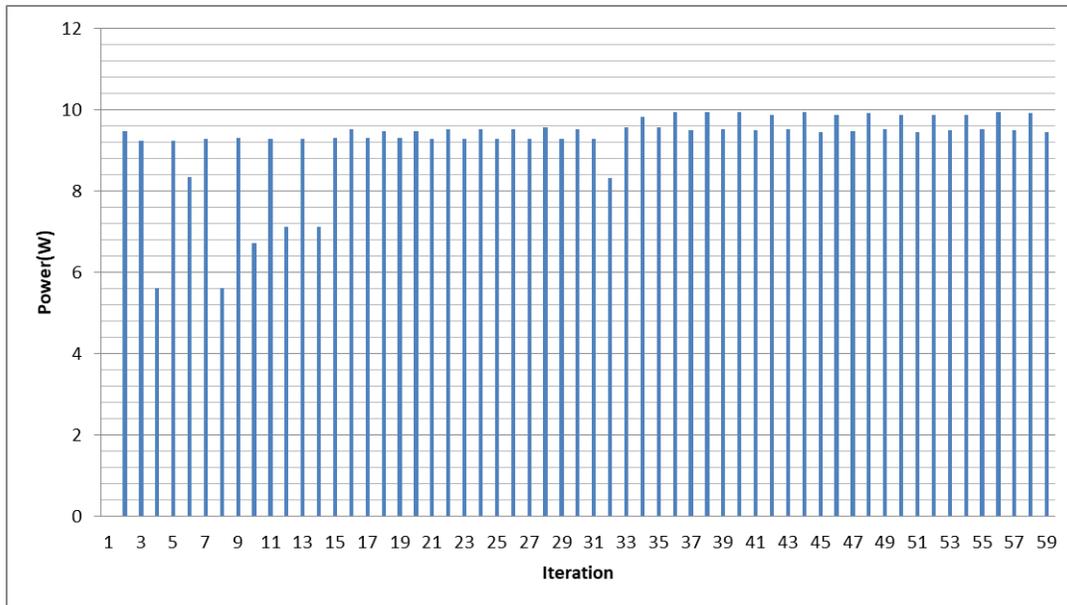
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 112-115 present the results of the above plots for shading pattern 3.



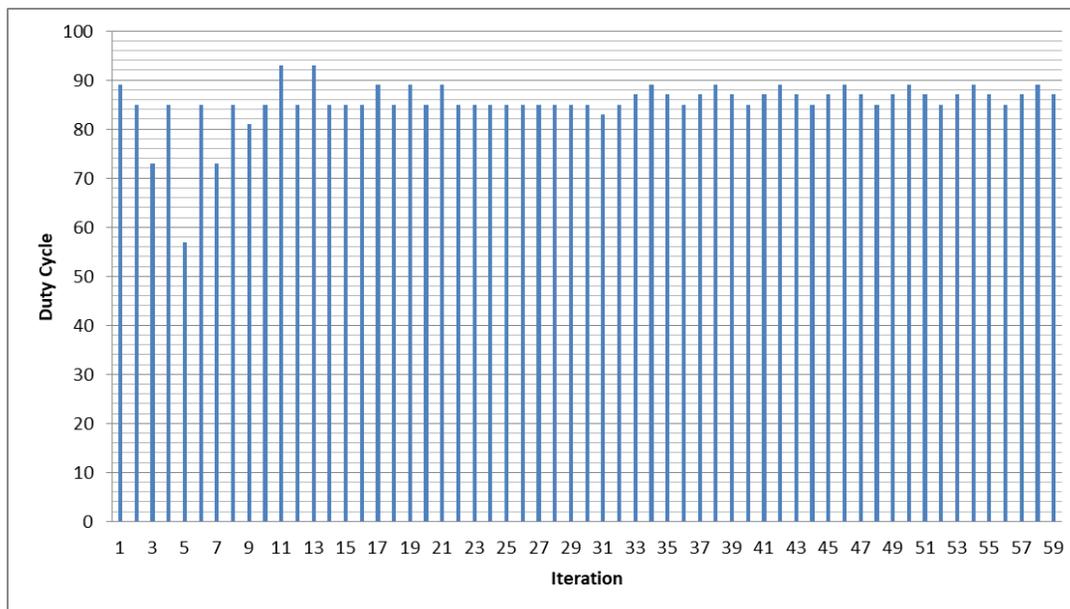
**Figure 112.** The PV array output voltage versus the number of search step.



**Figure 113.** The PV array output current versus the number of search step.



**Figure 114.** The PV array output power versus the number of search step.



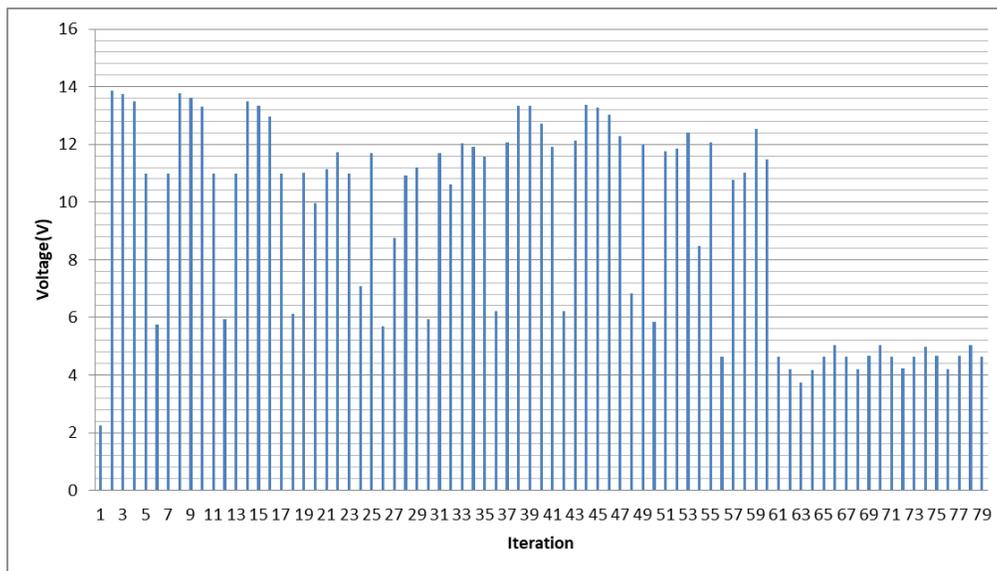
**Figure 115.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. It is considered that the exploration is over, when the oscillations start. The oscillations start at the 30<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP

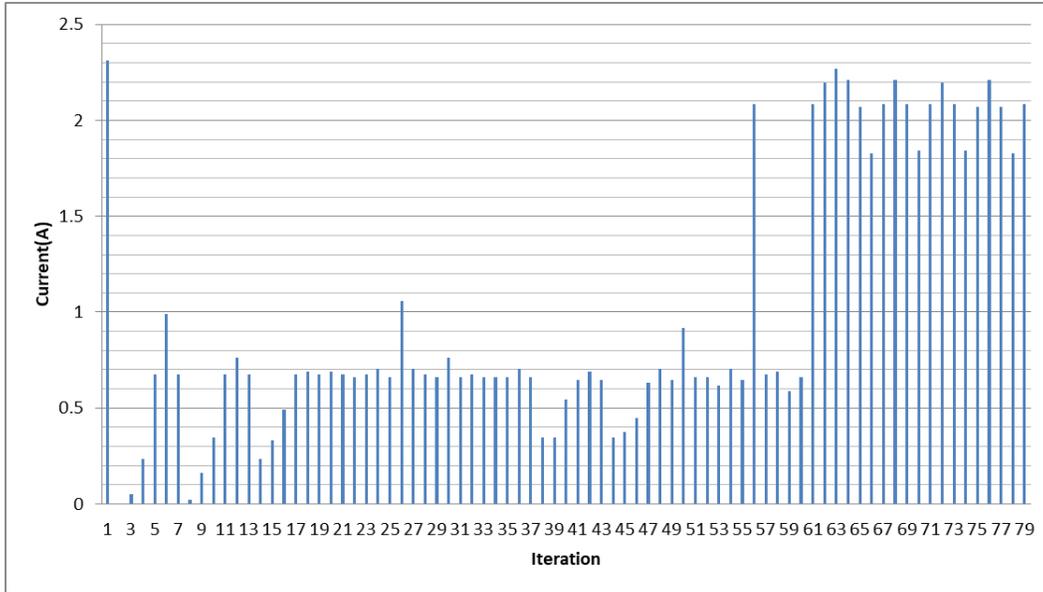
and the P&O algorithm starts to operate for fine-tuning. The oscillations can be seen on all four graphs. The algorithm explored both peaks and converged correctly at the global MPP. It also explored the smaller peak in less iterations. This means that the algorithm is very accurate and does not “waste” time exploring peaks that are not the global MPPs.

### Results of the PSO GMPPT algorithm:

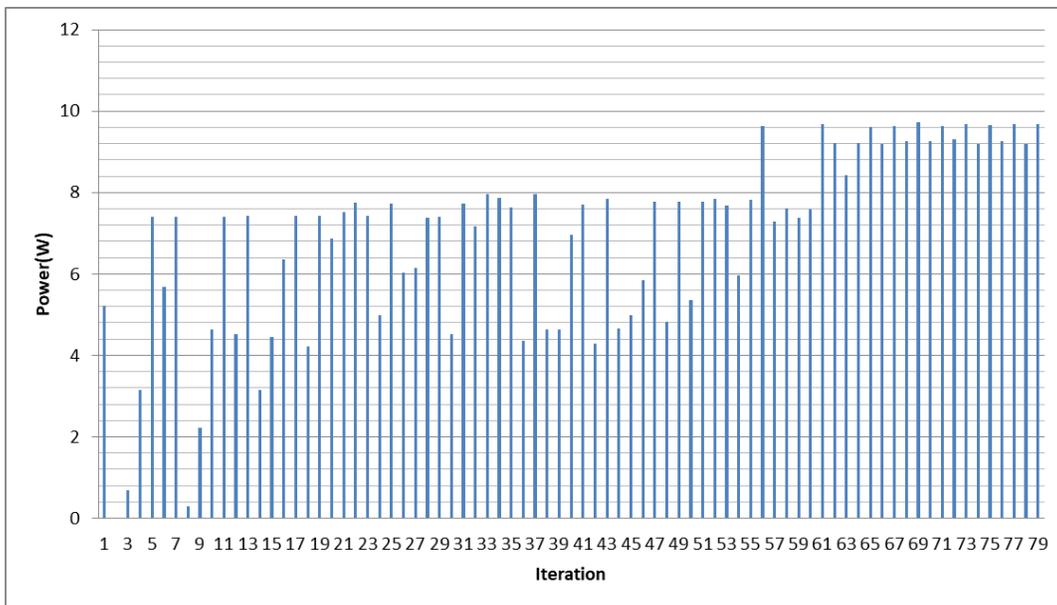
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 116-120 present the results of the above plots for shading pattern 3.



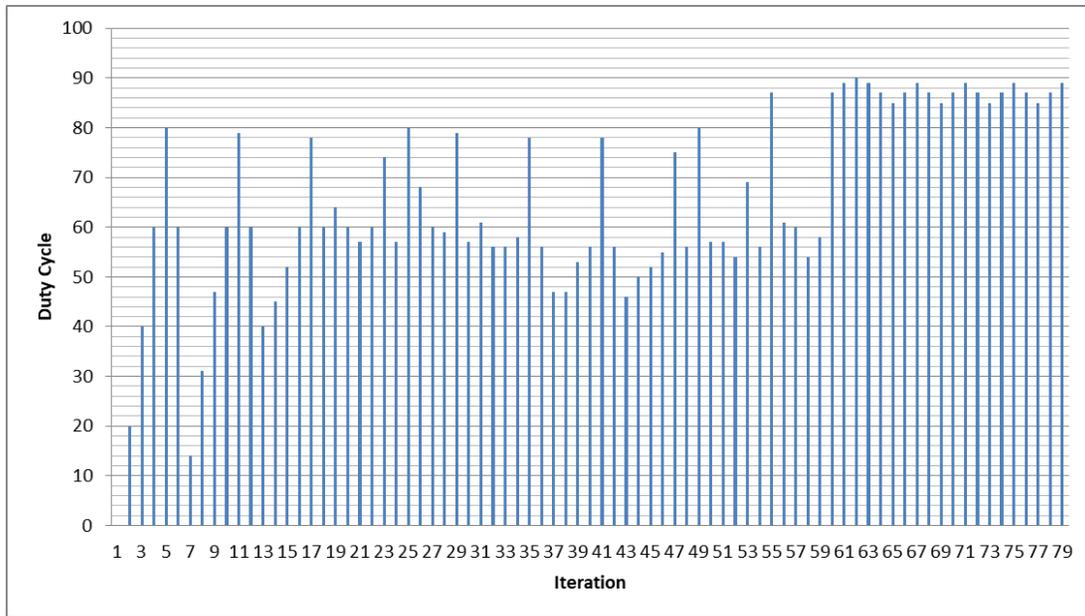
**Figure 116.** The PV array output voltage versus the number of search step.



**Figure 117.** The PV array output current versus the number of search step.

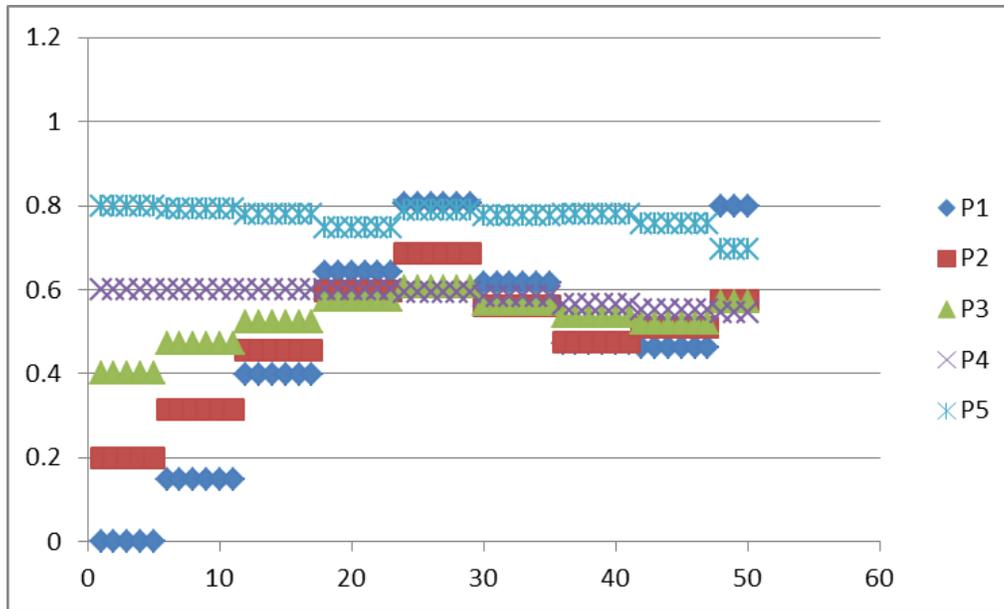


**Figure 118.** The PV array output power versus the number of search step.



**Figure 119.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is when the P&O algorithm starts to operate for fine-tuning. The oscillations start at the 61<sup>st</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs.



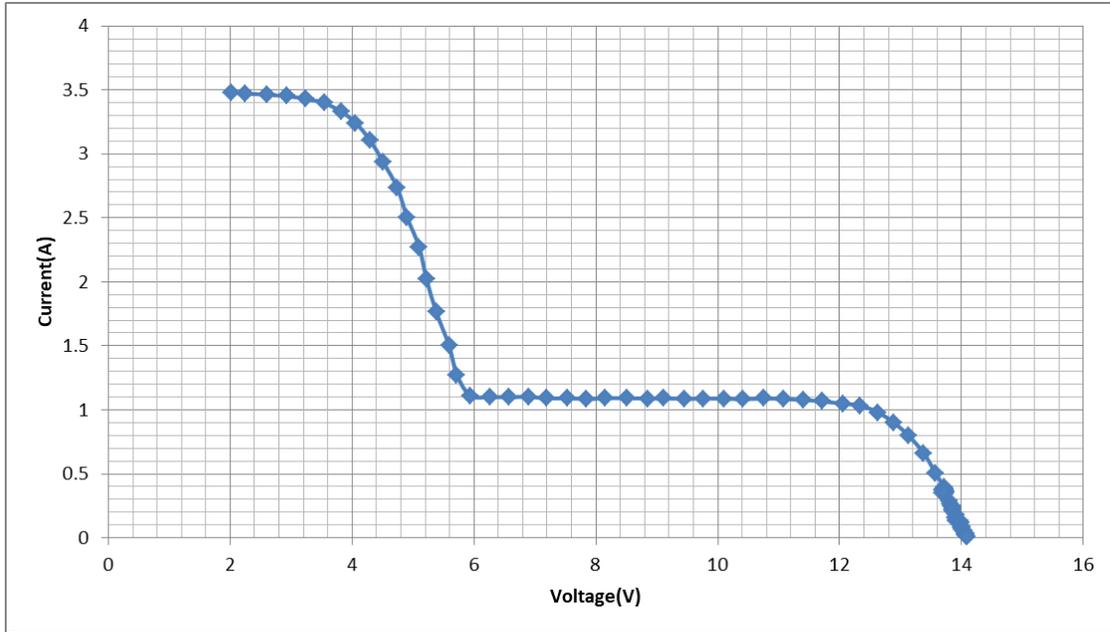
**Figure 120.** The Particle position versus the number of search step.

As shown in Figure 120, only particle 1 converges close to the MPP which is why the P&O algorithm needs a few iterations to fine tune the duty cycle.

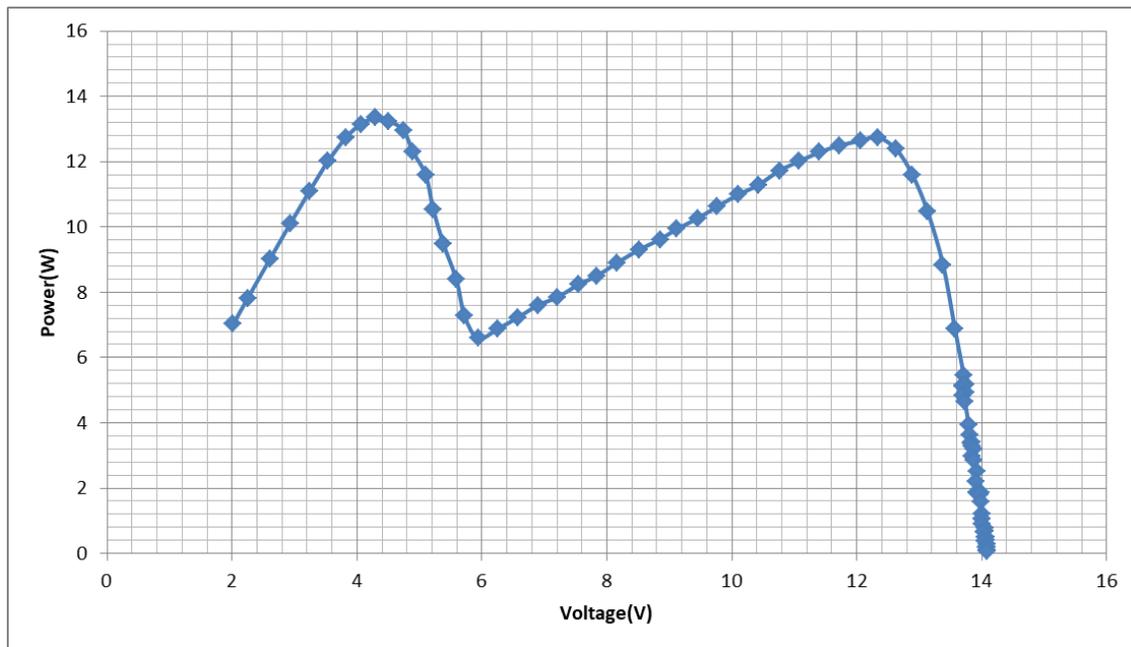
### **Experiment No 3: Failed PSO**

**Solar irradiance for the two photovoltaic modules PV1: 865 W/m<sup>2</sup>, PV2:370 W/m<sup>2</sup> and Ambient temperature: 23°C,**

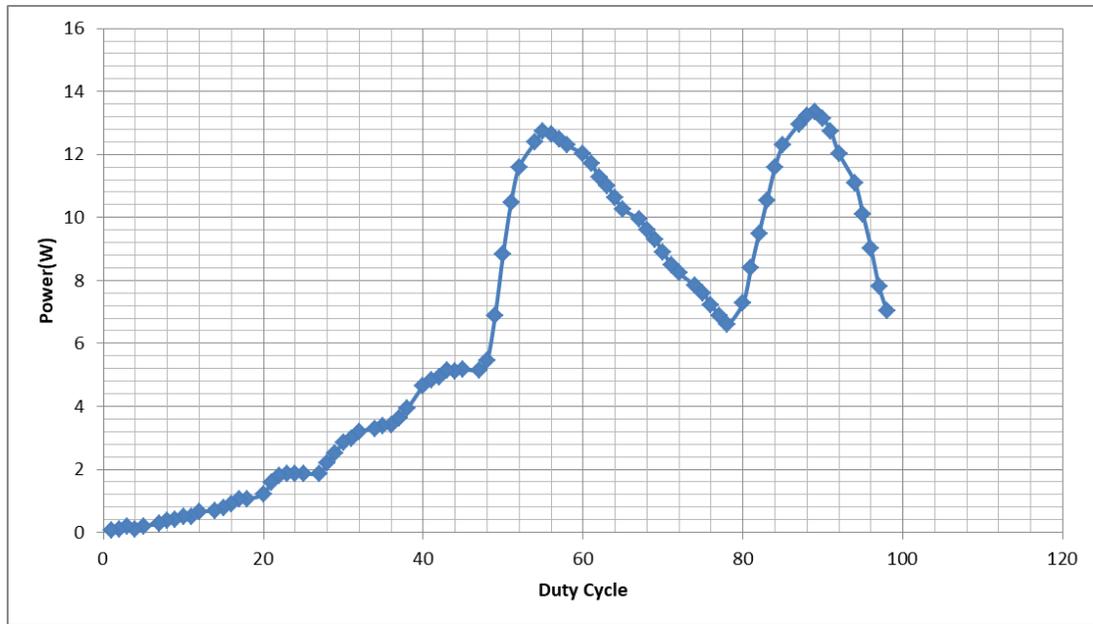
In Fig.121 the PV array output current-voltage curve is presented and in Fig. 122 the PV array output current-voltage curve is presented. The PV array operates under a non-uniform incident solar irradiance pattern. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 123 which helps to detect in which duty cycle value the MPP is located.



**Figure 121.** The PV array output current-voltage curve.



**Figure 122.** The PV array output power-voltage curve.

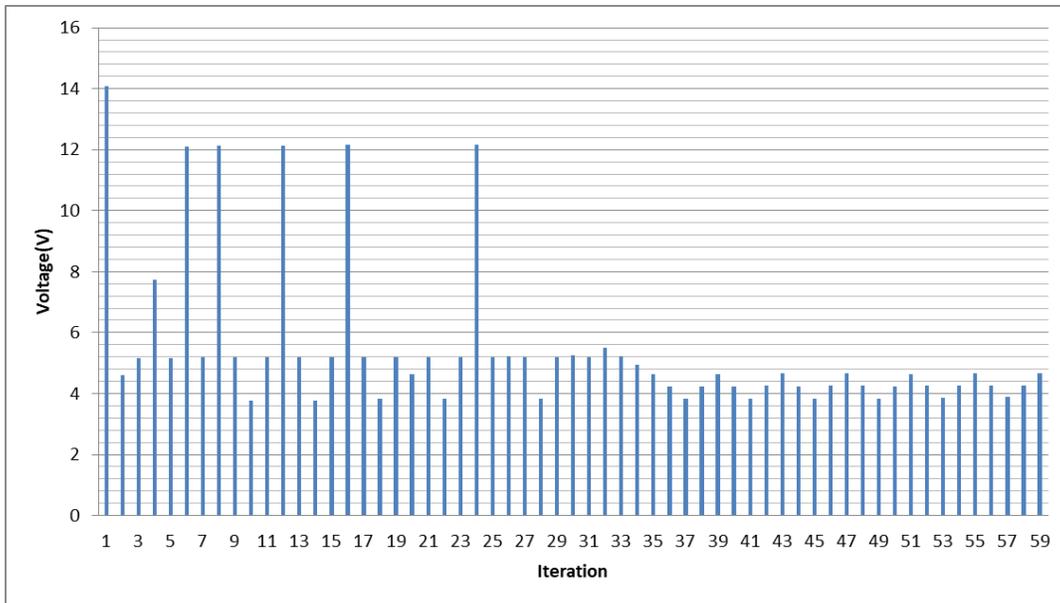


**Figure 123.** The PV array output power-duty cycle curve.

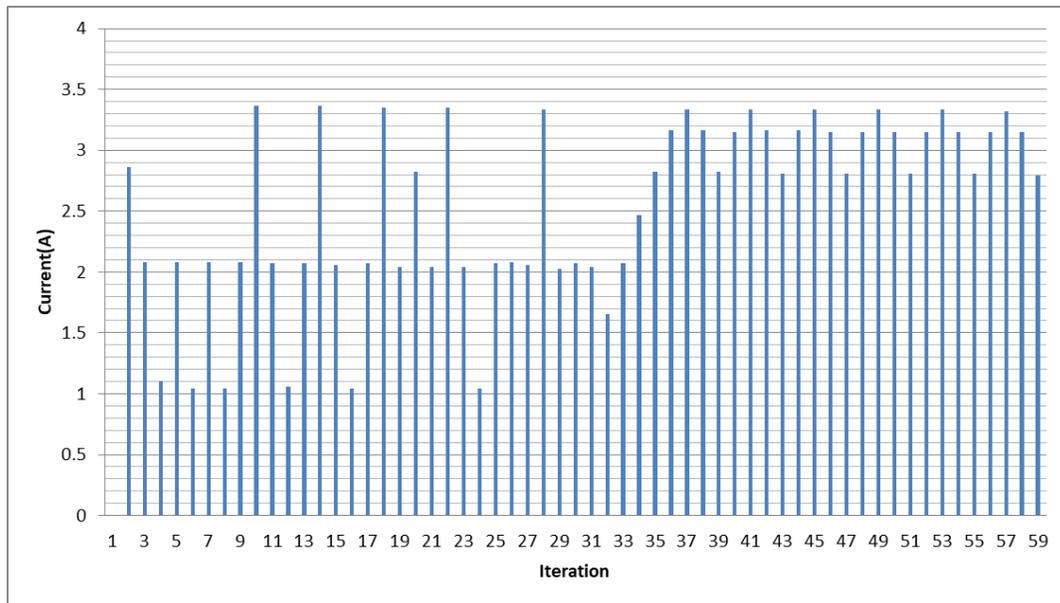
We can see that for those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun, the global MPP is located at approximately 88-89% duty cycle but since there is an angle between the two photovoltaic modules, a second peak appears with a power value at approximately 54% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module, as shown in Figure 123. The presence of a second peak can be observed in all three graphs.

#### **Results of the Q-learning GMPPT algorithm:**

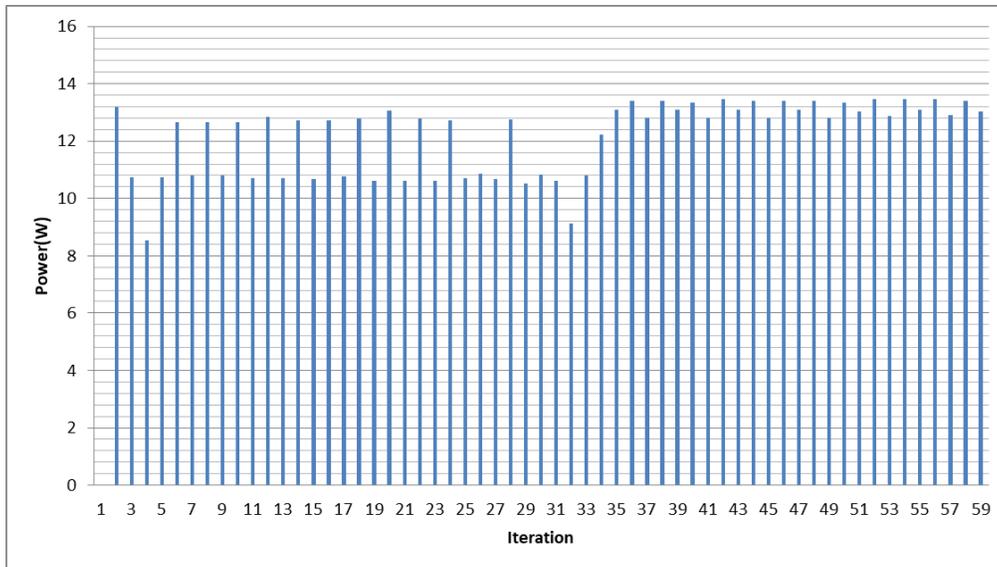
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 124-127 present the results of the above plots for shading pattern 3.



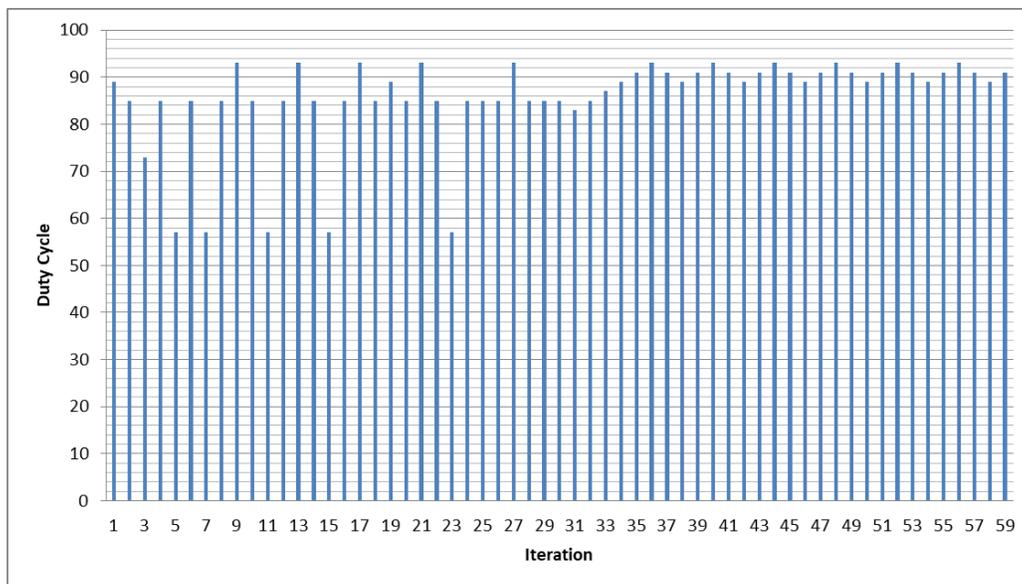
**Figure 124.** The PV array output voltage versus the number of search step.



**Figure 125.** The PV array output current versus the number of search step.



**Figure 126.** The PV array output power versus the number of search step..



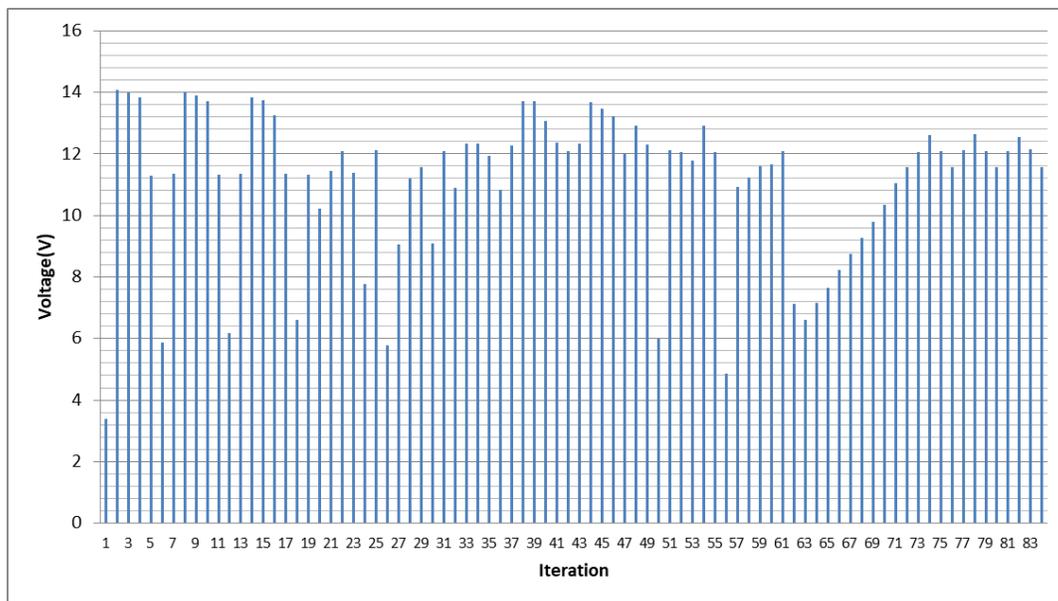
**Figure 127.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge. It is considered that the exploration is over, when the oscillations start. The oscillations start at the 34<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm starts to operate for fine-tuning. Oscillations are observed on all four

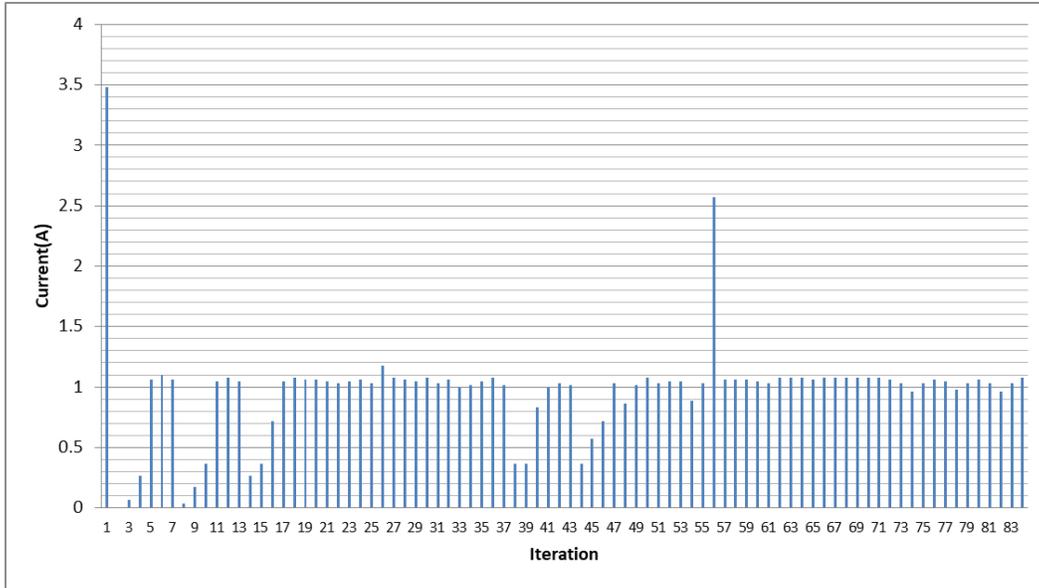
graphs. The algorithm explored both peaks and converged correctly at the global MPP. It also explored the lower power peak in less iterations. This means that the algorithm is very accurate and does not “waste” time exploring peaks that are not the global MPP.

### Results of the PSO GMPPT algorithm:

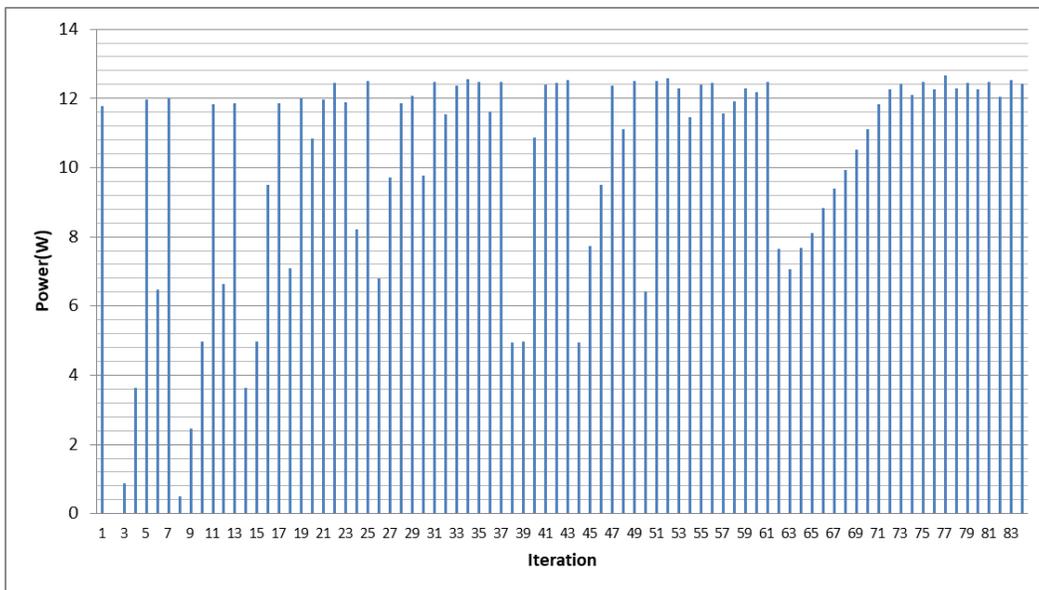
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 128-132 present the results of the above plots for shading pattern 3.



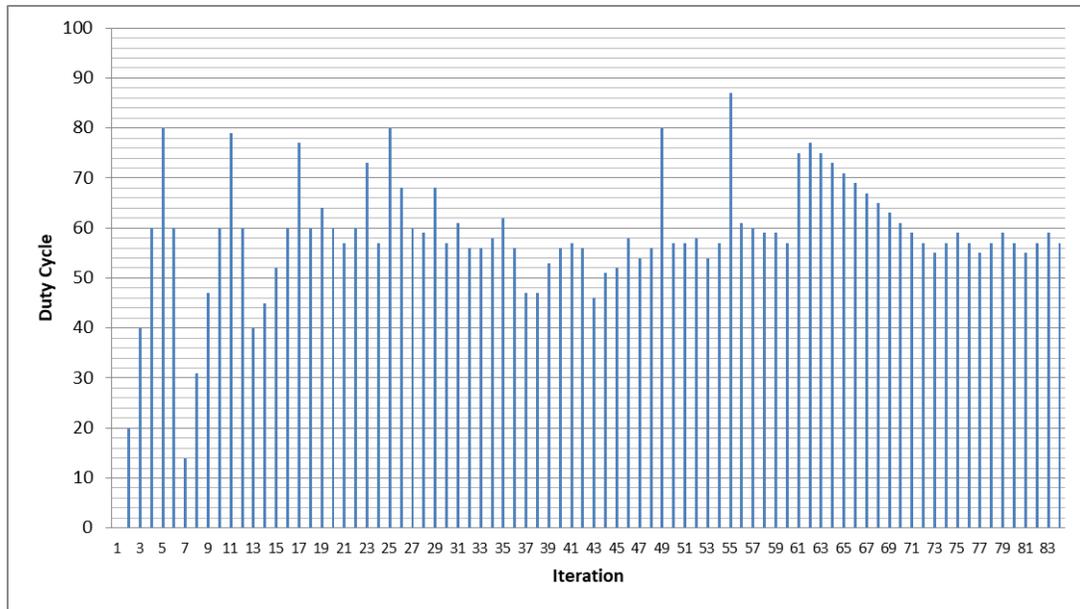
**Figure 128.** The PV array output voltage versus the number of search step.



**Figure 129.** The PV array output current versus the number of search step.

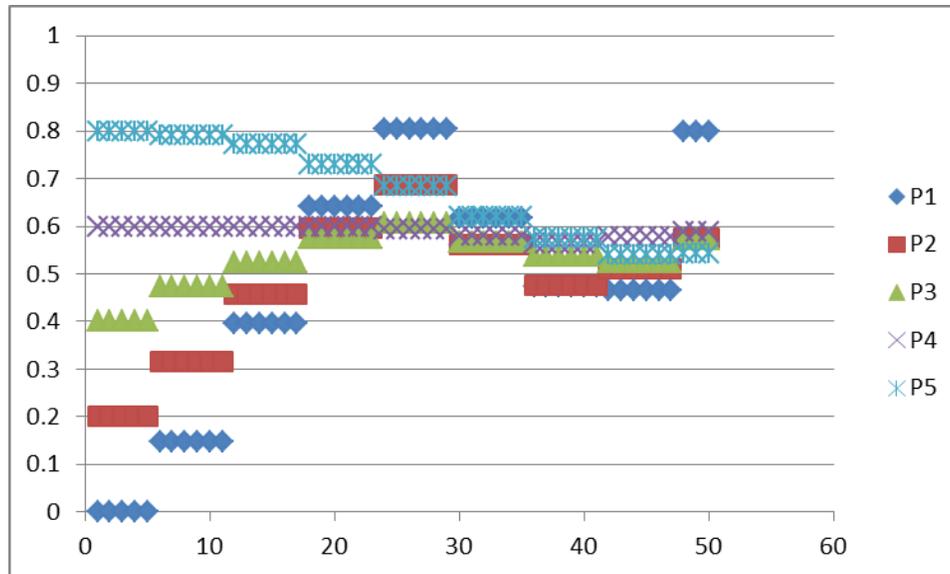


**Figure 130.** The PV array output power versus the number of search step.



**Figure 131.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm and it also converges at the wrong peak. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is when the P&O algorithm takes over for fine-tuning. The oscillations start at the 73<sup>rd</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs. Amongst the measurements that had two peaks with very close power values but different duty cycles the PSO algorithm failed to converge correctly in two measurements and one of them is the above. The reason why the PSO algorithm did not converge correctly is the same as why it was not very accurate in the second shading pattern and has to do with its search-space exploration capability. If the velocity that updates the particle position is not appropriate, then the PSO algorithm might never explore the global MPP but areas around it and decide that those areas have lower power than the second (i.e. local) MPP, since the two peaks have very close power values.



**Figure 132.** The particles position versus the number of search step.

As shown in Figure 132, none of the particles converge close to the global MPP which is why the PSO algorithm fails to converge correctly and instead converges at the second (i.e. local) MPP.

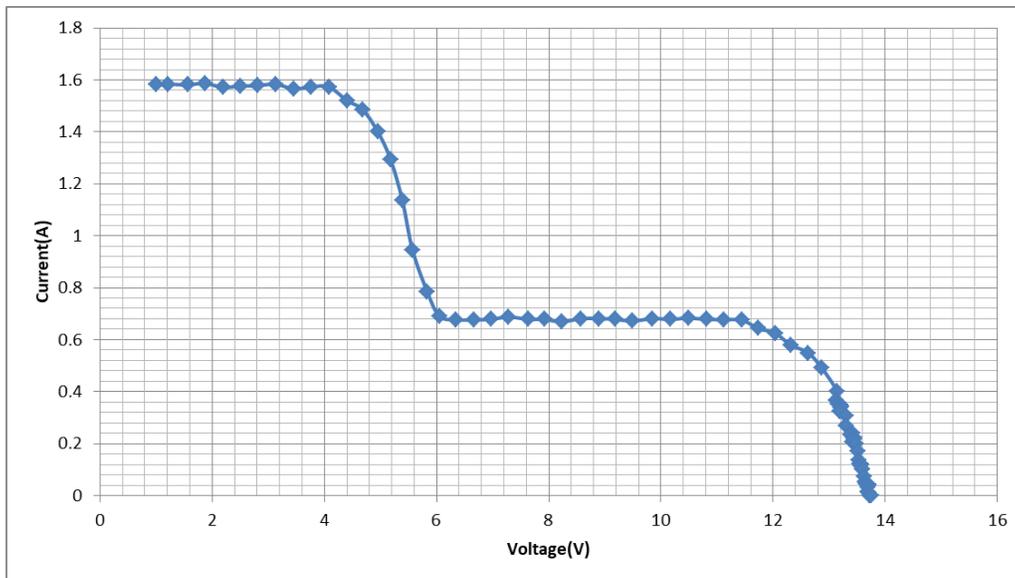
## 5.5 Analysis of the results for shading pattern 4

In this shading pattern, it will be observed how a 20-45 degree angle between the photovoltaic modules can affect the I-V and P-V curves of the photovoltaic modules, the speed and accuracy of the Q-learning as well as the PSO algorithm. The difference between this pattern and pattern 2 is that for this set of measurements the angle at which the solar irradiation was incident on the photovoltaic modules was much lower than that at shading pattern 2, meaning that the photovoltaic modules did not face the sun directly but were installed at a high tilt angle.

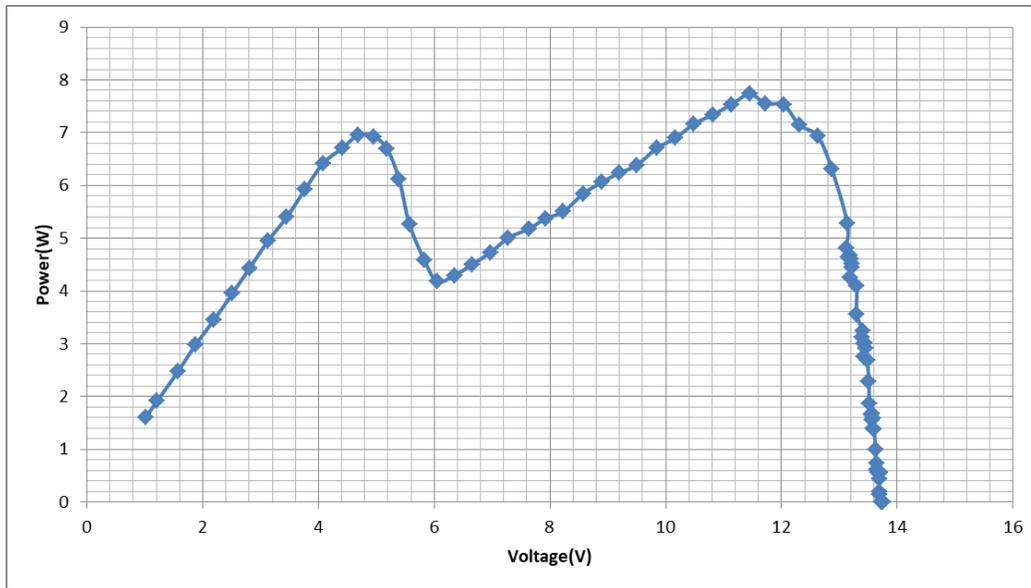
### Experiment No 1:

Solar irradiance for the two photovoltaic module  $sPV1$ : 260  $W/m^2$ ,  $PV2$ :450  $W/m^2$  and Ambient temperature: 24°C.

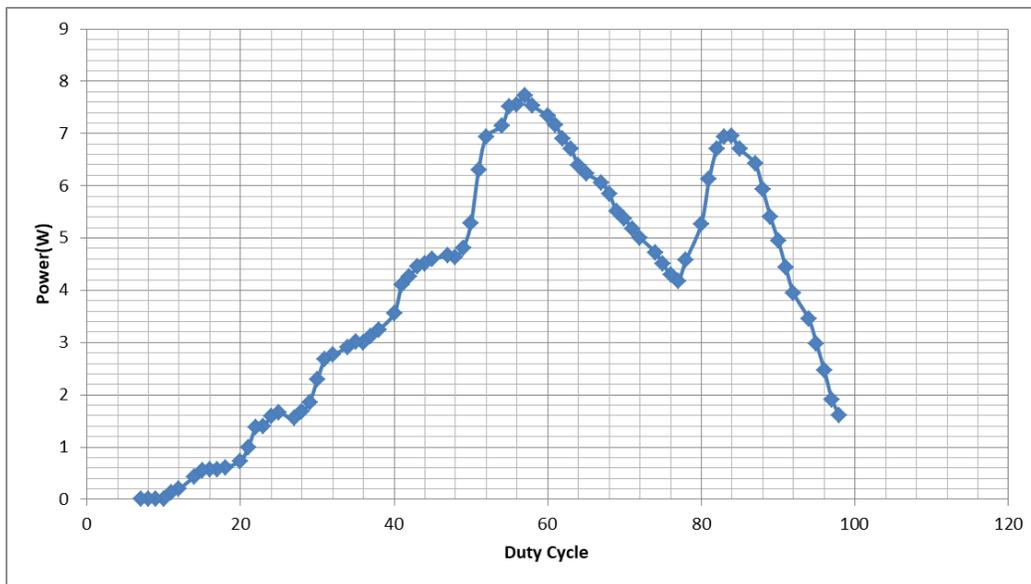
In Fig. 133 the PV array output current-voltage curve is presented and in Fig. 134 the PV array output power-voltage curve is presented. The PV array operates under a non-uniform incident solar irradiance pattern. Before presenting the results of the Q-learning and PSO algorithms, the power-duty cycle curve is depicted in Fig. 135 which helps to detect in which duty cycle value the MPP is located.



**Figure 133.** The PV array output current-voltage curve.



**Figure 134.** The PV array output power-voltage curve.



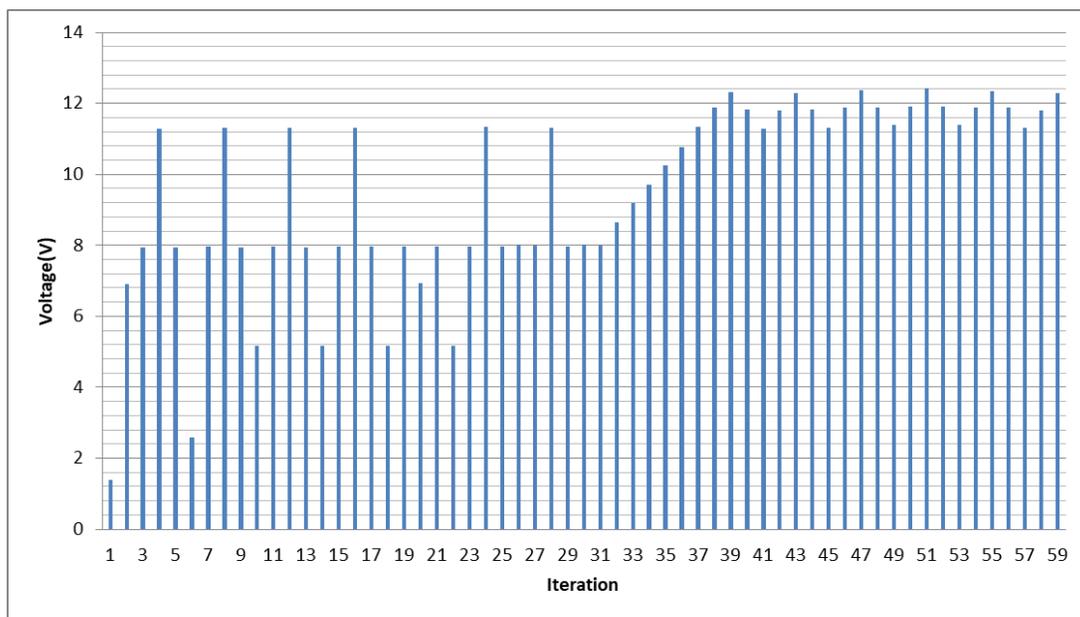
**Figure 135.** The PV array output power-duty cycle curve.

For those specific irradiance values, angle between the photovoltaic modules and angle between the photovoltaic modules and the sun, the global MPP is located at approximately 57-58% duty cycle but since there is an angle between the two photovoltaic modules, a second peak appears

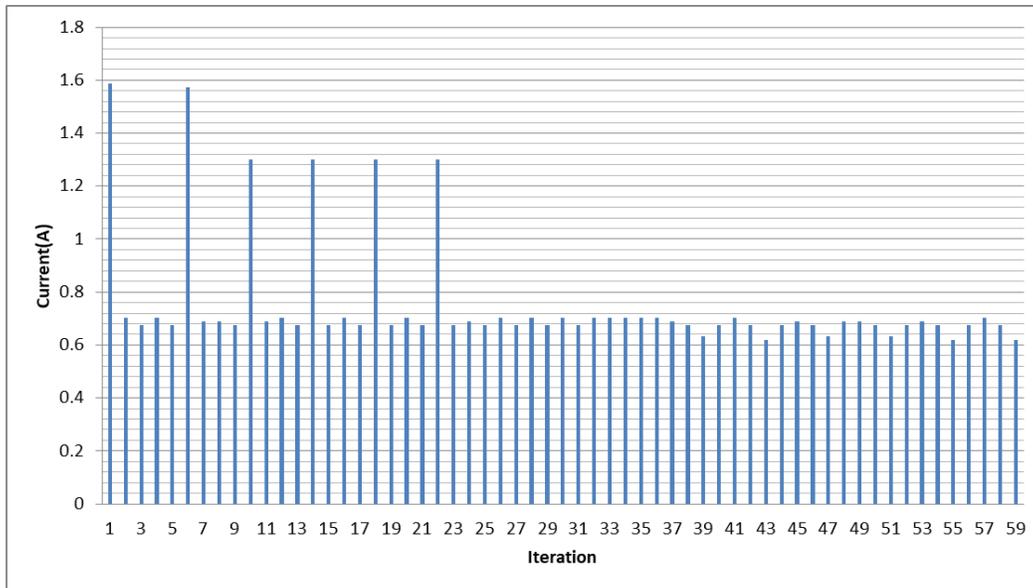
with a power value at approximately 83-84% duty cycle which is a local MPP and that is due to the different irradiance on each photovoltaic module, as shown in Figure 135. The presence of a second peak can be observed in all three graphs.

### Results of the Q-learning GMPPT algorithm:

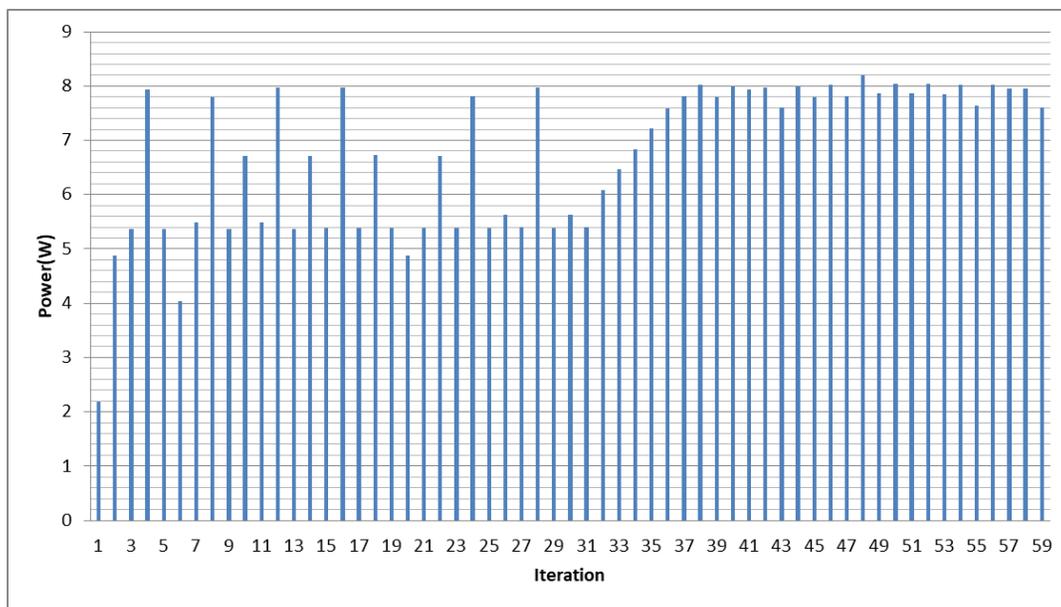
Four plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot and the duty cycle versus the number of steps until convergence plot. Figs. 136-139 present the results of the above plots for shading pattern 1.



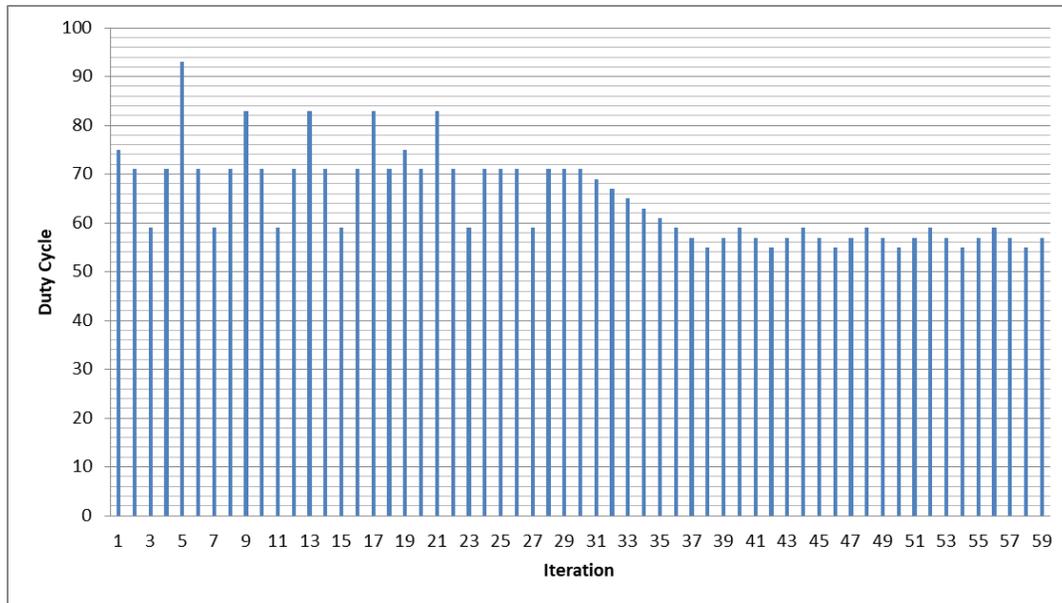
**Figure 136.** The PV array output voltage versus the number of search step.



**Figure 137.** The PV array output current versus the number of search step.



**Figure 138.** The PV array output power versus the number of search step.

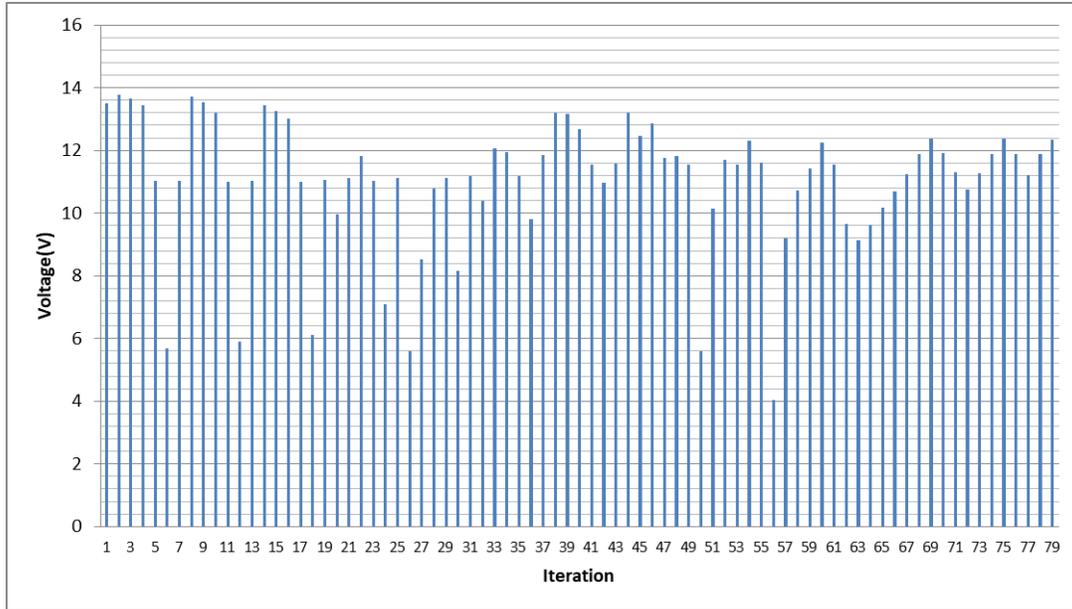


**Figure 139.** The PV array output duty cycle versus the number of search step.

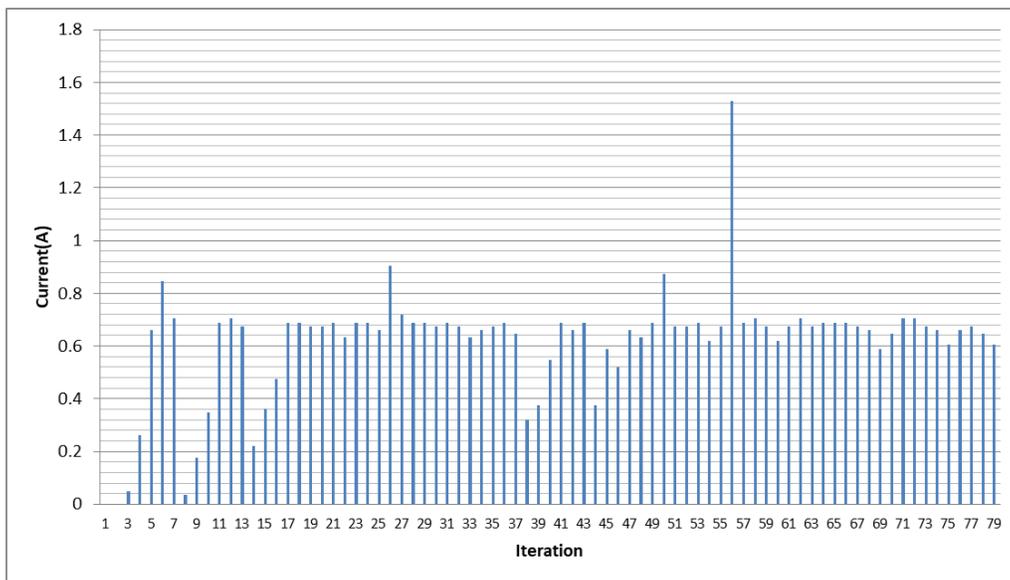
It is observed that the algorithm does not need a relatively long amount of time in order to converge. It is considered that the exploration is over, when the oscillations start. The oscillations start at the 39<sup>th</sup> iteration when the Q-learning algorithm converges to the right MPP and the P&O algorithm takes over for fine-tuning. The oscillations can be seen on all four graphs. The algorithm explored both peaks and converged correctly at the global MPP.

#### **Results of the PSO GMPPT algorithm:**

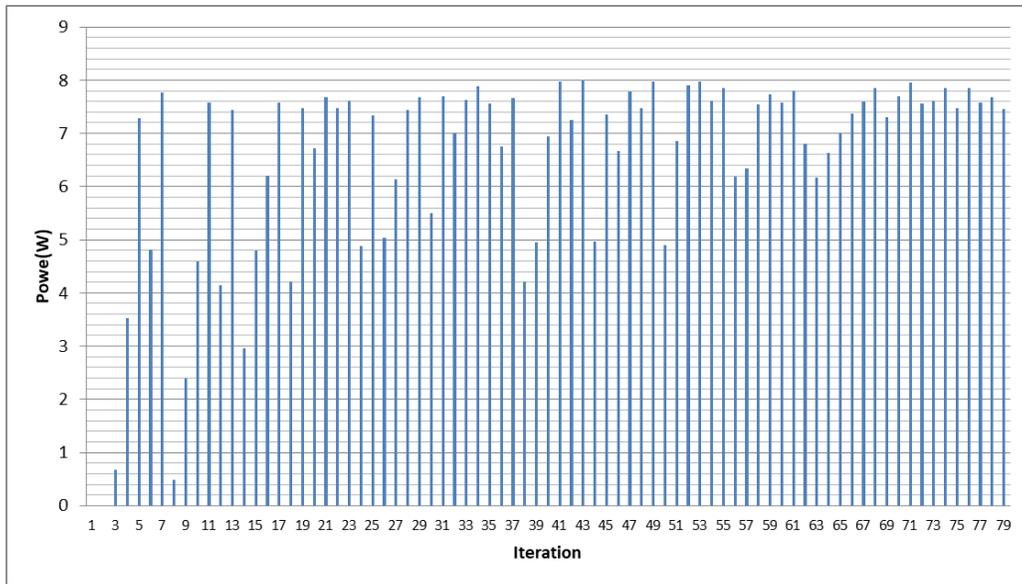
Five plots were exported, the PV array output voltage versus the number of steps until convergence plot, the PV array output current versus the number of steps until convergence plot, the PV array output power versus the number of steps until convergence plot, the duty cycle versus the number of steps until convergence plot and finally all particles positions versus the number of steps until convergence plot. Figs. 140-144 present the results of the above plots for shading pattern 2.



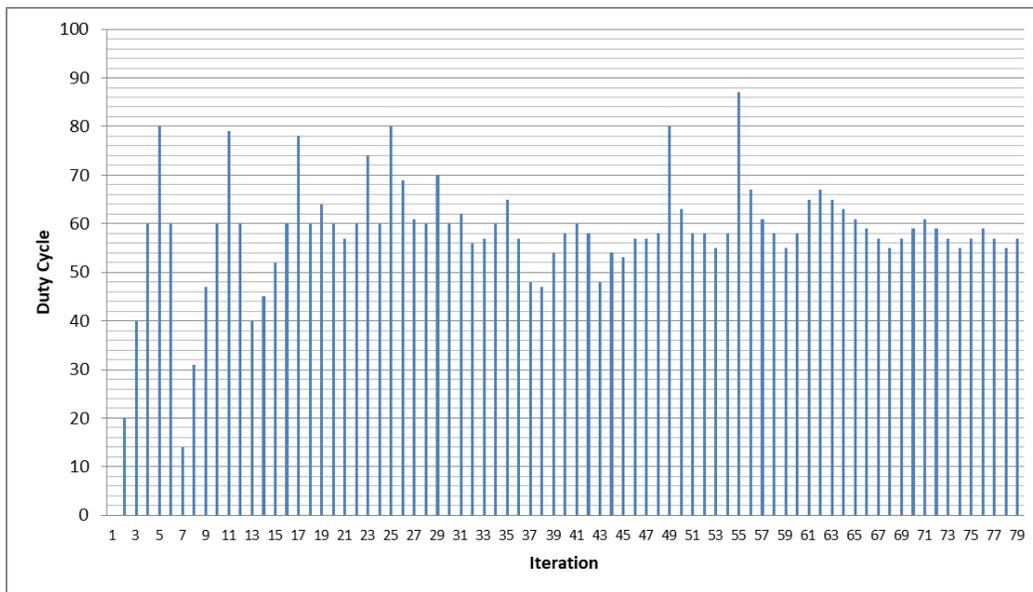
**Figure 140.** The PV array output voltage versus the number of search step.



**Figure 141.** The PV array output current versus the number of search step.



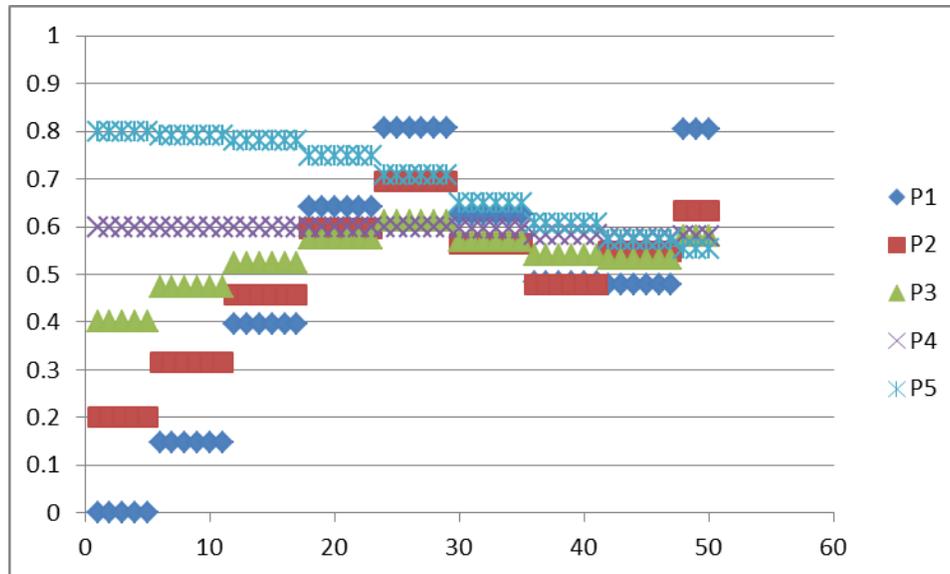
**Figure 142.** The PV array output power versus the number of search step.



**Figure 143.** The PV array output duty cycle versus the number of search step.

It is observed that the algorithm does not need a relatively long amount of time in order to converge however it takes longer than the Q-learning algorithm. It is considered that the exploration is over when the number of iterations reach 50 (5 particles \* 10 generations) which is

when the P&O algorithm starts to operate for fine-tuning. The oscillations start at the 75<sup>th</sup> iteration when the P&O algorithm converges. The oscillations can be seen on all four graphs.



**Figure 144.** The particles position versus the number of search step.

As shown in Figure 144, the particles 3, 4 and 5 converge close to the MPP which is why the P&O algorithm does not need many iterations to fine tune the duty cycle.

This shading pattern is very similar to pattern number 2. The only difference is that for this pattern the power value of the second (i.e. local) MPP is very close to the value of the global MPP instead of being much lower.

## 5.5 Summarized results

A comparative summary of the experimental results is presented in Table 5. In the first column the number of the experiment can be seen. In the second column the duty cycle at which the GMPP is located is presented. In the third and fifth columns, the duty cycle at which each algorithm converged to is presented. In the fourth and sixth columns, the number of search steps required by each algorithm for convergence to the global MPP can be seen. The number of search steps required by the P&O algorithm which is executed next, are also included in these results. Lastly in the final column, it is presented how faster the Q-learning algorithm is compared to the PSO algorithm for convergence to the global MPP, also taking into account the number of search steps required by the P&O algorithm which is executed next. It is observed that the Q-learning GMPPT algorithm is at its worst 99% accurate (before the execution of the P&O algorithm). Additionally, in combination with the P&O algorithm which was executed next, it was capable to derive the GMPP significantly faster than the PSO algorithm (also combined with the P&O algorithm).

**Table 5.** Comparative summary of the experimental results of the two GMPPT algorithms.

<b>Exp . No</b>	<b>Duty cycle at the GMPP</b>	<b>Duty cycle after convergence of the Q-learning algorithm</b>	<b>Q-learning number of search steps</b>	<b>Duty cycle after convergence of the PSO algorithm</b>	<b>PSO number of search steps</b>	<b>Q-learning speed-up %</b>
1	62-63%	63%	35	64%	59	40.6%
2	67-68%	67%	32	68%	69	53.6%
3	62-63%	64%	32	64%	68	52.9%

4	58-59%	60%	37	58%	68	45.5%
5	58-59%	57%	40	58%	68	41.1%
6	59%	60%	36	61%	68	47%
7	89-90%	88%	33	89%	61	45.9%
8	85-86%	86%	30	87%	61	50.8%
9	88-89%	88%	34	56%	73	53.4%
10	57-58%	57%	39	57%	75	48%

## 6. CONCLUSIONS

---

For maximizing the power produced by photovoltaic arrays under uniform and non-uniform solar irradiation conditions, the tracking of the position of the GMPP is definitely one major factor. The process of detecting the duty cycle that will allow the photovoltaic system to operate at this point is being called MPP tracking process. This process, however, can sometimes be ineffective. Due to the use of the bypass diodes, this may occur during the partial shading of the PV panels, where several peaks appear on the power-voltage curve of the PV array.

The design of a PV MPPT system was carried out in this thesis, which is based on reinforcement learning. For that reason, an MPP tracking method based on Q-learning was implemented. This algorithm was developed in the Arduino IDE. For the implementation of the PV system, a DC/DC Boost converter, a photovoltaic array, an MPP tracking control unit and a load (batteries), were used.

The photovoltaic system was tested for multiple shading patterns of the photovoltaic array. Uniform and non-uniform solar irradiation patterns were both included in these shading patterns. The Q-learning-based MPP tracking method was tested in 4 different cases. In the first case, the PV array was subject to uniform incident solar irradiance and only one peak occurred on the power-voltage curve, that peak being of course the global MPP. In the second case, the PV array was arranged in a way that a slight angle existed between the two photovoltaic modules, resulting in a power-voltage curve that had two peaks. One of them being the global MPP and the second being a lower, local MPP. In the third case, the PV array was arranged in a way that a high angle existed between the two photovoltaic modules, resulting in a power-voltage curve that had two similar peaks. One of them being the global MPP and the second being a slightly lower,

local MPP. The difference between the third and fourth shading patterns was on the position of the global MPP with respect to the local MPP. In the third case, the global MPP was the left peak on the power-voltage curve however in the fourth case, the global MPP was the right peak on the power-voltage curve.

A PSO-based MPPT approach was also implemented and applied for the same 4 shading patterns of the photovoltaic array for comparison purposes. According to the results of the PV system experimental testing under uniform and non-uniform conditions of solar irradiation, the Q-learning MPPT algorithm was always capable to detect accurately the global MPP and also converged faster to the global MPP. Regarding the PSO MPPT algorithm, it almost always detected the global MPP accurately. This is due to the fact that it is impossible for the particles to explore every duty cycle value. However, its major disadvantage is that it cannot reduce its convergence time no matter the shading pattern. The experimental results indicated that the Q-learning MPPT algorithm did not have a fixed convergence time, which was depending on the shading pattern applied.

## 7. REFERENCES

---

- [1] K. Lappalainen, S. Valkealahti, "Effects of PV array layout, electrical configuration and geographic orientation on mismatch losses caused by moving clouds", *Solar Energy*, 144 (2017), pp. 548-555.
- [2] O. Wasynczuk, "Dynamic behaviour of a class of photovoltaic power systems", *IEEE Transactions on Power Apparatus Systems*, 102 (1983), pp. 3031-3037.
- [3] P. Bhatnagar, R. K. Nema, "Maximum power point tracking control techniques: state-of-the-art in photovoltaic applications," *Renewable and Sustainable Energy Reviews*, 23 (2013), pp. 224–241.
- [4] T. Esum, P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques", *IEEE Transactions on Energy Conversion*, 22 (2007), pp. 439-449.
- [5] S. Surawdhaniwar, R. Diwan, "Study of Maximum Power Point Tracking Using Perturb and Observe Method", *International Journal of Advanced Research in Computer Engineering & Technology*, 1 (2012), pp. 106–110.
- [6] T. P. Sahu, T. V. Dixit, "Modelling and analysis of perturb and observe and incremental conductance MPPT algorithm for PV array using Cuk converter", *IEEE Students' Conference on Electrical, Electronics and Computer Science*, 4 (2014), pp. 213–224.

- [7] K. Ishaque, Z. Salam, "A deterministic particle swarm optimization maximum power point tracker for photovoltaic system under partial shading condition", *IEEE Transactions on Industrial Electronics*, 60 (2013), pp. 3195-3206.
- [8] Y. H. Liu, S. C. Huang, J. W. Huang, W. C. Liang, "A particle swarm optimization based maximum power point tracking algorithm for PV systems operating under partially shaded conditions", *IEEE Transactions on Energy Conversion*, 27 (2012), pp. 1027–1035.
- [9] K.-H. Chao, L.-Y. Chang, H.-C. Liu, "Maximum power point tracking method based on modified particle swarm optimization for photovoltaic systems," *International Journal of Photoenergy*, (2013), pp. 1-6 .
- [10] V. Mnih, K. Kavukcuoglu, D. Silver, "Human-level control through deep reinforcement learning", *Nature*, 518 (2015), pp. 529–533.
- [11] S. Shyamsundar, T. Mannucci and E. van Kampen, "Reinforcement learning based algorithm with Safety Handling and Risk Perception," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, (2016), pp. 1-7.
- [12] R. Sutton, A. Barto, "Reinforcement learning", MIT Press, 2018, pp. 1-552.
- [13] C. J. Watkins, P. Dayan, "Q-learning", *Machine learning*, 8 (1992), pp.279-292.

- [14] C. Kalogerakis, "Design of an Electronic Control System for Maximizing the Energy Production of Photovoltaic Arrays based on Artificial Intelligence Techniques", Technical University of Crete, Diploma thesis, (2019), pp. 1-152.
- [15] E. Durán, J. M. Andújar, J. Galán, M. Sidrach-De-Cardona, "Methodology and experimental system for measuring and displaying I-V characteristic curves of PV facilities," *Progress in Photovoltaics: Research and Applications*, 17 (2009), pp. 574–586.
- [16] E. Durán, J. Galan, M. B. Ferrera, J. M. Andujar, M. Sidrach-de-Cardona, "A new application of duty cycle sweep based on microcontroller to obtain the I-V characteristic n curve of photovoltaic modules", *IEEE International Conference on Industrial Technology*, (2008), pp. 1023–1028.
- [17] M. Abdulkadir, A. H. M. Yatim, S. T. Yusuf, "An Improved PSO-Based MPPT Control Strategy for Photovoltaic Systems", *International Journal of Photoenergy*, (2014), pp. 1-11.
- [18] S. Sengupta, S. Basak, R. A. Peters II, "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives", *Machine Learning and Knowledge Extraction*, 1 (2019), pp. 157-191.
- [19] Y. He, W. J. Ma, J. P. Zhang, "The Parameters Selection of PSO Algorithm influencing On performance of Fault Diagnosis", *MATEC Web Conference*, 63 (2016), pp. 1-5.