

Article

# A Low-Cost Maximum Power Point Tracking System Based on Neural Network Inverse Model Controller

Carlos Robles Algarín \* , Deimer Sevilla Hernández and Diego Restrepo Leal

Facultad de Ingeniería, Universidad del Magdalena, Carrera 32 No. 22-08, 470004 Santa Marta, Colombia; el.deimkydj@gmail.com (D.S.H.); diegorestrepoleal@gmail.com (D.R.L.)

\* Correspondence: carlosarturo.ing@gmail.com; Tel.: +57-5-421-7940

Received: 11 November 2017; Accepted: 30 December 2017; Published: 4 January 2018

**Abstract:** This work presents the design, modeling, and implementation of a neural network inverse model controller for tracking the maximum power point of a photovoltaic (PV) module. A nonlinear autoregressive network with exogenous inputs (NARX) was implemented in a serial-parallel architecture. The PV module mathematical modeling was developed, a buck converter was designed to operate in the continuous conduction mode with a switching frequency of 20 KHz, and the dynamic neural controller was designed using the Neural Network Toolbox from Matlab/Simulink (MathWorks, Natick, MA, USA), and it was implemented on an open-hardware Arduino Mega board. To obtain the reference signals for the NARX and determine the 65 W PV module behavior, a system made of a 0.8 W PV cell, a temperature sensor, a voltage sensor and a static neural network, was used. To evaluate performance a comparison with the P&O traditional algorithm was done in terms of response time and oscillations around the operating point. Simulation results demonstrated the superiority of neural controller over the P&O. Implementation results showed that approximately the same power is obtained with both controllers, but the P&O controller presents oscillations between 7 W and 10 W, in contrast to the inverse controller, which had oscillations between 1 W and 2 W.

**Keywords:** neural network inverse model; nonlinear autoregressive network with exogenous inputs; maximum power point tracking MPPT; dc-dc converter; photovoltaic module

---

## 1. Introduction

In recent years, the use of alternative energy sources to reduce pollution, global warming, and environmental impact has become a priority for governments around the world. In this context, PV energy is becoming a very attractive solution for regions with high solar density. Unfortunately, the energy conversion efficiency of PV modules is still low, reducing the cost-benefit ratio of PV systems implementation. Consequently, increasing this energy efficiency is the aim of numerous research projects. There are different alternatives to optimize the performance of a PV system, such as solar trackers, hybrid systems, and algorithms for maximum power point tracking (MPPT).

Solar trackers are one or two-axis devices that change their orientation throughout the day to track the sun's path and maximize energy capture. In one-axis trackers, the surface rotates on a horizontal or vertical axis [1,2]; while, in the two-axis trackers, the surface is always perpendicular to the sun rays [3–5]. The energy obtained depend on the type of tracker used.

The function of hybrid systems is to maintain the temperature in the manufacturer's recommended value for PV module, which is generally 25 °C. There are several studies in modeling, simulation, fabrication and implementation of thermal solar systems [6–8].

MPPT controllers ensure the highest available power to a load; regardless of variations in climatic conditions [9–12]. The traditional solution for these controllers is the Perturb and Observe (P&O) algorithm. With this method, the operating point of the PV module is disturbed by increasing or

decreasing the duty cycle of a dc-dc converter in order to measure the output power before and after the disturbance. But, it has been shown that it presents oscillation problems around the operating point [13]. For this reason, improving the results obtained with the P&O algorithm has become an important goal to reach for researchers. Some of them use fuzzy logic [14–17], artificial bee colony based algorithm [18], adaptive control [19], glowworm swarm optimization [20], ant colony optimization [21], and neural networks [22–24].

With neural networks, any continuous nonlinear function can be approximated by a multilayer neural network with one or more hidden layers. In addition, neural networks have a parallel architecture that is composed of many processing elements with a very simple structure, which is useful for implementation using embedded systems [25,26].

This work presents the modeling and implementation of a controller for tracking the maximum power point of a PV module using an inverse neural network model, where the main novelty is the use of a NARX based on the low cost platform Arduino with a buck converter as a control device. The main reasons why the Arduino was selected are focused on its low cost, simple programming environment, expandable hardware, open source, and extensible software through C libraries [27]. This project is part of a set of intelligent control techniques that are being developing in the Magma Ingeniería research group in order to implement a low cost MPPT controller that can be used in the PV systems of the Universidad del Magdalena in Santa Marta Colombia [28].

The modeling of the PV module, buck converter, and inverse neural network is presented in Section 2. The implementation of each of these components is presented in Section 3. The analysis and discussion of simulation and implementation results are presented in Section 4. Finally, the conclusions are presented in Section 5.

## 2. PV System Model

### 2.1. PV Module

The PV module was modeled using Equation (1) [5]. This model has the advantage that it is only necessary to calculate the curve fitting parameter ( $b$ ), which can be obtained from I-V equation. The other parameters are obtained from the electrical data of the PV module.

$$I(V) = \frac{I_x}{1 - e^{(\frac{V}{bV_x} - \frac{1}{b})}} \quad (1)$$

where  $V_x$  and  $I_x$  correspond to the open circuit voltage and the short-circuit current, which are represented by Equations (2) and (3).

$$V_x = s \frac{E_i}{E_{in}} T C_v (T - T_N) + s V_{max} - s (V_{max} - V_{min}) e^{\left( \frac{E_i}{E_{in}} \ln \left( \frac{V_{max} - V_{oc}}{V_{max} - V_{min}} \right) \right)} \quad (2)$$

$$I_x = p \frac{E_i}{E_{in}} [I_{sc} + T C_i (T - T_N)] \quad (3)$$

where,  $s$  is the number of PV modules in series;  $p$  is the number of PV modules in parallel;  $E_i$  is the solar irradiance;  $E_{in}$  is  $1000 \text{ W/m}^2$ ;  $T$  is the operating temperature;  $T_N$  is  $25^\circ\text{C}$ ;  $T_{cv}$  is the temperature coefficient of voltage;  $T_{ci}$  is the temperature coefficient of current;  $V_{oc}$  is the open circuit voltage;  $I_{sc}$  is the short-circuit current;  $V_{max}$  is  $103\%V_{oc}$ ; and,  $V_{min}$  is  $85\%V_{oc}$ .

Table 1 shows the electrical parameters of the 65 W PV module used (Yingli Solar, Baoding, China). Because the value of  $b$  is in the range of 0.01 to 0.18 [29,30], the approximation shown in Equation (4) can be made.

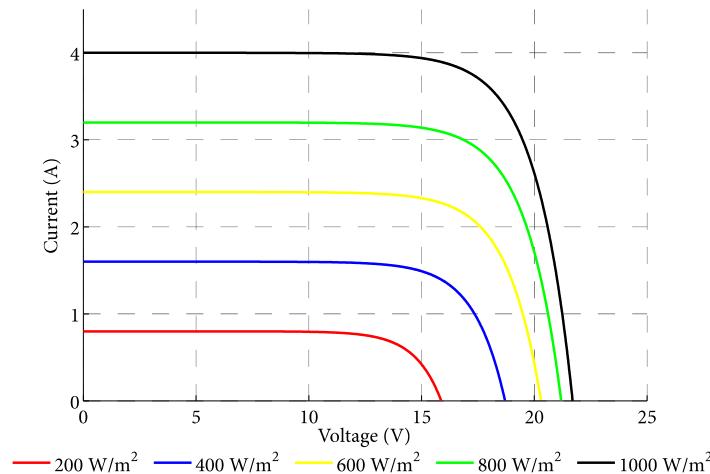
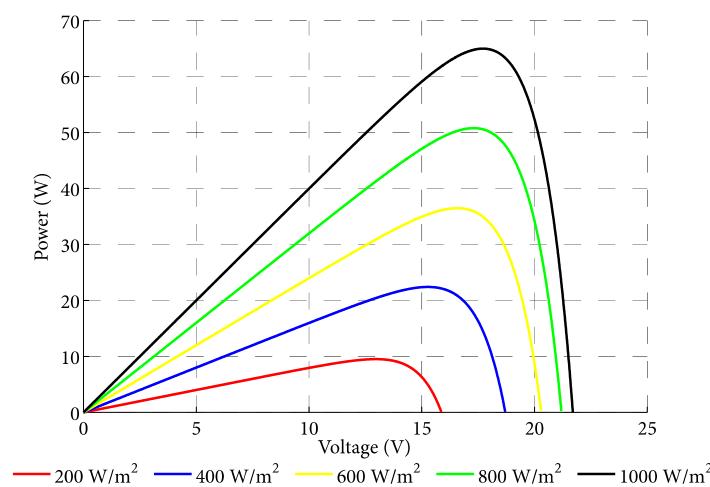
$$1 - e^{(\frac{-1}{b})} \approx 1 \quad (4)$$

Using the electrical parameters presented in Table 1,  $V_x = 21.7 \text{ V}$ ;  $I_x = 4 \text{ A}$ ;  $I = 3.71 \text{ A}$ ; and,  $V = 17.5 \text{ V}$ , the value of  $b$  is obtained,  $b = 0.07375$ .

**Table 1.** Electrical parameters of the photovoltaic (PV) module type YL65P-17b.

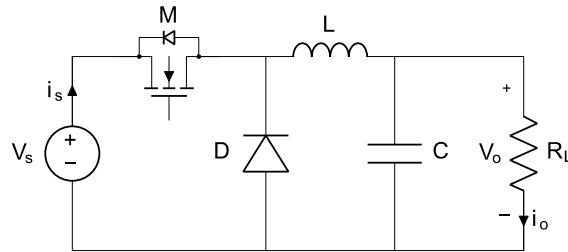
Parameter	Value
Short-circuit current ( $I_{sc}$ )	4 A
Open circuit voltage ( $V_{oc}$ )	21.7 V
Voltage at $P_{max}$ ( $V_{pmax}$ )	17.5 V
Current at $P_{max}$ ( $I_{pmax}$ )	3.71 A
Temperature coefficient of voltage ( $T_{cv}$ )	0.0802 V/ $^{\circ}$ C
Temperature coefficient of current ( $T_{ci}$ )	0.0024 A/ $^{\circ}$ C
Maximum voltage ( $V_{max}$ )	22.35 V
Minimum voltage ( $V_{min}$ )	18.44 V

Figures 1 and 2 show the I-V and P-V curves that are obtained with the electric model of the PV module in Simulink, with variable irradiation and constant temperature of 25 °C. For this purpose, irradiation and temperature were used as inputs, while current and voltage were used as outputs.

**Figure 1.** I-V curve of the PV module under different solar radiances.**Figure 2.** P-V curve of the PV module under different solar radiances.

## 2.2. DC-DC Converter Model

A buck converter was used as the control device, as shown in Figure 3.



**Figure 3.** Buck converter circuit.

Equations (5) and (6) are the ripples of the inductor current for the on and off states of the converter.

$$\Delta I_L(+) = \frac{(V_s - V_o)}{L} T_{on} \quad (5)$$

$$\Delta I_L(-) = \frac{V_o}{L} T_{off} \quad (6)$$

Equating Equations (5) and (6), using  $T = t_{on} + t_{off}$ ; the duty cycle is obtained. See Equation (7).

$$D = \frac{T_{on}}{T} = \frac{V_o}{V_s} \quad (7)$$

To operate in the continuous conduction mode, the critical output current must be greater than or equal to half the inductor current ripple [31]. See Equation (8).

$$i_0(\text{crit}) \geq \frac{\Delta I_L}{2} \quad (8)$$

Using Equations (5) and (8), Equation (9) for the design of the inductor is obtained.

$$L \geq \frac{V_o(1 - \frac{V_o}{V_s})T}{2i_0(\text{crit})} \quad (9)$$

To calculate L, the following values are used:  $V_o = 12$  V,  $V_{s\_max} = 17.71$  V,  $f = 20$  KHz e  $I_{o\_max} = 5.41$  A. Assuming a ripple of 10% for a maximum output current, Equation (10) is obtained.

$$\Delta i_L = 0.1 \times I_o(\text{max}) = 0.541 \text{ A} \quad (10)$$

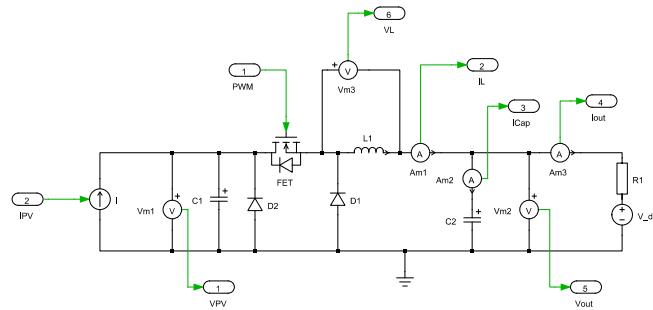
Using Equations (8) and (10), the critical output current is obtained. Subsequently, the value of the inductor is calculated, as shown in Equation (11).

$$L \geq \frac{12 \times \left(1 - \frac{12}{17.71}\right) \times 50 \mu\text{s}}{2 \times 0.2705} \geq 357.57 \mu\text{H} \quad (11)$$

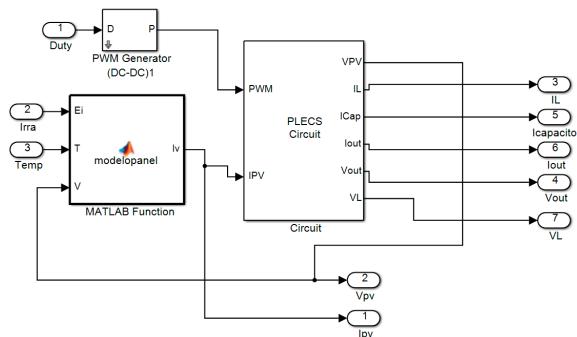
Assuming that the ripple of the inductor current flows through the capacitor and that the dc component flows through the load, the value of the capacitor for a ripple of 0.1% is shown in Equation (12).

$$C \geq \frac{\Delta i_L \times T_s}{8(0.001 \times \Delta V)} \geq \frac{0.5368 \text{ A} \times 50 \mu\text{s}}{8 \times 0.012 \text{ V}} \geq 279.63 \mu\text{F} \quad (12)$$

where the inductor current ripple was calculated using Equation (5) and the value obtained for L in Equation (11). Figure 4 shows the model of the buck converter implemented in Simulink with the Plecs blockset, for which a current source that generates a voltage in the input capacitor was used. Figure 5 shows the model in Simulink of the PV module with the buck converter.

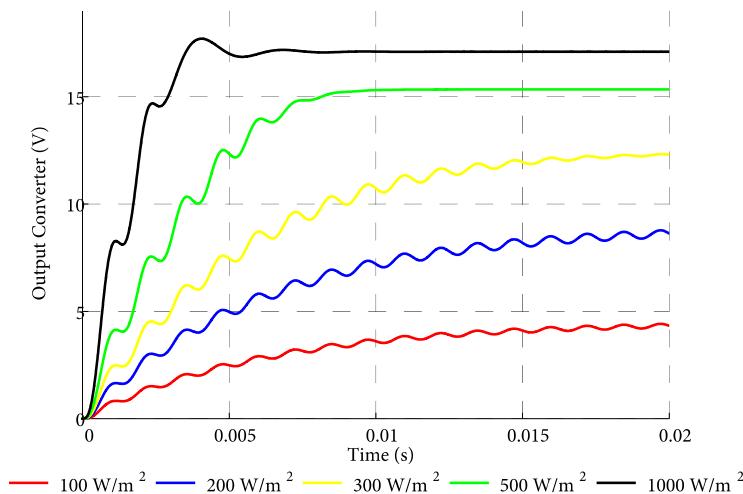


**Figure 4.** Buck converter model.



**Figure 5.** PV module and buck converter.

Figure 6 shows the voltage response of the converter for a duty cycle  $D = 0.9$ , resistive load of  $10 \Omega$ , constant temperature of the PV module of  $25^\circ\text{C}$ , and dynamic values of irradiation between  $100 \text{ W/m}^2$  and  $1000 \text{ W/m}^2$ .



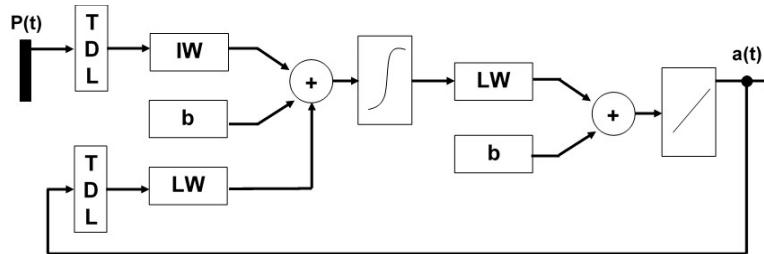
**Figure 6.** Output voltage of buck converter.

### 2.3. Neural Network Inverse Model Controller

The inverse control method was used to ensure maximum power transfer from the PV module to the battery. With this control technique, the dynamic of the plant is canceled by introducing an element between it and the input signal. The simplest strategy to implement the inverse control with artificial neural networks (ANNs) is the direct inverse model, in which an ANN is placed between the reference signal and the plant [32].

A NARX was trained, which can be mathematically defined, as shown in Equation (13); where  $u(t)$  and  $y(t)$  are the input and the output of the network at time  $t$ . Generally, a NARX has a hidden layer with hyperbolic tangent activation function, an output layer with linear activation function and two tapped delay line (TDL) [33,34]. See Figure 7.

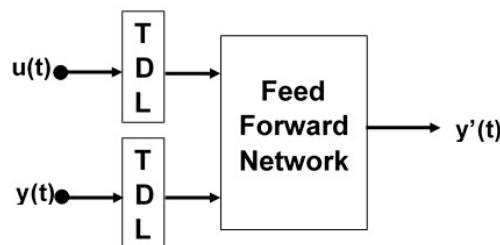
$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (13)$$



**Figure 7.** Recurrent dynamic network with exogenous inputs (NARX).

NARX is a dynamic artificial neural network (DNN) with delays and feedbacks that can modify the impact of synaptic weights and bias on the network response. A DNN is used in problems involving time series, prediction, system identification, filtering, or dynamic modeling. The ability of this type of network to approximate the dynamic of a system, makes them suitable to implement a controller using the inverse model method.

Because the true output is available during the network training, a serial-parallel architecture was implemented using the true output instead of feeding back the estimated output, as shown in Figure 8. This configuration allowed the use of static backpropagation [25], with Levenberg-Marquardt optimization [33] for the network training.



**Figure 8.** Series-parallel architecture for training a NARX.

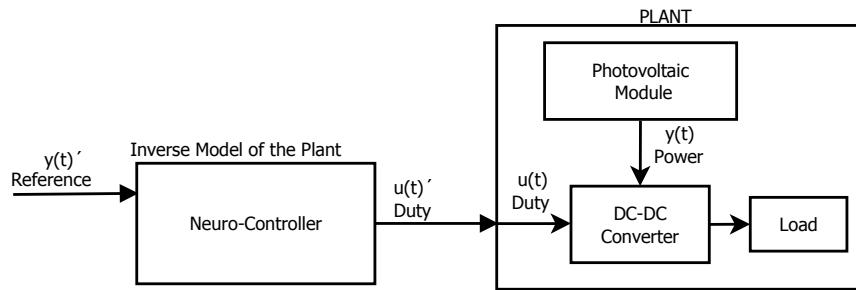
Where the new input  $P$  and the network target  $T$  are shown in Equation (14).

$$P = \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n_u) \\ y(t-1) \\ y(t-2) \\ \vdots \\ y(t-n_y) \end{bmatrix}, T = [y(t)] \quad (14)$$

Training was performed with the `tstool` from Matlab/Simulink Neural Network Toolbox (MathWorks, Natick, MA, USA). The NARX was configured with an input (power) and an output

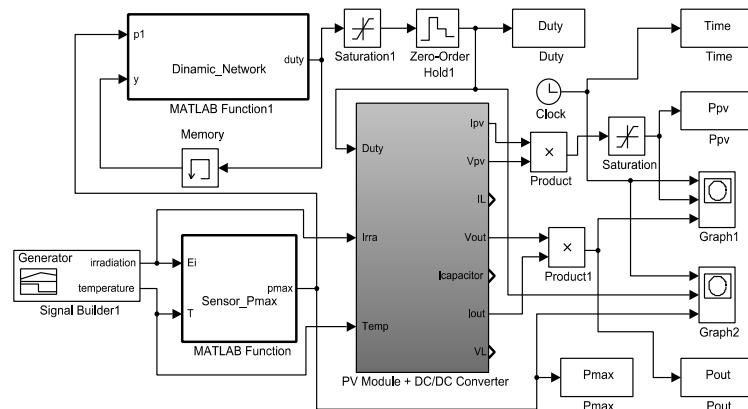
(duty cycle). The network is made of a hidden layer of 10 neurons with a Tansig (Tangent sigmoid) transfer function and an output layer of a neuron with pureline transfer function. For the reference signal and the feedback signal, four delays were used.

A block diagram of the neurocontroller-plant system is shown in Figure 9. In the model of the plant, the output signal is the power of the PV module, while the input signal is the percentage of duty cycle that is applied to the pulse width modulation (PWM) signal of the dc-dc converter.



**Figure 9.** Neurocontroller-plant system.

Figure 10 shows the PV system implemented in Matlab-Simulink, which consists of the PV module, dc-dc converter, and inverse neural network model.



**Figure 10.** PV system in Simulink.

#### 2.4. Training of the Neural Network

In order to obtain the training data, the plant to be controlled was modeled, which has as input the variation of the duty cycle of a PWM signal, and as output the power extracted from the photovoltaic module. See Figure 9. The data were captured and stored in Excel for a week, for which duty cycle variations were made between 80% and 93.8%, where it was observed that only the values of the intervals of 86% up to 93.8% were useful for training. With the above, it was concluded that only maximum power values would be obtained in this range of percentages. In this way, a total of 101,862 data were used to perform the controller training: 50,931 data correspond to the power, and the remaining 50,931 data correspond to the duty cycle.

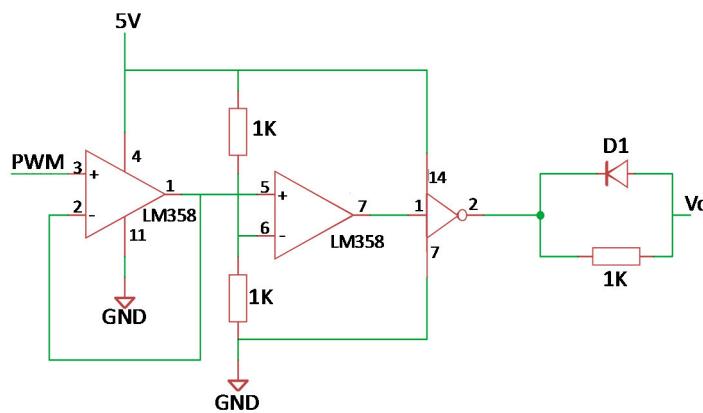
Once the training data set was obtained, the functions of the Matlab Neural Network Toolbox (NNT) to create the architecture of the NARX network were used. The number of neurons in the hidden layer, the number of delays for input and feedback, the training algorithm and its optimization were defined. The training data of the output of the plant and the objective of the neural network were imported from the file in Excel. After this, the NNT applies the arrangement shown in Figure 8 for training the NARX network (a dynamic network) with the static backpropagation

with Levenberg-Marquardt optimization as if it were a static network. The results for the network training were verified with data obtained from the simulation carried out previously in the plant. After validating the training, the implementation was carried out.

### **3. Implementation of the PV System**

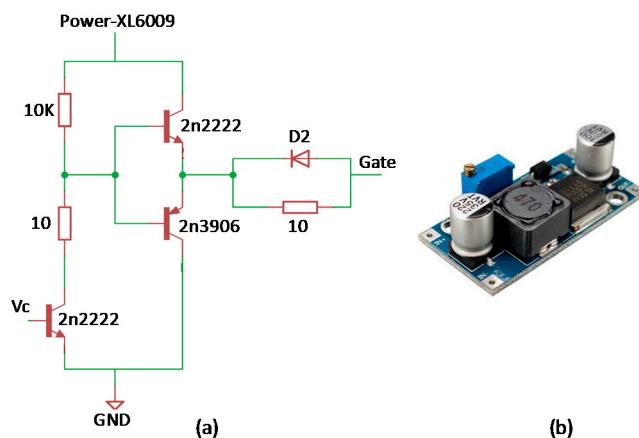
### 3.1. Buck Converter Implementation

The converter was implemented in two stages. The first stage conditions the PWM signal from the Arduino board, compares it with a reference voltage level and generates a square output with a CMOS inverter gate activated by Schmitt Trigger. See Figure 11.



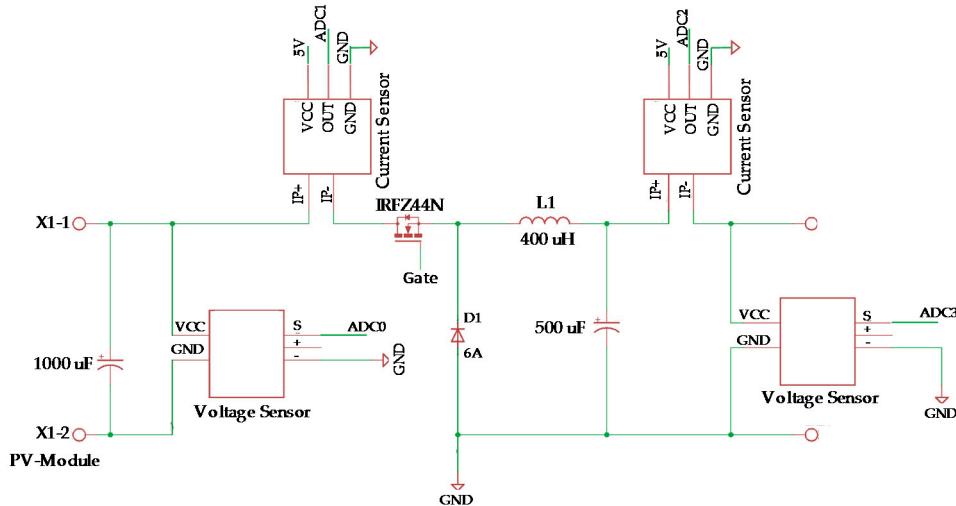
**Figure 11.** Schematic diagram for conditioning the PWM signal from the Arduino board.

In stage 2, to activate the Mosfet IRFZ44N (Infineon Technologies, Neubiberg, Germany), a totem pole driver supplied by a low-power boost XL6009 converter (KylinChip Electronic, Shanghai, China), was used. Additionally, an inductor was constructed using a ferrite core with type J material; in order to avoid the losses due to the Foucault effect and hysteresis. See Figure 12.

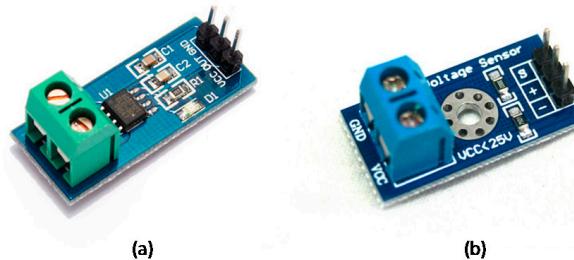


**Figure 12.** (a) Schematic diagram for totem pole driver; (b) XL6009 converter.

Figure 13 shows the schematic diagram of the dc-dc converter implemented. To measure the current of the PV module and the battery, two hall effect current sensors ACS712 (Allegro MicroSystems, Worcester, MA, USA) of 30 A were used. To measure the voltage, two E3-01M-1 modules (Vistronica, Fusagasugá, Colombia) were used, which are responsible for regulating voltages from 25 V to 5 V. See Figure 14.



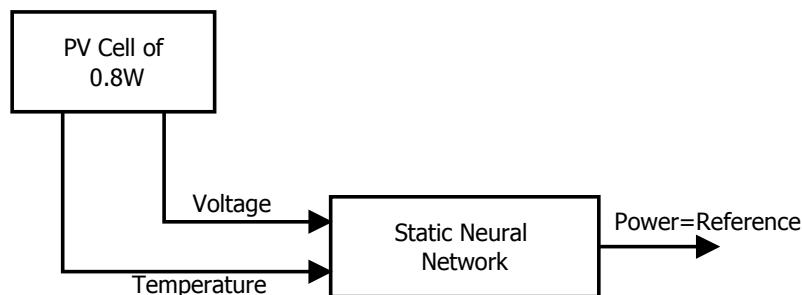
**Figure 13.** Buck converter with current and voltage sensors.



**Figure 14.** (a) Hall-effect current sensor; and, (b) Voltage sensor.

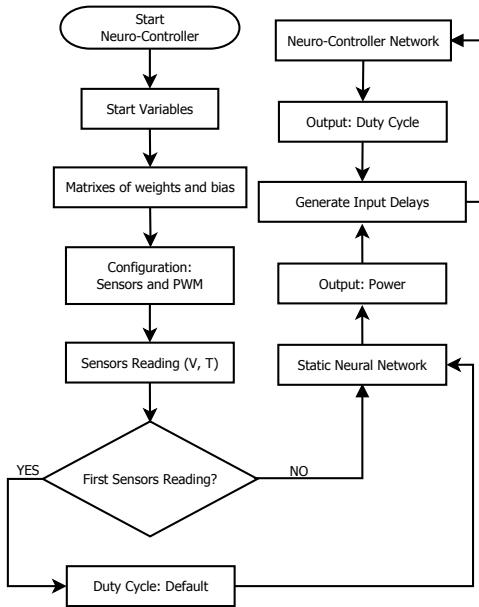
### 3.2. Neural Network Inverse Controller Implementation

It was necessary to carry out the acquisition of a reference signal to obtain a duty cycle for the MPPT. In order to reduce costs, a static neural network was used to generate the reference signal, which has as inputs the temperature and voltage of a PV cell of 0.8 W and as output the maximum power obtained from the PV module. In this way, it was possible to acquire the reference signal. See Figure 15.



**Figure 15.** Neurocontroller reference signal.

Figure 16 shows the flow diagram of the inverse neural controller that was implemented using the low cost Arduino board. Table 2 shows the main specifications of the Arduino Mega board used, which is based on the ATmega 2560 (Microchip, Chandler, AZ, USA) microcontroller [27].



**Figure 16.** Flowchart of the neurocontroller algorithm.

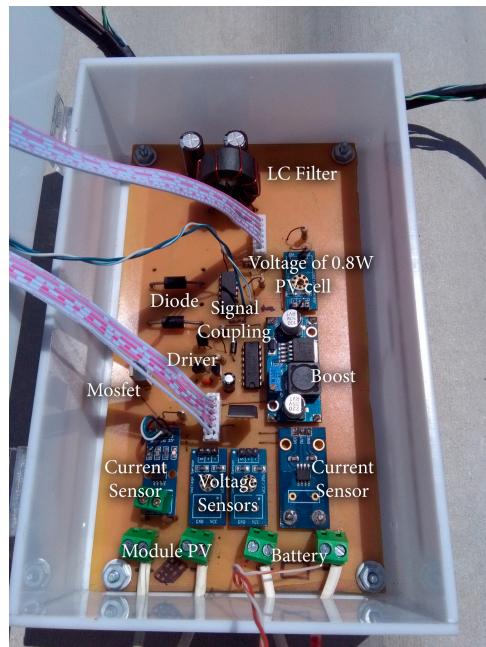
**Table 2.** Arduino mega specifications.

Specifications	Values
Microcontroller	ATmega 2560
Operating Voltage	5 V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
Clock Speed	16 MHz
Flash Memory	256 KB
SRAM-EEPROM	8 KB–4 KB
Communication Interfaces	UART, SPI, I <sup>2</sup> C

Figure 17 shows the PV system, which consists of the PV module, sensor with a static neural network, battery, buck converter, and Arduino neural controller. Figure 18 shows in detail the components used for the buck converter.



**Figure 17.** PV system hardware.

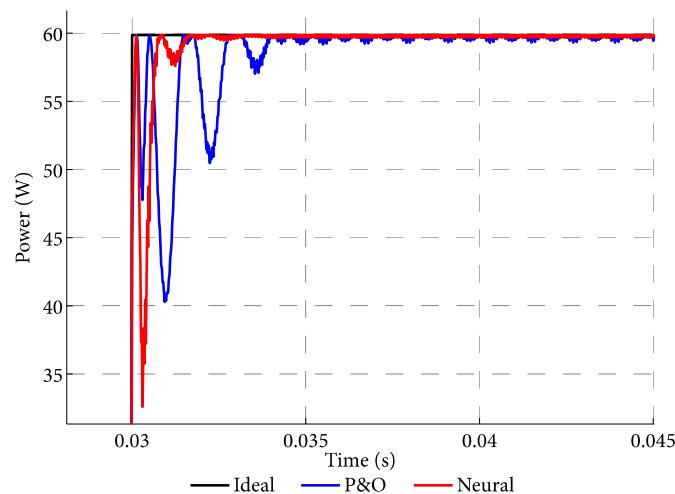


**Figure 18.** Buck converter and sensors used.

#### 4. Results and Discussion

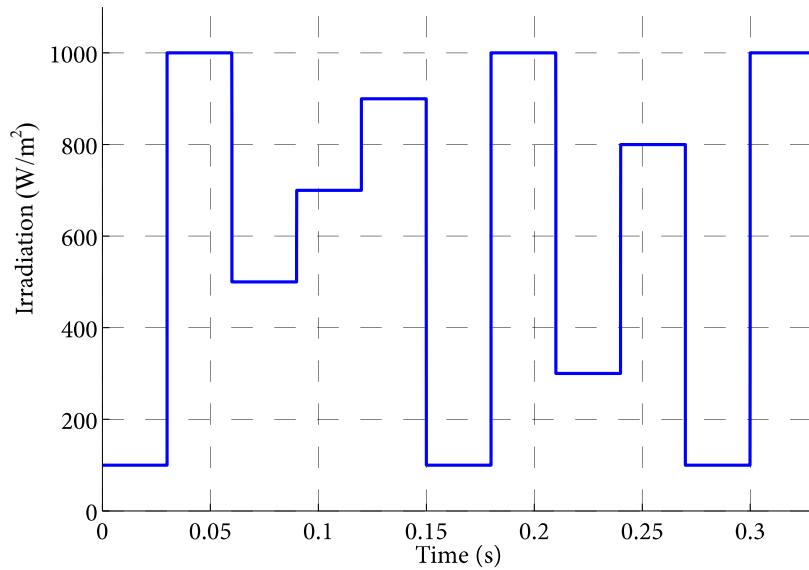
##### 4.1. Simulation Results

Figure 19 shows the output power of the PV module for the inverse neural controller and the P&O algorithm, using a constant temperature of  $50^{\circ}\text{C}$  and a change in irradiation of  $100$  to  $1000\text{ W/m}^2$ . It can be observed that the neural controller is more efficient when compared to the P&O, since it eliminates the oscillations around the operating point and presents a better time response. The neuronal controller stabilizes in a time of  $0.032$  s, while the P&O decreases oscillations in a time of  $0.034$  s; however, it does not eliminate them completely.



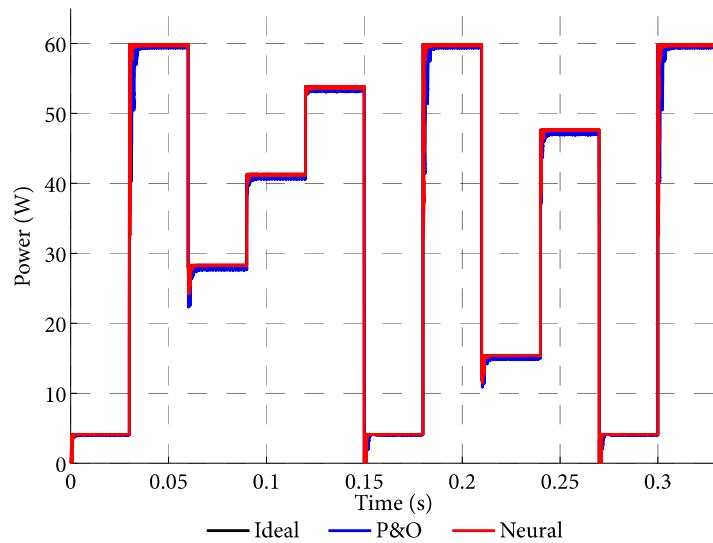
**Figure 19.** Output power of the PV module for an irradiation change from  $100$  to  $1000\text{ W/m}^2$ ,  $50^{\circ}\text{C}$ .

A constant temperature of  $50^{\circ}\text{C}$  and an irradiation signal with sudden changes over seven possible values, at  $0.03$  s,  $0.06$  s,  $0.09$  s,  $0.12$  s,  $0.15$  s,  $0.18$  s,  $0.21$  s,  $0.24$  s,  $0.27$  s, and  $0.30$  s, were used to evaluate the performance of the system. See Figure 20.



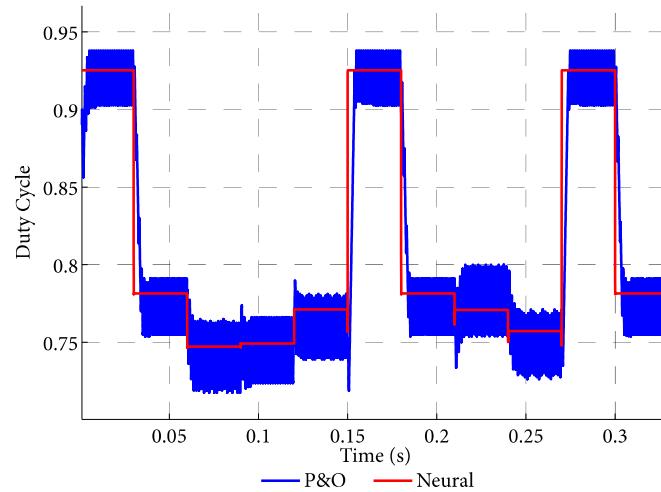
**Figure 20.** Variable irradiation.

For these test conditions, Figure 21 shows the output power that was obtained from the PV module using the neural controller and the P&O algorithm. The results demonstrate that the neural controller accurately tracks the maximum power point without oscillations and power losses. In contrast, the P&O algorithm presents oscillations in each change of irradiation and power losses of 5.56%, 1.50%, 3.97%, 2.17%, 2.08%, 5.56%, 1.50%, 5.88%, 1.89%, 5.56%, and 1.50%. These results can be compared with the work presented in [20], where power losses between 2.3% and 2.5% were obtained during the evaluation of a MPPT control using a P&O algorithm and a glowworm swarm optimization for PV systems.



**Figure 21.** Output power of the PV module under variable irradiation.

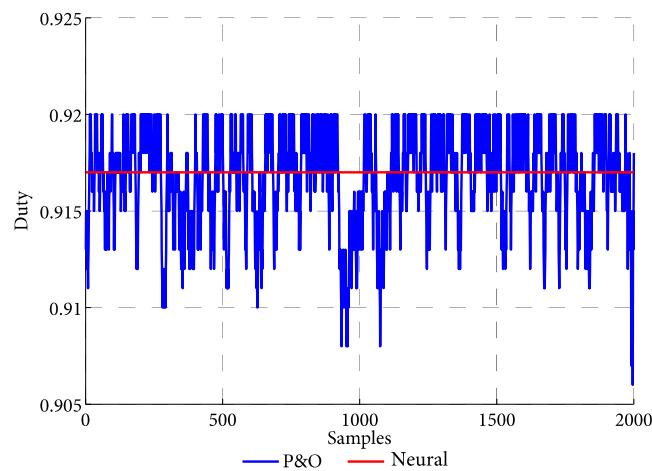
Figure 22 shows in detail the variations of the duty cycle presented with the traditional P&O algorithm in contrast to the stability shown by the inverse neural controller.



**Figure 22.** Duty cycle under variable irradiation.

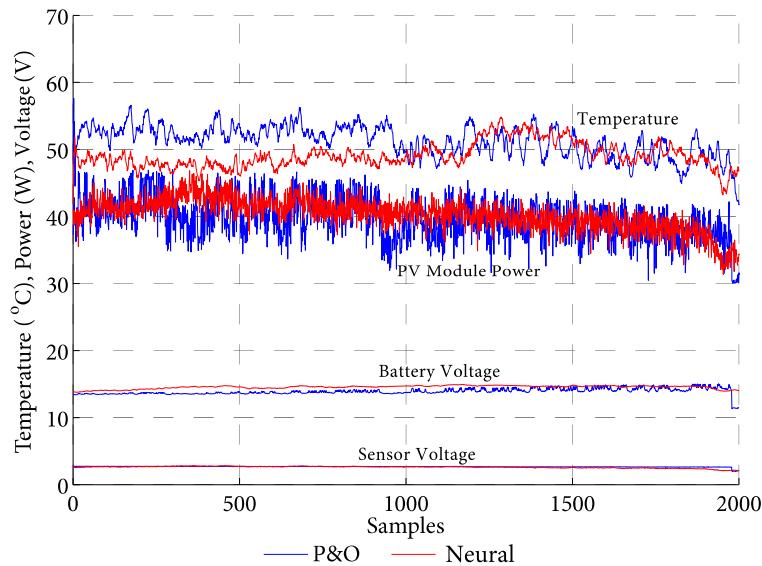
#### 4.2. Implementation Results

Figure 23 shows the implementation results that were obtained for the duty cycle. It can be seen that the neural controller has a constant duty cycle, while the P&O has oscillations. Figure 24 shows the results of temperature, power, battery voltage, and sensor voltage, for the two controllers. By analyzing the obtained power, between the samples 500 to 1000, it can be observed that the voltage of the sensor is practically equal in the neuronal and P&O controllers. It is logical that both of the controllers when reaching the MPP extract the same power of the PV module, however the P&O controller presented oscillations between 7 W and 10 W, while the power obtained from the neural control presents oscillations between 1 W and 2 W.

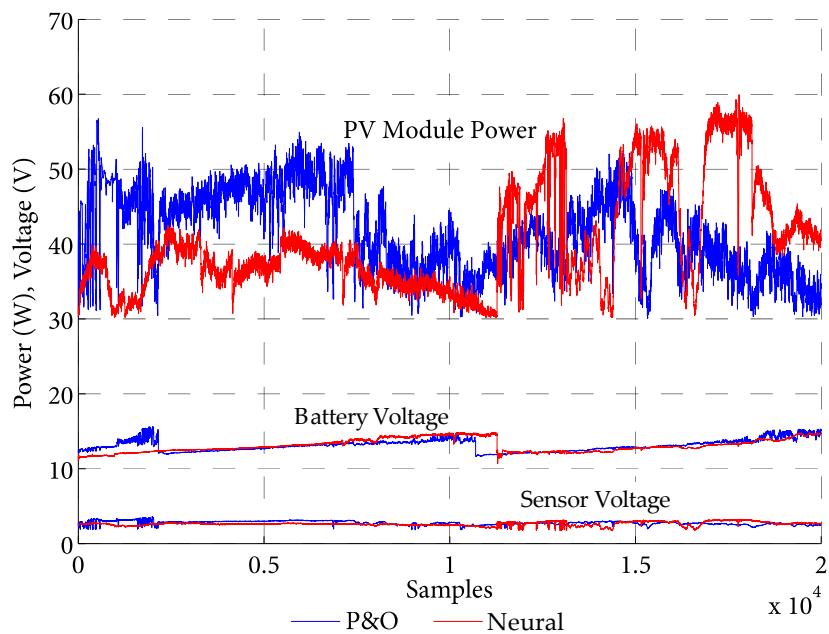


**Figure 23.** Real system duty cycle.

Figure 25 shows the results obtained in the implementation of the controllers during two days in partially cloudy conditions. In the power curve, fluctuations can be observed due to the variations obtained with the voltage sensor. The increase in the battery voltage curve reflects that the battery is being properly charged with the use of the two controllers; highlighting the advantage of neuronal control that presents minimal oscillations. Additionally, in Figures 24 and 25, it is observed that with P&O a lower voltage level is obtained in the battery. This is because the resistance to the charge current is low, and therefore there is a greater current flow.



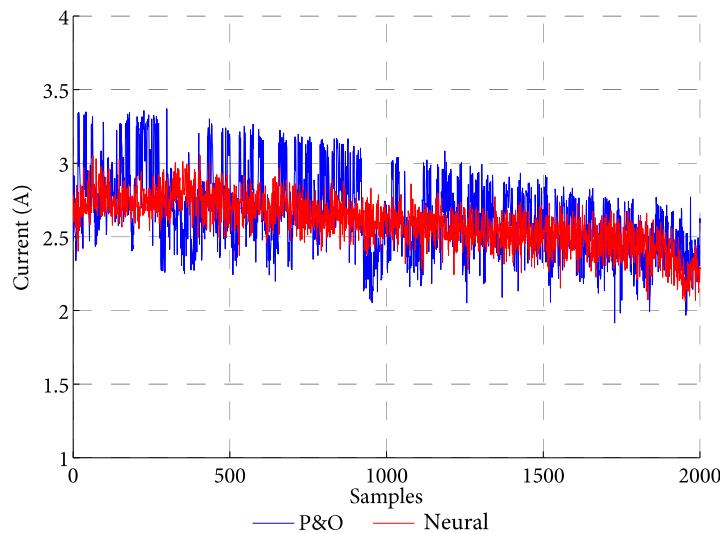
**Figure 24.** Implementation results for the neural controller and Perturb and Observe (P&O) algorithm.



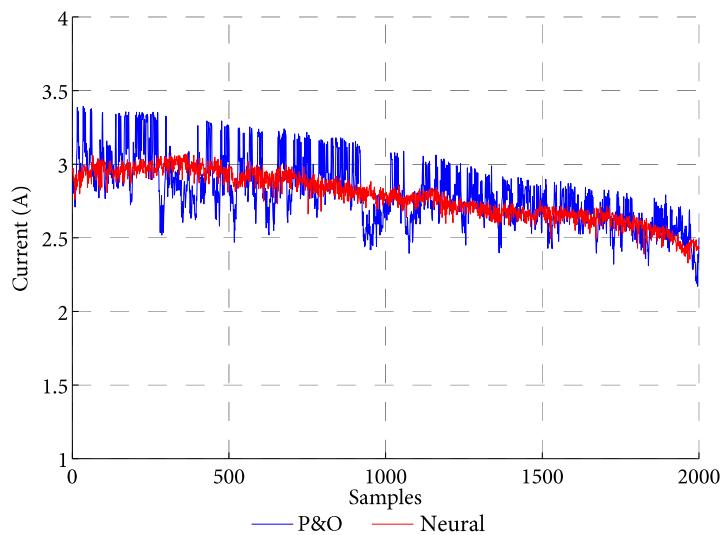
**Figure 25.** Implementation results for two days for the neural controller and P&O.

Figure 26 shows the current extracted from the PV module and Figure 27, the current flowing to the battery. In Figure 26, higher current levels are observed due to the low resistance to the charge current, in contrast to the current values that are observed in Figure 27. In both cases, greater oscillations occur with P&O when compared with the neural control.

It should be noted that the results that are obtained in the simulation and implementation of the MPPT controller are based on the implementation of a buck converter with low losses due to the equivalent resistance  $R_s$  and the switching of the Mosfet, and the implementation of a dynamic recurrent neural network, which is ideal to be used in systems with dynamic behavior, as is the case of PV modules. In this way, the main contribution of this manuscript is the guarantee of supplying the maximum possible power to a battery without oscillations that generate power losses and affect the performance of the system. Negligible oscillations, use of low cost tools, and easy implementation are the main advantages of the proposed neural controller.



**Figure 26.** Current obtained from the PV module.



**Figure 27.** Current delivered to the battery.

## 5. Conclusions

With this work, it was demonstrated that the neural network inverse model control strategy allows for tracking the maximum power point of a PV module in a more efficient way than the traditional P&O algorithm, minimizing the oscillations around the operating point. The NARX presented a better response to disturbances due to the ability of dynamic neural networks to address problems involving dynamic systems. The use of static neural network was a good choice, since data were obtained to establish the dynamics of the PV module in order to perform the appropriate training for the NARX.

The buck converter was designed to operate in the continuous conduction mode, with a switching frequency of 20 KHz and a high permeability of the ferrite toroid, which allowed for decreasing the losses due to the equivalent resistance. A low permeability of the ferrite core produces heating in the coil, leading to greater losses in the dc-dc converter.

The Neural Network Toolbox was an important tool to design and train the neural network. Besides, Simulink provides an appropriate environment to perform the simulations and approach the behavior of the real system. Additionally, the implementation of the controller in the Arduino platform allowed for evaluating the relevance of this type of tools in intelligent control systems.

**Acknowledgments:** This work was supported by the Vicerrectoría de Investigación of the Universidad del Magdalena.

**Author Contributions:** Carlos Robles Algarin designed the dc-dc converter, the P&O controller and wrote the paper. Deimer Sevilla Hernández performed the hardware implementation. Diego Restrepo Leal designed the inverse neural controller. All authors read and approved the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Stamatescu, I.; Fagarasan, I.; Stamatescu, G.; Arghira, N.; Iliescu, S. Design and Implementation of a Solar Tracking Algorithm. *Procedia Eng.* **2014**, *69*, 500–507. [[CrossRef](#)]
2. Vieira, R.; Guerra, F.; Vale, M.; Araújo, M. Comparative performance analysis between static solar panels and single-axis tracking system on a hot climate region near to the equator. *Renew. Sustain. Energy Rev.* **2016**, *64*, 672–681. [[CrossRef](#)]
3. El Kadmiri, Z.; El Kadmiri, O.; Masmoudi, L.; Najib, M. A Novel Solar Tracker Based on Omnidirectional Computer Vision. *J. Sol. Energy* **2015**, *2015*, 1–6. [[CrossRef](#)]
4. Phong, J.; Rahim, N.; Al-Turki, Y. Performance of Dual-Axis Solar Tracker versus Static Solar System by Segmented Clearness Index in Malaysia. *Int. J. Photoenergy* **2013**, *2013*. [[CrossRef](#)]
5. Robles, C.; Ospino, A.; Casas, J. Dual-Axis Solar Tracker for Using in Photovoltaic Systems. *Int. J. Renew. Energy Res.* **2017**, *7*, 137–145.
6. Chen, F.; Yin, H. Fabrication and laboratory-based performance testing of a building-integrated photovoltaic thermal roofing panel. *Appl. Energy* **2016**, *177*, 271–284. [[CrossRef](#)]
7. Lamnatou, C.; Mondol, J.; Chemisana, D.; Maurer, C. Modelling and simulation of Building-Integrated solar thermal systems: Behaviour of the coupled building/system configuration. *Renew. Sustain. Energy Rev.* **2015**, *48*, 178–191. [[CrossRef](#)]
8. Beker, M.; Riffat, S. Building integrated solar thermal collectors—A review. *Renew. Sustain. Energy Rev.* **2015**, *51*, 327–346. [[CrossRef](#)]
9. Rezaee, A. Maximum power point tracking in photovoltaic (PV) systems: A review of different approaches. *Renew. Sustain. Energy Rev.* **2016**, *65*, 1127–1138. [[CrossRef](#)]
10. Saadsaoud, M.; Abbassi, H.; Kermiche, S.; Ouada, M. Study of Partial Shading Effects on Photovoltaic Arrays with Comprehensive Simulator for Global MPPT Control. *Int. J. Renew. Energy Res.* **2016**, *6*, 413–420.
11. Filippini, M.; Molinas, M.; Oregi, E.O. A Flexible Power Electronics Configuration for Coupling Renewable Energy Sources. *Electronics* **2015**, *4*, 283–302. [[CrossRef](#)]
12. Yaden, M.F.; Melhaoui, M.; Gaamouche, R.; Hirech, K.; Baghaz, E.; Kassmi, K. Photovoltaic System Equipped with Digital Command Control and Acquisition. *Electronics* **2013**, *2*, 192–211. [[CrossRef](#)]
13. Sheik, S.; Devaraj, D.; Imthias, T. A novel hybrid Maximum Power Point Tracking Technique using Perturb & Observe algorithm and Learning Automata for solar PV system. *Energy* **2016**, *112*, 1096–1106. [[CrossRef](#)]
14. Chen, Y.; Jhang, Y.; Liang, R. A fuzzy-logic based auto-scaling variable step-size MPPT method for PV systems. *Sol. Energy* **2016**, *126*, 53–63. [[CrossRef](#)]
15. Hocksun, T.; Wu, X. Maximum power point tracking using a variable antecedent fuzzy logic controller. *Sol. Energy* **2016**, *137*, 189–200. [[CrossRef](#)]
16. Nabipour, M.; Razaz, M.; Seifossadat, S.; Mortazavi, S. A new MPPT scheme based on a novel fuzzy approach. *Renew. Sustain. Energy Rev.* **2017**, *74*, 1147–1169. [[CrossRef](#)]
17. Hassan, S.Z.; Li, H.; Kamal, T.; Arifoğlu, U.; Mumtaz, S.; Khan, L. Neuro-Fuzzy Wavelet Based Adaptive MPPT Algorithm for Photovoltaic Systems. *Energies* **2017**, *10*, 394. [[CrossRef](#)]
18. Benyoucef, A.; Chouder, A.; Kara, K.; Silvestre, S.; Ait, O. Artificial bee colony based algorithm for maximum power point tracking (MPPT) for PV systems operating under partial shaded conditions. *Appl. Soft Comput.* **2015**, *32*, 38–48. [[CrossRef](#)]
19. Pradhan, R.; Subudhi, B. Design and real-time implementation of a new auto-tuned adaptive MPPT control for a photovoltaic system. *Int. J. Electr. Power* **2015**, *64*, 792–803. [[CrossRef](#)]
20. Hou, W.; Jin, J.; Zhu, C.; Li, G. A Novel Maximum Power Point Tracking Algorithm Based on Glowworm Swarm Optimization for Photovoltaic Systems. *Int. J. Photoenergy* **2016**, *2016*. [[CrossRef](#)]

21. Titri, S.; Larbes, C.; Youcef, K.; Benatchba, K. A new MPPT controller based on the Ant colony optimization algorithm for Photovoltaic systems under partial shading conditions. *Appl. Soft Comput.* **2017**, *58*, 465–479. [[CrossRef](#)]
22. Kofinas, P.; Dounis, A.; Papadakis, G.; Assimakopoulos, M. An Intelligent MPPT controller based on direct neural control for partially shaded PV system. *Energy Build.* **2015**, *90*, 51–64. [[CrossRef](#)]
23. Aymer, M.; Romero, H.; Carmona, J.; Gomand, J. Maximum Power Point Tracking Using P&O Control Optimized by a Neural Network Approach: A Good Compromise between Accuracy and Complexity. *Energy Procedia* **2013**, *42*, 650–659. [[CrossRef](#)]
24. Messalti, S.; Harrag, A.; Loukriz, A. A new variable step size neural networks MPPT controller: Review, simulation and hardware implementation. *Renew. Sustain. Energy Rev.* **2017**, *68*, 221–233. [[CrossRef](#)]
25. Hu, Y.H.; Hwang, J.N. *Handbook of Neural Network Signal Processing*, 1st ed.; CRC Press: Seattle, WA, USA, 2001; pp. 12–13.
26. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1998; pp. 23–28.
27. Robles Algarín, C.; Callejas Cabarcas, J.; Polo Llanos, A. Low-Cost Fuzzy Logic Control for Greenhouse Environments with Web Monitoring. *Electronics* **2017**, *6*, 71. [[CrossRef](#)]
28. Robles Algarín, C.; Taborda Giraldo, J.; Rodríguez Álvarez, O. Fuzzy Logic Based MPPT Controller for a PV System. *Energies* **2017**, *10*, 2036. [[CrossRef](#)]
29. Gil, O. Modelado y Simulación de Dispositivos Fotovoltaicos. Master's Thesis, Universidad de Puerto Rico, San Juan, Puerto Rico, 2008.
30. Ortiz, E. Modeling and Analysis of Solar Distributed Generation. Ph.D. Thesis, Michigan State University, East Lansing, MI, USA, 2006.
31. Robles, C.; Villa, G. Control del punto de máxima potencia de un panel solar fotovoltaico, utilizando lógica difusa. *Telematique* **2011**, *10*, 54–72.
32. Hussain, M.; Ali, J.; Khan, M. Neural Network Inverse Model Control Strategy: Discrete-Time Stability Analysis for Relative Order Two Systems. *Abstr. Appl. Anal.* **2014**, *2014*, 1–11. [[CrossRef](#)]
33. Hagan, M.T.; Demuth, H.B.; Beale, M.H.; De Jesús, O. *Neural Network Design*, 2nd ed.; Oklahoma State University: Stillwater, OK, USA, 2014; pp. 892–895.
34. Medsker, L.R.; Jain, L.C. *Recurrent Neural Networks: Design and Applications*, 1st ed.; CRC Press: Seattle, WA, USA, 2001; pp. 142–145.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).