Available online at www.sciencedirect.com

ScienceDirect

Solar Energy 85 (2011) 265–277

SOLAR ENERGY

www.elsevier.com/locate/solener

# Maximum power point tracking using a GA optimized fuzzy logic controller and its FPGA implementation

A. Messai [a], A. Mellit [b,*,1], A. Guessoum [c], S.A. Kalogirou [d]

[a] CRNB Ain Oussera, P.O. Box 180, 17200 Djelfa, Algeria
[b] Department of Electronics, Faculty of Sciences and Technology, Jijel University, Ouled-aissa, P.O. Box 98, Jijel 18000, Algeria
[c] Department of Electronics, Faculty of Sciences Engineering, Blida University, Blida 90000, Algeria
[d] Department of Mechanical Engineering and Materials Science and Engineering, Cyprus University of Technology, P.O. Box 50329, Limassol 3603, Cyprus

Received 24 May 2010; received in revised form 29 November 2010; accepted 2 December 2010
Available online 5 January 2011

Communicated by: Associate Editor Nicola Romeo

## Abstract

Maximum power point tracking (MPPT) must usually be integrated with photovoltaic (PV) power systems so that the photovoltaic arrays are able to deliver the maximum power available. In this paper details of the work, carried out to optimize and implement a fuzzy logic controller (FLC) used as a maximum-power-point tracker for a stand-alone PV system, are presented. The near optimum design for membership functions and control rules were found simultaneously by genetic algorithms (GAs) which are search algorithms based on the mechanism of natural selection and genetics. These are easy to implement and efficient for multivariable optimization problems such as in fuzzy controller design. The FLC thus designed, as well as the components of the PV control unit, were implemented efficiently on a Xilinx reconfigurable field-programmable gate array (FPGA) chip using VHDL Hardware Description Language. The obtained simulation results confirm the good tracking efficiency and rapid response to changes in environmental parameters.


*Keywords:* Maximum power point tracking (MPPT); Photovoltaic (PV); Fuzzy logic controller (FLC); Genetic algorithms optimization (GAO); Field-programmable gate array (FPGA); Very high speed integrated circuit Hardware Description Language (VHDL)

## 1. Introduction

Photovoltaic (PV) cells are an attractive source of energy. Abundant and ubiquitous, this source is one of the important renewable energy sources that have been increasing worldwide year by year (Goetzberger and Hoffmann, 2005). Unfortunately, PV generation systems have two major problems; the conversion efficiency of electric power generation is very low (from 12% in ordinary units up to a maximum of 42.8% in very special setups), especially under low irradiation conditions, and the amount of electric power generated by solar arrays depends on many extrinsic factors, such as isolation (incident solar radiation) levels, temperature, ageing and load conditions (Godoy and Franceschetti, 1999).

In order to extract maximum power from a PV module, a maximum-power-point tracker (MPPT), which is a DC/DC converter associated with the control unit, is usually connected between the photovoltaic module and the load (Kida et al., 1991).

Solar irradiance varies quickly with time, which means that the maximum power point also moves to another curve quickly. Therefore, the MPPT controller should also be capable of reaching the MPP as quickly as possible in order to reduce the output power oscillations and the system power loss.

* Corresponding author. Tel.: +213 551998982.
*E-mail address:* a.mellit@yahoo.co.uk (A. Mellit).
[1] Associate member at the International Centre for Theoretical and Physics (ICTP), Trieste, Italy.

0038-092X/$ - see front matter 
doi:10.1016/j.solener.2010.12.004

The issue of MPPT in the context of PV systems has been addressed in many different ways in the literature (Calais and Hinz, 1998; Hohm and Ropp, 2003; Feel-soon Kang et al., 2005; Kobayashi et al., 2006; Xiao and Dunford, 2004; Salas et al., 2006; Enrique et al., 2007). Among the various techniques proposed, the use of intelligent techniques-based MPP trackers should be noted. These were recently developed and used to improve or to replace conventional tracking techniques based on perturb and observe method (P&O), the incremental conductance method (IncCond) or the hill climbing method (HC) (Bahgat et al., 2005; Mellit and Kalogirou, 2008; Chu and Chen, 2009).

Due to their heuristic nature associated with simplicity and effectiveness, for both linear and non-linear systems, fuzzy logic controller (FLC) methods have showed their salient features in implementations for MPP seeking (Khaehintung et al., 2004; Veerachary et al., 2003; Altas and Sharaf, 2008; Mellit et al., 2010a). Hence, many studies and applications have been proposed, combining MPP tracking and FLC (Godoy and Franceschetti, 1999; Mellit and Kalogirou, 2008). For example, fuzzy logic control has been used for MPPT by Won et al. (1994), who demonstrated the fine performance of the fuzzy-based MPPT under different environmental operating conditions. In a similar approach, the experimental results obtained via the fuzzy tracker is presented by Khaehintung and Sirisuk (2004) who have shown that the MPPT was more than eight times better in terms of tracking speed over the conventional MPPT using the P&O method. These results have also revealed that a PV system based upon the proposed controller can reach a power efficiency of about 85%. Furthermore, Gounden et al. (2009) presented a MPPT controller for a grid-connected PV generation system, which is a representative example of this category and proved that the fuzzy logic control is an effective tool to extract maximum power to the grid.

However, in spite of the good results provided by all the aforementioned works, all have one common drawback in the design of the FLC employed, which was done according to the trial-and-error method rather than a guided approach. In this traditional design, the presence of an expert knowledge is compulsory; conversely, in the absence of such knowledge, their design is usually slow and not optimized (Linkens and Nyongesa, 1995). To provide a way of surmounting this shortcoming, Larbes et al. (2009) choose the Genetic Algorithms (GAs) tool to optimize the FLC of their MPP tracker. They applied GAs to calculate accurately the base lengths and the peak locations of the membership functions in the FLC for which the rule-base have already been created. The proposed solution leads to a good performance improvement of the MPP tracker addressed. Nevertheless, a literature review in the area of FLC's design (Homaifar and McCormick, 1995) reveals that in such a situation, the designed FLC is not yet optimal and still requires the use of an expert's experience to design the control rules.

In this work, we propose a more efficient design for a FLC-based MPPT planned to be used in stand-alone PV systems. Our strategy is based on GAs which choose optimally and simultaneously both membership functions and control rules for the FLC. The procedure followed makes the design of this type of MPP tracker simpler and more efficient.

Additionally, a software implementation of the designed FLC on general purpose computer cannot be considered as a suitable design solution for this type of application, especially when it has to be used as MPPT controller for stand-alone PV systems installed in remote rural areas. In such a situation, the FLC was usually implemented in microcontrollers (Khaehintung et al., 2004; Eakburanawat and Boonyaroonate, 2006) and/or in digital signal processors (DSPs) (Akkaya et al., 2007). In fact, these chips can provide a reasonable performance, but they do not provide the advantages that field-programmable gate array (FPGA) chips can potentially offer to the implementation of the MPPT control unit (Khaehintung et al., 2006). In comparison with DSP and microcontroller implementations, FPGAs offer lower cost since the various electronic functions required by the control unit can be integrated onto the same FPGA chip (Jiménez et al., 1995; Mekki et al., 2010; Mellit et al., 2010b) as opposed to the former chips which can only perform software DSP-related computations. Due to this advantage, an FPGA implementation of the GAs optimized FLC is also proposed at the end of this work.

The paper is organized as follows: Next section presents the MPPT in a stand-alone PV system, whereas a brief background on fuzzy logic controllers and genetic algorithm theory is presented in Section 3. Section 4 introduces GAs as the tool used for optimizing the fuzzy controller and its application for the MPPT control design. Section 5 provides the simulation results of the new designed GA-FLC. Finally, Section 6 outlines the steps for the implementation of the designed GA-FLC on FPGA and the performance obtained.

## 2. MPPT in stand-alone PV systems

As shown in Fig. 1, a generic stand-alone PV system includes a solar array, DC/DC converter, resistive load and an MPPT control unit.

The PV array converts solar energy to electric energy. In its V–P characteristic curve, there is a maximum point commonly called the maximum power point (MPP), at which the module operates with maximum efficiency and produces the maximum power output $P_{\max}$ (see Fig. 2). In order to obtain an operating point close to this MPP, a controlled DC/DC converter is usually employed to interface the energy flow from the solar module to the load.

The main task of MPP seeking and DC/DC converter operation is provided by a control unit (MPPT Controller), where one of the known MPP tracking algorithms is usually implemented. This continuously checks the output (current and voltage) of the PV module, compares them to

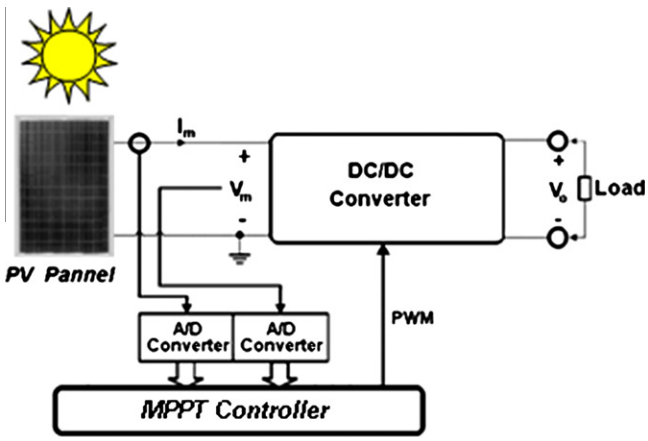


Fig. 1. The considered stand-alone PV system.

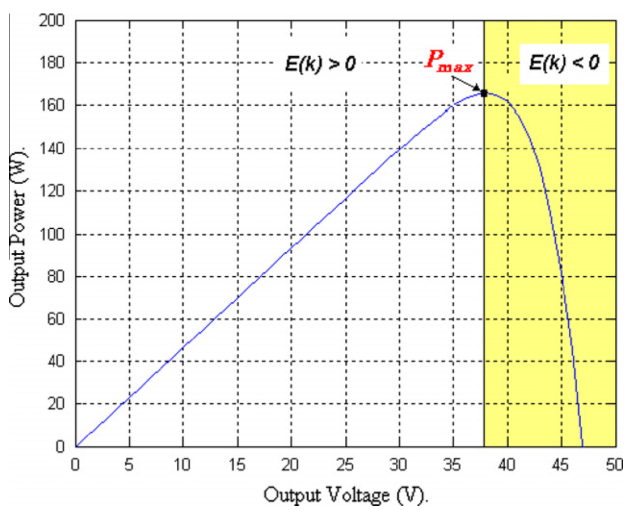


Fig. 2. Power curve under constant irradiance and temperature.

the previous values, and fixes what is the best power that PV module can produce to supply the power to the DC load under all operating and weather conditions (Goetzberger and Hoffmann, 2005). In the present application, a fuzzy logic controller was used as the core of the control unit.

## 3. Fuzzy MPPT for PV systems

In this section it is not intended to give the theoretical background information about fuzzy logic; we will just give the necessary concepts that will allow the understanding of the analysis that follows. Fuzzy control is based on the principles of fuzzy Logic developed by Zadeh in 1965 (Timothy, 2004). It is a non-linear control method, which attempts to apply the expert knowledge of an experienced user to the design of a fuzzy-based controller. Generally, as depicted in Fig. 3, a FLC contains four main components:

(a) The fuzzifier that maps crisp values into input fuzzy sets to activate rules.
(b) The rules which define the controller behavior by using a set of IF–THEN statements.

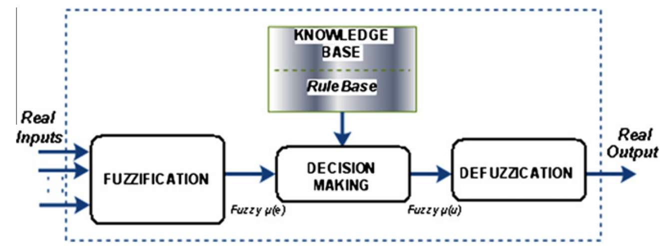


Fig. 3. Fuzzy inference system.

(c) The inference engine which maps input fuzzy sets into output fuzzy sets by applying the rules, and
(d) The defuzzifier that maps output fuzzy values into crisp values.

The rules describing the FLC operation are expressed as linguistic variables represented by fuzzy sets. The controller output is obtained by applying an inference mechanism (Jiménez et al., 2004), which define:

(1) the kind of membership functions,
(2) the connections used to link the rules antecedents,
(3) the implication function chosen, and
(4) the rule aggregation operator.

In the case of fuzzy controllers hardware implementation, which is of interest here, the shapes of the membership functions associated to the FLC linguistic variables are often piece-wise linear functions (triangular or trapezoidal).

It should be noted that the number and shape of the membership functions of each fuzzy set as well as the fuzzy logic inference mechanism was initially selected based on trial-and-error methods, in a manner that the region of interest is covered appropriately by the input data. The idea behind the chosen reasoning in this paper was; if the last change in the control signal ($D$) caused the power to rise, keep moving in the same direction; otherwise, if it has caused the power to drop move it in the opposite direction. Subsequently, as it will be explained in the following sections, a genetic algorithms (GAs) technique was used to tune both of membership functions and the rule-base set.

The MPPT using the Mamdani's FLC approach, which uses the min–max operation fuzzy combination law, is designed in a manner that the control task try to continuously move the operation point of the solar array as close as possible to the maximum power point (MPP) (Khaehintung and Sirisuk, 2004). The two inputs of the proposed fuzzy controller are the tracking error ($E$) and the change of the error ($\Delta E$), which are defined by:

$$E(n) = \frac{p(n) - p(n-1)}{V(n) - V(n-1)} \tag{1}$$

$$\Delta E(n) = E(n) - E(n-1) \tag{2}$$

where $E$ and $\Delta E$ are the error and change in error, $n$ is the sampling time, $p(n)$ is the instantaneous power of the PV generator, and $V(n)$ is the corresponding instantaneous voltage. These inputs are chosen so that the instantaneous value of $E(n)$ shows if the load operation power point is located on the right or in the left compared to the $P_{max}$ actual position. While $\Delta E(n)$ expresses the moving direction of this operation point (see Fig. 2).

The output variable is the pulse width modulation (PWM) signal called $D$, which is transmitted to the boost DC/DC converter to drive the load. After the rules have been applied, the center of area as the defuzzication method is used to find the actual value of ($D$) as a crisp output.

In our application, the range of the power error is (−35.9 to 4.75 W/V) and their linguistic variables are considered as negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB) whereas change of power error range is (−40 to 40 W/V) and its linguistic variables are selected as negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB). The output variable is the PWM signal driver whose range is (−2.85 to 2.85 V) and its linguistic variables are chosen as negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB).

## 4. Genetic algorithms (GAs) as a tool of FLC optimization for a MPPT

The genetic algorithms are based on concepts of evolutionary theory, and provide an effective way of searching a large and complex solution space to give close to optimal solutions much faster than random trial-and-error methods. They are also generally more effective at avoiding local minima than differentiation-based approaches.

The basic mechanism of a GA can be simply described as follows (Timothy, 2004):

(1) *Define the string of a chromosome*: The string of searching parameters for the optimization problem should be defined first. These parameters are genes in a chromosome, which can be binary coded or real coded. Different chromosomes represent different possible solutions.
(2) *Define the fitness function*: The fitness function is the performance index of a GA to resolve the viability of each chromosome. The design of the fitness function is according to the performance requirements of the problem, e.g., convergence value, error, rise-time, etc.
(3) *Generate an initial population*: A set of chromosomes should be randomly generated before using a GA operation. These chromosomes are called the initial population. The size of the population is chosen according to the complexity of the optimization problem. Generally speaking, larger values require fewer generations to converge to a solution. However, the total computation time depends also on the number ($N$) of used generations to reach the algorithm's convergence.
(4) *Generate the next generation or stop*: GAs use the operations of reproduction, crossover and mutation, as detailed in Fig. 4, to generate the next generation. From generation to generation, a maximum value of the fitness value is achieved.

Genetic algorithms as just described can be used to tune the fuzzy controller parameters like the structure of rules and membership functions to find those parameter values that are optimal with respect to the design criteria (Karr and Gentry, 1993). The optimization of these two entities can be done separately, which may result in a suboptimal solution, because the design parts are mutually dependent as is presented by Homaifar and McCormick (1995) who demonstrated clearly that by using GA's to design both parameters simultaneously, the two elements of fuzzy controllers can be fully integrated to deliver a more finely tuned, high performance controller. Consequently, this last methodology is followed to design simultaneously the membership functions and their rule-sets for the FLC.

Generally, designing a fuzzy logic controller involves two major steps; structure identification and parameter identification. Structure identification is the process of choosing a suitable controller structure, such as the size of the fuzzy rule-base. Parameter identification then determines the value of the parameters of a fuzzy controller, such as the shape of the fuzzy membership functions and the contents of the fuzzy rule-base. In the following section a demonstration is given of how to get over the step of parameter identification by using the GAs optimization approach, in a manner that optimal or near optimal fuzzy rules and membership functions can be selected without a human operator's experience or a control engineer's knowledge. The assumptions used and the constraints introduced to simplify this process are also explained.

### 4.1. Coding of the FLC parameters

By coding the coefficients of the membership functions and the fuzzy logic rule-set, FLC design can be developed and optimized by using GAs. The coded FLC design population can be found by the entire string termed "chromosome", each of which has randomly generated "bits", termed "genes". Then the GA process is used to reproduce

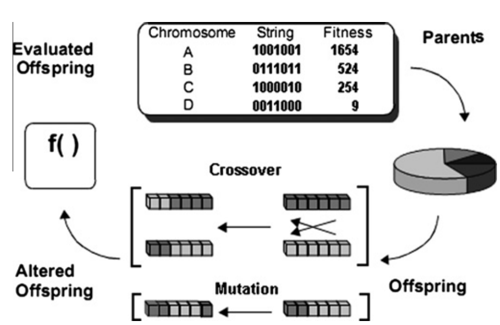


Fig. 4. Basic mechanism of genetic algorithms.

and select the "fittest" individual, i.e., the optimal solution of FLC design. To do this, either binary or real-valued coding can be used. In this case, the binary coding is chosen, where each parameter (rule & membership function) is converted into a binary string. These strings are concatenated and the genetic operations are performed on this concatenated string.

In the case of the triangular fuzzy sets, used in this work, three characteristic points (center and two widths) are generally used as the parameters to be coded (Timothy, 2004). Nevertheless, the number of these parameters can be reduced if certain constraints are imposed on the fuzzy set partition. Most fuzzy systems employ normalized fuzzy sets that require the membership values $\mu_{Ai}(x)$ of all fuzzy sets $A_i$ to sum up to unity, as shown by:

$$\forall\ x : \sum_{i}^{N} \mu_{Ai}(x) = 1 \tag{3}$$

This can literally be explained as a permission to overlap at most two active rules between adjacent fuzzy sets. It is therefore sufficient to define only the center points $C_1$, $C_2, \ldots, C_N$ of the normalized triangular membership functions, see Fig. 5, in order to specify the entire fuzzy partition of these variables.

In the present application, the domain intervals for input and output variables are $\{-35.9, 4.75\}$, $\{-40, 40\}$ and $\{-2, 2\}$. We note here that these intervals were obtained by calculating the maximum and the minimum values allowed for each used variable in our simulated environment for a stand-alone PV system. The four parameters for each input signal are encoded into six binary bits and those related to the output signal are also encoded into six binary bits for each parameter (see Fig. 6). If we consider that $X_i$ is the abscissa of $C_i$ (the center point of the $i$th fuzzy set), the decoding mapping (from binary to decimal) is obtained by using (Timothy, 2004):

$$X_i = X\min_i + \frac{b}{(2^L - 1)}(X\max_i - X\min_i) \tag{4}$$

where $b$ is the number in decimal form that is represented in binary form, $L$ is the length of the bit string (i.e., the number of bits in each string), and $X_{\max}$ and $X_{\min}$ are user-defined constants between which $X_i$ vary linearly. These parameters depend on the considered problem.

For the fuzzy rule-set, each fuzzy rule parameter is encoded into three binary bits that cover the range from

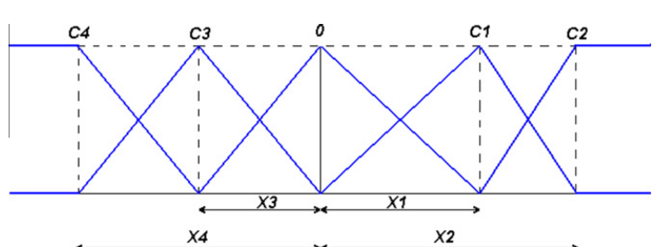


Fig. 5. Information which will be coded using binary coding ($X_1$, $X_2$, $X_3$, $X_4$).

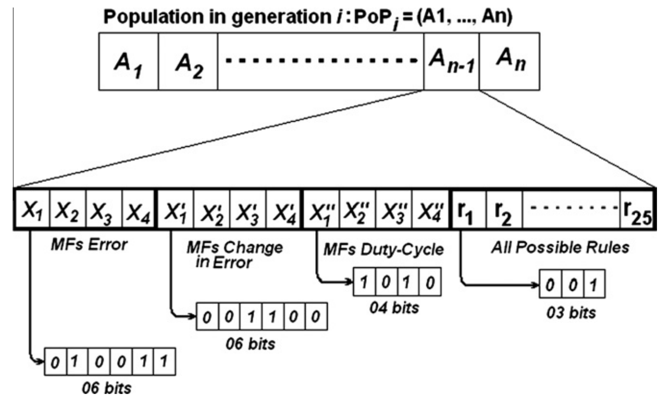


Fig. 6. Structure of the used chromosome with binary coding.

0 to 7. The rules considered here have 25 parameters that range over five fuzzy levels and are coded by: 0 = None (unused), 1 = NB (Negative Big), 2 = NS (Negative Small), 3 = ZE (Zero), 4 = PS (Positive Small), 5 = PB (Positive Big), 6 = None (unused), 7 = None (unused), (see Fig. 6).

The string produced by concatenating all the encoded parameters forms a genotype of $(25 \text{ rules} \times 3 \text{ bits})_{25\ \text{rules}} + (8\ \text{center-point-positions} \times 6\ \text{bits})_{\text{two inputs}} + (4\ \text{center-point-positions} \times 4\ \text{bits})_{\text{one output}} = 139$ bits. Each genotype specifies an individual member in the population. Evaluation of each string is based upon a fitness measure that is problem dependent.

### 4.2. Steps followed for the FLC design

The GA operation starts with a population of randomly generated solutions (chromosomes) and advances toward better solutions by applying the genetic operators. In each generation, relatively good solutions propagate to give offspring that replace the relatively inferior solutions. The fitness function plays the role of the environment in distinguishing between good and bad solutions. In order to find the optimum value for the adjustment factor, we can use the integral absolute error (IAE) performance index as the objective function, which is explained below. This performance index can estimate the dynamic and static characteristics of the control system synthetically. The procedure is as follows:

– *Generating the initial population*: ($N = 60$) sets of chromosomes are randomly generated before using a GA operation. These chromosomes are called the initial population.
– *Evaluation of the individual fitness*: For each individual chromosome (a complete string) in the population, it is necessary to establish a measure of its fitness, $f(x)$, that is often used to accurately evaluate the performance of the controller and will be used to generate a probability according to which the individual in question will be selected for reproduction or not. The task of defining the fitness function is always application specific. In this

paper, the objective of the controller is to drive the output to the desired set-point with a minimum overshoot and minimum settling time. Therefore, the fitness function of the GA for each individual is defined as follows:

The optimization done by GA is based on the minimization of integral absolute error (IAE) given by:

$$\mathrm{IAE} = \int_0^{\infty} |e(t)|dt \tag{5}$$

where $e(t) = P(t)_{\text{expect}} - P(t)_{PV}$

$P(t)_{expect}$ is the maximal theoretical delivered power at STP, i.e., $T = 25\,°\mathrm{C}$ and $G = 1000\ \mathrm{W/m^2}$ (For the units used-BP SX150S, $P(t)_{\text{except}} = 150\ \mathrm{W}$).

$P(t)_{PV}$ is the instant power provided by the PV module.

The fitness values are scaled so as to distinguish the individuals for which the fitness values are calculated.

We also use the fitness measure defined by:

$$\mathrm{Fitness} = 1000 - \mathrm{IAE} \tag{6}$$

One can see that since (IAE) is relatively small compared to 1000, so minimizing IAE given by Eq. (5) is equivalent to maximize fitness expression in Eq. (6). Fig. 7 illustrates the Matlab–Simulink model used to evaluate fitness.

– *Evaluating of the next generation or stop*: The operations of reproduction, crossover and mutation are used in order to generate the next generation. From generation to generation the maximum value of the fitness value is achieved. Table 1 summarizes the parameters used of the GA.

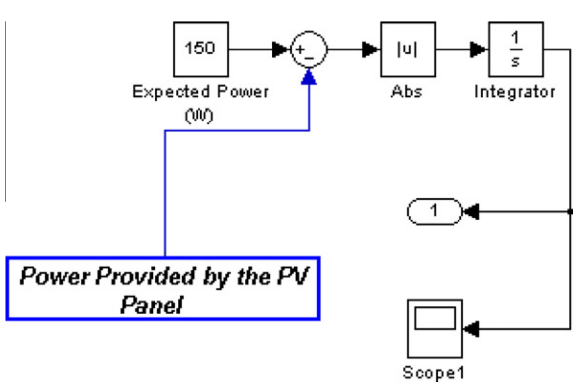


Fig. 7. Fitness evaluation with Matlab–Simulink model.

Table 1
Parameters of the genetic algorithm used.

| Parameter | Value |
|---|---|
| Representation | Binary |
| Chromosome size | 139 bits |
| Population size | 60 |
| Generations | 100 |
| Selection method | Roulette wheel |
| Rate of crossover | 0.8 |
| Mutation method | Gaussian |
| Rate of mutation | 0.03 |

### 4.3. Parameters of the optimal FLC obtained

Fig. 8 shows the evolution of highest and average fitness for generations 1–100. The results show that a better fitness value is achieved from generation to generation. The optimal chromosomes of the FLC were found at approximately generation number 50.

The optimal solution obtained, represented by the chromosome with the highest fitness in the last generation (100th generation), gives the shape of the membership functions as well as the table of the rule-sets shown in Fig. 9 and Table 2 respectively.

All these parameters are included in the solution surface shown in Fig. 10, which describes the dynamics of the optimized FLC. As it can be seen from this 3D curve, the entire output set surface is represented with respect to the entire span of the input sets.

## 5. Simulation results

In order to test the robustness of the designed fuzzy-based MPP tracker, the various parts of the *PV* system have been modeled by separate blocks using the Matlab–Simulink model shown in Fig. 11.

The main parts are:

- photovoltaic panel "BP SX150" – specifications shown in Table 3,
- the GA designed fuzzy-based MPPT,
- the PWM generator, and
- the DC/DC boost converter connected to a resistive load.

The first test to carry out, after the integration of the new designed FLC in the simulated stand-alone PV system, was the verification of the inter-compatibility of the new designed MPP tracker with the ubiquitous constituents of a standard stand-alone PV system (PV panel, DC/DC converter, PWM generator, etc.). Fig. 12 shows that the system

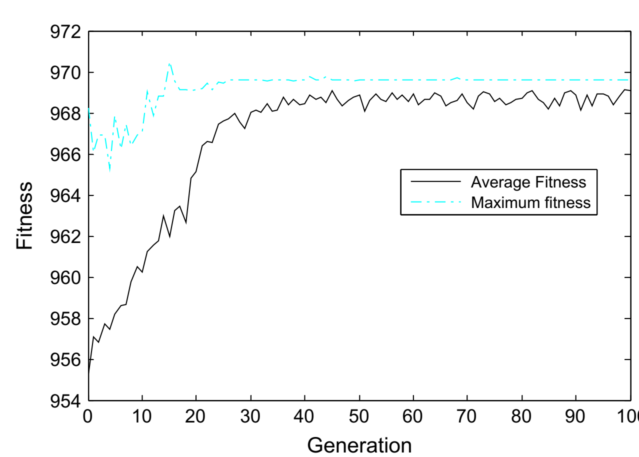


Fig. 8. Evolution of GA to evolve the FLC (average fitness, maximum fitness).

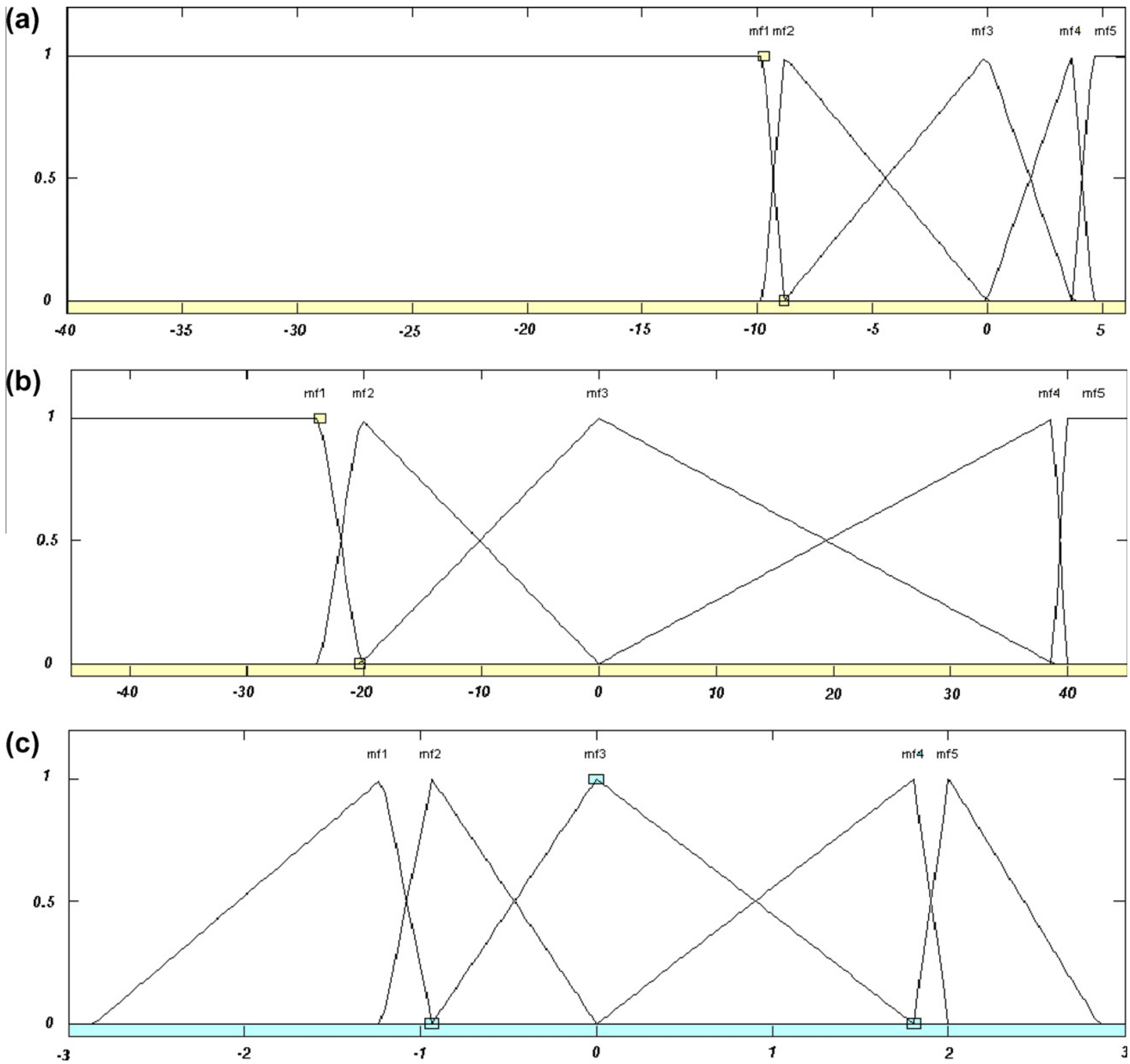


Fig. 9. Best membership functions obtained for the system variable error (a), Δerror (b), and D-cycle (c).

Table 2
Control rule table of the designed fuzzy controller.

| Output (*D*) | | Change in error | | | | |
|---|---|---|---|---|---|---|
| | | NB | NS | ZE | PS | PB |
| Error | NB | *PS* | *PS* | *PB* | *PS* | *PS* |
| | NS | *PS* | *PS* | *ZE* | *NS* | *ZE* |
| | ZE | *NS* | *ZE* | *ZE* | *NB* | *PB* |
| | PS | *ZE* | *ZE* | *NS* | *NS* | *PS* |
| | PB | *ZE* | *PS* | *NS* | *NS* | *PS* |

responds normally to an external excitation. The control and the duty cycle signals perform as in an ordinary tracking task.

The second characteristic, which is very important for every controller, is the controller's step response. For this reason, the GAs optimized FLC is also tested for rapid variation of solar irradiation parameter (from 0 to 1000 W/m$^2$ in 0.1 s – fast moving cloud cover). The results are shown in Fig. 13. As can be seen, the system exhibit a minimum degree of overshoot, undershoot and steady-state error. The magnitudes of the essential performance parameters under this dynamic condition are summarized in Table 4.

Additionally, when abrupt changes of the solar irradiation levels are present, the curves obtained, shown in Fig. 14, indicate that the GAs optimized FLC behave exactly as expected and almost exact MPPT is achieved in response to the considered variations.

The next simulation aim to compare the performance of the designed GA-FLC based tracker and the conventional P&O tracker. The signal representing the solar irradiation is turned on at $t = 0$ s to simulate the insolation power of 1000 W/m$^2$ and temperature of 25 °C. The MPP corre-

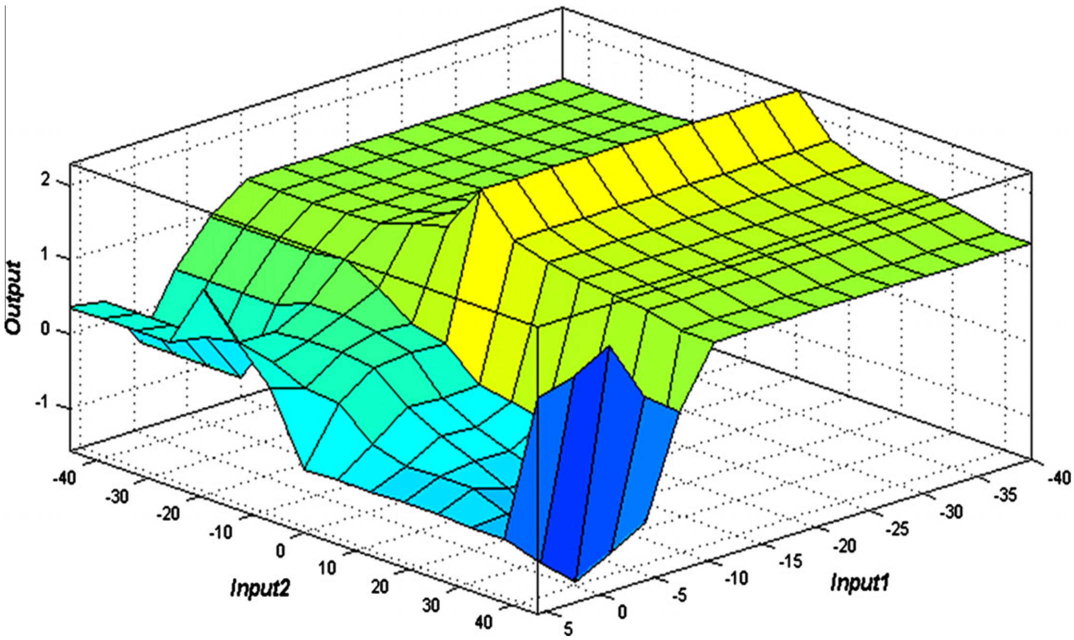


Fig. 10. Control surface for the fuzzy model found by the GA.

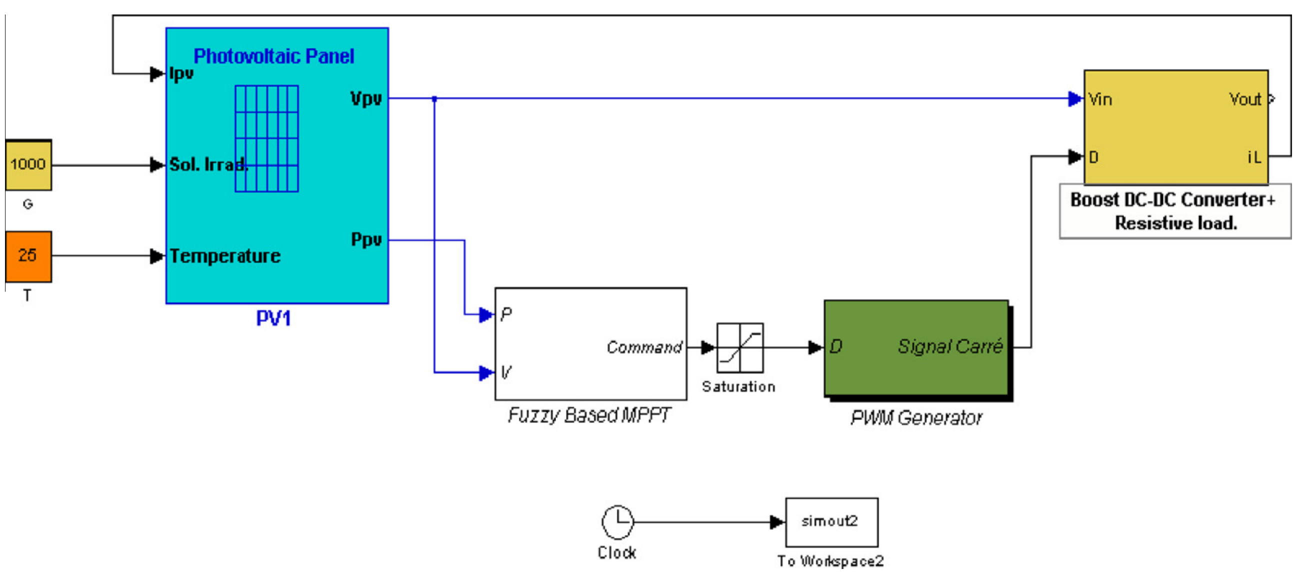


Fig. 11. Simulation of the fuzzy MPPT with resistive load.

Table 3
Specifications of the solar panels used in the simulation stage.

| Designation | BP SX150 (BP solar) |
|---|---|
| Maximum power ($P_{max}$) | 150 W |
| Voltage at $P_{max}$ ($V_{mp}$) | 34.5 V |
| Current at $P_{max}$ ($I_{mp}$) | 4.35 A |
| Warranted minimum ($P_{max}$) | 140 W |
| Short-circuit current ($I_{sc}$) | 4.75 A |
| Open-circuit voltage ($V_{oc}$) | 43.5 V |

sponding to this operating condition is about 150 W. Fig. 15 shows the result of the tracked power by the two controllers.

As it is clearly shown the optimized GA-FLC gives a fast response since it reaches its target at 0.723 s, while the P&O based tracker requires more time (1.752 s) to reach the same target. Furthermore, the response of the P&O based controller presents oscillations around the operating point even at steady state (see Fig. 15).

In the same way, the last simulation has been performed when both solar irradiation (changes from 800 W/m$^2$ to 1000 W/m$^2$ in 0.1 s and changes back to 800 W/m$^2$ in 0.1 s) and temperature ($T$) (rises from 18 °C to 35 °C in 0.1 s transition time) are changing. As it can be seen from Figs. 16a and 16b the effect of the simultaneous variation

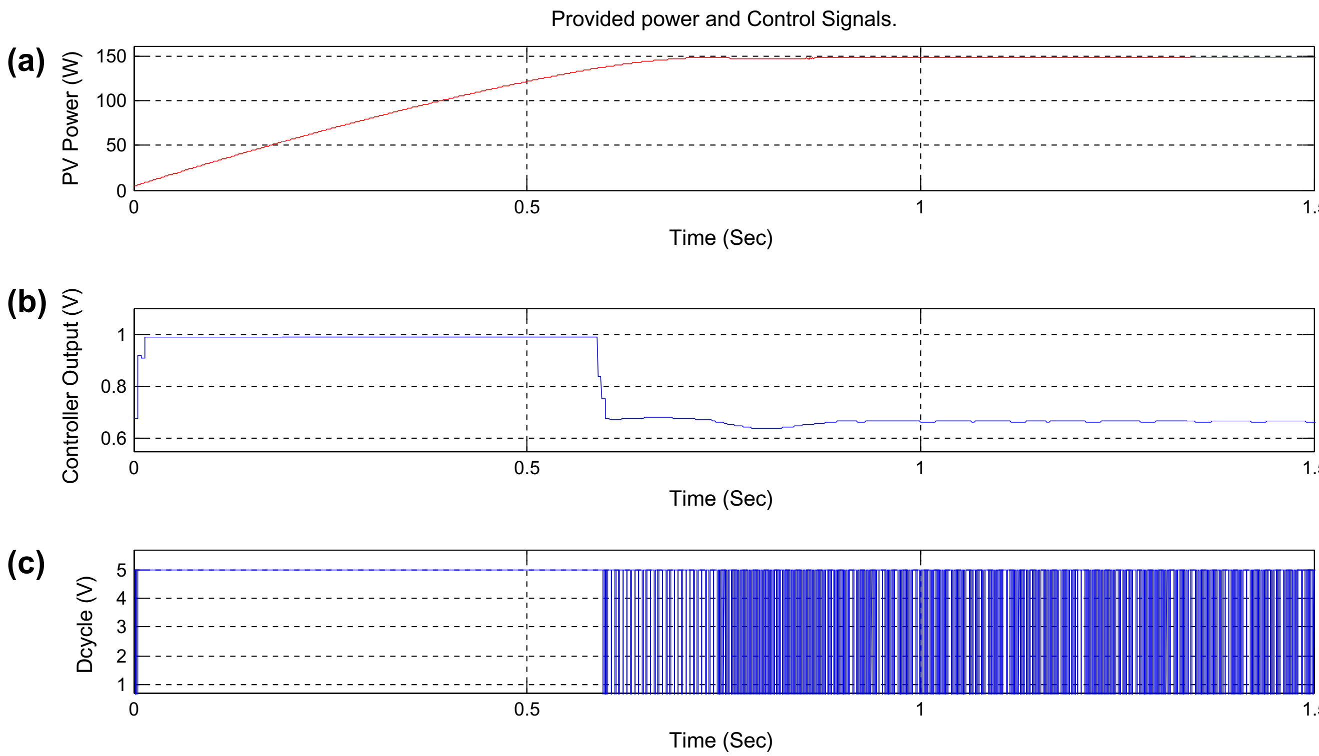

Fig. 12. (a) PV power, (b) control signal, (c) rectangular duty cycle signal.

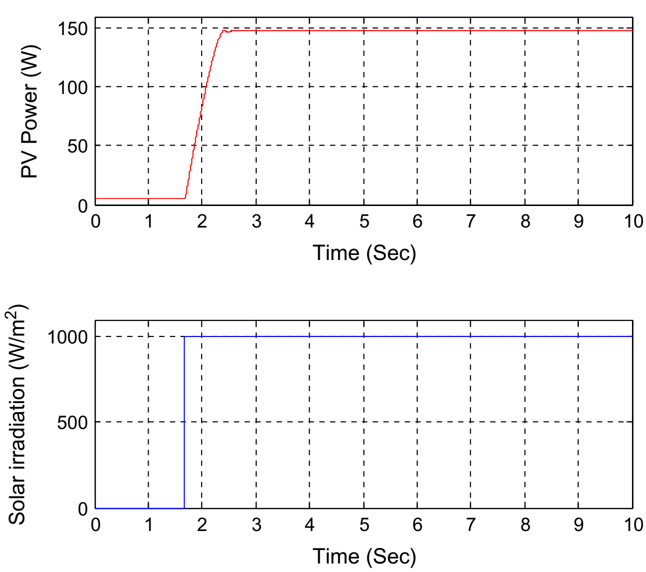

Fig. 13. Step responses of the optimized FLC based-GA.

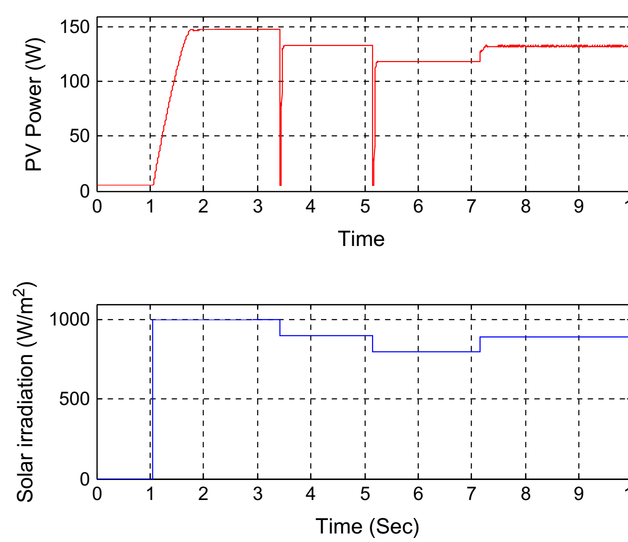

Fig. 14. MPPT response for different irradiation levels.

Table 4
Measured performance of the optimized FLC.

| Characteristic | Value |
|---|---|
| Overshoot | 0.004% |
| Rise time | 0.715 s |
| Steady-state error | 0.016 |

for theses atmospheric parameters influence correctly the response of the designed GA-FLC.

All the abovementioned simulations results show that the followed GA approach to optimize a specific FLC is efficient, reliable and effective for obtaining an optimal MPP tracker.

## 6. FPGA implementation of the optimized FLC based GA

The motivation behind the implementation of the MPP tracker presented in previous sections, is the application in a FPGA chip. This was driven by the need to create an

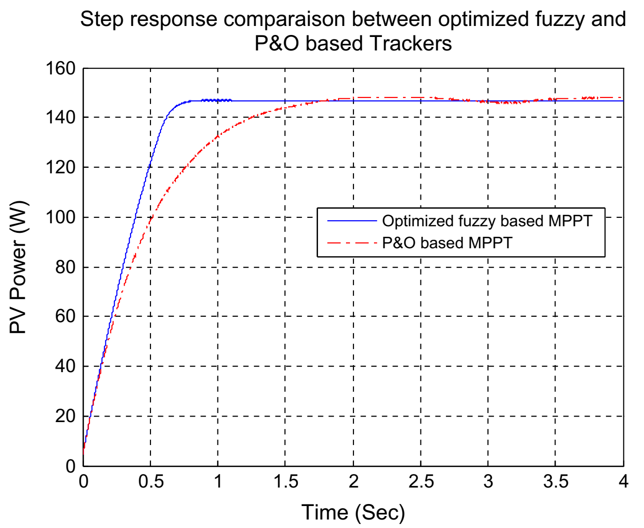


Fig. 15. Step response comparison between P&O and the optimized GA-FLC MPPT (BP solar 150; 25 °C & 1000 W/m$^2$).

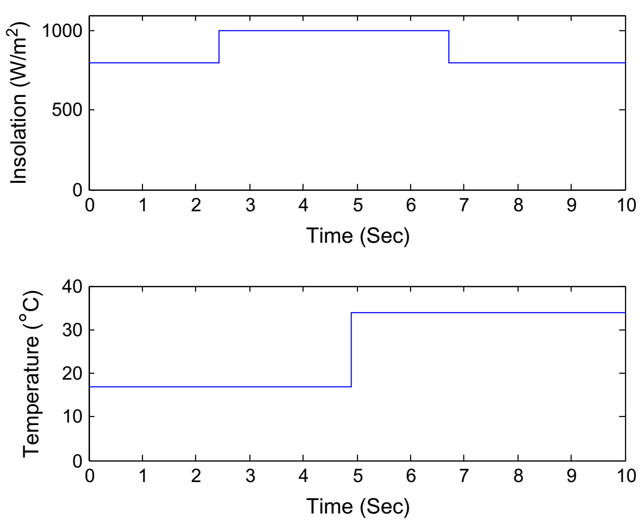


Fig. 16a. The change of solar irradiation and temperature in simulation.

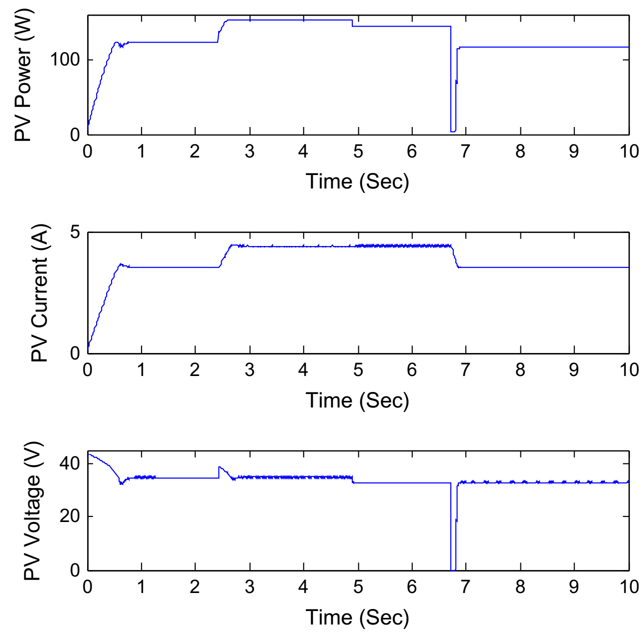


Fig. 16b. The current, voltage and output power of PV array based on GAs tuned fuzzy MPPT.

inexpensive hardware implementation for use in stand-alone PV systems placed in remote rural areas. To do this, a Virtex-II V2MB1000 Development Kit (from Memec design) has been employed. This provides a complete solution for developing designs and applications based on the Xilinx Virtex 2 FPGA family. The controller was designed using Very high speed integrated circuit Hardware Description Language (*VHDL*) integrated with the *Xilinx* Foundation ISE 7.1i tools. The *ModelSim Xilinx* Edition-III (MXE-III) v6.0a was used for the simulation purposes.

To get the benefits from FPGA solutions for the implementation of FLCs, we have used the well known functional description approach (Deliparaschos et al., 2006) which increase the fuzzy controller performance as well as the direct access to fuzzy membership, simultaneous rules activation and implementation of arithmetic functions.

The block diagram of the FPGA implemented MPPT is shown in Fig. 17. As can be seen the MPPT includes four principal units:

(1) fuzzification unit which convert a crisp input into a fuzzy term set,
(2) a rule selector unit which stores fuzzy rules describing how the fuzzy system performs,
(3) an inference engine unit which performs approximate reasoning by associating input variables with fuzzy rules, and finally
(4) the defuzzification unit which convert the FLC's fuzzy output to a crisp value representing the control action.

In addition to the standard units, shown in Fig. 17, we have also used an input pre-calculation unit, which provides the real inputs to the controller, the error and its change depicted as $E$ and $\Delta E$ respectively. These errors are calculated according to the expressions given by Eqs. (1) and (2). A sequencer is also used, operating as a manager of the control signals which synchronize the tasks of all units quoted above. This sequencer, named as control unit in Fig. 17, is driven by a frequency divider unit used to adapt the frequency of the FPGA board and the sampling rate for the considered process. The FLC's output, i.e., the crisp value is used to drive a simple PWM generator also implemented on the same chip.

The different sub-units described above were implemented separately on a Virtex II (XC2v1000-4fg456) FPGA chip from Xilinx, shown in Fig. 18. This chip was largely sufficient to implement all the constituents of the MPPT controller addressed in this work as it contains

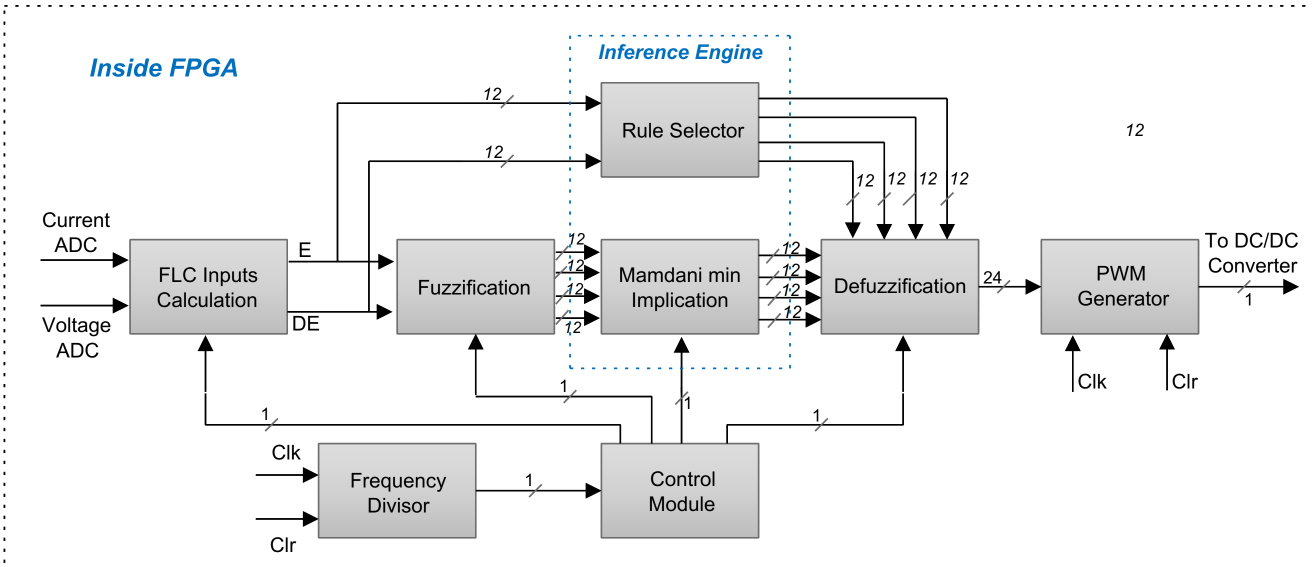


Fig. 17. Block diagram of the FPGA implemented MPPT.

5120 slices and 10,240 logic cells as well as forty 18 × 18 multipliers. Table 5 shows the FPGA logic resources used to develop the controller.

To evaluate the MPPT performance a test is carried out, related to the examination of the implemented controller's robustness with respect to the rapid change of the solar irradiation. A sample from the curves obtained at the end of this operation is illustrated in Fig. 19, from which it is clearly shown that the controller ensures fast convergence and robust performance against rapid solar irradiation variations. The system stabilizes after a relatively short time and seeks the maximum power transfer in all operating conditions.

The simulation was carried out in a manner such that to each value of the duty cycle ratio ($D$) the signal generated at the output of the controller, corresponds a single point

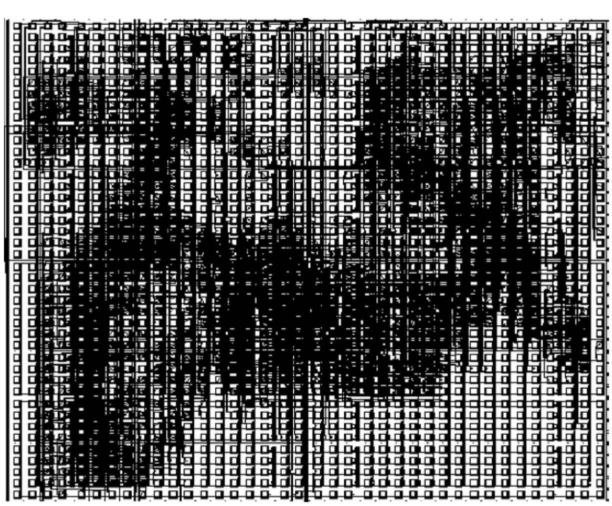

Fig. 18. Floor planning of the FPGA implementing the MPPT controller and interface circuits (the used space in the FPGA).

Table 5
Device utilization summary.

| Selected device | xc2v1000 |
|---|---|
| Number of slices | 1964 out of 5120, 38% |
| Number of slice flip flops | 2668 out of 10,240, 26% |
| Number of 4 input LUTs | 1928 out of 10,240, 18% |
| Number of bonded IOBs | 27 out of 324, 8% 32 |
| Number of MULT18X18s | 9 out of 40, 22% |
| Number of GCLKs | 4 out of 16, 25% |
| Maximum frequency | 97.040 MHz |

with ($I$, $V$) coordinates on the memorized $I$–$V$ characteristic of the considered PV panel (BP SX150). The digitized values representing current and voltage respectively, were supplied into the two inputs of the controller in fixed time periods (10 ms in our case). During each acquisition period, the controller generates the corresponding new duty cycle ratio ($D$) signal which sets the new position of the tracked MPP and vice versa.

The results presented here have shown that the advantages of the system developed are the adaptation of the GA-FLC parameters for fast response, good transient performance, and robustness to variations in external disturbances.

## 7. Conclusion and perspectives

The application of genetic algorithms to fuzzy logic controllers design holds a great deal of promise in overcoming two of the major problems in fuzzy controller design; design time and design optimization. As it is shown, they were successfully used in this work, to improve the performance of a fuzzy logic-based MPPT controller by optimizing simultaneously both the membership functions and the fuzzy control rules. From the simulation results, it is

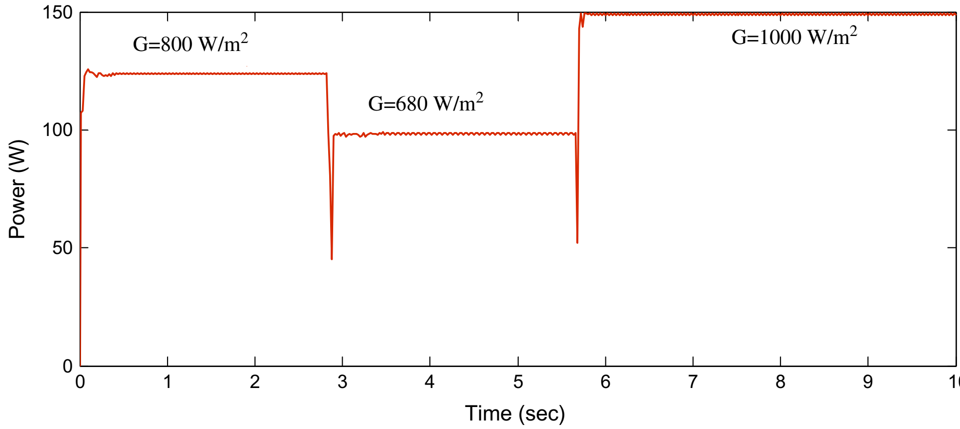


Fig. 19. The evolution of the MPP-based optimized FLC for different solar irradiation values (800, 680 and 1000 W/m$^2$, 25 °C) implemented on FPGA board.

concluded that the performance of the new GA-FLC-based MPPT is better than the ones obtained with classical P&O controller, since the response time in the transitional state is shortened and the fluctuations in the steady state are considerably reduced. In addition to the software simulations, an FPGA chip was also used to implement the hardware prototype of the whole MPP tracker including the designed GA-FLC.

This work is the first step before a final field installation to experimentally validate the effectiveness of the proposed GA-FLC in a stand-alone photovoltaic system.

## Acknowledgments

The second author would like to thank the International Centre for Theoretical Physics (ICTP), Trieste (Italy) for providing the materials and the computer facilities for performing the present work. This work was also supported by the Ministry of Higher Education & Scientific Research (Algiers) under Project Number: J0201720080012.

## References

Akkaya, R., Kulaksız, A.A., Aydoğdu, Ö., 2007. DSP implementation of a PV system with GA-MLP NN based MPPT controller supplying BLDC motor drive. Energy Conversion and Management 48 (1), 210–218.

Altas, I.H., Sharaf, A.M., 2008. A novel maximum power fuzzy logic controller for photovoltaic solar energy systems. Renewable Energy 33 (3), 388–399.

Bahgat, A.B.G., Helwa, N.H., Ahmad, G.E., El Shenawy, E.T., 2005. Maximum power point tracking controller for PV system using neural networks. Renewable Energy 30, 1257–1268.

Calais, M., Hinz, H., 1998. A ripple-based maximum power point tracking algorithm for a single-phase, grid-connected photovoltaic system. Solar Energy 63 (5), 277–282.

Chu, C., Chen, C., 2009. Robust maximum power point tracking method for photovoltaic cells: a sliding mode control approach. Solar Energy 83 (8), 1370–1378.

Deliparaschos, K.M., Nenedakis, F.I., Tzafestas, S.G., 2006. Design and implementation of a fast digital fuzzy logic controller using FPGA technology. Journal of Intelligent and Robotic Systems 45, 77–96.

Eakburanawat, J., Boonyaroonate, I., 2006. Development of a thermoelectric battery–charger with microcontroller-based maximum power point tracking technique. Applied Energy 83 (7), 687–704.

Enrique, J.M., Duran, E., Cardona, M.S., Andujar, J.M., 2007. Theoretical assessment of the maximum power point tracking efficiency of photovoltaic facilities with different converter topologies. Solar Energy 81, 31–38.

Feel-soon Kang, F., Cho, S.E., Park, S., Kim, C., Ise, T., 2005. A new control scheme of a cascaded transformer type multilevel PWM inverter for a residential photovoltaic power conditioning system. Solar Energy 78 (6), 727–738.

Godoy, M.S., Franceschetti, N.N., 1999. Fuzzy optimization based control of a solar array system. IEE Proceedings – Electric Power Applications 146 (5), 552–558.

Goetzberger, A., Hoffmann, V., 2005. Photovoltaic Solar Energy Generation. Springer, Germany.

Gounden, N.A., Peter, S.A., Nallandula, H., Krithiga, S., 2009. Fuzzy logic controller with MPPT using line-commutated inverter for three-phase grid-connected photovoltaic systems. Renewable Energy 34, 909–915.

Hohm, D.P., Ropp, M.E., 2003. Comparative study of maximum power point tracking algorithms. Progress in Photovoltaics: Research and Application 11, 47–62.

Homaifar, A., McCormick, E., 1995. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Transactions on Fuzzy Systems 3 (2), 129–139.

Jiménez, C.J., Sànchez Solano, S., Barriga, A., 1995. Hardware implementation of a general purpose fuzzy controller. IFSA'95, pp. 185–188.

Karr, C.L., Gentry, E.J., 1993. Fuzzy control of pH using genetic algorithms. IEEE Transactions on Fuzzy System 1 (1), 46–53.

Khaehintung, N., Sirisuk, P., 2004. Implementation of maximum power point tracking using fuzzy logic controller for solar-powered light-flasher applications. The 47th IEEE International Midwest Symposium on Circuits and Systems, Hiroshima, July 25–28, pp. 171–174.

Khaehintung, N., Pramotung, K., Tuvirat, B., Sirisuk, P., 2004. RISC-microcontroller built-in fuzzy logic controller of maximum power point tracking for solar-powered light-flasher applications. In: ICON The 30th Annual Conference of the IEEE Industrial Electronics Society, pp. 2673–2678.

Khaehintung, N., Wiangtong, T., Sirisuk, P., 2006. FPGA implementation of MPPT using variable step-size P&O algorithm for PV applications. IEEE International Symposium on Communication and Information, IEEE-ISCIT'06, pp. 212–215.

Kida, J., Tokuda, K., Ishihara, Y., Todaka, T., 1991. Analysis of DC–DC converter for the maximum power point control of photovoltaic. In: INTELEC'91, IEEE Proceedings, pp. 291–295.

Kobayashi, K., Matsuo, H., Sekine, Y., 2006. An excellent operating point tracker of the solar-cell power supply system. IEEE Transactions on Industrial Electronics 53 (2), 495–499.

Larbes, C., Aït Cheikh, S.M., Obeidi, T., Zerguerras, A., 2009. Genetic algorithms optimized fuzzy logic control for the maximum power point tracking in photovoltaic system. Renewable Energy 34, 2093–2100.

Linkens, D.A., Nyongesa, H.O., 1995. Genetic algorithms for fuzzy control. Part 1: offline system development and application. IEEE Proceedings Control Theory and Applications 142 (3), 161–176.

Mekki, H., Mellit, A., Kalogirou, S.A., Messai, A., Furlan, G., 2010. FPGA-based implementation of a real time photovoltaic module simulator. Progress in Photovoltaics: Research and Application 18, 115–127.

Mellit, A., Kalogirou, S.A., 2008. Artificial intelligence techniques for photovoltaic applications: a review. Progress in Energy and Combustion Science 34, 574–632.

Mellit, A., Rezzouk, H., Messai, A., Medjahed, B., 2010a. FPGA-based real time implementation of MPPT controller for photovoltaic systems. Renewable Energy. doi:10.1016/j.renene.2010.11.019.

Mellit, A., Mekki, H., Messai, A., Salhi, H., 2010b. FPGA-based implementation of an intelligent simulator for stand-alone photovoltaic system. Expert Systems with Applications 37, 6036–6051.

Salas, V., Olias, E., Barrado, A., Lazaro, A., 2006. Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems. Solar Energy Mater & Solar Cells 90, 1555–1578.

Timothy, J. Ross, 2004. Fuzzy Logic with Engineering Applications, second ed. John Wiley & Sons Ltd.

Veerachary, M., Senjyu, T., Uezato, K., 2003. Neural-network-based maximum-power-point tracking of coupled-inductor interleaved-boost-converter-supplied PV system using fuzzy controller. IEEE Transactions on Industrial Electronics 50 (4), 749–758.

Won, C.Y., Kim, D.H., Kim, S.C., 1994. A new maximum power point tracker of photovoltaic array using fuzzy controller. In: Proceedings of 25th Annual IEEE Power Electronics Specialists Conference, PESK'94, pp. 396–403.

Xiao, W., Dunford, W.G., 2004. A modified adaptive hill climbing MPPT method for photovoltaic power systems. IEEE Power Electronics Specialists Conference 3, 1957–1963.