

Research Article

Hardware Implementation of a Fuzzy Logic Controller for a Hybrid Wind-Solar System in an Isolated Site

Aymen Jemaa ¹, Ons Zarrad,¹ Mohamed Ali Hajjaji ^{2,3} and Mohamed Nejib Mansouri¹

¹Unit of Industrial Systems Study and Renewable Energy (ESIER), National Engineering School, University of Monastir, Monastir, Tunisia

²Laboratory of Electronic and Microelectronic, University of Monastir, Monastir, Tunisia

³Higher Institute of Applied Sciences and Technology of Kairouan, University of Kairouan, Kairouan, Tunisia

Correspondence should be addressed to Aymen Jemaa; jemaa_aymen@yahoo.fr

Received 1 November 2017; Revised 19 March 2018; Accepted 19 April 2018; Published 4 July 2018

Academic Editor: Joaquín Vaquero

Copyright © 2018 Aymen Jemaa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, two main contributions are presented to manage the power flow between a wind turbine and a solar power system. The first one is to use the fuzzy logic controller as an objective to find the maximum power point tracking, applied to a hybrid wind-solar system, at fixed atmospheric conditions. The second one is to respond to real-time control system constraints and to improve the generating system performance. For this, a hardware implementation of the proposed algorithm is performed using the Xilinx System Generator. The experimental simulation results show that the suggested system presents high accuracy and acceptable execution time performances. The proposed model and its control strategy offer a proper tool for optimizing the hybrid power system performance which we can use in smart house applications.

1. Introduction

Wind and solar energies present the most famous renewable energy sources which attract attention due to the decreasing fossil fuel reserves and environmental property impact. The use of wind energy systems may not be technically viable at all sites because of low wind speeds and being more unpredictable than solar energy [1, 2]. The hybridization of these renewable energy sources can ensure the continuity of energy production and can be an economic solution for countries to develop and decrease the consumption of fossil (fuel and nuclear) energy sources. Research and development efforts in solar, wind, and other renewable energy technologies are necessary to continue improving their performance. Among the most important factors to consider, we can cite the maximum power point tracking (MPPT), which presents an essential factor of system performances.

In general, MPPT methods can be classified into two principal categories. The first one uses classic algorithms

such as hill-climbing (HC), perturbation and observation (P&O), and incremental conductance (IncCond). For the second type, it is based on intelligent methods such as fuzzy logic, artificial neural network (ANN), and neurofuzzy, which is a combination between the two previous methods.

The idea to use the fuzzy logic controller is to control, respectively, the proportional-integral controller for wind turbines and the duty cycle (d) for solar energy to regularize successively the optimal rotor speed and the pulse-width modulation in the boost converter. This algorithm does not require a specific detailed mathematical model or linearization, about an operating point, and it is independent from system parameter variations.

For a subsystem wind turbine, the pitch angle of the turbine is synchronized according to the measured wind speed values in a fuzzy logic controller. For the solar subsystem, the impedance of the photovoltaic cell's output is equal to the values of the impedance measured on the load impedance in the fuzzy logic controller. Both controllers are applied to boost the performance. The control of the

two subsystems presents one of the main objectives of this manuscript.

To get the optimum performance and effective power of the wind-solar turbine at fixed atmospheric conditions, a second objective is studied, which is the implementation of both controllers. This implementation is designed on Xilinx, which has the Xilinx System Generator (XSG) tools to facilitate the design. The implementation on a field-programmable gate array (FPGA) circuit with XSG allows us to find the solution of complexity, parallelism, and calculation time.

The XSG handles much of the routing and placement timing. The FPGA design flow eliminates the complex and time-consuming floor planning, placement and routing, and timing analysis. The FPGA runs up to 500 MHz with a superior performance. The unprecedented logic density increases, and a host of other features like embedded processors, DSP blocks, clocking, and high-speed serial raise the performance.

This paper is planned on six sections: The first section is an introduction which contains a generality of a wind-solar system and the purpose of this work. Section 2 is the related work and the model description of the hybrid system components. Section 3 describes the XSG tools, the conception, and the hardware architecture of the different blocks of a hybrid system with an XSG design. Section 4 contains the results and discussions of the simulation. The results of a fuzzy logic controller implementation on FPGA are given in Section 5 before the conclusion which is the final part of this paper.

2. Related Work and Proposed Hybrid-System Model

2.1. Related Work. Tracking the various MPPT algorithms permits fixing and extracting the maximum power, which has been dealt with in the literature.

In [3], a photovoltaic- (PV-) wind system was controlled and compared with two approaches. The first one was a hybrid system with HC and P&O algorithms. The second one used a fuzzy logic controller. The based power generation of the system was 3.1 kW.

In [4], two connection models of a solar-wind system were studied with fuzzy logic controllers. The first model had just one hybrid system connected to the main grid. The second one had the same system with a 75 kW load connected to the grid.

In [5], a standalone PV-wind energy conversion system was sized and optimized. P&O and fuzzy logic algorithm controllers were used to get the optimum mechanical speed of turbine and duty cycles of a DC-DC converter.

2.2. Proposed Solar-Wind System. In this paper, a hybrid power system is designed with two renewable energy sources, solar and wind, which are connected with a conventional energy source. A wind system with a solar photovoltaic one is the best hybrid combination of all renewable energy systems and is suitable for most applications [6]. Furthermore, the system is dedicated to isolates sites that are not

connected to the grid, so the only energy source is the hybrid power system.

Figure 1 describes the schematic diagram of the wind-solar hybrid energy conversion system studied in this paper. Its principal blocks are a PV generator, a wind turbine, a permanent magnet synchronous generator (PMSG), a rectifier, boost converters, a continuous DC-DC bus, and fuzzy logic controllers.

2.3. Wind Turbine Model. Depending on the aerodynamic characteristics, the wind turbine mechanical power is given by

$$P = \frac{1}{2} \pi \rho C_p(\lambda, \beta) R^2 V_v^3, \quad (1)$$

where ρ is the air density, C_p is the power coefficient, λ is the tip-speed ratio, β is the pitch angle, R is the turbine radius, and V_v is the wind speed.

The form of the used power coefficient C_p is given in

$$C_p(\lambda) = -0.2121\lambda^3 + 0.0856\lambda^2 + 0.2539\lambda. \quad (2)$$

It depends only on one variable, which is the tip-speed ratio λ . The pitch angle β is usually the angle between the turbine blades and its longitudinal axis. In this work, β is set to zero. The tip-speed ratio λ is considered as the linear speed form of the rotor to the wind speed. The power value extracted from the wind turbine system will be at its utmost when the power coefficient C_p is at its maximum at a defined value of the tip-speed ratio λ .

Accordingly, for each wind speed, there is an optimum rotor speed value where the maximum power is extracted from the wind. Therefore, we can say that C_p is maximal at a particular λ_{opt} . The expression of the tip-speed ratio λ is presented in

$$\lambda = \frac{\Omega R}{V_v}, \quad (3)$$

where Ω is the turbine's angular speed.

The variable-speed wind form studied in this work is expressed in (4) and illustrated in Figure 2.

$$V_v(t) = 7.5 + 0.2 \sin(0.1047t) + 2 \sin(0.2665t) + \sin(1.2930t) + 0.2 \sin(3.6645t). \quad (4)$$

The analyzed wind turbine and PMSG have their electric specifications given, respectively, in Tables 1 and 2.

Figure 3 presents the plot of the $C_p(\lambda)$ characteristic where a maximum point wind speed with the coordinate (C_{pmax}, λ) is equal to (0.15, 0.78).

2.4. Permanent Magnet Synchronous Generator (PMSG). To define the PMSG model, we use the following simplifying assumptions [7]:

- (i) The stator is connected on a star and is neutral in the air to eliminate the homopolar component of currents.

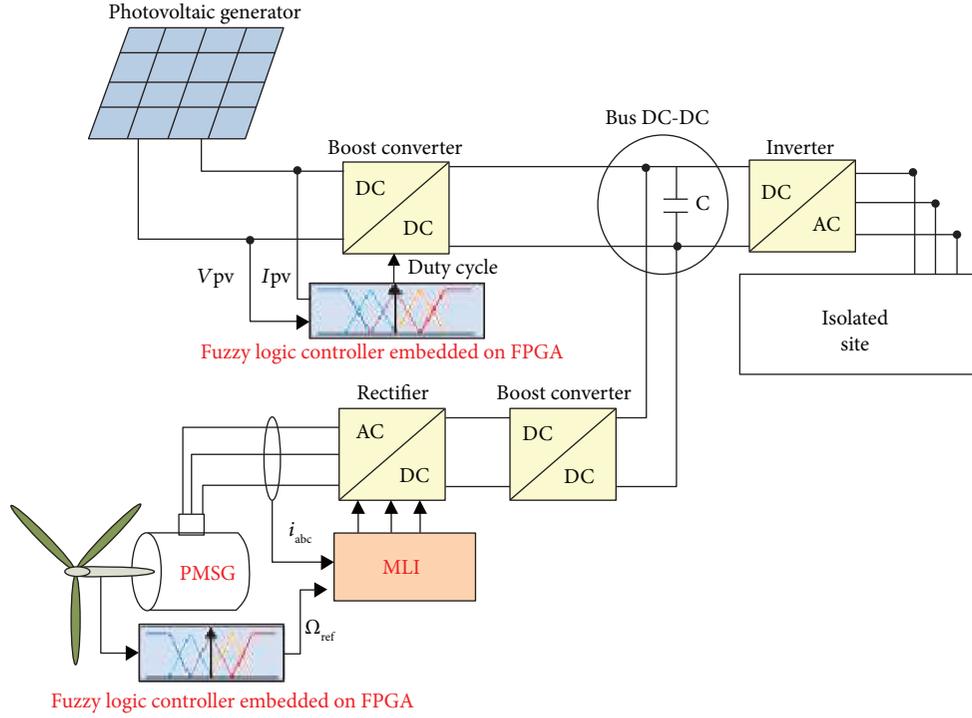


FIGURE 1: Hybrid wind-solar energy conversion system for an isolated site.

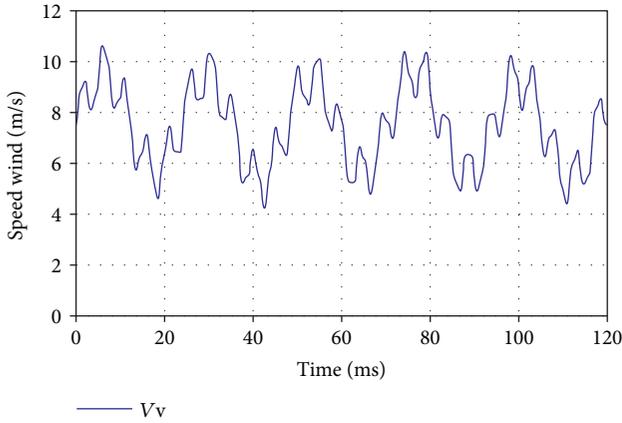


FIGURE 2: Allure of variable-speed wind.

TABLE 1: Wind turbine specifications.

Air density " ρ "	1.2 kg/m ³
Turbine radius " R "	0.5 m
Turbine height " H "	2 m
Inertia constant " J "	16 kg/m ²
Friction factor " f "	0.01 kg-m/rad
Maximum coefficient of power " C_{pmax} "	0.15
Optimal tip-speed ration " λ_{opt} "	0.78

TABLE 2: PMSG specifications.

Number of rotor pole pairs	2
Stator phase resistance	0.137 Ω
Stator phase inductance	0.0027 H
Inertia constant	0.1 kg-m ²
Friction factor	0.06 kg-m/rad

- (ii) The saturation of the magnetic circuit is neglected, which leads to expressing the magnetic fluxes as linear functions of the phase currents.
- (iii) The distribution of the electromotive force (f.e.m) in the air gap is sinusoidal, and the harmonics of space are then neglected.
- (iv) The variation in resistance as a function of temperature is neglected.
- (v) Hysteresis and current losses are neglected.

The generator is modeled by the following voltage equations given by (5) in the rotor reference frame (d, q) axes.

$$\begin{aligned}
 V_d &= R_s i_d + L_d \frac{di_d}{dt} - p\Omega L_q i_q, \\
 V_q &= R_s i_q + L_q \frac{di_q}{dt} + p\Omega L_d i_d + p\Omega \phi_f,
 \end{aligned} \tag{5}$$

where $i_d, i_q, v_d,$ and v_q are the currents and voltages, respectively; L_d and L_q are the equivalent stator inductances in the (d, q) axes, respectively; R_s is the stator resistance; $\omega = p\Omega$

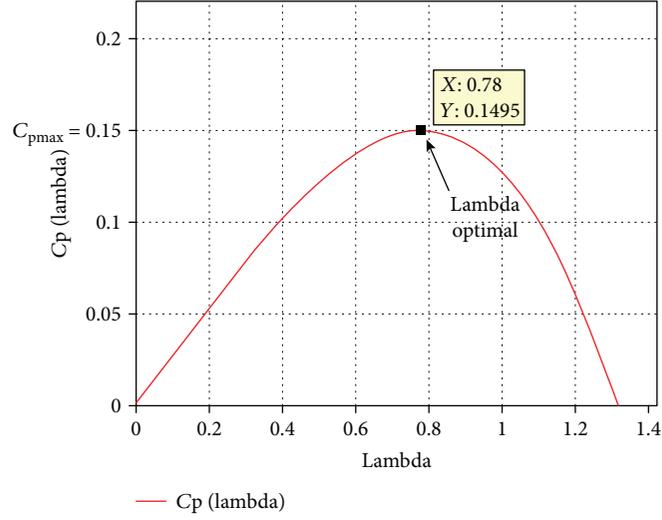


FIGURE 3: Characteristic curve of $C_p(\lambda)$ in MATLAB/SIMULINK environment.

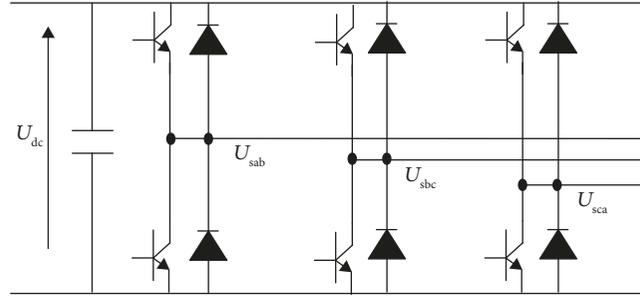


FIGURE 4: PWM rectifier circuit diagram.

is the electric frequency related to the mechanical speed, and ϕ_f is the permanent magnetic flux produced by the rotor magnets.

The electrical torque (T_e) applied to the PMSG rotor can be expressed by

$$T_e = p i_q ((L_d - L_q) i_d + \phi_f), \quad (6)$$

where p is the pair pole number of the machine. The magnetic flux is a constant that depends on the material used for the realization of the magnets.

The mechanical expression is represented by

$$J \frac{d\Omega}{dt} + f\Omega = T_e - T_r \quad (7)$$

where J is the value of the total inertia of the rotor, f is the viscous friction coefficient, and T_r is the electromagnetic torque.

2.5. Pulse-Width Modulation. Pulse-width modulation (PWM) is a static converter used to rectify an alternating signal and then transform it into a continuous signal. In order to reduce the simulation time and simplify the modeling, the rectifier is modeled by ideal switches where there is a

zero resistance in the on state, an infinite resistance in the off state, and an instant response to control signals.

PWM is composed of six components; each one comprises two switching cells consisting of a transistor and a diode, as depicted in Figure 4. Accordingly, the current passes in both directions.

The main function of the PWM switches is to provide the connection between the AC voltage produced by the wind subsystem and the DC bus.

The switches' states are complementary, as defined by

$$\begin{aligned} S &= 0 & \text{if } i_j &= +1, \\ S &= 1 & \text{if } i_j &= -1, \end{aligned} \quad (8)$$

$j = a, b, c.$

In general, the voltage equation of the outputs can be written in the form expressed by

$$U_n = U_{dc} \left(S_n - \frac{1}{3} \sum_{n=a}^c S_n \right), \quad (9)$$

where S_n is equal to 0 or 1 depending on the state of the switches, and n is a, b , or c .

The equations of voltages for the balanced three-phase system without a neutral are given by

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = R \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + L \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix}, \quad (10)$$

where e_n and i_n are, respectively, the voltages and currents of the inputs a , b , and c .

2.6. Solar Model. The solar cells can be divided into two types, bulk and thin film [2, 8–11]. Solar cells are made of different materials and have different efficiency values. Based on cost and generation efficiency, we choose monocrystalline silicon for our work among the different materials provided in Table 3.

The characteristic I - V for a PV module is given by

$$\begin{aligned} I = n_p I_{PV} \\ - n_p I_0 \left(\frac{T_c}{T_{ref}} \right)^3 e^{(qE_g/ak) \left((1/T_{ref}) - (1/T_c) \right)} \left[e^{(q(VI+IR_S)/akt_c n_s) - 1} \right] \\ - \frac{V + IR_S}{R_p}, \end{aligned} \quad (11)$$

where I_0 is the reverse saturation current of the diode, E_g is the value of the energy band of the diode material, a is the ideality factor of the diode, k is the Boltzmann's constant, q is the electron charge, and T_c is the cells' temperature in Kelvin.

The analyzed PV generator has the electric specifications given in Table 4.

2.7. Boost DC-DC. Generally, the objective of the DC-DC boost converter is to help the power renewable energy voltage [13], which is in our work the solar and wind energies, to converge the reference value provided by the MPPT algorithm. Perturbing the duty ratio of a PV generator signal, which is fed into the converter, can minimize the proportion between the input voltage and the output desired voltage. The general topology of the boost DC-DC converter is shown in Figure 5.

The equation system of the boost DC-DC converter is done by

$$\begin{aligned} i_L &= i - c_1 \frac{dv_i}{dt}, \\ i_0 &= (1-d)i_L - c_2 \frac{dv_0}{dt}, \\ v_i &= (1-d)v_0 + L \frac{di_L}{dt}. \end{aligned} \quad (12)$$

We use two DC-DC boost converters in the output of the solar and wind subsystems to rebuild optimized voltages. The currents of the DC-DC boost converters are transmitted to the DC-DC bus.

TABLE 3: Comparison of solar cell efficiency [10].

Solar cell	Monomer efficiency (%)	Module efficiency (%)
Monocrystalline silicon	22	10–15
Polycrystalline silicon	18	9–12
Boron-phosphorus compound	30	17
Thin-film amorphous silicon	13	10
Thin-film Cu-In	19	12
Thin-film Cd-Te compound	16	9

TABLE 4: Photovoltaic generator specifications [12].

Rated power	60 W
Current at maximum point	3.25 A
Voltage at maximum point	16.8 V
Short circuit current	3.56 A
Short circuit voltage	21.6 V
Number of cells in parallel	1
Number of cells in series	36

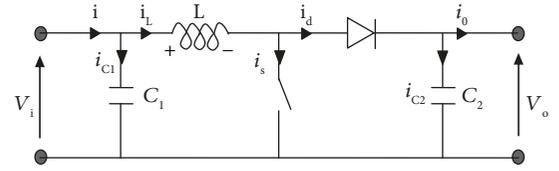


FIGURE 5: Circuit diagram of boost converter.

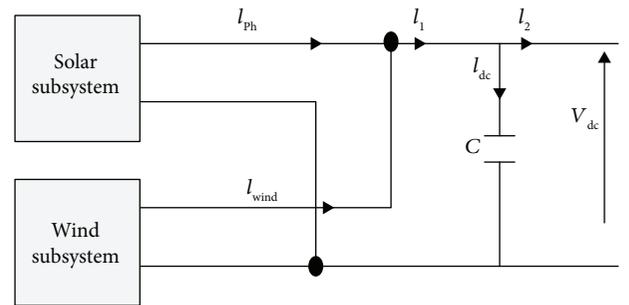


FIGURE 6: Electrical diagram of the continuous bus.

2.8. DC-DC Bus. The coupling of the two subsystems (solar and wind) is provided via a continuous bus. As illustrated in Figure 6, the DC bus is represented by the capacitor C connected to both subsystems. The role of the C capacitor integrated in the bus ensures the regulation of the ripple.

In our system, the photovoltaic energy is connected to the load via an MPPT-controlled DC-DC converter. The wind energy is connected to the load by the controlled PWM rectifier.

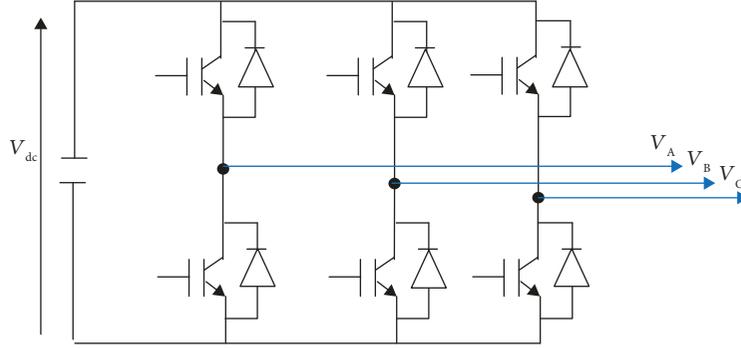


FIGURE 7: Structure of a three-phase voltage PWM inverter.

On the basis of Figure 6 and the equation of meshes, we can establish the following relations.

$$\begin{aligned} \frac{dV_{dc}}{dt} &= \frac{1}{c} I_{dc}, \\ I_1 &= I_{Ph} + I_{wind} = I_{dc} + I_2, \\ I_{dc} &= I_1 - I_2, \\ V_{dc} &= \frac{1}{c} \int_{t_2}^{t_1} I_{dc} dt + V_{dc0}. \end{aligned} \quad (13)$$

2.9. Inverter. The inverter is the last output of our system. The role of this block is to transform the single-phase to a three-phase signal.

The inverters are static converters of electrical energy from DC to AC. The function of an inverter is the inverse of a rectifier; that is, for a DC voltage given at the input, the inverter performs the voltage cutting by means of semiconductors (transistors or thyristors) in order to obtain an AC voltage that can be adjusted in frequency and effective values.

The AC waveform of the output voltage is determined by the ripple system. The most used inverters are in PWM. The type of inverter chosen for our work is presented in Figure 7.

By applying the mesh equations, we obtain the compound voltages between phases given by

$$\begin{aligned} V_{AB} &= (F_1 - F_2) V_{dc}, \\ V_{BC} &= (F_2 - F_3) V_{dc}, \\ V_{CA} &= (F_3 - F_1) V_{dc}, \end{aligned} \quad (14)$$

where F_1 , F_2 , and F_3 respectively represent the state of the switches K1, K2, and K3, and V_{dc} presents the output of a continuous bus.

Assuming that the input load is balanced, the output constitutes then a balanced system given in

$$V_A + V_B + V_C = 0. \quad (15)$$

The input current I_{dc} is expressed as a function of current outputs I_A , I_B , and I_C , as expressed in

$$I_{dc} = F_1 I_A + F_2 I_B + F_3 I_C. \quad (16)$$

2.10. MPPT Fuzzy Logic Controllers. The implementation of classic algorithms is simple and is independent from turbine

characteristics [14], but there still exist issues like the selection of the step size. The use of a big step size can define the MPPT fast, but it can result in severe oscillations around the MPPT. Reducing the perturbation step size slows down the MPPT process mostly where the wind speed varies fast despite the fact that the oscillations around the MPPT can be minimized [15, 16]. To solve this conflicting situation, we use in this paper a fuzzy logic control algorithm that can realize a variable step-size control.

The fuzzy logic control can use a large step size when the operating point is far away from the maximum power point, whereas the step can be minimized when the algorithm converges to the maximum power point [17, 18]. Thus, we can say that the fuzzy logic control can dynamically change its step depending on the energy input conditions.

Generally, The MPPT controller is a functional element of the PV and wind systems, which enables searching the operating point of the PV generator and the wind turbine under variable load and atmospheric conditions.

For the solar subsystem, the MPPT is based on the circuit maximum power transfer requirements: the objective is to check the point where the PV cell's output impedance is equal to the load impedance. The duty cycle is controlled by the MPPT controller for the pulse-width modulation block. This controls the power converter (DC-DC) to deliver a maximum power to the DC load bus.

The structure of the fuzzy controller used in the solar subsystem is depicted in Figure 8. In a MATLAB/SIMULINK (V.R2012b) environment, there is a fuzzy toolbox that allows the user to manage this structure and formulate fuzzy rules. Using this tool, we can configure the used command.

A controller based on a fuzzy logic algorithm is composed of three stages: fuzzification, rule base, and defuzzification. During fuzzification, the numerical input variables P_{pv} and V_{pv} are converted into linguistic variables based on a membership function. There is a block for calculating the error (E) and the change of the error (dE), expressed, respectively, in (17) and (18), at sampling instant k , as follows:

$$E(k) = \frac{dP}{dV} = \frac{P(k) - P(k-1)}{V(k) - V(k-1)}, \quad (17)$$

$$dE(k) = E(k) - E(k-1), \quad (18)$$

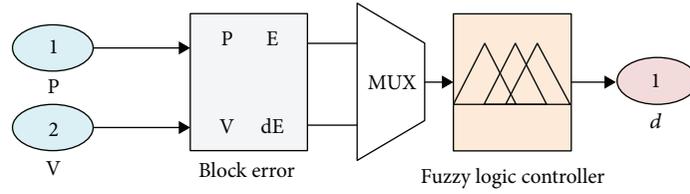


FIGURE 8: SIMULINK model of fuzzy logic controller of solar subsystem.

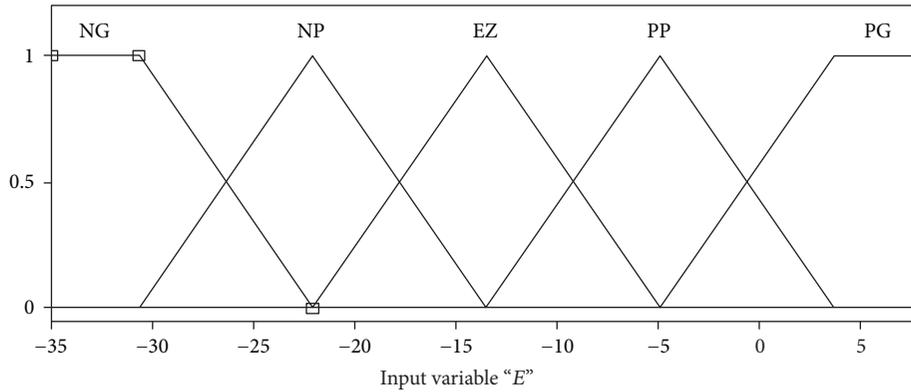


FIGURE 9: Member functions of solar input E .

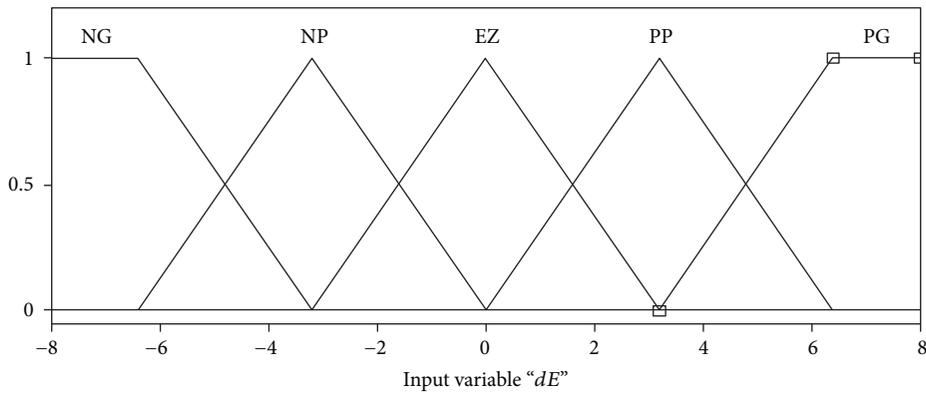


FIGURE 10: Member function of solar input dE .

where $P(k)$ and $V(k)$ are the power and the terminal voltage delivered, respectively, by a PV module.

The value of the error $E(k)$ determines the exact MPPT controller output according to the sign. The MPPT controller can decide in this way what the variation in the duty cycle (decrease or increase the speed of convergence) will be, which must be imposed on the DC-DC boost converter to approach the maximum power point. The value result of $E(k)$ and $dE(k)$ are converted to the linguistic variables, then the output of the fuzzy logic controller, which is d , can be looked up in a rule-base table.

The member function of the input variables (E and dE) of the solar fuzzy logic controller with MATLAB fuzzy tools is five member functions for each one. They are parametrized for E and dE , as shown in Figures 9 and 10, respectively.

TABLE 5: Fuzzy logic controller inference rules.

E	dE				
	NB	NS	ZE	PB	PS
NB	PB	PB	PB	PB	PB
NS	PB	PS	PS	PS	ZE
ZE	PS	PS	ZE	NS	NS
PB	NS	NS	NS	NS	NS
PS	NB	NS	NS	NS	ZE

The linguistic variables assigned to the duty factor d for the different combinations of $E(k)$ and $dE(k)$ are established according to our knowledge given in Table 5.

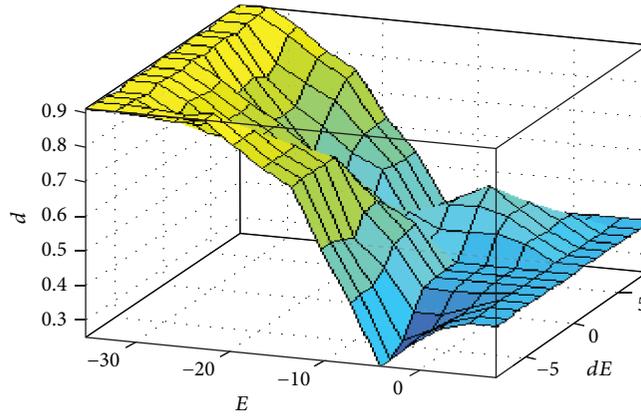


FIGURE 11: 3D surface of output duty cycle (d) values.

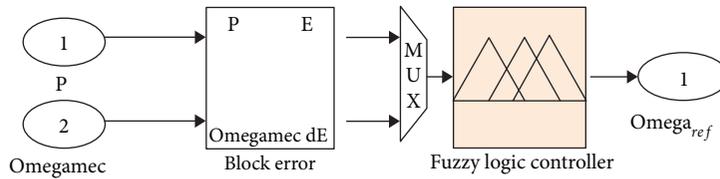


FIGURE 12: SIMULINK model of a fuzzy logic controller of a wind subsystem.

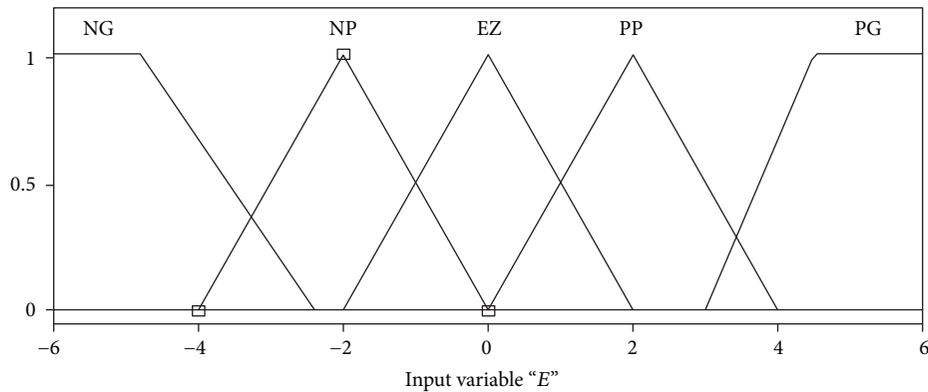


FIGURE 13: Member function of wind input E .

In this table, the linguistic variables used for the fuzzy logic controller are PB (positive big), PS (positive small), ZE (zero), NS (negative small), and NB (negative big).

The Mamdani type and inference rules with logical Min–Max operators are chosen for the fuzzy logic controller.

By applying the rules, the fuzzy tools generate the result output of each pair (E, dE) , as described in Figure 11.

In the second case of the wind subsystem, to control the MPPT, we use the same block error inputs where $P(k)$ is the power output delivered by the wind turbine module and $\Omega(k)$ is the wind speed of the module. According to the sign of $E(k)$, the MPPT controller can decide what the variation in the wind speed (Ω_{ref}) will be, which must be imposed.

We choose the same five linguistic variables for the MPPT controller [19]. For the inference rules of the MPPT

controller, we choose the Mamdani type inference rules with logical Min–Max operators.

On defuzzification, the fuzzy logic controller output is converted from a linguistic variable to a numerical one while still using the membership function [20].

In Figure 12, the structure of the MPPT fuzzy logic controller designed to command the wind subsystem is shown.

Each member function of the input variables (E and dE) of the wind fuzzy logic controller with MATLAB fuzzy tools is parametrized for E and dE , as illustrated in Figures 13 and 14, respectively.

By applying the rules, the fuzzy tools generate the result output of each pair (E, dE) , as described in Figure 15.

All the other blocks (the PV generator, the wind turbine, the PMSG, the rectifier, the boost converters, the DC-DC

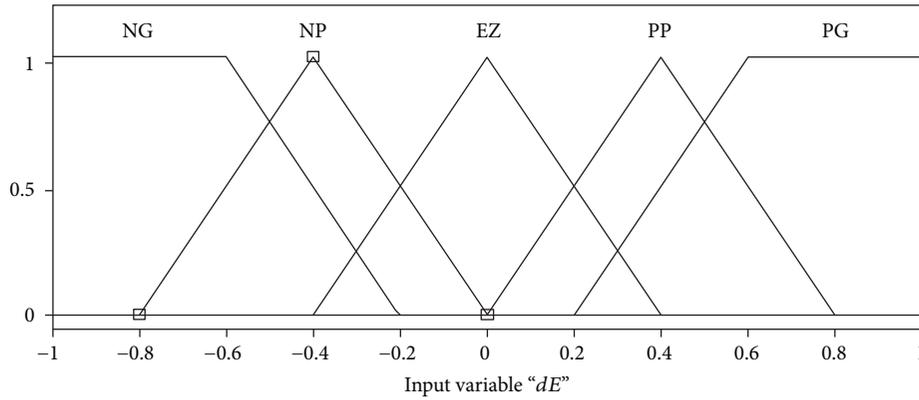


FIGURE 14: Member function of wind input dE .

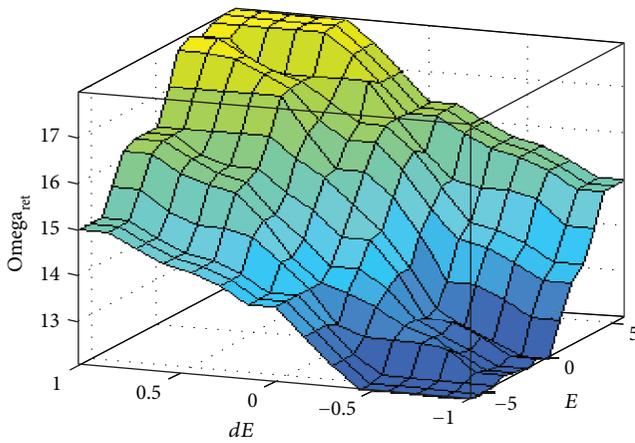


FIGURE 15: 3D surface of output wind speed reference values.

bus, etc.) are designed according to the mathematical equations of each component, as described above.

3. XSG Conception

3.1. Introduction. The classic FPGA implementation methodology consists of two main steps [21]. In the first one, the command is modeled and simulated using the MATLAB/SIMULINK software tools in our work. The second step is dedicated to the hardware architecture design and the HDL description which is performed manually. The designers that are not familiar with the HDL-coding process can consume a lot of time in this step. In this paper, we use the XSG tool for the development of a digital signal processor (DSP). It allows a high-level implementation of DSP algorithms on an FPGA circuit. It consists of a graphical interface library on MATLAB/SIMULINK used for modeling and simulating a dynamic system process [22]. We can find the library of logic cores, specific for FPGA implementation, provided to be configured and used according to the designer's requirements. Next, the implementation of system generator blocks, the Xilinx Integrated Software Environment (ISE) is used to create "netlist" files. The latter serves to convert the logic design into a physical file and generate the bitstream that will

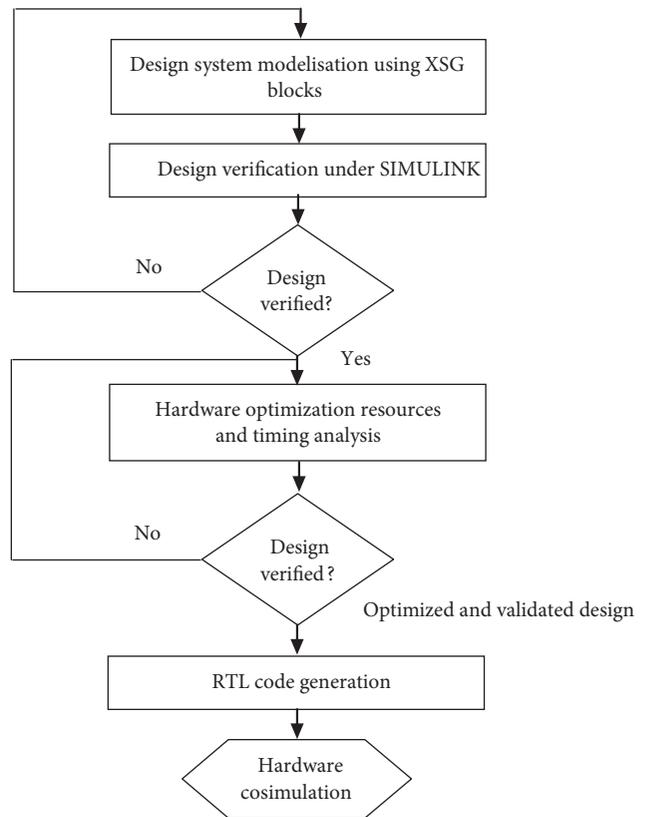


FIGURE 16: Flow of system generator tool.

be transferred to the target device through a standard JTAG (Joint Test Action Group) connection. The simulation in MATLAB XSG allows the test of the FPGA application with the same conditions as with the physical device and with the same data width of variables and operators. The simulation result can be bit-to-bit and cycle accurate, which means that each single bit is accurately processed and each operation will take the exact number of cycles to process.

Figure 16 presents the processing flow of the system generator tool from the hardware architecture design to the generation and cosimulation process. In this paper, the

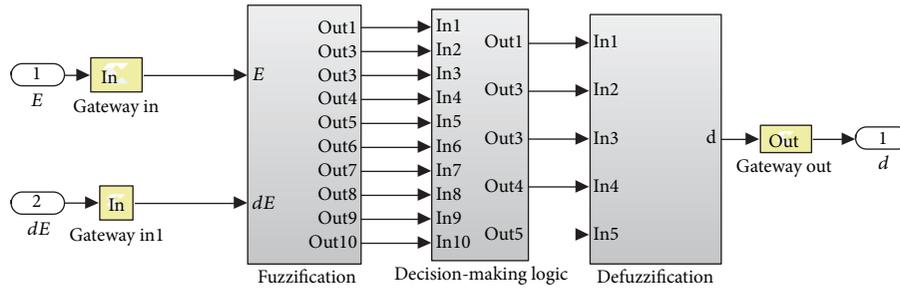


FIGURE 18: Architecture of solar fuzzy logic controller with XSG.

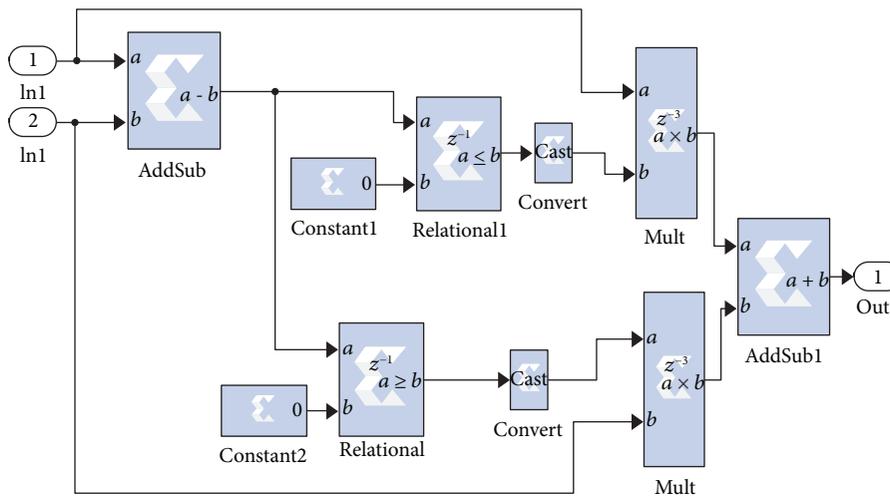


FIGURE 19: Min block built with Xilinx library.

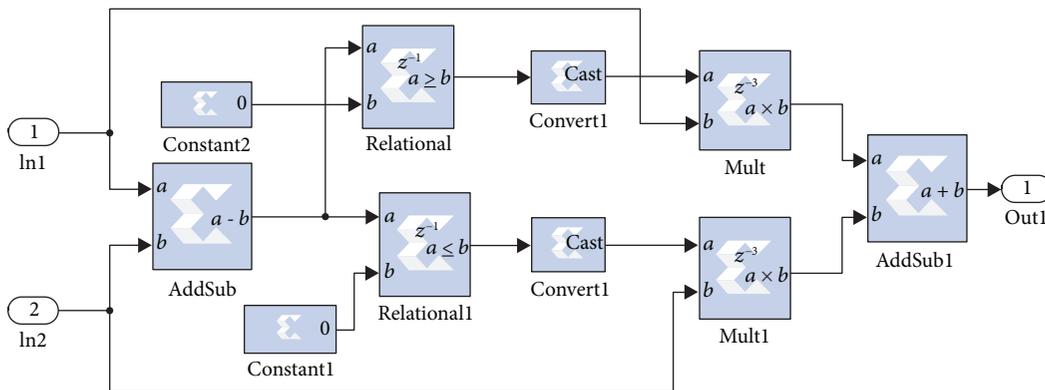


FIGURE 20: Max block built with Xilinx library.

(E , dE), a gateway output (d), and three subsystems which are fuzzification, decision-making logic, and defuzzification, where they function with the same steps used by the MATLAB/SIMULINK tool. The same procedure is applied to control the pitch angle for the wind fuzzy logic controller.

In the subsystem fuzzification, we model the five linguistic variables; then in the decision-making logic subsystem, we apply the Min–Max inference to the linguistic variables. For each gateway input, we have 25 rules, hence 50 rules for both

gateway inputs. We reuse the Min–Max inference until we have only one gateway output.

Figures 19 and 20 respectively present the Min and Max blocks built with the Xilinx library.

For linguistic variables, we cite PS as an example, which is described in Figure 21.

During the defuzzification, a MUX is used to convert the gateway output to a numeric value. Figure 22 shows the defuzzification block built with XSG components.

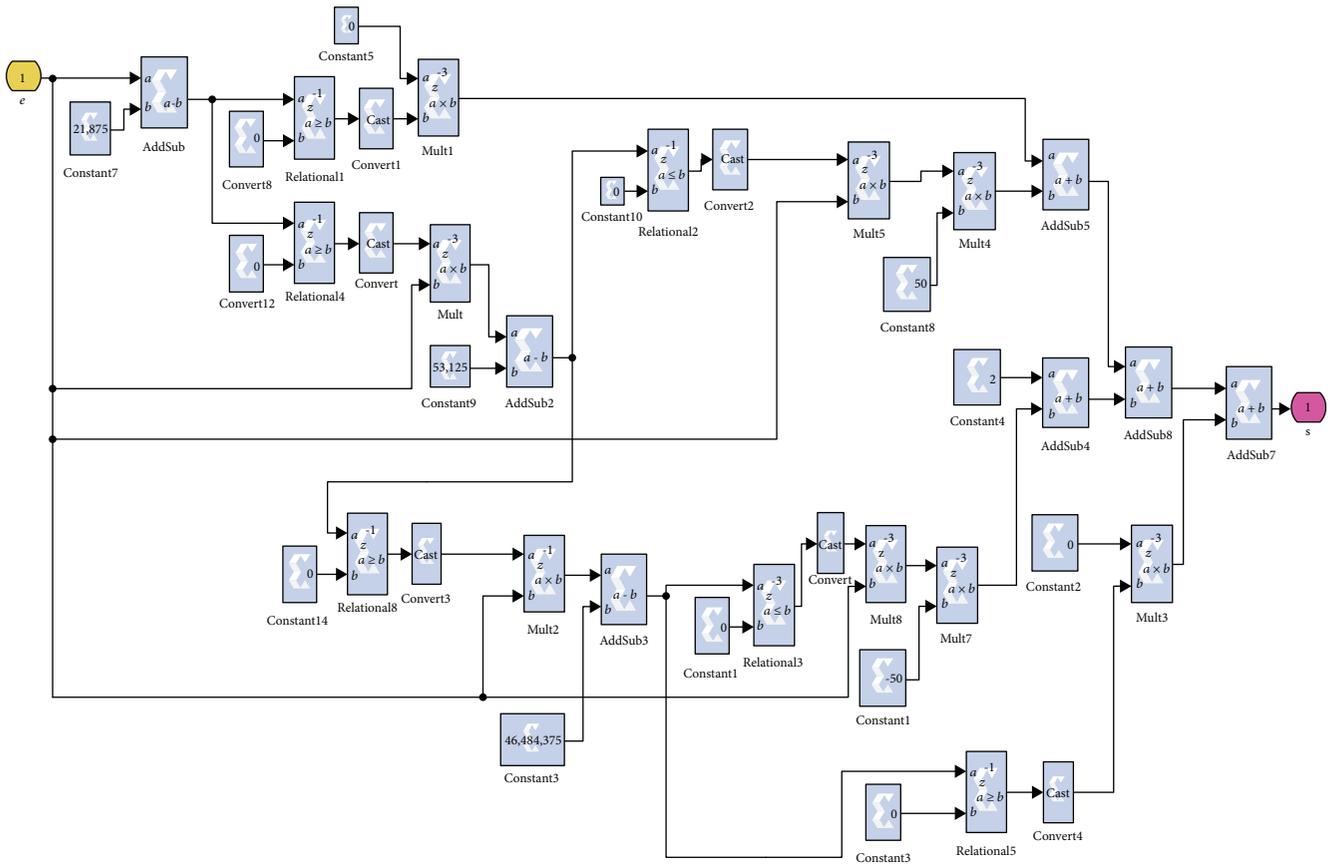


FIGURE 21: Linguistic variable PS built with Xilinx library.

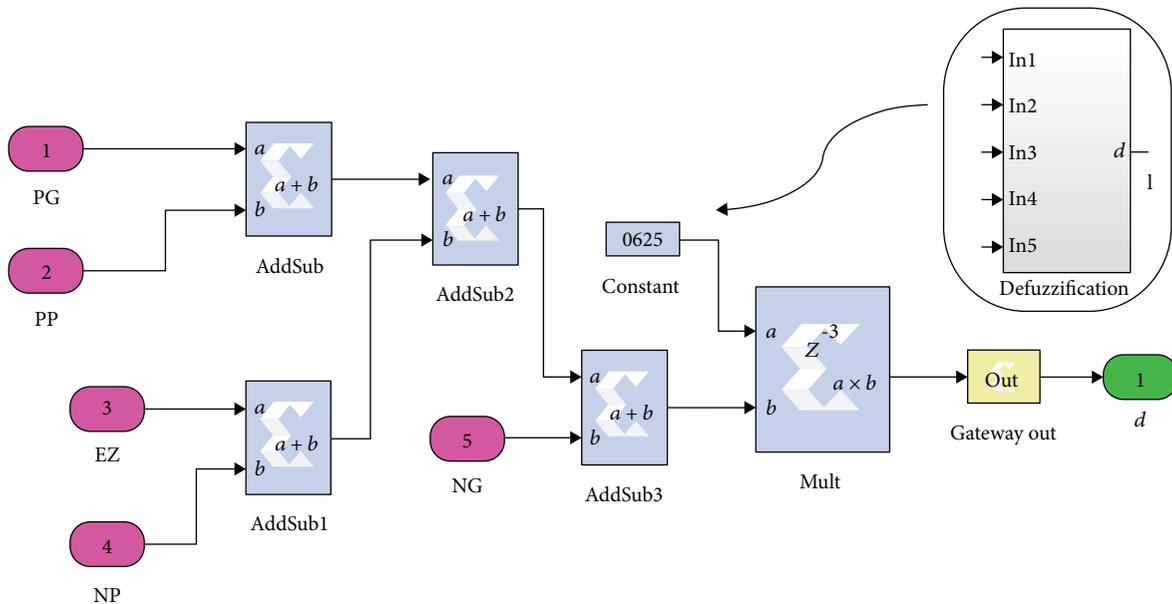


FIGURE 22: Defuzzification block built with Xilinx library.

4. Simulation Results

4.1. Simulation of Controller Output. Figures 23 and 24 respectively present the simulation results of the duty cycle

output of the solar fuzzy logic controller and the wind speed (ω_{ref}) output of the wind fuzzy logic controller. As can be seen, the outputs are numerical values applied to adjust the performance of subsystems.

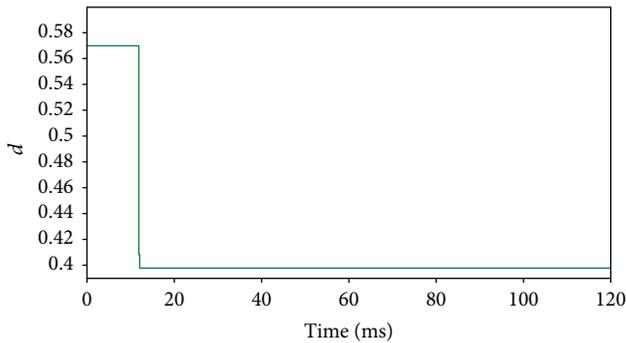


FIGURE 23: Fuzzy logic controller output of solar subsystem.

The duty cycle (d) and the wind speed (ω_{ref}) stabilization values are, respectively, 0.41 and 15 m/s.

4.2. Simulation of Power Output. To analyze the results, we compare the system with XSG and the system designed with MATLAB/SIMULINK block simulations on fixed atmospheric conditions.

The values of the different atmospheric conditions used for the simulation of the designed wind-solar system are given on Table 6.

Figure 25 compares between power versus time evolution obtained with MPPT controllers in the two cases: with simple block MATLAB/SIMULINK (PHyb) and with XSG. As is noted, the maximum power value is obtained at the same time, but the stabilization is quicker for the system with XSG. Thus, we can say that the system model with XSG uses its advantage of parallel processing to reach the stabilization form.

The curve of the hybrid power output with XSG using fuzzy logic controllers shows a good accuracy of the proposed algorithm since the stabilization at time is 12 ms.

In the system with the XSG library, the forms are the same as in the system with simple blocks. Consequently, we can conclude that the synchronization between the XSG blocks is perfect.

4.3. Simulation of Inverter Output. The intersection of the signals (carrier and modulator) gives the switching times. Figure 26 illustrates the switch output signal of the inverter switches.

The output signal is 1 if the modulator is larger than the carrier and 0 in the opposite case. The output signal therefore changes the state (0 or 1) at each intersection of both signals. Figure 27 describes the two signals to be compared.

The three voltages of the resulting phases of the inverter are phase shifted by $2\pi/3$. The frequencies of the carrier and modulating signals are successively 50 Hz and 5 Hz. The purpose of this choice is to extract a maximum intersection. Figure 28 schematically depicts the appearance of a phase voltage at the output of the inverter.

These results of the inverter output coincide with the system modeling by using the MATLAB/SIMULINK library. In this step, we can say that the cosimulation with XSG is satisfied.

The picture of the experimental setup is shown in Figure 29.

To ensure greater efficiency of our model, with fuzzy logic controllers, we compare it with other work results.

In Table 7, a comparison with some references in terms of power delivered by the hybrid system and the inverter signal output is provided.

As represented in Table 7, a comparison with some references uses different topologies in terms of tracking efficiency and response time. After ensuring that the system progress is validated and evaluated, we can move now to implement the commands on FPGA circuits.

5. Implementation on FPGA

In the first time, we start with a standard 36-bit fixed-point format in various blocks of the XSG wind-solar system so as to ensure the same performance as the floating-point format of MATLAB simulation [24]. In a second step, by using the test-error approach, we try to reduce the width of the data while keeping the performance and the synchronization between blocks. The utilized general fixed-point format is 24 bits [25, 26] which guarantees the same performance as MATLAB.

Once completed and verified, the XSG architecture can be automatically mapped to hardware implementation on FPGA. We choose Virtex-6-XC6VLX315T and we use the “generate” button from the settings window of the system generator tool. After that, the VHDL is automatically generated [27]. The characteristics of the FPGA Virtex-6-XC6VLX315T that we use in the prototype are shown on Table 8.

We use an HDL code generated into the XILINX ISE design software to prepare the transfer of the command controller to the FPGA support.

Synthesis is the step where the code HDL transforms to electronic components and the register transfer level will be generated.

The different resources of device utilization provided by the Xilinx ISE are shown in Table 9.

6. Conclusion

In this paper, we have put forward a hardware implementation of fuzzy logic controllers for a solar-wind system targeted for isolated sites.

First, we have designed the hybrid system with MATLAB/SIMULINK and XSG blocks. In general, the maximum power point has achieved forms of the wind-solar system with simple blocks: MATLAB/SIMULINK and XSG blocks are practically the same. The system with XSG has stabilized more quickly because of the parallel advantage of the XSG architecture. The maximum power point has achieved time in two analyzed cases, which has been about 0.4 ms.

In the second step, the performance of the proposed hardware implementation in terms of processing latency has been evaluated relatively to the first achieved maximum time and stabilization form. The XSG tool has been used

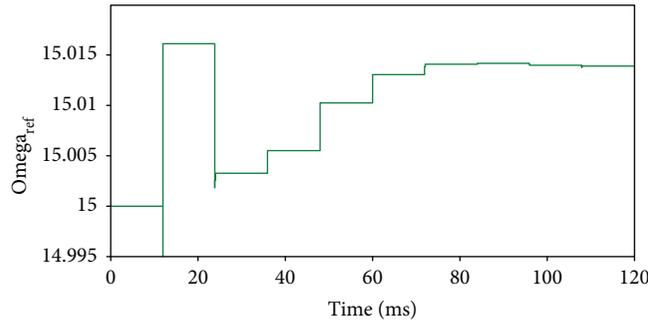


FIGURE 24: Fuzzy logic controller output of wind subsystem.

TABLE 6: Atmospheric wind and solar values.

Atmospheric condition	Values
Wind speed	10 m/s
Sunlight	1000 W/m ²
Temperature	300 K

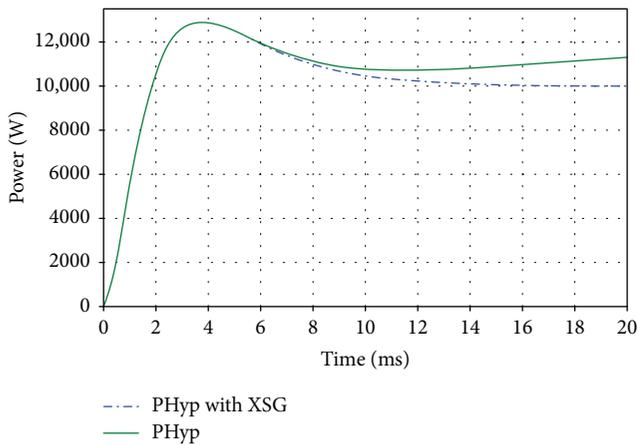


FIGURE 25: Power of wind-solar system with XSG and MATLAB/SIMULINK on fixed atmospheric conditions.

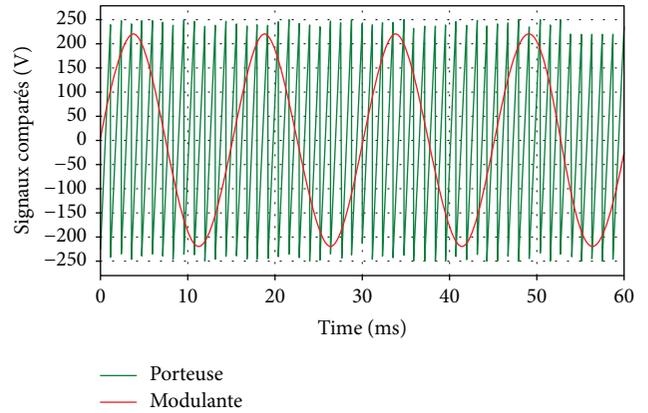


FIGURE 27: Aspect of modulated and carrier signals.

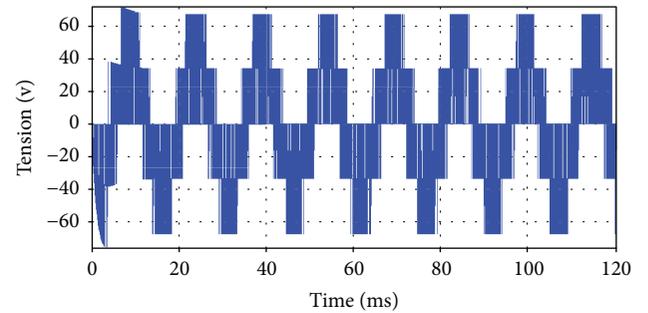


FIGURE 28: Voltage of one phase at inverter output.

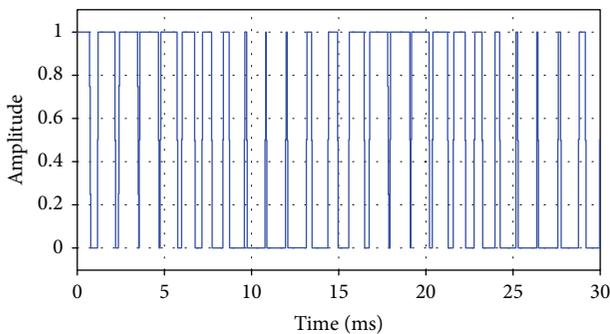


FIGURE 26: Switching signal of inverter switches.

for the system development. The use of this tool demonstrates that it has benefits in terms of conception time, since the developed design has been utilized firstly for the software validation and for the hardware system generation.

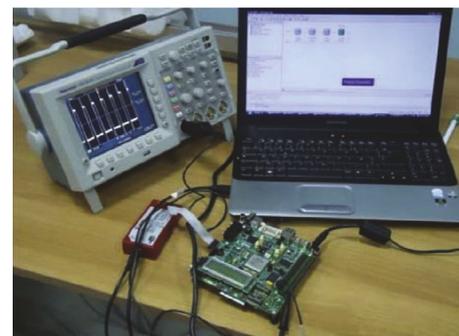


FIGURE 29: Picture of the experimental setup.

TABLE 7: Performance comparison between proposed work and other works.

References	Algorithms	Tracking efficiency η_{MPPT} (%)	Response time (s)
Proposed algorithm	Fuzzy logic	99.61	0.40
Izadbakhsh et al. [23]	ANFIS for solar fuzzy logic for wind subsystems	99.34	0.80
Suresh and Dhandapani [3]	Fuzzy logic	99.40	0.58
	HC and P&O	98.32	1.7
Rezvani et al. [4]	Fuzzy logic without load	99.25	0.80
	Fuzzy logic with load		1.1
Belmili et al. [5]	Fuzzy logic	99.22	0.80
	P&O	96.98	2.95

TABLE 8: Characteristics of the FPGA Virtex-6-XC6VLX315T.

Logic cells	314,880	
	Slices	49,200
Configurable logic blocks (CLBs)	Max distributed RAM (Kb)	5,090
DSP48E1 slices	1,344	
Interface blocks for PCI express	2	
Total I/O banks	18	
Max user I/O	720	
Total number of configuration bits	104,465,888	
Price	\$667.29	

TABLE 9: FPGA circuit device utilization.

Slice logic utilization	Used	Available	Utilization (%)
Number of slice registers	21,286	393,600	5
Number of slice LUTs	37,009	196,800	18
Number used as logic	27,480	196,800	13
Number used as memory	5851	81,440	7
Number of occupied slices	11,300	49,200	22
Number with unused flip flop	22,350	39,794	56
Number with unused LUT	2785	39,794	6
Number of bonded IOBs	257	600	42
Number of DSP48E1s	1056	1344	78

Moreover, the timing response and tracking efficiency of the complete design on the XSG architecture indicate that a high performance in terms of execution time can be reached. Indeed, the suggested architecture largely respects the timing and tracking constraints of the used hybrid system. The performance of the proposed solar-wind system supports increases with respect to material specifications. The experimental simulation results in terms of achieved tracking time, efficiency, and algorithm utilization are compared to those of other existing systems.

At the end of the paper, we have implemented and optimized the designed system on an FPGA circuit, Virtex-6-XC6VLX315T, by using the Xilinx ISE.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

All authors helped in conceiving the experiments. Aymen Jemaa designed and performed the experiments. At the same time, Aymen Jemaa and Mohamed Ali Hajjaji wrote the main part of the paper. Ons Zarrad and Mohamed Nejib Mansouri contributed in interpreting the results and revising and writing of the paper.

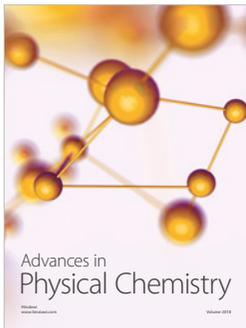
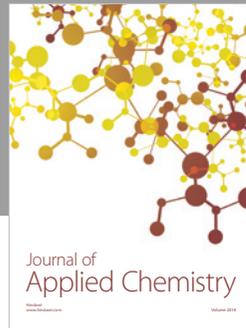
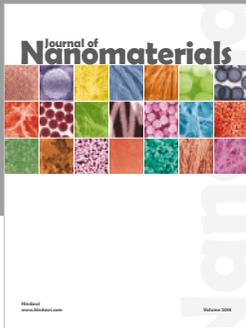
Acknowledgments

This work was partially supported by ESIER.

References

- [1] P. Nema, R. K. Nema, and S. Rangnekar, "A current and future state of art development of hybrid energy system using wind and PV-solar: a review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 8, pp. 2096–2103, 2009.
- [2] J.-H. Chen, H.-T. Yau, and J.-H. Lu, "Implementation of FPGA-based charge control for a self-sufficient solar tracking power supply system," *Applied Sciences*, vol. 6, no. 2, 2016.
- [3] P. Suresh and K. Dhandapani, "Modeling of an efficient hybrid wind and PV system for DC bus voltage regulation," *IJCTA*, vol. 9, pp. 2517–2525, 2016.
- [4] A. Rezvani, M. Izadbakhsh, and M. Gandomkar, "Enhancement of hybrid dynamic performance using ANFIS for fast varying solar radiation and fuzzy logic controller in high speeds wind," *Journal of Electrical Systems*, vol. 11, pp. 11–26, 2015.
- [5] H. Belmili, S. Boulouma, B. Boualem, and A. M. faycal, "Optimized control and sizing of standalone PV-wind energy conversion system," *Energy Procedia*, vol. 107, pp. 76–84, 2017.
- [6] S. K. A. Shezan, N. H. Khan, M. T. Anowar et al., "Fuzzy logic implementation with MATLAB for solar-wind-battery-diesel hybrid energy system," *Imperial Journal of Interdisciplinary Research*, vol. 2, pp. 574–583, 2016.
- [7] M. Rosyadi, S. M. Muyeen, R. takahashi, and J. Tamura, "A design fuzzy logic controller for a permanent magnet wind generator to enhance the dynamic stability of wind farms," *Applied Sciences*, vol. 2, no. 4, pp. 780–800, 2012.

- [8] Z. Ons, J. Aymen, A. Craciunescu, and M. Popescu, "Comparison of hill-climbing and artificial neural network maximum power point tracking techniques for photovoltaic modules," in *2015 Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, pp. 19–23, Sliema, Malta, 2015.
- [9] K. K. Tse, B. M. T. Ho, H. S. H. Chung, and S. Y. R. Hui, "A Comparative study of maximum-power-point trackers for photovoltaic panels using switching-frequency modulation scheme," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 410–418, 2004.
- [10] C.-H. Huang, H.-Y. Pan, and K.-C. Lin, "Development of intelligent fuzzy controller for a two-axis solar tracking system," *Applied Sciences*, vol. 6, no. 5, 2016.
- [11] C. Marisarla and K. R. Kumar, "A hybrid wind and solar energy system with battery energy storage for an isolated system," *International Journal of Engineering and Innovative Technology*, vol. 3, pp. 99–104, 2013.
- [12] J. Aymen, Z. Ons, A. Crăciunescu, and M. Popescu, "Comparison of fuzzy and neuro-fuzzy controllers for maximum power point tracking of photovoltaic modules," *Renewable Energy and Power Quality Journal*, vol. 1, pp. 796–800, 2016.
- [13] S. Hassan, H. Li, T. Kamal, U. Arifoğlu, S. Mumtaz, and L. Khan, "Neuro-fuzzy wavelet based adaptive MPPT algorithm for photovoltaic systems," *Energies*, vol. 10, no. 3, 2017.
- [14] X.-P. Li, W.-L. Fu, Q.-J. Shi, J.-B. Xu, and Q.-Y. Jiang, "A fuzzy logical MPPT control strategy for PMSG wind generation systems," *Journal of Electronic Science and Technology*, vol. 11, pp. 72–77, 2013.
- [15] E. Cam, G. Gorel, and H. Mamur, "Use of the genetic algorithm-based fuzzy logic controller for load-frequency control in a two area interconnected power system," *Applied Sciences*, vol. 7, no. 3, 2017.
- [16] M. Lotfy, T. Senjyu, M. Farahat, A. Abdel-Gawad, and H. Matayoshi, "A polar fuzzy control scheme for hybrid power system using vehicle-to-grid technique," *Energies*, vol. 10, no. 8, 2017.
- [17] A. Aksjonov, K. Augsburg, and V. Vodovozov, "Design and simulation of the robust ABS and ESP fuzzy logic controller on the complex braking maneuvers," *Applied Sciences*, vol. 6, no. 12, 2016.
- [18] S. Cheng, W.-b. Sun, and W.-L. Liu, "Multi-objective configuration optimization of a hybrid energy storage system," *Applied Sciences*, vol. 7, no. 2, 2017.
- [19] W. Na, P. Chen, and J. Kim, "An improvement of a fuzzy logic-controlled maximum power point tracking algorithm for photovoltaic applications," *Applied Sciences*, vol. 7, no. 4, 2017.
- [20] P. Suresh and B. C. Sujatha, "Maximum power point tracking of a hybrid solar-wind power generation system for a smart grid using fuzzy logic control," *International Journal of Scientific & Engineering Research*, vol. 7, no. 5, pp. 165–170, 2016.
- [21] R. Hmida, A. Ben Abdelali, and A. Mtibaa, "Hardware implementation and validation of a traffic road sign detection and identification system," *Journal of Real-Time Image Processing*, vol. 15, no. 1, pp. 13–30, 2018.
- [22] R. Hmida, A. Ben Abdelali, F. Comby, L. Lapierre, A. Mtibaa, and R. Zapata, "Hardware implementation and validation of 3D underwater shape reconstruction algorithm using a stereo-catadioptric system," *Applied Sciences*, vol. 6, no. 9, 2016.
- [23] M. Izadbakhsh, A. Rezvani, and M. Gandomkar, "Improvement of microgrid dynamic performance under fault circumstances using ANFIS for fast varying solar radiation and fuzzy logic controller for wind system," *Electrical Engineering*, vol. 63, no. 4, pp. 551–578, 2014.
- [24] M. Bahoura, "FPGA implementation of blue whale calls classifier using high-level programming tool," *Electronics*, vol. 5, no. 1, 2016.
- [25] R. Nagaraj and B. K. Panigrahi, "Simulation and hardware implementation of FPGA based controller for hybrid power system," *International Journal of Electrical Energy*, vol. 3, no. 2, pp. 86–93, 2015.
- [26] P. Balamurugan, A. D. Praveenraj, I. Kaleeswarn, R. M. Karthik, and A. Tamilarasi, "FPGA implementation of MPPT hybrid algorithm for photovoltaic system," in *International Conference on Engineering Innovations and Solutions (ICEIS-2016)*, pp. 107–113, Tamilnadu, India, 2016.
- [27] A. P. Ruiz, M. Cirstea, W. Koczara, and R. Teodorescu, "A novel integrated renewable energy system modelling approach, allowing fast FPGA controller prototyping," in *2008 11th International Conference on Optimization of Electrical and Electronic Equipment*, pp. 395–400, Brasov, Romania, 2008.



Hindawi

Submit your manuscripts at
www.hindawi.com

