



FPGA Implementation of a PV Generator and a Boost Converter Controllers for a Virtual Laboratory Environment

Oscar Bautista¹ · Álvaro Bernal¹ · Andrés Pantoja² · Javier Revelo²

Received: 27 December 2022 / Accepted: 31 January 2024
© The Author(s) 2024

Abstract

This paper presents an FPGA hardware implementation of two different components of a microgrid's electrical systems. Small and isolated microgrids are currently a frequent solution for electricity coverage in remote areas, where detailed studies, simulations, and realistic emulations are required to design appropriate systems according to each location. Hardware implementations based on a Labview environment of 240 W and 330 W generic photovoltaic generators, a Boost converter, and P & O algorithm were performed. The results obtained from the panels and converter were compared to the Simulink response. Unified systems, with and without controls, were compared. The systems were first compiled individually on an FPGA NI PCIe 7841R and later unified. The data and performance obtained from the emulated environment were verified using Simulink models, and the desired correspondence was obtained.

Keywords FPGA · Microgrids · HIL · Photovoltaic · Boost converter · P& O

1 Introduction

Microgrids are a solution to introduce new sources of generation to provide to the electrical system greater autonomy without altering the reliability and avoiding blackouts and aggressive energy rationing programs. This option is especially growing in areas of difficult access or non-interconnected zones (NIZ), where on-site generation becomes a valid option, given the difficulty of transporting energy from large generators. The NIZ of many developing countries require these new sources of generation, since there are many rural areas that have only 4 or 5 h of service using diesel plants. The objectives of public policies are focused on guaranteeing the supply of energy, reducing solutions based on expensive liquid fuels that are difficult to transport and with continuous maintenance

requirements. To address the lack of coverage, distributed generation is considered altogether with information systems that allow the management scheme, an intelligent measurement, control of the quality of the power, optimization of the primary resources, and implementation of energy efficiency programs. These characteristics are the main topics related to both the rural microgrids and the principles of smartgrids (González et al. 2008).

According to Bloomberg and the United Nations report, in a wind and solar hybrid system, only a 10% of the investment is associated to the equipment for the solar plan, while a large percentage is used in maintenance and redesign (Moslener et al. 2017). Then, appropriate planning and careful designs are crucial for money savings and reliable energy systems to attend the more vulnerable population. In a technical sense, emulations of highly realistic scenarios based on hybrid generation systems and their respective couplings using converters will allow predicting the behavior of the system in a reliable way and in real time. Hence, the studies and designs avoid cost overruns, failures, or malfunctioning of the design, and facilitate technical sustainability of the projects.

Consequently, several technical efforts have focused on developing systems able to simulate and/or emulate the conditions of real microgrids. These systems allow

✉ Oscar Bautista
oscar.bautista@correounivalle.edu.co

¹ School of Electrical and Electronic Engineering, Universidad del Valle, Ciudad Universitaria Melendez, Santiago de Cali 76001, Valle del Cauca, Colombia

² Electronics Department, Universidad de Nariño, Ciudad Universitaria Torobajo, San Juan de Pasto 52001, Nariño, Colombia

reaching scenarios such as the control of maximum power extraction, synchronization, analysis of energy quality, and the management of resources associated with the dispatch of generation units in remote areas. Currently, there are quite robust commercial systems to perform these simulations, but they are expensive and complex. In consequence, highly specialized systems are not appropriate solutions to design networks of few users, like the ones in isolated areas. It is necessary to explore cheap and reliable control mechanisms that guarantee the proper functioning of a microgrid, through an adequate implementation of models, algorithms, and hardware capable of representing the real system. In addition to technical learning for the people who interact with the maintenance of these systems within the communities, since in the event of system failures they are exposed to blackouts. So, easy-to-use systems are needed that allow them to familiarize themselves and know these systems without the difficulty of large interaction environments that can be confusing.

In specific cases, solid solutions are formulated considering systems based on Field Programmable Gate Arrays (FPGAs), since they provide high performance, parallel compute, and involve low physical costs and space, compared to large laboratories for large electric systems.

FPGAs are integrated into different systems, which allow emulating different parts on an electrical grid, such as converters, controllers, generators, control strategies using tools such as: DSP Toolbox from Altera (Parma and Dinavahi Apr 2007), Virtual Test Bed (VTB-RT) (Bin et al. 2007), RTDS (McLaren et al. 2011), eMEGAsim (Snider et al. 2010), coprocessing between DSP and FPGA (Adler et al. 2012), OPAL-RT (Yamane and Abourida 2015), and Typhoon HIL (Jonke et al. 2016).

2 Field Programmable Gate Array

The Field Programmable Gate Array (FPGA) concept refers to an integrated circuit technology, which is organized and reconfigured into logical elements (LE) by interconnecting these units, obtaining two-dimensional arrangements over an area of the chip and intersections of these units which are connected by fuse technology to obtain more complex functions (Duman et al. 2014). Also, by having embedded RAM blocks and given its computational core, simulations can be achieved in real time, with the support of parallel processing and simultaneous execution of multiple threads, basing their performance on the optimization of low-level elements such as logical cells, DSP blocks and memory blocks. It should be noted that a more detailed modeling; the computational load will be greater (Li et al. 2017). Moreover, synchronous FPGAs are useful when multiFPGAs are used and a common time step

is sought. Multiprocessors based on FPGAs are also used to offer better performance and allow computational parallelism (Matar et al. 2011).

Previously, FPGA programming was done only by hardware description languages (HDLs), which are slow and require experience in chip design. Currently, there are tools without the complexity of the HDL languages, such as Xilinx System Generator Tools (XSG), a high level software that uses MATLAB/Simulink environments to create and verify hardware designs for Xilinx FPGAs (Xilinx 2009). Another example of a more user-friendly language is Labview FPGA, where block functions and data flows are used into a graphical programming compiled by FPGAs modeled as state diagrams or flow diagrams for general visualization (Washington and Dolman 2010).

Many HIL system are based on FPGA technologies, and these systems provide nanoseconds sampling times (Bélanger et al. 2010), so the input and output devices based on FPGAs can obtain high time resolution, allowing real simulations of the plant components with the option of easy reconfiguration of the hardware arrangements without modifications of the electronic system (Washington and Dolman 2010). For example, in Makinen et al. (2014) a microgrid with photovoltaic (PV) system and a battery was simulated on FPGAs, and that microgrid with their simulation times could become an RTS (Li et al. 2017). Figure 1 shows the different parts of a microgrid that can be emulated by integrating system components into FPGAs using HIL.

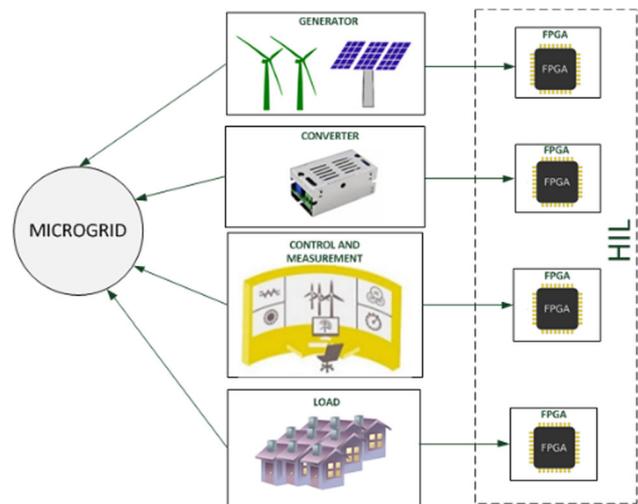


Fig. 1 HIL through FPGAs

3 Methodological Proposal

The methodology is composed of eight steps as shown in Fig. 2. The first step involves a model approach which includes a photovoltaic system and Boost converter. In second step, the models from step 1 are solved using a numerical method. In third step, the models and their solution are described in Labview in their simulated environment, and in fourth step, the models are compiled in the Labview FPGA tool. In fifth step, the preliminary results of each model are obtained, the photovoltaic panel model is compared with the curves provided by Simulink for two different panel designs, the same comparison is made for a standard Boost converter. In sixth step, the systems are unified, in seventh step, a system P &O algorithm is carried out, and in last step, the behavior of the systems with and without control is compared, with the input of an irradiance curve. In the implementation of this methodology, a Dell Precision 3600 computer was used with an Intel Xeon CPU E5-1620 at 3.60 GHz, with a 16 GB ram

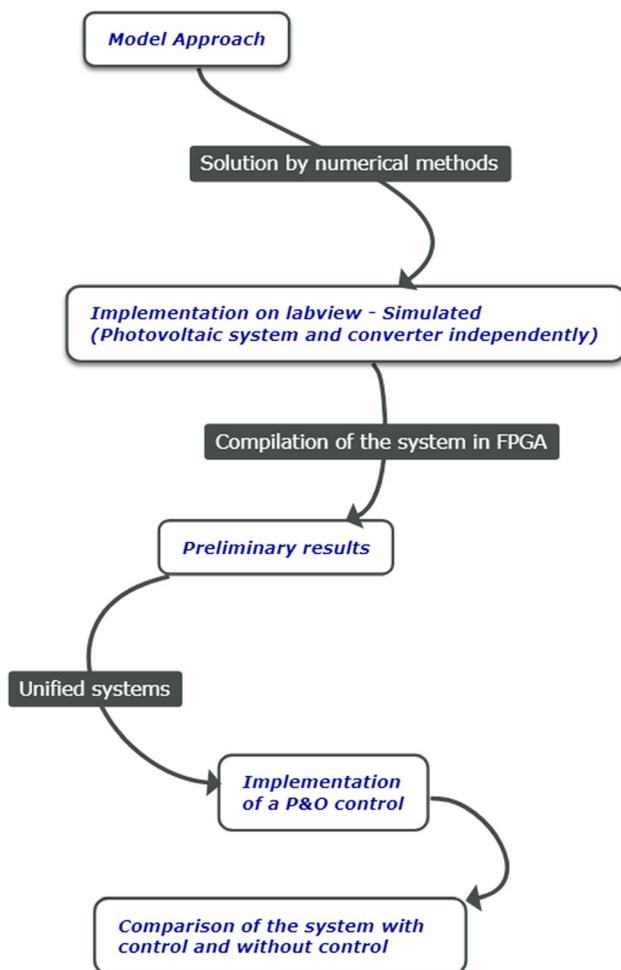


Fig. 2 Flowchart of the proposed methodology

memory, Windows 7 64-bit operating system. Labview software with all its modules and MATLAB R2020a Simulink were used for testing and the development board used was the NI PCIe 7841R with a FPGA Virtex 5 LX30 (National Instruments Corporation 2009).

4 Formulation of the Models

4.1 Photovoltaic Model

A technology that is currently booming is photovoltaic generation, which allows adjusting peak loads, flattening load curves. In the literature, different studies are presented about how the behavior of a photovoltaic panel can be modeled, some takes into account the effect of solar radiation (Fernández et al. 2008), and other authors propose panel models considering the effect of temperature (Granda-Gutiérrez et al. 2013). A proposed system contemplates solar radiation and temperature, which through the use of genetic algorithms allows obtaining physical parameters not provided by the manufacturer that standardize their behavior in the field (Ospino et al. 2014). In this study, an exponential model with temperature and irradiance is presented, this model is observed in Fig. 3 Villalva et al. (2009).

From Fig. 3, the equation shown in (1) is described

$$I = I_{pv} - I_0 \left[\exp\left(\frac{V + R_s I}{V_t a}\right) - 1 \right] - \frac{V + R_s I}{R_p} \quad (1)$$

where R_p is a shunt resistor, R_s is a series resistor, V is the panel output voltage, I is the panel output current, I_{pv} is the photovoltaic current, $I_0[\exp(\frac{V+R_s I}{V_t a}) - 1]$ is the diode current, I_0 is the saturation current of the array and V_t is the thermal voltage and is equal to $\frac{N_s k T}{q}$ with N_s cells connected in series, k is the Boltzmann constant ($1.381 \times 10^{-23} \frac{J}{K}$), a is the ideality factor of the diode, T is the junction temperature of the panel ($^{\circ}K$), and q is the charge of the electron ($1.602 \times 10^{-19} C$).

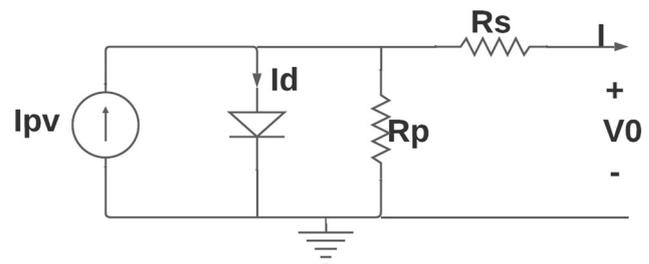


Fig. 3 Equivalent circuit of a practical photovoltaic device (Villalva et al. 2009)

Connecting the cells in parallel increases current and serial connection increases the voltage at the output of the panel.

In Eq. (2), the temperature and irradiance dependence of the photovoltaic current is observed

From Fig. 3, the equation shown in (1) is described

$$I_{pv} = (I_{pv,n} + K_I \Delta T) \frac{G}{G_n} \quad (2)$$

where $I_{pv,n}$ is the photovoltaic current under nominal conditions ($25^\circ C$ y $1000 W/m^2$), K_I is the short circuit current/temperature coefficient, T and G are the current temperature and irradiance of the device, respectively. Equation (1) has no direct solution, so it is solved using Newton–Raphson method and the relationship shown in (3) is obtained.

$$I_{(n+1)} = I_n + \frac{I_{pv} - I_0 \left[e^{\frac{(V+R_s I_n)}{V_{t,a}}} - 1 \right] - \frac{(V+R_s I_n)}{R_p}}{1 + \frac{R_s}{V_{t,a}} I_0 \left[e^{\frac{(V+R_s I_n)}{V_{t,a}}} \right] + \frac{R_s}{R_p}} \quad (3)$$

4.2 Boost Converter Model

In applications based on photovoltaic systems, it is common to use Boost converters, given their advantage over other converters in their stability and efficiency (Hayat et al. 2016) and its objective is to raise the voltage level. The Boost converter is made up of an inductor, a capacitor, a switch, and a diode; its configuration can be seen in Fig. 4.

For the continuous conduction mode (CCM), the converter works in three states: switch is on and the diode is off, this structure is called C, and when the switch is off and the diode is on, this structure is called D (Kazimierczuk 2008).

4.2.1 C Structure

This structure can be seen in Fig. 5a, with the switch on and the diode off, at this point the inductor is storing energy from the source and the capacitor is discharging through the load resistor.

In (4), (5) and (6), we can see the equations corresponding to this circuit.

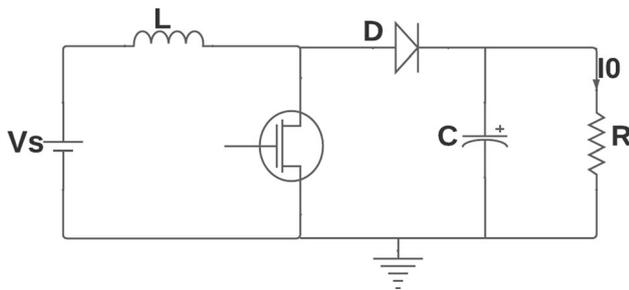


Fig. 4 Boost converter general model (Kazimierczuk 2008)

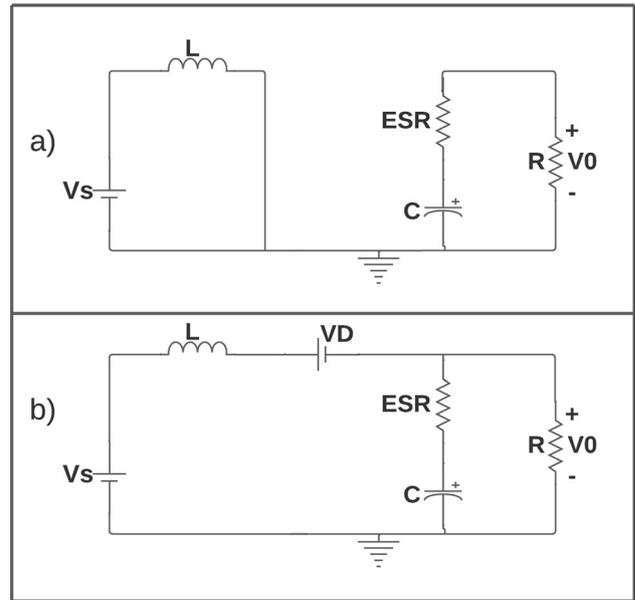


Fig. 5 Boost converter: a C Structure, b D Structure

$$\frac{dI_L}{dt} = \frac{V_s}{L} \quad (4)$$

$$\frac{dV_C}{dt} = \frac{-V_0}{C * R} \quad (5)$$

$$V_0 = \frac{R}{R + ESR} * V_C \quad (6)$$

4.2.2 D Structure

This structure can be seen in Fig. 5b, with the switch off and the diode on, in this state the inductor delivers its stored charge to the capacitor as to the load resistor. In (7), (8), and (9), we can see the equations corresponding to this circuit.

$$\frac{dI_L}{dt} = \frac{V_s - V_0}{L} \quad (7)$$

$$\frac{dV_C}{dt} = \frac{1}{C} \left(I_L - \frac{V_0}{R} \right) \quad (8)$$

$$V_0 = \frac{R}{R + ESR} * (I_L ESR + V_C) \quad (9)$$

To solve this system adapts the small-signal ac state-space averaged model that approximates the dynamic behavior of the system. In Estrada et al. (2020), the base form of this system is specified, and it is described in Eqs. (10) and (11).

$$\begin{aligned} & \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ &= \begin{bmatrix} -D'(R \parallel \text{ESR}) & -D' \frac{R}{R + \text{ESR}} \\ D' \frac{R}{R + \text{ESR}} & -\frac{1}{R + \text{ESR}} \end{bmatrix} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ &+ \begin{bmatrix} 1 & I_L(R \parallel \text{ESR}) + V_C \frac{R}{R + \text{ESR}} \\ 0 & -I_L \frac{R}{R + \text{ESR}} \end{bmatrix} \begin{bmatrix} \hat{v}_g(t) \\ \hat{d}(t) \end{bmatrix} \\ \hat{v}_0(t) &= \begin{bmatrix} D'(R \parallel \text{ESR}) & \frac{R}{R + \text{ESR}} \\ & \end{bmatrix} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ &+ [I_L(R \parallel \text{ESR})] \hat{d}(t) \end{aligned} \tag{10}$$

In Eq. (10) there is a system with the control variables $\hat{v}_g(t)$ and $\hat{d}(t)$. An adjustment must be made to this system by introducing a current control $\hat{i}_L(t)$ that passes through the load R ; the modified system is observed in Eq. (12).

$$\begin{aligned} & \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ &= \begin{bmatrix} -D'(R \parallel \text{ESR}) & -D' \frac{R}{R + \text{ESR}} \\ D' \frac{R}{R + \text{ESR}} & -\frac{1}{R + \text{ESR}} \end{bmatrix} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ &+ \begin{bmatrix} 1 & 0 & I_L(R \parallel \text{ESR}) + V_C \frac{R}{R + \text{ESR}} \\ 0 & 1 & -I_L \frac{R}{R + \text{ESR}} \end{bmatrix} \begin{bmatrix} \hat{v}_g(t) \\ \hat{i}_0(t) \\ \hat{d}(t) \end{bmatrix} \end{aligned} \tag{12}$$

4.2.3 Solution with Numerical Method

The system shown in (12) cannot be solved directly, so a numerical method is used as is the general case of Euler’s method (Butcher 2008) shown in Eq. (13). There is an analysis similar to this system of equations in the literature for a Buck system (Estrada et al. 2020); this work exposes it for a Boost converter.

$$X_{n+1} = X_n + h\dot{X}_n \tag{13}$$

Where h is a time step of the method and \dot{X}_n is the derivative of X_n .

The system shown in Eq. (12) must be taken to the form shown in (14) to be solved by the method shown in (13).

$$[\dot{X}_n] = [K]^{-1}[A][X_n] + [B][\mu] \tag{14}$$

For the system shown in Eq. (14), the equalities shown in (15) are used.

$$\begin{aligned} [K] &= \begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \\ [\dot{X}_n] &= \frac{d}{dt} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ [A] &= \begin{bmatrix} -D'(R \parallel \text{ESR}) & -D' \frac{R}{R + \text{ESR}} \\ D' \frac{R}{R + \text{ESR}} & -\frac{1}{R + \text{ESR}} \end{bmatrix} \\ [X_n] &= \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_c(t) \end{bmatrix} \\ [B] &= \begin{bmatrix} 1 & 0 & I_L(R \parallel \text{ESR}) + V_C \frac{R}{R + \text{ESR}} \\ 0 & 1 & -I_L \frac{R}{R + \text{ESR}} \end{bmatrix} \\ [\mu] &= \begin{bmatrix} \hat{v}_g(t) \\ \hat{i}_0(t) \\ \hat{d}(t) \end{bmatrix} \end{aligned} \tag{15}$$

Afterward, the panel and converter implementations in Labview are compared with the curves provided by Simulink and the error in the curves is found using the equation shown in (16)

$$\text{Error} = \left| \frac{\int_0^t P_{\text{simulink}} dt - \int_0^t P_{\text{labview}} dt}{\int_0^t P_{\text{simulink}} dt} \right| * 100 \tag{16}$$

In Eq. (16) $\int_0^t P_{\text{simulink}} dt$ is the power delivered by Simulink, while $\int_0^t P_{\text{labview}} dt$ is the power delivered by Labview.

4.3 P & O Algorithm Model

In the study of photovoltaic systems, it is relevant to extract the maximum power from the solar panels, for this there are different systems designed to monitor the tracking maximum power point (MPPT) such as perturb and observer (P &O), incremental conductance (INC), constant voltage (CV), parasitic capacitance (PC), among others (Hohm and Ropp 2000). For this study, the method perturbation and observation (P &O) is used, since it has been widely used for the ease of its structure and consists of measuring voltage, current, and power outputs and with a given disturbance that causes an increase or decrease in matrix power, then the next disturbance is performed depending on the previous result, as displayed in Fig. 7. In this way, the MPPT continuously searches for the maximum power (Hua and Shen 1998). For the implementation

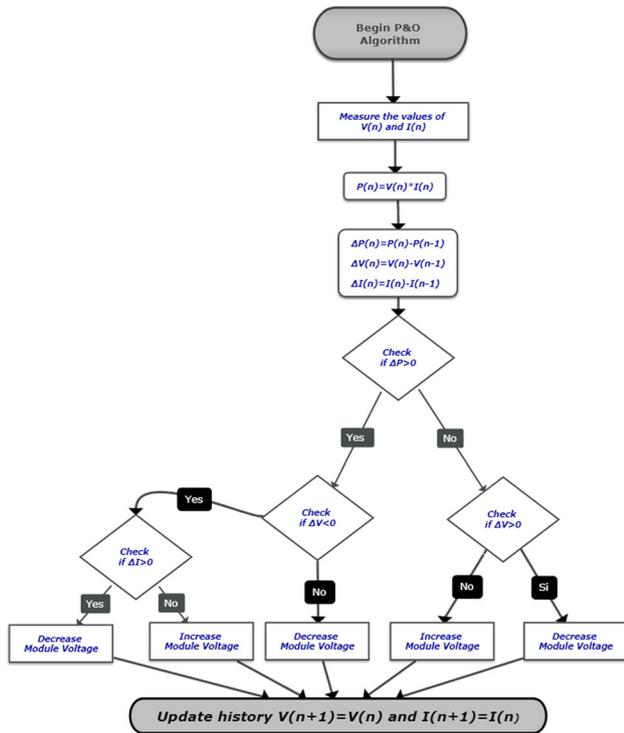


Fig. 6 Flowchart of P & O algorithm (Mahendran and Ramabadran 2017)

of this method, the flow shown in Fig. 6 is implemented, in which each step of the algorithm is described, starting with the measurement of the variables, continuing with the census and verification of the voltage behavior, and depending on this increase or decrease the voltage module which in this case is done by interacting with the duty cycle: when the adjustment is increased it is $D = D + \Delta D$ and when it decreases it is $D = D - \Delta D$ (Francis et al.

2014). The verification of the current variable allows to increase the duty cycle that can eliminate the problem of drift when approaching the operating point to the MPP and reduces the oscillation, since the conventional model presents these problems. In the same way, Fig. 7 shows the adjustments made by the algorithm in the different parts of the curve for each irradiance level, going from point “a” to “b” and following that order until reaching point “g” (Mahendran and Ramabadran 2017).

With the implementation of the algorithm, it is decided to compare the curves with control and without control by comparing performance following Eq. (17), in this way we can observe how much the performance of the curve with the implemented algorithm improved or worsened.

$$\eta = \left(1 - \frac{\int_0^t P_{P\&O} dt}{\int_0^t P_{converter} dt} \right) * 100 \quad (17)$$

In Eq. (17) $\int_0^t P_{P\&O} dt$ represents the power curve with control, while $\int_0^t P_{converter} dt$ represents the curve without control.

Given the hardware limitations present in the experimentation setup, we were confronted with the need to implement a maximum power point tracking (MPPT) solution that would maximize system efficiency while maintaining low computational cost. However, a significant portion of the FPGA’s available resources had already been allocated to the implementation of the photovoltaic panel models and the Boost converter. Furthermore, we faced the challenge of developing the low-level implementation in Labview (even the implementation of a simple exponential operator demands resources and a level of

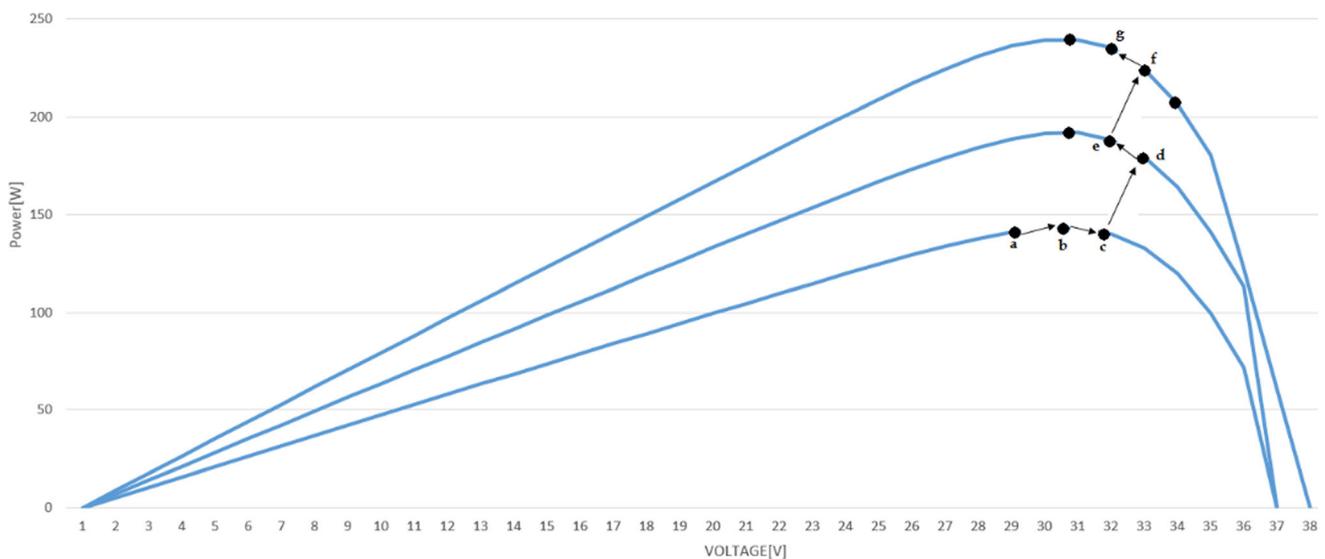


Fig. 7 Analysis using P & O algorithm (Mahendran and Ramabadran 2017)

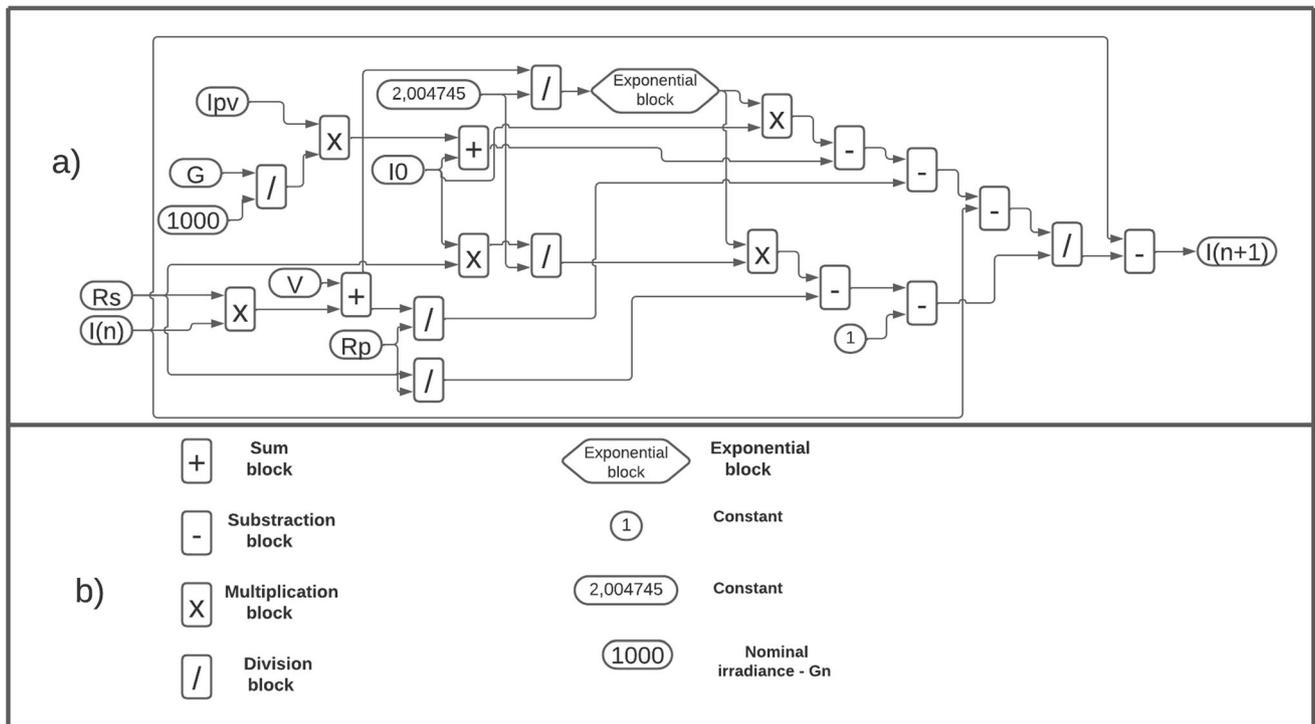


Fig. 8 a Block diagram of the Newton–Raphson function for the PV system, b description of the blocks of 8a

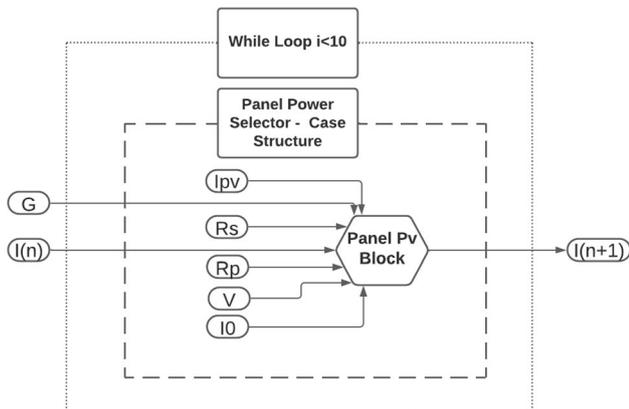


Fig. 9 Block diagram of the PV system

complexity, as these specific blocks are not readily available for such implementations).

Consequently, after evaluating all possibilities, it was concluded that the perturb and observe (P &O) MPPT method was the most suitable for the system’s MPPT needs. While this method may not exhibit the same precision as others in certain cases, it is less intricate and consumes fewer resources, aligning perfectly with the imposed limitations. By implementing the P &O MPPT method, we achieve optimal energy efficiency and prevent system overload.

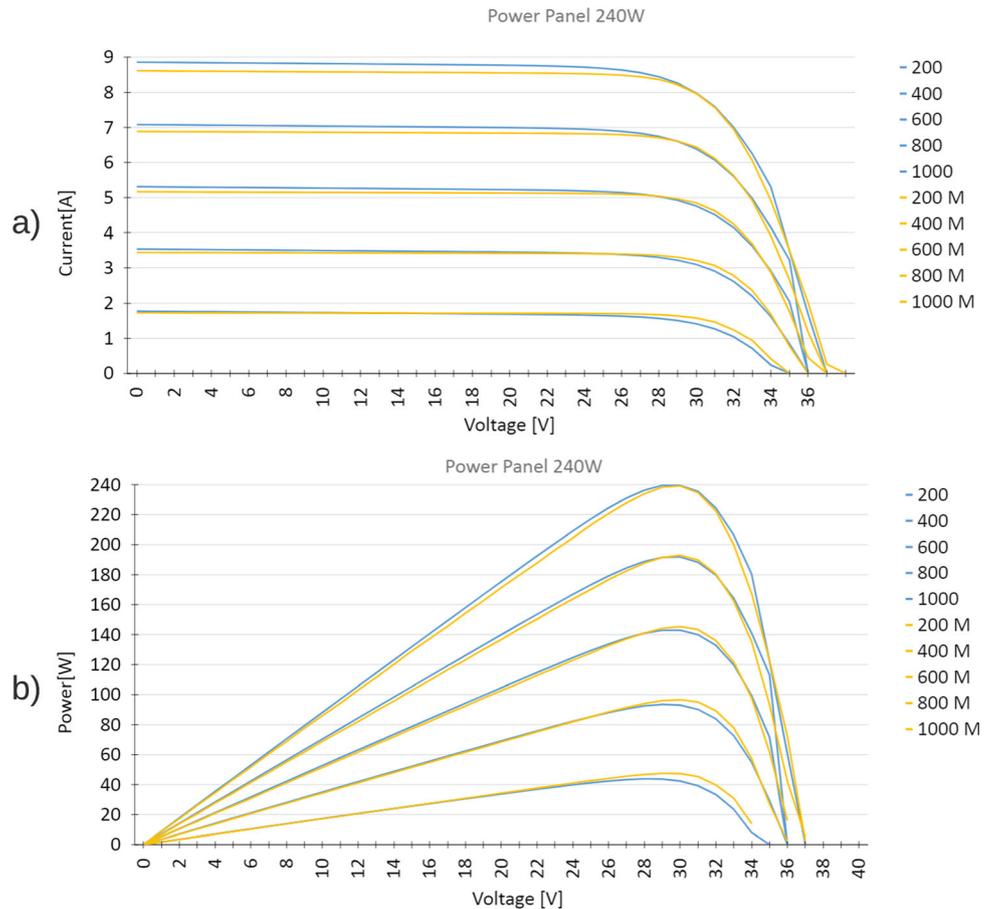
Table 1 Input values for the simulation of the 240 W and 330 W panels under nominal conditions ($T_n = 25^\circ C$, $G_n = 1000 \frac{W}{m^2}$)

Variable	PV (240 W)	PV (330 W)
$I_0 [A]$	$4.124 * 10^{-10}$	$7.089 * 10^{-11}$
$I_{(pv, n)} [A]$	8.181	8.83
$R_p [\Omega]$	337.822	342.515
$R_s [\Omega]$	0.354	0.354
$I_{max} [A]$	8.07	8.65
$V_{max} [V]$	29.7	38.15
N_s	60	60
a	1.3	1.3

5 Models Implementation

The implementation is carried out on the Labview software, where the variables are of the fixed-point representation type, for their subsequent implementation on the FPGA. The setting of the configuration is done by means of word length and integer word length, the first one determines the size of the word (depending on whether it is signed or unsigned) and the second one determines the word ranges from minimum to maximum value. In Labview, the variables can be controlled variables, constants, and indicators, the first two are normally used as inputs of an algebraic operation and the last one that are normally

Fig. 10 Curves of a 240 W at constant temperature (Simulink yellow, Labview blue):
a Current x Voltage. **b** Power x Voltage



used as outputs and is used to visualize the value obtained from an algebraic operation.

The following implementation is not the sole method for carrying out this type of implementation. It can be executed on any hardware description language (HDL), provided that a procedure similar to the one outlined in this article is adhered to. For this type of implementation, the system of equations is established and discretized to be compatible with the data accepted by an FPGA. However, the flexibility of the FPGA in the Labview environment allows for graphical programming and the generation of an end-user friendly interface. Additionally, Labview provides easy interconnection with instrumentation and hardware systems, making the project ready to connect with other signal acquisition systems. At the end of this article, a comparison is made with systems implemented in HDL language, and the results presented here improve upon previous works.

5.1 Implementation of the Photovoltaic Model

As shown in Fig. 8a, Eq. (3) was implemented on the Labview block diagram, leaving all the input variables as controlled variables and only I_{n+1} as the output indicator. In addition, it shows the implementation of the

mathematical model where the triangles are mathematical operations, the variables seen on the left (I_o , R_s , I_{pv} , V , $I(n)$, and R_p) are controlled variables and the output of the system is an indicator. Figure 8b shows a description of the blocks of 8-a). In the implementation of this model, it should be noted that operations such as Euler and natural logarithm, which are common operations in the basic Labview environment, are not common in the Labview FPGA environment due to their fixed-point representation and their significant digits. Therefore, for its representation, it was necessary to create a block that approximates these functions, limiting their significant digits, but obtaining reliable margins, given that the more exact its representation, the more space it will occupy inside the FPGA.

The system set out in Fig. 8a is converted into a small block that can be seen in Fig. 9 (box with number 2) and the controlled variables are converted into constants and are introduced into a "Case Structure" that gives versatility to the system, allowing to add more panel capacities. For this design, two power panels of 240 W (1Soltech 1STH-240-WH) and 330 W (Znshine PV-tech ZXM6-72-330-M) are used. This system is located within a "while loop," facilitating the convergence of the system.

Fig. 11 Curves of a 330 W at constant temperature (Simulink yellow, Labview blue):
a Current x Voltage. **b** Power x Voltage

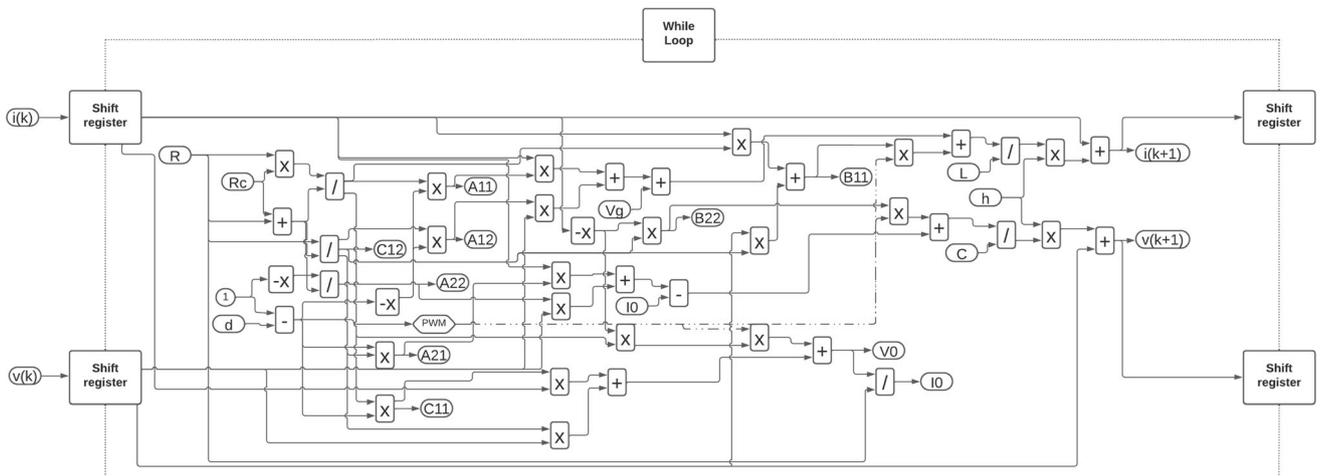
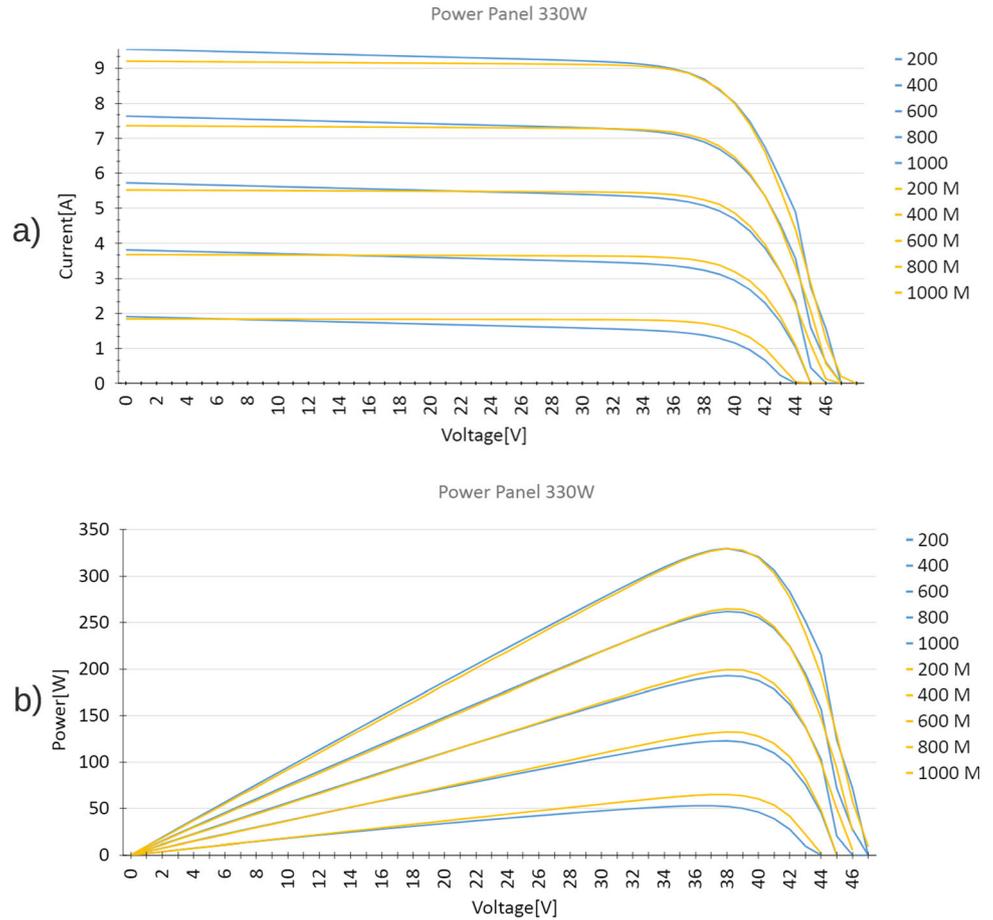


Fig. 12 Block diagram of the Euler's method for the Boost Converter

The constants used from the system are data obtained from National Renewable Energy Laboratory (NREL) System Advisor Model (National Renewable Energy Laboratory), which is used in Simulink for "PV Array." These input values are summarized in Table 1.

Using Eq. 2, the appropriate adjustments are made, leaving the temperature constant and varying the irradiance at values of 200, 400, 600, 800, and 1000 $\frac{W}{m^2}$. The behavior of the panel is compared with the graphs produced by Simulink. Figures 10 and 11 show the graphs of

Table 2 Input values for the simulation of the Boost converter

Variable	Value
V_S [V]	24
I_O [A]	16
R [Ω]	3
h [μ s]	9
L [μ H]	250
C [μ F]	100
ESR [Ω]	0.01

CurrentxVoltage and PowerxVoltage for the design of each panel: 240 W and 330 W.

The implementation is compiled in the Labview FPGA module; the results together with those of the other implementations are summarized at the end of this report.

5.2 Implementation of the Boost Converter Model

In the implementation of the Boost converter, a methodology used for a Buck converter (Estrada et al. 2020) was used; the block diagram of this implementation is shown in Fig. 12 where Eq. (14) is implemented with the equalities expressed in Eq. (15), for this system the input variables are constant, except the controlled variables V_S and I_O . The output variables are indicators $I(k+1)$ and $V(k+1)$.

Using the reported values (Kazimierczuk 2008; Vacheva et al. 2018; Valderrama et al. 2013), the data shown in Table 2 is chosen.

In addition to the input data in Table 2, the input values of I_L and I_C are equal to 0. Figure 13 show the Boost converter simulations for a) Voltage, b) Current, and c) Power; the yellow line corresponds to the input values obtained in Labview, the blue line corresponds to the input values obtained in Simulink, the orange line corresponds to the output data from Labview, and the green line corresponds to the output values obtained from Simulink.

The implementation is compiled in the Labview FPGA module, the results together with those of the other implementations are summarized at the end of this report.

5.3 Unified Implementation

In this subsection, the systems are unified, initially the panel and the converter and later the algorithm MPPT is attached. Subsequently, an irradiance curve is imported from Excel that moves from zero to 1000 W/m^2 and again returns to 0, this curve enters the panel in the circuits with control and without control and the behavior of the output variables is observed and an analysis of these final results is

made. For these implementations, the data observed in Table 3 is used:

5.4 Photovoltaic System Linked to the Boost Converter

After testing each of the previous systems, they are joined and the system shown in Fig. 14 is obtained, as can be seen in the left part, each of the panels is selected and the respective adjustments are made in the converter for the two power levels. Following the implementation made with the photovoltaic system, it is decided to compact the system of Fig. 12 into a block within Fig. 14, the controlled variables are converted into constants and the PWM was configured with a duty cycle of 50% and a frequency of 5 KHz.

The implementation is compiled in the Labview FPGA module, the results together with those of the other implementations are summarized at the end of this report.

5.5 Implementation of Algorithm P &O

The implementation of the model is carried out following the instructions shown in the algorithm of Fig. 6, measuring the variables V , I y P . The implementation carried out can be shown in Fig. 15.

Finally, the implementation is compiled in the Labview FPGA module, the results together with those of the other implementations are summarized at the end of this report. After introducing the systems in the FPGA, it interacts directly with the files generated in the FPGA and by importing an irradiance curve, the behavior of the two proposed systems with and without control is observed, obtaining a comparison of these results for voltage, current, and power shown in Figs. 17 and 18, for the powers of 240 W and 330 W, respectively. Figure 16 shows the implementation of the imported files from the FPGA and the irradiance curve.

Figures 17 and 18 show the curves generated by the system for voltage and current in the upper part and power in the lower part of each figure, while Fig. 17 shows the curves for a 240 W panel, and Fig. 18 shows the curves for a 330 W panel. The blue curves show the behavior of the uncontrolled system and the gray curves show the system with a P &O algorithm, where it is observed that the oscillations are reduced allowing a follow-up to nominal values of the panel at different irradiance values from zero to 1000 W/m^2 and again returns to 0, it is remarkable to observe how quickly the circuit reacts to irradiance differences, making good power monitoring, which was what was expected in this implementation.

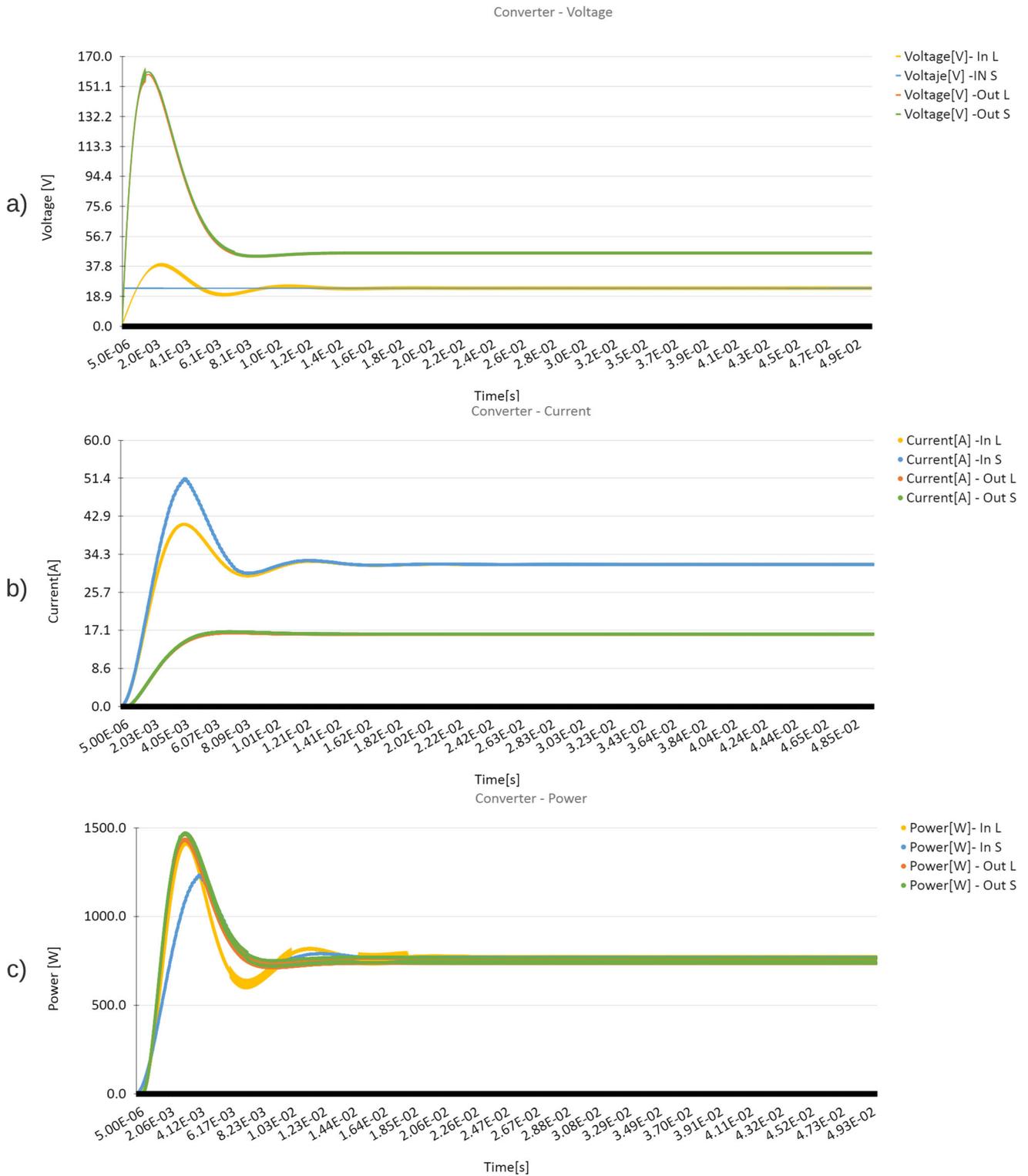


Fig. 13 Inputs and outputs of the Boost converter: **a** Voltages, **b** Currents, **c** Powers

Table 3 Input values in the converter variables used for the 240 W and 330 W panels

Variable	Value panel 240 W	Value panel 330 W
$R[\Omega]$	25	10
$h[\mu s]$	5	5
$L [mH]$	20	20
$T_m [ms]$	90	90
$C[\mu F]$	4	4
$ESR[\Omega]$	0.001	0.001

6 Results

6.1 Results of the Photovoltaic Systems

Table 4 shows the output variables of the system implementing in Fig. 9.

To find the relative error produced by this simulation, the reference values of the PV Array of Simulink are used. It is observed that despite using small data type configuration, given the need to minimize resources within the FPGA, values are obtained very close to this experienced simulator, it is important to show that the relative error of the variables is less than the 0.1. Using Eq. (16), the relative error is found for different levels of irradiance at constant temperature for each of the panels, these are summarized in Table 5.

Table 5 shows a relative error of each power curve, where it is observed that for low irradiance values the error

is higher and when the irradiance increases the error becomes smaller, for irradiance values greater than $600 \frac{W}{m^2}$ the error is less than 2%. In addition, in Figs. 10 and 11 it is observed that when the curves approach high voltage values the error increases, when the curve saturates and when the power starts to drop its behavior is influenced by the saturation current; in these two cases, it is because the blocks need a configuration for a fairly high level of accuracy, which in FPGA terms requires more space in the design. It was found that by varying this value the error in these parts of the curves can be reduced at the cost of space in the FPGA, since large blocks are needed for the implementation of such low values, as the values increase, the implementation becomes simpler. After this compilation process, results are verified and compared with the simulated variables previously analyzed and data was obtained with the same value for each simulation, due to choosing the appropriate data type configuration.

6.2 Results of the Boost Converter

Given the conditions specified by the PWM, it can be seen that the values of the signals when they take a stable value, meet the standards required for a type of converter of this type. Since it doubles the voltage signal, the current is reduced to the half and the power is almost the same. It is reduced a bit due to the associated losses of the element, satisfying the expectations of the implementation. The results are observed in Table 6.

Using Eq. (16), the error between the converter power curves obtained from Simulink and Labview of Fig. 14 is

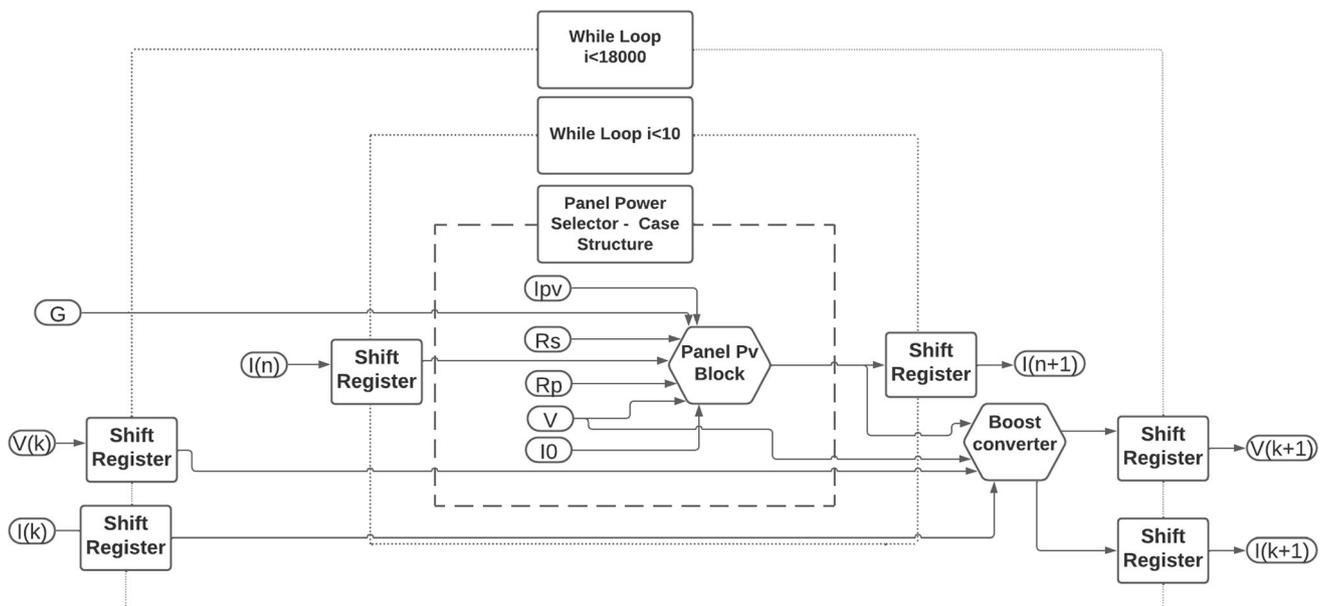


Fig. 14 Block diagram of the unified system

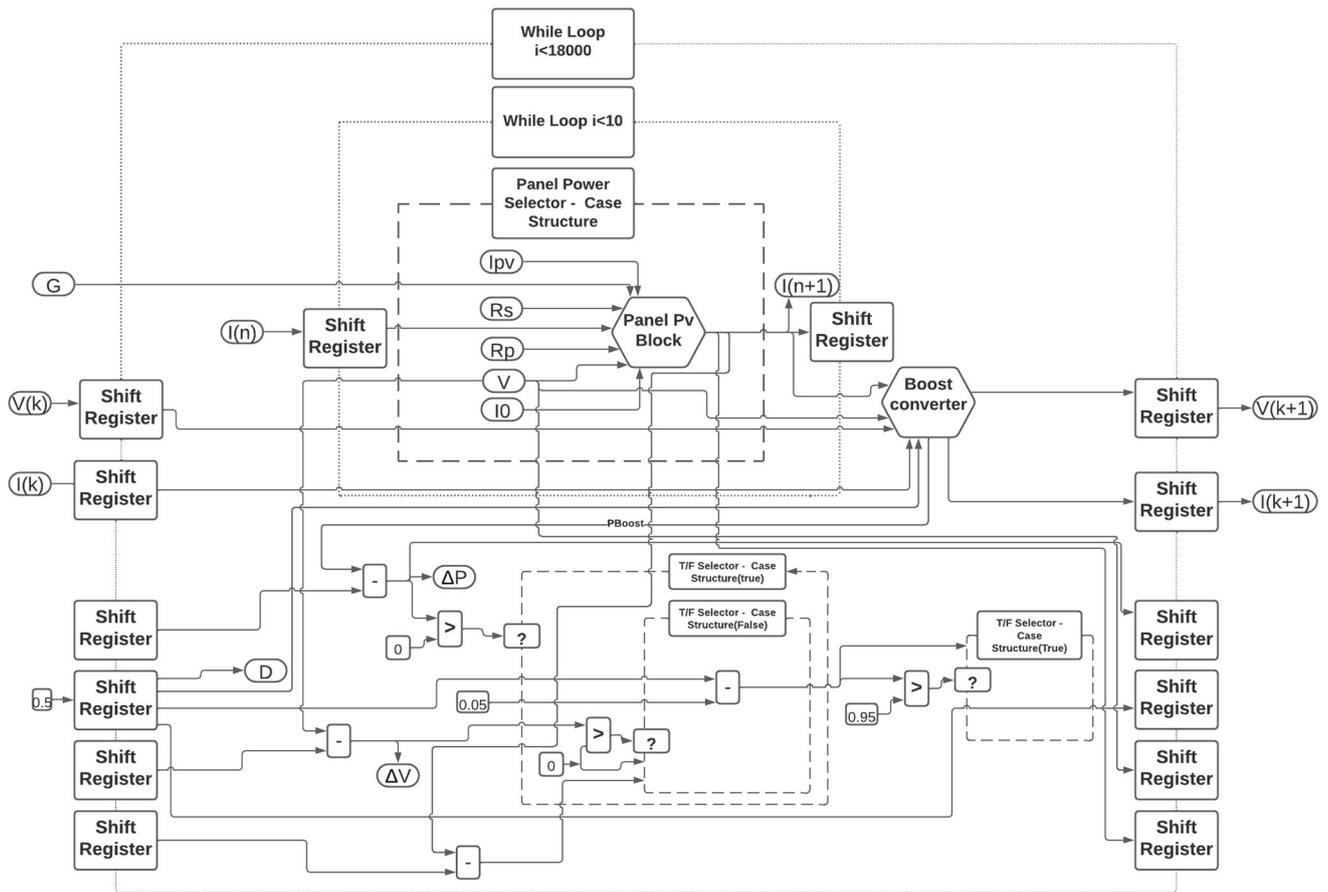


Fig. 15 Block diagram of the unified system with algorithm P & O

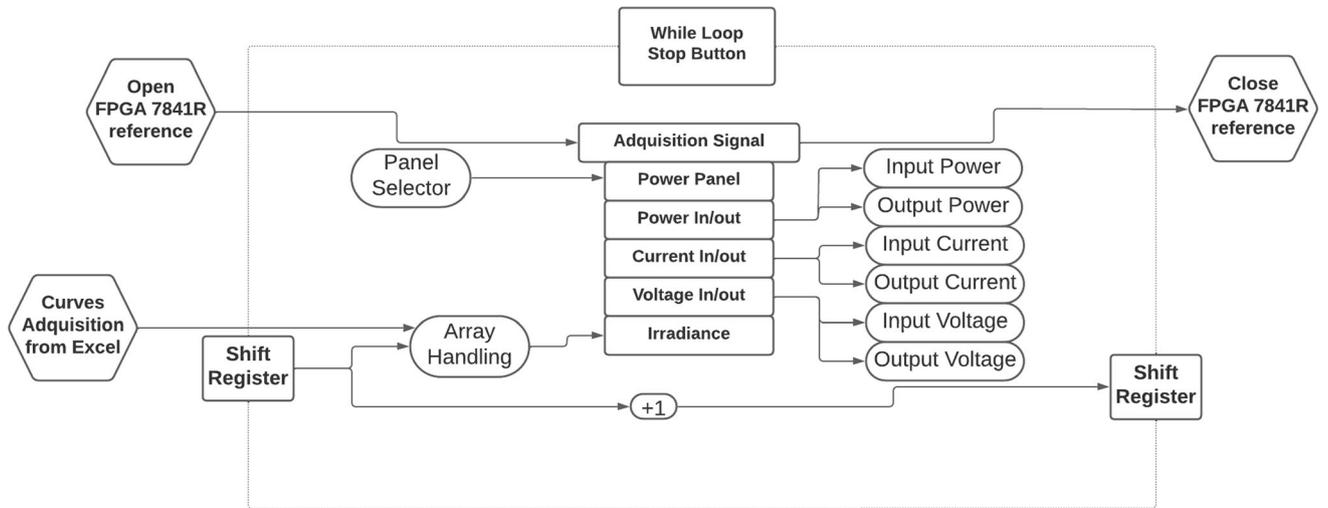


Fig. 16 Block diagram of the irradiance signal acquisition system and implementations on FPGAs

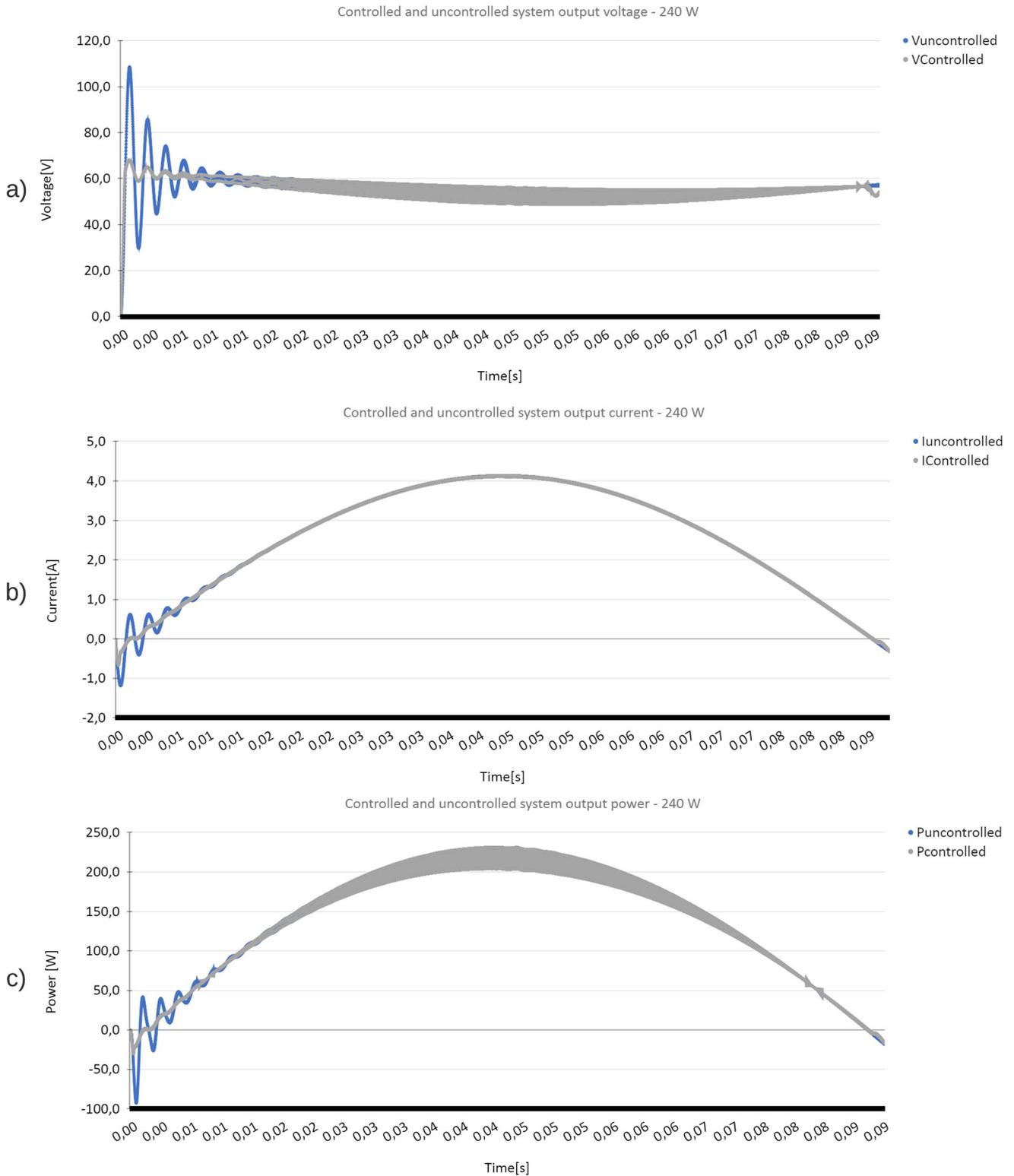


Fig. 17 Controlled and non-controlled voltage, current, and power comparative diagrams for a 240 W panel: **a** Voltage, **b** Current, **C** Power

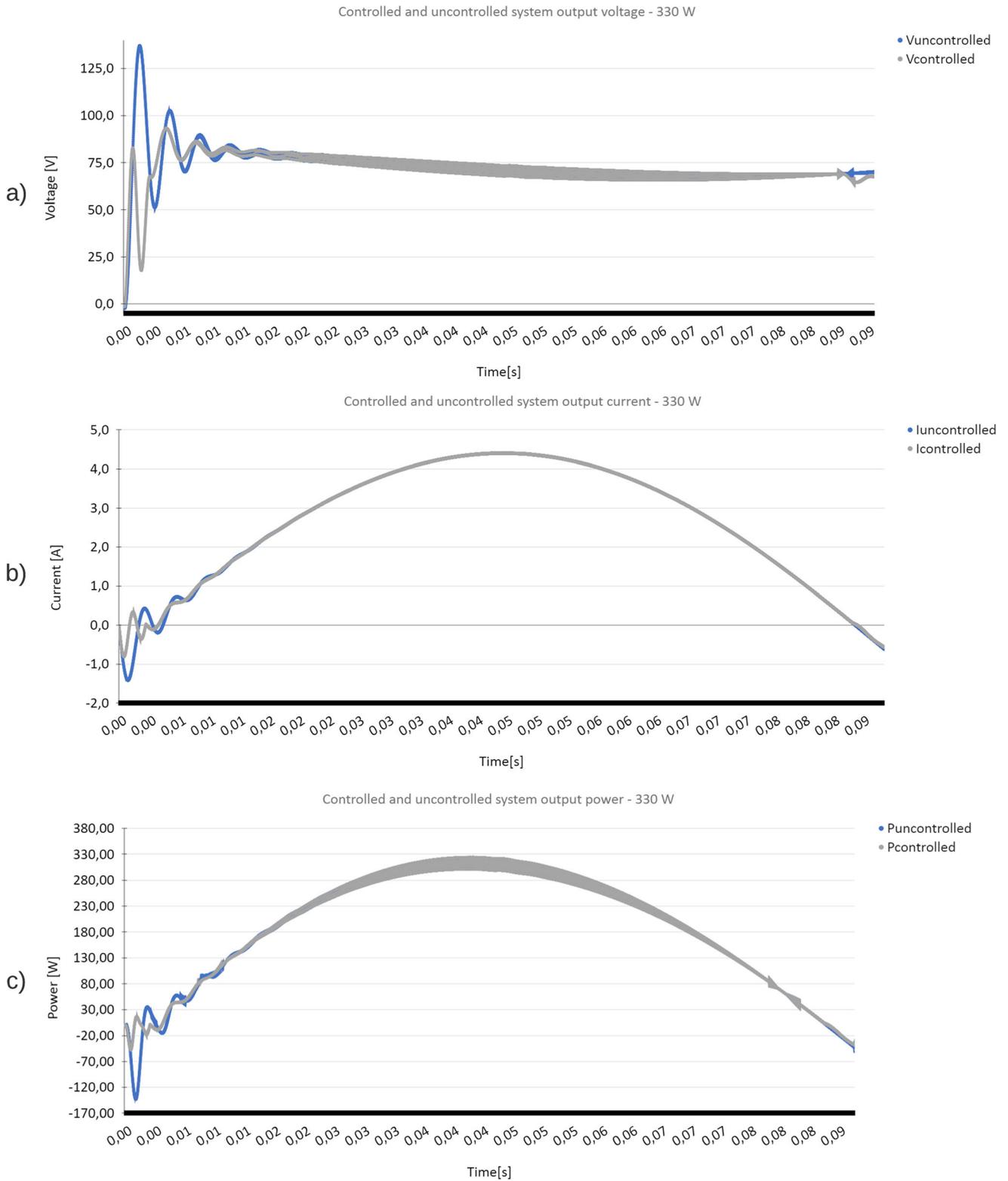


Fig. 18 Controlled and non-controlled voltage, current and power comparative diagrams for a 330 W Panel: **a** Voltage, **b** Current **C** Power

Table 4 Output values for the simulation of the 240 W and 330 W panels under nominal conditions ($T_n = 25^\circ\text{C}$, $G_n = 1000 \frac{\text{W}}{\text{m}^2}$)

Variable	Value PV (240 W)	Value PV (330 W)
I [A]	8.076	8.647
V [V]	29.699	38.148
P [W]	239.849	329.878

Table 5 Relative error in power curves with different irradiation values

Irradiance	PxV 240 W	PxV 330 W
$200 \frac{\text{W}}{\text{m}^2}$ [%]	5.388	15.968
$400 \frac{\text{W}}{\text{m}^2}$ [%]	0.685	4.290
$600 \frac{\text{W}}{\text{m}^2}$ [%]	0.606	1.624
$800 \frac{\text{W}}{\text{m}^2}$ [%]	0.843	0.339
$1000 \frac{\text{W}}{\text{m}^2}$ [%]	1.681	1.613

Table 6 Summary of the average values of the inputs and outputs of the converter

Variable	In	Out
Voltage [V]	24.032	47.856
Current [A]	32	15.952
Power [W]	769.0304	765.146

Table 7 Summary of the performance of the system with control and without control

Variable	System 240 W	System 330 W
$P_{(PO)}$ [W]	12.2297	16.8279
$P_{\text{Converter}}$ [W]	12.2032	16.7142
η [%]	2.1650	6.8035

found, it is observed that for the inputs there is an error of 0.622%, while for the output there is an error of 0.819%, this gives a good balance to the implementation made in Labview that delivers errors less than 1%. Once this process has been carried out, it is decided to compile the model within the FPGA, with the results that will be reflected at the end of this report.

6.3 Final Results

Similar behaviors occur in the operation of the two panels, uncontrolled voltages are observed to have a greater oscillation when irradiance values are low, and when values close to the nominal are reached, the system has a good behavior, something similar happens with the current and power graphs; this algorithm MPPT presents a good follow-up of the reference signal, allowing the values to be close to those expected for each irradiance value. For the operation of the controls of each curve, Eq. (17) is used, through the integrals for the two curves the performance is found, in Table 7 is the data found for this simulation.

In Table 7, it is observed that the performance improves with the use of the algorithm MPPT, for the case of the 240 W system by 2.165% and for the 330 W system by 6.80%, which for large generation blocks are significant values that would give value to implementation. Table 8 summarizes the space used on the FPGA for the different implementations made; the individual implementations of the panel and the converter have high values since in the different tests they were carried out with controlled variables and indicators were used in different sections of the implementation. In addition to using “Word length of data type configuration” values that are large enough to obtain high reliability in the results, after performing this process and observing the behavior of the variables, the panel and the converter are joined and subsequently the control to reduce the weight of the implementation. Also, the controlled variables are converted into constants and the value of the “Word length” is reduced to the values that allow obtaining a good sensitivity in the tests, in this way it is observed that the joined systems have smaller spaces. Another important point is that the compiler used by Labview performs an optimization in the distribution of resources in the FPGA, since the software allows optimizing around FPGA space or simulation time. It is important to mention that the implementation was carried out on a Virtex 5 LX30 system (National Instruments Corporation 2009) that has 4800 slices, 19200 slice registers, 19200 slice LUTs, 32 block RAMs, 32 DSP48s, and in timing a MiteClk by 33 MHz and an Onboard Clock by 40 MHz. In Table 8, it should be noted that the final results for the complete system show a use of approximately half of the FPGA space, which given the complexity of the models and algorithms is quite remarkable that will also allow in the future to include new parts that can be implemented in this system, which by acquiring data from Labview will facilitate HIL.

Table 8 Space used by the FPGA in the implementations

Variable	Panel PV	Boost converter	Panel + converter	Panel + converter + MPPT
Total slices [%]	40.8	41.6	54.9	55.8
Slice registers [%]	18.7	16.6	20.8	22.2
Slice LUTs [%]	31.2	31.3	42.5	43.5
Block RAMs [%]	0	0	0	0
DSP48s [%]	87.5	100	81.2	81.2
Max MiteClk [MHz]	105.16	96.74	110.10	107.38
Max onboard clock [MHz]	40.25	46.54	42.01	42.12

Table 9 Space used by the FPGA in different implementations

Article	Description	FPGA	FPGA space (LUT)	FPGA space used (LUT)	FPGA space (Registers)	FPGA space used (Registers)
Çelikel and Gundogdu (2021)	MPPT P &O	EP4CE115F29C7	114.480	456	114.480	195
Çelikel and Gundogdu (2021)	MPPT INC			5125		195
Morales-Caporal et al. (2018)	MPPT MPC			40.339		7.246
Tornez-Xavier et al. (2013)	NN Optimization	XC6SL34	27.288	977	54.576	2.857
Own Implementation	Panel	Virtex 5 LX30	19.200	5.990	19.200	3.590
	Converter			6.010		3.187
	Panel+Converter			8.160		3.994
	Panel+Converter+P &O			8.352		4.262

7 Comparison with Previous Studies

It is difficult to establish a reliable comparison between the space used on the FPGAs and their respective models, since the simulations depend on the configuration of each device and the factors that are prioritized in each compiler. For example, in Labview there are two ways to compile, the first is for the simulation time and the second for the space used on the FPGA and each option returns different values. In addition to the extrapolation of the models and the way of implementing them, the results also modify the space occupied within the FPGA. In the literature, most articles focus on the results obtained using an FPGA, but there are few that address its performance regarding the space used in the FPGA for the different applications, in this study some studies were found that focused on this matter. In Table 9, there is a compilation of different studies where FPGAs were used. The study citation, a description, the FPGA used, the number of LUTs and

registers it contains, and the LUTs and registers used are listed.

Among the articles that were compiled in Table 9, in Çelikel and Gundogdu (2021) that show a study carried out for a 10 kW panel system with Trina TSM panels - 250PA05.08 which are 250W arrays for MATLAB/Simulink and implemented in a DE2-115 Altera Cyclone IV EP4CE115F29 FPGA, in this study the PO and INC control models are implemented independently. In Tornez-Xavier et al. (2013), a 5W panel system is proposed that uses neural networks in MATLAB to optimize the weights and deviations to improve the modeling of the response of a solar panel, this system is implemented in a Spartan 6 XC6SL45 FPGA with VHDL and the XILINS ISE. And in Morales-Caporal et al. (2018), a system is proposed that solves the disadvantages of P &O and INC, which is model predictive control (MPC) and is responsible for compensation of the nonlinearities associated with power electronic converters; it offers a fast dynamic response, with a high margin of stability, allowing it to operate in dynamic

environmental conditions. This system produces 880W maximum theoretical and was developed using MATLAB/Simulink on an Altera Cyclone IV EP4CE115F29E DE2-115 FPGA. The last four systems are those carried out in this paper. In Table 9, it can be seen that the FPGA used in the own implementation contains less resources compared to others that have large space, achieving in addition to the implementation of the P &O algorithm, a photovoltaic panel and a Boost converter and that this system is fully functional, since being divided by blocks can easily interact with the physical outputs of the FPGA. As explained in the article that every time an additional block is added the system is optimized; it can be seen that the space occupied by the P &O system is 192 LUTs, which represents a little less than half that of the one implemented in Çelikel and Gundogdu (2021), which is a very similar system and at the level of registers 268 were used, 73 registers more than those were used in the same comparative article. It is a good general balance within the FPGA used, since the control system occupies a value close to 1% of your total space, which is a very similar system and at the level of registers 268 were used, 73 registers more than those used in the same comparative article. It is a good general balance within the FPGA used, since the control system occupies a value close to 1% of your total space.

8 Conclusion

In this article was presented a strategy to implement and to interact with the different variables of a photovoltaic system, a Boost converter, and a P &O modified algorithm; this starts from their mathematical models and their respective solutions. The Labview FPGA module only works with basic operations so operations like Euler or natural logarithm are not supported, due to their fixed-point representation and their significant digits. Therefore, for its representation, it was necessary to create a block that approximates these functions, limiting their significant digits, but obtaining reliable margins.

The use of a system such as Labview offers the advantage of being able to easily observe the behavior of the system when changes are made. When these models are compiled using the Labview FPGA module, it allows the variables to have a physical output. In this work, it was decided to read the variables from the FPGA. In future work, it is anticipated that the system will interact with physical signals through hardware-in-the-loop (HIL) testing. It is important to note that the implementation conducted in this study is not the only method and can also be carried out using other tools such as MATLAB/Simulink or any HDL language. In this particular study, Labview was utilized due to its flexibility in graphical programming and

its ability to generate a user-friendly graphical interface. Additionally, the seamless interconnection with instrumentation and hardware systems allows for easy integration with other signal acquisition systems, facilitating future work as described.

It was observed that compared to existing stable models such as those used by Simulink, the implemented models have a good behavior, with small errors in the data sweep, a better performance can be obtained from these implementations using a more robust system that allows a larger word size in the data type configuration, which means more accuracy in the models. It can be noted that this type of information (required space in an FPGA) is not very common in implementations made with Labview, the articles that focused on the space used by the FPGA in other environments were summarized in the paper, showing the features and the implementations that were carried out. In addition, it was possible to observe the behavior of the P &O algorithm and the effect it causes on the system, considerably improving the performance of the systems, reducing the oscillation, and extracting the greatest amount of power from the system.

Acknowledgements This work was supported by CEIBA Foundation

Author Contributions OB was involved in conceptualisation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft. AB helped in conceptualization, formal analysis, supervision, writing—review. AP assisted in conceptualization, formal analysis, supervision, writing—review. JR contributed to conceptualization, formal analysis, supervision, writing—review.

Funding Open Access funding provided by Colombia Consortium.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adler F, Benigni A, Stagge H, Monti A, De Doncker Rik W (2012) A new versatile hardware platform for digital real-time simulation: verification and evaluation. In: 2012 IEEE 13th workshop on

- control and modeling for power electronics (COMPEL). IEEE, pp 1–8
- Bélangier J, Venne P, Paquin J-N (2010) The what, where, and why of real-time simulation. Planet RT, pp 37–49
- Bin L, Xin W, Figueroa H, Monti A (2007) A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *IEEE Trans Industr Electron* 54(2):919–931
- Butcher JC (2008) Numerical methods for ordinary differential equations. Wiley, Chichester
- Çelikel R, Gundogdu A (2021) Comparison of PO and INC MPPT methods using FPGA in-the-loop under different radiation conditions. *Balkan J Electr Comput Eng* 9(2):114–122
- Duman E, Can H, Akin E (2014) FPGA based hardware-in-the-loop (HIL) simulation of induction machine model. In: 2014 16th International power electronics and motion control conference and exposition, number August. IEEE, pp 616–621
- Estrada L, Vázquez N, Vaquero J, de Castro Á, Arau J (2020) Real-time hardware in the loop simulation methodology for power converters using LabVIEW FPGA. *Energies* 13(2):373
- Fernández H, Martínez A, Guzmán VM, Giménez María I (2008) Modelo genérico de celdas fotovoltaicas Universidad, Ciencia y Tecnología, 12(48):157–162
- Francis WK, Beevi SS, Mathew J (2014) MATLAB/simulink PV Module Model of P & O And DC Link CDC MPPT algorithms with Labview real time monitoring and control over P & O technique. *Int J Adv Res Electr Electron Instrument Eng* 92–101
- González JMM, Daza CAD, Ureña CHG (2008) Análisis del esquema de generación distribuida como una opción para el sistema eléctrico colombiano. *Revista Facultad de Ingeniería* 44:97–110
- Granda-Gutiérrez EE, Orta-Salomón OA, Díaz-Guillén JC, Jimenez MA, Osorio M, González MA (2013) Modelado y Simulación de Celdas y Paneles Solares. ISSN:1405-2172. *Congr. Int. Ing. Electrón. Mem. Electro* 2013, (October):17–22
- Hayat A, Faisal A, Javed MY, Hasseb M, Rana Riaz A (2016) Effects of input capacitor (cin) of boost converter for photovoltaic system. In: 2016 international conference on computing, electronic and electrical engineering (ICE Cube). IEEE, pp 68–73
- Hohm DP, Ropp ME (2000) Comparative study of maximum power point tracking algorithms using an experimental, programmable, maximum power point tracking test bed. In: Conference record of the twenty-eighth IEEE photovoltaic specialists conference-2000 (Cat. No.00CH37036), volume 2000-Janua. IEEE, pp 1699–1702
- Hua C, Shen C (1998) Comparative study of peak power tracking techniques for solar storage system. In: APEC '98 thirteenth annual applied power electronics conference and exposition, vol 2. IEEE, pp 679–685
- Jonke P, Stockl J, Miletic Z, Brundlinger R, Seitzl C, Andren F, Lauss G, Strasser T (2016) Integrated rapid prototyping of distributed energy resources in a real-time validation environment. In: 2016 IEEE 25th international symposium on industrial electronics (ISIE). IEEE, pp 714–719
- Kazmierczuk MK (2008) Pulse-width modulated DC–DC power converters. Wiley, Chichester
- Li P, Wang Z, Wang C, Xiaopeng F, Hao Yu, Wang L (2017) Synchronisation mechanism and interfaces design of multi-FPGA-based real-time simulator for microgrids. *IET Gener Transm Distribut* 11(12):3088–3096
- Mahendran V, Ramabadran R (2017) Impact of modified perturb and observe control on MPPT of PV/battery fed three—port DC-DC converter. *Acta Sci Technol* 39(3):313
- Makinen AS, Messo T, Tuusa H (2014) Power hardware in-the-loop laboratory test environment for small scale wind turbine prototype. In: 2014 16th European conference on power electronics and applications. IEEE, pp 1–10
- Matar M, Saeedifard M, Etemadi A, Irvani R (2011) An FPGA-based hardware-in-the-loop simulator for multilevel converter systems. In: International conference on power systems transients (IPST2011), pp 1–6
- McLaren P, Nayak O, Langston J, Steurer M, Sloderbeck M, Meeker R, Lin X, Yu M, Forsyth P (2011) Testing the “Smarts” in the Smart T & D Grid. In: 2011 IEEE/PES power systems conference and exposition. IEEE, pp 1–8
- Morales-Caporal M, Rangel-Magdaleno J, Peregrina-Barreto H, Morales-Caporal R (2018) FPGA-in-the-loop simulation of a grid-connected photovoltaic system by using a predictive control. *Electr Eng* 100(3):1327–1337
- Moslener U, McCrone A, D’Estais F, Grüning C, Louw A, Boyle R, Strahan D, Collins B, Stopforth K (2017) Global trends in renewable energy investment 2017
- National Renewable Energy Laboratory (NREL) System Advisor Model (SAM)
- National Instruments Corporation (2009) NI R Series Multifunction RIO User Manual
- Ospino CA, Adalberto Carlos RA, Pabón D (2014) Modelado y simulación de un panel fotovoltaico empleando técnicas de inteligencia artificial. *Ingeniería Energetica* 35(3):225–233
- Parma GG, Dinavahi V (2007) Real-time digital hardware simulation of power electronics and drives. *IEEE Trans Power Delivery* 22(2):1235–1246
- Snider L, Belanger J, Nanjundiah G (2010) Today’s power system simulation challenge: high-performance, scalable, upgradable and affordable COTS-based real-time digital simulators. In: 2010 Joint international conference on power electronics, drives and energy systems and 2010 power India. IEEE, pp 1–10
- Tornez-Xavier GM, Gomez-Castaneda F, Moreno-Cadenas JA, Flores-Nava LM (2013) FGPA development and implementation of a solar panel emulator. In: 2013 10th International conference on electrical engineering, computing science and automatic control (CCE), number 405. IEEE, pp 467–472
- Vacheva G, Hinov N, Zlatev Z (2018) Modelling of DC/DC boost converter in visual programming environments. In: 2018 41st International spring seminar on electronics technology (ISSE), vol 2018-May. IEEE, pp 1–4
- Valderrama F, Moreno H, Vega HH, Henry MC, Héctor HV (2013) Análisis, simulación y control de un convertidor de potencia DC-DC tipo boost. *Ingenium Revista de la facultad de ingeniería*, 12(24 SE - Artículo de investigación científica):1–12
- Villalva MG, Gazoli JR, Filho ER (2009) Modeling and circuit-based simulation of photovoltaic arrays. In: 2009 Brazilian power electronics conference, vol 14. IEEE, pp 1244–1254
- Washington C, Dolman J (2010) Creating next generation HIL simulators with FPGA technology. In: 2010 IEEE AUTOTEST-CON. IEEE, pp 1–6
- Xilinx (2009) System Generator for DSP User Guide, vol 640
- Yamane A, Abourida S (2015) Real-time simulation of distributed energy systems and microgrids. In: 2015 International conference on sustainable mobility applications, renewables and technology (SMART). IEEE, pp 1–6