FPGA-based real-time simulation of mismatched photovoltaic arrays

Alejandro Carabali-Isajar a, Martha Lucia Orozco-Gutierrez a,b, Edinson Franco-Mejía a, Giovanni Spagnuolo b, Jose Alex Restrepo c

a Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Calle 13 #100-00, Santiago de Cali, 76001, Colombia
b Università degli Studi di Salerno, Via Giovanni Paolo II 132, Fisciano (SA), 84084, Italy
c Universidad Simón Bolívar, Valle de Sartenejas, Caracas, 1080-A, Venezuela

A R T I C L E   I N F O

Keywords:
Real-time simulation
Mismatching conditions
Digital hardware architecture
Parallelism

A B S T R A C T

This paper proposes an approach to the real time simulation of photovoltaic (PV) arrays that are subjected to mismatching conditions, e.g. partial shadowing. The method, which has been named Model by Zone (MBZ), adopts the best PV model depending on the operating conditions of the cells in the module: it switches among single-diode model (SDM), linear model and constant voltage model. An optimized digital hardware architecture exploiting parallelism of operations over a FPGA system is exploited to effectively implement the proposed model. It reduces the computation time and the use of hardware resources. The good trade-off between accuracy and computation time of the proposed technique has been demonstrated in two cases of study: by evaluating the long-term PV power production of a PV field subjected to dynamic shadowing conditions and by analyzing the model performance in a maximum power point tracking (MPPT) application. In the former case, the proposed approach improves the computation time by 182.5 % with respect to methods that are available in recent literature, with a Relative Error (RE) at the Global Maximum Power Point (GMPP) lower than 0.39 %. In the MPPT application, the proposed technique allows to achieve a MAPE of 0.0319 % and 0.1892 % in the string voltage and power calculation, respectively.

1. Introduction

In the transition towards renewable energies, photovoltaic (PV) and wind installations play a significant role being the technologies, not including hydropower, that have experienced the highest growth in recent years. In 2020 the solar PV capacity in the world reached 760 GW [1]. Although, the last two years of pandemic negatively impacted the renewable energy market, several stand-alone solar-hybrid systems are being installed as an energization solution in remote and rural areas of developing countries [1].

A PV installation mainly consists of a PV generator and electronic converters to drive the power. A PV generator is composed of PV modules connected in a particular way, being the most typical series-parallel interconnection. PV installations suffer from power losses due to mismatching conditions, generated mainly by shadowing. A PV panel manufacturer installs bypass diodes among cells to reduce power loss during mismatching conditions. But inflection points appear in the I-V curves, increasing the computational burden in maximum power point tracker algorithms [2]. In the design stage of PV installations, simulation tools’ features impact the accuracy of power predictions and the time of studies. Although some commercial software such as Matlab provides a block to simulate a PV array, it is expensive. Moreover, the simulation of large installations with mismatched conditions needs granular PV modules, which take a long time to test. This problem has led several authors to propose algorithms for the simulation of large PV installations. For example, in [3] a method for obtaining the I-V curve of strings during mismatching conditions is presented. The lossless single-diode model (SDM) describes the behavior of each module, and a Newton-Raphson (NR) method allows to solve a set of non-linear equations to represents strings. That proposed technique was validated in Matlab using a PV array of four strings of nine modules each one. In [4], the SDM including series and parallel resistances has been used to obtain the string I-V curve. The non-linear equations solved by the NR method run faster with an explicit inverse Jacobian matrix using the Shür complement. However, the computation time for simulating a string of 30 modules is close to twelve seconds in Matlab.

* This article is a part of the “Design, Operation and Reliability of Large PhotoVoltaic Systems” Special issue.
* Corresponding author.
E-mail address: martha.orozco@correounivalle.edu.co (M.L. Orozco-Gutierrez).

https://doi.org/10.1016/j.heliyon.2022.e09969
Received 30 March 2022; Received in revised form 16 May 2022; Accepted 13 July 2022

2405-8440/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
On the one hand, there is a need for real-time simulation methods as an alternative for building a complete photovoltaic setup. Besides, several problems arise from operating PV strings in mismatched conditions. Among these problems, the more relevant are developing strategies for maximum-power-point tracking (MPPT), power converter topologies able to search all the photovoltaic string curves, and topologies for string reconfiguration among others [5, 6]. On the other hand, developing real-time simulations of PV strings must consider different intricacies involved in the simulation. For example, convergence time, accuracy, computational cost, and technology price [7, 8]. Table 1 shows the features of real-time simulation methods proposed in literature, those implemented in embedded systems. In all of them, the SDM has been used to represent the behavior of the PV modules. The second and third columns show the PV generator and mismatching/homogeneous conditions used to validate those methods. Also, other columns show the technology, accuracy and convergence time. In [9, 10], a solution at module level subjected to homogenous or mismatching conditions is given. Meanwhile, in [11] and [12] methods to simulate strings under mismatching conditions have been proposed and more practical situations are afforded therein. The proposed methods have been implemented in embedded high level digital systems such as SoC, FPGA or DSP, which impacts the convergence time. However, the accuracy depends more on the solution method of the PV models. For example, [12] and [11] implemented the proposed NRJ method of [4] in a DSP and SoC systems, respectively. The authors in [12] solve the inverse jacobian matrix in an NR method sequentially. In that work, a PV array simulation for evaluating one point of the I-V curve, using an ADSP21369 board, took less than 34.75 μs using IEEE single precision-32 bits, and 8.4 ms with IEEE double precision-64 bits. The accuracy, measured in terms of the mean absolute percentage error (MAPE), were 0.0315 % and 0.0297 % for the corresponding cases. Meanwhile, [11] is focused on implementing the LambertW function in the XC7Z020 SoC system using a high level synthesizes tool to translate C++ code to a Hardware Description Language (HDL). The other functions such as the NR method with the inverse jacobian matrix calculation proposed in [4], was executed in a Cortex ARM processor available in the SoC. A mismatched PV string of 30 modules was used as validation case, this spending 3.69 ms for computing 722 points of the string I-V curve, that is an average of 5.11 μs for each point. In that work, the accuracy measured as Mean Percentage Error was lower than 0.2 × 10⁻⁴ %. Therefore, the implementation of [4] proposed in [11] is faster than the one proposed in [12], due to the capabilities of digital systems used and the parallelism of the digital implementation.

Regarding real-time simulation methods at module level, the authors in [9, 10] were able to represent homogeneous conditions, although expensive hardware has been chosen. The authors in [9] used the implicit SDM, but they did not specify the numerical solution method employed. The XSG design tool from Xilinx allowed to generate automatically the FPGA code from a Simulink model. The relative errors (RE) for the I-V curve in the linear and nonlinear zone were between 1 % and 13 % respectively. The errors were impacted by the fixed point data implementation on the Virtex5 FPGA system. Meanwhile the authors in [14] used an RT-LAB system. Their proposed method allows to solve the explicit SDM using the LambertW function, which is developed with the Halley formulation (LWH). The results are compared with a data-set, obtaining a Mean Relative Error (MRE) of 0.62 %. An optimization in the calculations, which consists on precalculate fixed or constant values (PFV) involved in the SDM, allows to save computation time. The time spent computing each point of the I-V curve was 2.45 μs using the LWH, and 1.75 μs using the LWH in combination with the PFV. The authors in [13] proposed a solution implemented in a MVC33 DSP system. They used an optimization for a division operation involved in the NR computation. For this, two main alternatives were evaluated: successive subtractions and NR to determine an inverse, which converge in 34 μs and 11 μs, respectively. The coding of the method was done in assembly language. The calculation time to obtain one point of the I-V curve was 50 μs. So, more calculation time is required by the DSP than for the FPGA implementations, for either the string or the module levels.

On the other hand, [10] provides a technique able to simulate a PV module under mismatching conditions with or without bypass diode. The SDM model is used, in which the photo-induced current is affected by a factor representing the shaded and unshaded cells. The derivative discrete of the function f, which is defined in the NR method, is used to decrease the computational effort of the calculations. A graphical Systems Modeling Language (SysML) to implement the PV model, and the High Level Specification of Embedded Systems (HiLEs) software for automatic VHDL code generation were used for implementing the system. The SysML tools allow to define the components employed to implement the model in an XC7Z020 FPGA system, and taking advantages of parallel processing. Although, the graphical SysML facilitates the programming procedure, there is not complete control of the digital architecture to be executed in the FPGA system. The accuracy measured in terms of Mean Square Error (MSE) was 0.096 × 10⁻⁴, and the calculation time to obtain one I-V point was 8.3 μs.

Still the SDM is used in the methods presented in Table 1, the accuracy feature differs in all of them. The errors do not depend on the type of digital processor system, rather it depends on data precision and the numerical method used to solve the string model. On the other hand, the computation time depends on the type of digital processor system, the digital architecture of the solution and the numerical solution. SoC and FPGA systems are able to compute in parallel simple or complex functions, and the digital architecture is developed by a software or by a programmer. Although, there are different tools to generate optimized FPGA code from Simulink, an expert programmer could provide a better solution in terms of time and resources. In general, trying to maintain an acceptable cost for the embedded system used in the implementation, accuracy is sacrificed to improve the computation time. However, the evaluation of an adequate trade-off between computation time and accuracy depends on the application.

The present work proposes a method called Model by Zone (MbZ), to simulate real-time mismatched PV strings. The model representing the modules in the PV installation commute among three possibilities,
i.e., SDM, linear, and fixed voltage. This is possible because the string voltage is obtained using the contribution voltage of each module in the string, and the magnitude of that contribution varies depending on which zone of the I-V curve the module is working on. Modules working at the right side of the I-V curve knee (see Fig. 3), have a contribution voltage to the string less than the contribution provided by modules working at the left side. Therefore, for modules contributing with large voltages, the SDM representation provides the best accuracy. Meanwhile, modules contributing with small voltages could be represented by a linear approximation, which do not produce a large impact on the error. At the same time, a fixed voltage, which corresponds to the bypass diodes in the on state, is enough representation for the bypassed modules. The proposed commutation of the models avoids the use any numerical method to solve the string model, and the solution is analytic. Additionally, this computation time is saved due to a hand-developed optimized digital hardware architecture, which exploits the parallelism feature of FPGA systems. The proposed architecture defines dedicated hardware to develop particular sub-functions of the model. The way in which these sub-functions are distributed allows partial results of these sub-functions to be used in a different sub-function, which improves the total execution time of the model. Because, the proposed method uses the explicit form of SDM with the Lambert W function, which has to be computed for each I-V point, then one of the main sub-functions does its computation. Also, an optimized digital hardware architecture has been proposed for developing the Lambert W function, which uses basic elements such as registers, multiplexers, and a control unit to configure the data flow. Additionally, the equations representing the model are written such that terms with no change between I-V points are grouped, to save hardware use. The performance of the MBZ, for representing a string under mismatching conditions, implementing the proposed hardware architecture in a medium level FPGA at 50 MHz, and using double-precision arithmetic, exhibits less computation time than those reported in the literature. Moreover, the good accuracy of the proposed method is demonstrated by using a real-time application such as MPPT.

A PV string installation subjected to a dynamic mismatching condition is used as case of study to show the good trade-off between computation time and accuracy provided by the proposed technique. The paper is organized as follows. Section 2 presents the mathematical model of the PV modules and strings, section 3 presents the proposed MBZ method to represent strings under mismatched conditions, then section 4 presents the optimized digital software architecture of the proposed method, and section 5 shows the execution time and accuracy. Paper ends with the conclusions.

2. Single-diode model (SDM) to represent PV modules and strings

Fig. 1 shows the PV’s single-diode model SDM [15] which is the most adopted model for representing the behavior of PV modules. Equation (1), obtained by applying theKirchhoff law, allows to obtain the output current I. The SDM has five parameters: diode saturation current $I_s$, series and shunt resistances $R_s$ and $R_{sh}$, thermal voltage $V_T$, and photo-induced current $I_{ph}$. Moreover, if a bypass diode $D_{sh}$ is included, two parameters are added.

![Fig. 1. Single diode model SDM used for representing a PV module.](image1)

![Fig. 2. PV string formed by groups of PV modules $G_j$ under different irradiance $S_j$ conditions.](image2)

$$I = -I_s \left( e^{\frac{V+I R_s}{R_{sh}}} - 1 \right) - \frac{V + I R_s}{R_{sh}} + I_{ph}$$

(1)

The LambertW($\Theta$) function allows the description of the SDM with an explicit equation [15], as shown in (2).

$$I = \frac{R_{sh}(I_{ph} + I_s) - V}{R_{sh} + R_s} - \frac{V_T \cdot LambertW(\Theta)}{R_s}$$

(2)

Where: $\Theta = \frac{R_{sh} R_s I_{ph}}{V_T R_s + R_{sh}}$ 

The series connection of several PV modules makes up a PV string, as shown in Fig. 2. This paper proposes ordering the PV modules in groups $G_j$, where $N_j$ modules are subjected to the same irradiance $S_j$ and temperature $T_j$. The PV modules in the same group (PVM) have the same short circuit current $I_{sc}$, modeled by equation (3), where $I_{scSTC}$ and $T_{STC}$ are the short circuit current and temperature at standard test conditions, and $a_j$ is the current variation coefficient, dependent on the temperature. Usually, the manufacturers provide the PV parameters at standard test conditions. A measurement system provides the irradiance $S_j$ and temperature $T_j$ or is assigned for the simulations as done in the present work.

The irradiance affects the PV modules’ open-circuit voltage $V_{oc}$, causing different open circuit voltages between groups in the string. Equation (4) gives each group’s $G_j$ open circuit voltage $V_{ocj}$, in which $N_j$ is the number of PV modules in the group. $I_{phj}$, $I_{scj}$ and $V_{cj}$ are the photo-current, saturation current, and thermal voltage of the group respectively.

$$I_{scj} = \left[ \frac{S_j}{S_{STC}} I_{scSTC} \left( 1 + a_j(T_j - T_{STC}) \right) \right]$$

(3)

$$V_{ocj} = N_j V_{cj} \log \left( 1 + \frac{I_{phj}}{I_{scj}} \right)$$

(4)

The string voltage $V_{ST}$ results of adding the contribution voltage of all groups in the string $V_{G1}, V_{G2}, \ldots, V_{Gj}$, as shown in Equation (5):

$$V_{ST} = N_j V_{cj} \log \left( 1 + \frac{I_{phj}}{I_{scj}} \right)$$

3
\[ V_{ST} = V_{G_1} + V_{G_2} + V_{G_3} + \ldots + V_{G_j} + \ldots + V_{G_j} \]  

(5)

Where: \( V_{G_j} = N_j \cdot V_j \)

Because in the previous string assumption, the PV modules in a group \( j \) have the same irradiance condition, then the current of a group is equal to the current of any PV module in that group. The string current \( I_{ST} \) is the same in any module or group in the string. Therefore, the voltage contribution \( V_j \) of any module in a group \( j \) is retrievable from the \( I_{ST} \)'s current expression, as shown in equation (6).

\[ I_j = I_{ST} = \frac{R_{h_j} (I_{ph_j} + I_j) - V_j \cdot \text{LambertW}(\Theta)}{R_{h_j} + R_j} \]  

(6)

\[ \Theta = \frac{R_{h_j} R_j}{R_{h_j} + R_j} \cdot \frac{I_j \cdot e^\left(\frac{-R_{h_j} (I_{ph_j} + I_j) - R_h}{R_j}ight)}{V_j} \]  

(7)

The literature has outlined some computational methods for solving the non-linear equation in (5), in which the groups are subjected to mismatching conditions, but they are time-consuming for large PV systems (4). The present paper proposes a method called Model by Zone (MbZ), which provides an analytic solution of (5), so a faster reconstruction of I-V curve is possible. In section 5 is demonstrated that the proposed technique is faster than the methods of literature reported in Table 1, at string level, and the error is lower than 1.5%.

3. Model by Zone Method (MbZ) to reconstruct the PV string I-V curve

Using a test/pilot example that employs three groups of modules subjected to different irradiance conditions, the proposed Model by Zone (MbZ) method used to calculate the I-V string characteristic is presented. The left side of Fig. 3 shows the I-V curves of three groups of modules \( G_1, G_2, \) and \( G_3, \) subjected to high, medium and low irradiance levels. The right-hand side of Fig. 3 shows the resulting I-V curve of those groups when connected in series. The string I-V curve shows inflection points at the short-circuit currents, \( I_{SC_1}, I_{SC_2}, \) and \( I_{SC_3}, \) of each PV group. Getting across the PV string I-V curve, from left to right at each inflection point, a group of modules starts working. This means that between \( I_{SC_1} \) and \( I_{SC_2}, \) the group subjected to the highest irradiance level, is working, and the groups \( G_2 \) and \( G_3 \) are in off state, then the contribution voltage for these last two groups corresponds to the sum of voltages of the activated bypass diodes, which is \( \sum V_{j}. \) In the range of current \( I_{SC_2} \) and \( I_{SC_1}, \) the groups subjected to high and medium irradiance levels are activated, and the group \( G_2 \) is in off state. In the last segment of the string I-V curve, this is from current \( I_{SC}, \) down to zero, the three groups of modules are activated, so all bypass diodes are in off-state. In each zone of the string I-V curve, which is limited between two short-circuit currents, there is a group of modules contributing more to the string voltage variations than the other groups, which could be named the dominant group. The group that has entered into activation on the right side of the zone is the dominant one. For example, between \( I_{SC_2} \) and \( I_{SC_1} \), the contributions from voltage variations of group \( G_1 \) are small compared to the contributions from voltage variations of \( G_2. \) Besides, the voltage contribution of \( G_1 \) is almost linear as seen in Fig. 3. In the same current interval, the voltage contribution of \( G_1 \) corresponds to the sum of bypass diode voltages in that group. A similar situation occurs in the current range between \( I_{SC_3} \) and zero, where the dominant group is \( G_2. \) Meanwhile, the voltage contribution of \( G_1 \) and \( G_2 \) is small and approximately linear.

The proposed MbZ method defines the set of equations shown in Table 2 to determine the contribution voltages of groups \( V_{G_j} \) depending on the string's current \( I_{ST} \) operating point. In Table 2, the current range varies by columns, and the voltage contribution of groups varies by rows. For example, column \( i = 1 \) corresponds to string interval current of \( I_{SC_2} \leq I_{ST} \leq I_{SC_1}, \) and the rows \( j = 1 \) until \( j = 3 \) gives the voltage contribution of groups \( V_{G_j} \), until \( V_{G_3}, \) for that range of current. The string voltage \( V_{ST} \) is calculated by adding the contribution voltage of groups, such as has been presented in (5). The number of current ranges in Table 2 is equal to the number of groups forming the string, e.g. three in the pilot example. Besides, those ranges of currents are determined by the short-circuits currents of groups.

Using Table 2 for the pilot example, the string voltage equations \( V_{ST} \) at the short circuit currents intervals are presented in (8). The short circuit current at \( J + 1 \) is always equal to zero, such is shown in Fig. 3.
The linear approximation of I-V curve is proposed as the difference of voltages at the short circuit currents divided by two, as it is observed in Table 2. For a particular \( V_{ST} \), which falls in a particular interval of current, the equation (8) has one unknown, e.g. \( V_{Gj} \), \( V_{G2} \) or \( V_{G3} \), so it can be found directly. Therefore, the solution of (8) does not require any numerical method. Once the voltage of a group has been calculated, e.g. \( V_{Gj} \), \( V_{G2} \) or \( V_{G3} \), and taking into account that \( V_{Gj} = N_j \cdot V_j \), then \( V_j \) is used in (6) to determine the string current \( V_{ST} \).

\[
V_{ST} = V_{Gj} + N_j \cdot V_d + N_S \cdot V_d, \quad I_{SC1} \leq I_{ST} \leq I_{SC2},
\]

\[
V_{ST} = \left\{ \frac{V_{i,1.2} + V_{j,1.2}}{2} \right\} + V_{Gj} + N_S \cdot V_d, \quad I_{SC3} \leq I_{ST} \leq I_{SC4},
\]

\[
V_{ST} = \left\{ \frac{V_{i,1.3} + V_{j,1.3}}{2} \right\} + \left\{ \frac{V_{i,2.1} + V_{j,2.1}}{2} \right\} + V_{Gj}, \quad 0 \leq I_{ST} \leq I_{SC5},
\]

(8)

As it has been reported in literature [16], the direct calculation of \( V_j(I_{un}) \) using the explicit expression (6), presents an overflow in the exponential term of the LambergW(9) function. Due this fact, the calculation of contribution voltages of each group at the short circuit currents of the other groups, they are \( V_{j,i}(I_{un}) \), are calculated by the implicit expression of SDM presented in (1), that solved by the Newton-Raphson method presented in (9). In (9), the guess condition \( V_{j,i}(I_{un}) \) is determined by the open-circuit voltage of the group, given by (4). Besides, the tolerance error of (9) is defined as \( \epsilon \leq |V_{j,i}(I_{un}) - V_{j,i}(I_{un})| \), which impacts the number of iterations; it is adjusted to reach the solution in six iterations \((k = 1, 2, ..., 6)\). The calculation of those \( V_{j,i}(I_{un}) \) must be realized just once, and they are saved in memory to be used in the calculation of the string voltage of Table 2.

\[
V_{j,i+1} = V_{j,i} - \frac{V_{j,i}}{\Delta V_{j,i}/\Delta V_{j,i}}
\]

(9)

Where: \( V_{j,i} = I_{j,i} \cdot e^{\frac{-V_{j,i}}{n_{j,i}} - 1} + V_{i,k} \cdot e^{\frac{-V_{j,i}}{n_{j,i}} + I_{j,i} - I_{i,j}} \)

\[
\Delta V_{j,i}/\Delta V_{j,i} = I_{j,i} \cdot e^{\frac{-V_{j,i}}{n_{j,i}} + I_{j,i} - I_{i,j}} + \frac{1}{n_{j,i}}
\]

The pseudo-code of the proposed MbZ method is shown in Algorithm 1. The method provides current and voltage points of a PV string \( V_{ST}(I_{ST}) \). The number of \( k \) points depends on the parameter setting of \( \delta \), which represents the variations of \( V_{ST} \). The parameters related to the string configurations are loaded in the first step, those are the number of groups \( J \) forming the string and the number of modules \( N \) forming each group \( j \). Besides, the irradiance \( S \) and temperature \( T \) conditions are configured for each group \( j \). The manufacturer SDM and bypass diode parameters, those required in equations (3), (4) and (6), are also loaded for each group \( j \). The short circuit currents of the groups are calculated by (3), then those are organized in descendent way. The voltages of each group at the short circuit current of the other groups, that is \( V_{j,i} \), are calculated once using (9). Next, sample \( k \) is initialized to one, which is incremented inside a while loop to calculate all \( K \) samples required to construct the string I-V curve. The first point to calculate is \( I_{ST} = I_{SC1} \), so the string I-V curve is constructed from the higher short circuit current until zero. For a particular string voltage \( V_{ST} \), the contribution voltages of groups \( V_{Gj} \) are calculated using Table 2. The unknown voltage \( V_{Gj} \), for \( j = i \), is obtained directly as it is shown in line 10 of the Algorithm 1. That voltage is used to calculate the string current \( I_{ST} \) using (6).

4. Optimized hardware architecture of MbZ method for a FPGA system

This section shows an optimized digital hardware design of the proposed MbZ that exploits the parallel calculation capacity of FPGA systems to improve the computation time and resource use. Also, FPGA features such as high precision control of the execution time, parallel execution paths, and the use of different clock frequencies [17][18] were capitalized. In FPGA implementations, in addition to high-level synthesis tools, the user can also employ a hardware description language (HDL) to code the digital hardware. The advantage of using a hand coded style is in the optimization incorporated by the user, thanks to its inner knowledge of the algorithm. By doing this, redundant calculations are avoided and dedicated functions are implemented by combining encoders, decoders, counters, multiplexers, state machines, among others, such that resources and execution time are saved. In contrast, using high-level synthesis tools save development time but sacrifices some possible optimizations. A detailed analysis about these aspects has been carry out in [19]. In the present work, the digital software architecture is optimized by using a hand coded style.

The proposed MbZ is able to reconstruct the string I-V curve in a fast way, which is helpful in long-term power production studies. Besides, a short computation time to determine a point of I-V curve is an important feature in real-time applications, such as Hardware-in-the-loop (HIL) to test control strategies or power converters. In the present section, the computation time and accuracy obtained from reconstructing an string I-V curve of a PV system using the proposed MbZ method are compared with those obtained from the NRIJ method [4]. Moreover, at the end of this section a modification of MbZ method is proposed, which allows to calculate a point of I-V curve, as required in real-time applications such as MPPT.

4.1. FPGA digital hardware architecture of the MbZ method

Fig. 4 depicts the proposed MbZ method, showing the high level partition into sub-blocks of the digital hardware, and the interconnection between the sub-blocks and the main block. In order to exploit the parallel computing capabilities of the FPGA, the MbZ algorithm was programmed with Verilog HDL. In Fig. 4, the sub-block \( I_{un} \) computes the \( I_{un} \) and \( V_{un} \) of the \( J \) groups forming the string. The \( V_{j,i} \) sub-block executes the Newton-Raphson algorithm presented in (9), to determine
the voltages of the J groups at the short-circuit currents. The \( \delta V \) sub-block calculates the contribution voltages of the J groups corresponding at the linear approximation zone, and the \( N \cdot V_d \) of the modules with bypass diode activated. Meanwhile, the \( LW(\theta) \) sub-block performs the LambertW(\( \theta \)) function involved in (6). The \( I_{\text{string}} \) block drives the previous described sub-blocks, so it is the control unit, which follows the state machine presented in Fig. 5. The different signals belonging to the block-diagram have been described in the Table 3. In Fig. 6, the data flow of the state machine of Fig. 5 is presented to highlight the parallel design of the different mathematical functions.

The main tasks developed by the states defined in Fig. 5 are:

- **S0**: It is the beginning of the state machine. In this state, the write-enable and address-valid signals allow writing the loaded data, by \( \text{data}_{\text{in}} \) signal, to the internal registers. Then, the start signal will be active.

- **S1**: This state enables other state machines associated with sub-blocks \( V_{ij} \), \( I_{ij} \) and \( \delta V \) to start parallel calculations of equations (9), (4), (9) and (10) for computing \( I_{ij}, V_{ij}, \) and \( V_j \). As soon as the results of the contribution voltages of the J groups corresponding to the linear approximation zone are reached, the signal \( \text{vp}_{\text{ready}} \) becomes active and the transition to state S2 occurs.

- **S2**: The parallel calculation of the string current is started. This is accomplished by splitting equation (6) into equations (10), (11) and (12). In addition, this state transitions to state S3, triggered by the \( \theta \) signal, when \( \theta \)'s calculation is completed. As seen in (6), \( \theta \) requires to make many calculations, which take up a significant amount of time to compute. In order to decrease the computation time, section 4.2 presents a digital hardware optimization of \( LW(\theta) \).

\[
I_{S1} = (I_{ph_j} + I_{i_j}) - V_j \tag{10} \\
I_{S2} = \frac{R_S}{R_{S} + R_j} \tag{11} \\
I_{S3} = \frac{V_j}{R_j} \tag{12}
\]

- **S3**: The computation of \( LW(\theta) \) and \( I_{ST_{\text{new}}} \) (following (13)) is developed. At the same time, \( V_{ST_{\text{old}}} = V_{ST_{\text{new}}} + \theta \) gets updated. After completing (13), the signal \( \text{sync} \) becomes high, triggering the transition to state S4.

\[
I_{ST} = I_{S1} \cdot I_{S2} - I_{S3} \cdot LW(\theta) \tag{13}
\]

- **S4**: It is possible to have a transition between three conditions. If the string current just calculated is higher than the short-circuit current of the next group, that is \( I_{S3} > I_{SC} \), the transition is to state S3. For the new calculations at state S3, the parameters of the modules remain equal to those in state S4; this saves computation time because computing (13) only needs new calculations for (10) and (7). In the other case, that is \( I_{ST} < I_{SC} \), then the transition is done to the S5 state, where new parameters must be used for the SDM calculations. The state machine returns to the S0 state when \( j > J \) condition is accomplished, this happens when the I-V curve is at zero current. In this case, \( I_{ST} \) and \( V_{ST} \) values are saved in internal registers and the state machine transitions to state S0.

- **S5**: In this state a new dominant group emerges, and \( j \) increments by 1, requiring the reading of updated parameters. The signal \( \text{sync}_{\text{new}} \) is active-high when those parameters are ready, so the transition to state S2 is done.

The data flow of Fig. 6 allows to observe the parallel calculation of functions. The transition signals of the state machine \( I_{\text{string}} \) are drawn first, which are activated at the end of their corresponding states. Initially the calculation of \( \theta \) and \( LW(\theta) \) are done in states S2 and S3, respectively. However, the hardware dedicated to the calculation of \( \theta \) and \( I_{ST} \) do calculations continuously. \( \theta \) and \( I_{ST} \) values required for the next calculation of string current are calculated in advance. As seen, \( \Theta_{S1} \) and \( I_{S1_j} \), are calculated during the current state S3, which is calculating the current \( I_{ST} \). Those \( \Theta_{S1} \) and \( I_{S1_j} \) values will be used in the next state S3, in which the string current \( I_{ST_{\text{new}}} \) is calculated. The other terms \( I_{S2} \) and \( I_{S3} \) required to the string current calculation remains equal unless there is a transition to state S5, such is observed in the data flow. A transition to state S5 means that the dominant group \( G \) of the string is changed, so the last calculations of \( \theta \) and \( I_{ST} \) are not used because new parameters are required in the SDM calculations.

Table 3 shows the proposed hardware architecture’s performance. This implementation used the Intel Modelsim-FPGA software with an Intel Arria 10AS066K3F40ES25G FPGA clocked at 50 MHz, and the math operations used 64 bits Intel floating point arithmetic. The tests conducted considered the following two cases: Case I consisted of a string formed by four groups \((J = 4)\), each one with twenty modules \((N_j = 20)\). Case II corresponds to a string formed by ten groups of ten modules each. Parameter \( S_j \) in Table 4 indicates the irradiance conditions for both cases. The variation of string voltage \( \theta \) is 0.3 V in both cases. Since case II is longer than the case I, its number of samples is higher. The same cases implemented in Matlab using the NRJ method proposed in [4] served as references to determine the accuracy because that method [4] uses SDM without any approximation. The (root-mean-square-error) RMSE and (mean-absolute-percentage-error) MAPE for the resulting currents in the I-V plane provide a measurement of the quality of the MBZ method against the NRJ method. Based on the RMSE and MAPE, the 64 bits floating-point arithmetic gives enough precision to perform the calculations. The linear approximation of the I-V curve has the highest impact on the accuracy of the MBZ method. The simulation time of Intel Modelsim-FPGA software, using a PC with Intel(R) core(TM) i7-8750H
CPU base speed 2.2 GHz with 12 processors and 24 Gb DDR4 ram memory with 2400 MHz speed to obtain one point of the I-V curve, are 2.8 μs and 2.3 μs for cases I and II, respectively. Those computation times correspond to improvements of 512.6 % and 586.1 % with respect to the proposed MbZ method implemented in Matlab.

Using the Intel Quartus Premier Pro software tool, it is possible to find out the FPGA resources used by the proposed MbZ method. As it is observed in Table 5, the sub-block \(\delta V\) uses the lowest amount of resources, while the sub-block \(V_{i,j}\), running the Newton-Raphson algorithm for (9), uses the biggest amount of resources. The optimization of the \(Lambert W(\Theta)\) function in the hardware architecture has led to a large cut in the used FPGA resources. The total amount of resources used from the selected Intel Arria 10AS066K3F40E2SG FPGA is 28 %.

### 4.2. Lambert\(W(\Theta)\) optimized hardware architecture

The current literature describes many ways to do the \(LW(\Theta)\) function \([16]\). This paper uses the implementation proposed in [11] because of its feasibility for parallel implementation and reduced execution time. For this work, the chosen \(LW(\Theta)\) implementation uses equations (14) and (15), where the theta range and the equations’ coefficients come from [20].

For \(\Theta < 0.145469\)

\[
\text{result} = \frac{\Theta + a_{11} \Theta^2 + a_{21} \Theta^3 + a_{31} \Theta^4 + a_{41} \Theta^5}{1 + b_{11} \Theta + b_{21} \Theta^2 + b_{31} \Theta^3 + b_{41} \Theta^4}
\]  (14)

For \(\Theta \geq 0.145469\)

\[
\text{result} = \frac{\Theta + a_{12} \Theta^2 + a_{22} \Theta^3 + a_{32} \Theta^4 + a_{42} \Theta^5}{1 + b_{12} \Theta + b_{22} \Theta^2 + b_{32} \Theta^3 + b_{42} \Theta^4}
\]  (15)
Table 4. Accuracy and Computation Time of MbZ Method.

| Case | I       | II      |
|------|---------|---------|
| Samples (k points) | 5627    | 7053    |
| Number of modules by group $N_j$ | 20,20,20,20 | 10,10,10,10,10 |
| Irradiance level $S_j$ (W/m²) | 1000,700,400,100 | 1000,900,800,700,600 |
| Variation Step of String Voltage $\theta$ (V) | 0.3     | 0.3     |
| RMSE of I-V Curve | 0.5313  | 0.3151  |
| MAPE (%) of I-V Curve | 1.4553  | 1.1298  |
| Convergence time of MbZ at Intel ModelSim-FPGA for I-V point (μs) | 2.8     | 2.3     |
| Convergence time of MbZ at Matlab for I-V point (μs) | 14.3    | 15.79   |
| Improvement of time comparing FPGA vs Matlab implementations (%) | 512.6   | 586.1   |

The digital hardware architecture of Fig. 7 has been developed for implementing the $LW(\Theta)$. As it can be seen the proposed structure has a three pipeline stages. Using multiplexers and multipliers, $\Theta^{-1}$ and $a\Theta^0$ are generated. The addition-accumulator structure formed by an adder and a register allow to obtain $1 + ... + b\Theta^{-1}$ and $a\Theta^1 + ... + a\Theta^0$. Those results are fed to a divisor to produce $LW(\Theta)$. The $mux3$, $mux4$ allow the utilization of the same multiplication, addition-accumulator and division structures for the equations (14) and (15), without a degradation on the performance. The physical synthesis of the digital hardware architecture of Fig. 7, using Quartus Prime Pro for an 10AS066K3F40E2SG FPGA, used a total of 26 DSP blocks, which is an acceptable amount elements compared with the implementation proposed by [11], which implements the same $LW(\Theta)$ function with 276 DSP blocks. Besides, the computation time of the proposed $LW(\Theta)$ implementation was 1.06 μs with a 50 MHz clock.

4.3. Modified MbZ method for real-time applications

Real-Time applications such as MPPT algorithms require to determine one operating point of the PV string instead of reconstructing the whole I-V curve at once. For that kind of applications, the stated MbZ method can be adapted using a slight modification, which consists in finding the inflection point voltages around the actual string voltage. For doing this, the voltage at the inflection points of the string I-V curve, those called here $V_{IF}$, must be computed. As it is observed in Fig. 3, the $V_{IF}$ voltages are determined by the $V_j$ voltages and the number of bypass diodes activated, which are calculated by (16).

$$V_{IF_j} = V_{I/j+1} - \sum_{j=1}^{J} N_j \cdot V_{Gj}, f o r j = 1, ..., J - 1$$

(16)

Comparing the $V_{ST}$ with the $V_{IF}$ voltages it is possible to determine the inflection points just to the right and to the left side of $V_{ST}$. Therefore, the $i$ value in Table 2 is determined, which allows to use the proposed MbZ method in the same way as it was explained in section 3. The proposed MbZ modification has not a severe impact on the memory requirements, meanwhile this saves computation time. The performance of the modified MbZ method is evaluated in the following section in a MPPT implementation case.

5. Validation of the proposed MbZ method

PV strings subjected to dynamic shadowing serve as a test case for the proposed MbZ method, which is evaluated for two different purposes: long-term power production and a MPPT application. Fig. 8 shows, hour-by-hour in a day, a PV installation of 32 PV modules (M1-M32) under different shadowing conditions depending on the sun’s trajectory and obstacles, that represented in gray fill. Table 6 shows the irradiance profile, which has been obtained from a local meteorological base station. Also, the PV modules subjected to shadowing conditions are included in the last column. The shadowed condition has a 30 % of incident irradiance in the area, forming two groups ($J = 2$) in the string.
5.1. Example 1: long-term power production

To determine the power production in a day, the proposed MbZ method is used to reconstruct the I-V curve of the PV installation and determining the global maximum power point (GMPP). As a reference, the test case has been developed using the NRiJ method proposed in [4]. The GMPP obtained by MbZ has been compared to the one obtained by NRiJ, both implemented in Matlab. The relative error (RE) at the global MPP is shown in the fourth column. The last column of Table 7 shows the MAPE of the proposed MbZ method, this taking all points of the string I-V curve. The worst precision at the GMPP was observed between hours 8 and 9, reaching 0.2544 % of RE. In the meantime, the most accurate results occur between 12 and 13 hours, with an RE of 0.0002 %. It is observed that the most inaccurate result occurs when most PV modules are shadowed, and the most accurate case for homogenous conditions.

Different shadowing levels are included in the previously described case, to test more sparsity mismatched conditions. Table 8 shows the considered shadowing cases, in which the irradiance levels received by the modules are presented in columns three and four, respectively. The hour range in the first row of Table 8 was selected because it has more shadowing conditions in the proposed PV installation. The shadowing factor column refers to a percentage in which $s_{\text{ho}}$ is the highest shadowing corresponding to 70 %, and $s_{\text{lo}}$ the lowest shadowing corresponding to 30 % of incident irradiance in the zone and received by the module. The modules without shadowing are indicated $s_{\text{un}}$.

Table 8 shows the execution time needed by the MbZ proposed technique to construct the I-V curve of sparse cases shown in Table 8. The second column of Table 9 shows the execution time for the NRiJ technique, developed in Matlab as proposed in the literature. Table 9 also shows the time taken to compute the proposed MbZ method first in Matlab and then using the ModelSim-Intel software. It is observed that for the Matlab implementation, the improvement in execution time of the proposed MbZ method versus the NRiJ is more than 16740 %. Also, the FPGA implementation of the MbZ proposed technique, respect to Matlab implementation, improves the calculation time in more than 477.32 %.

5.2. Example 2: MPPT application

In Fig. 9, a boost converter and a P&O algorithm is used to extract the power of the previous described PV installation. The same set-up is implemented using the PV array block from Matlab (blue dotted square)
Table 8. Sparse shadowing case.

| Hour-Range | Shadowing Factor | Irradiance W/m² | Modules |
|------------|------------------|-----------------|---------|
| 7-8        | $s_{h1}$         | 45.03           | M1, M2, M3, M4, M5, M6 |
|            | $s_{h2}$         | 105.07          | M7, M8, M9, M10, M11, M12 |
|            | $s_{h3}$         | 150.10          | M13-M32 |
| 8-9        | $s_{h4}$         | 97.32           | M1, M2, M3, M4 |
|            | $s_{h5}$         | 227.08          | M5, M6, M7, M8 |
|            | $s_{h6}$         | 324.40          | M9-M32 |
| 9-10       | $s_{h7}$         | 148.44          | M1, M2 |
|            | $s_{h8}$         | 346.36          | M3, M4 |
|            | $s_{h9}$         | 494.82          | M5-M32 |
| 10-11      | $s_{h10}$        | 182.1           | M1 |
|            | $s_{h11}$        | 424.90          | M2 |
|            | $s_{h12}$        | 607             | M3-M32 |

Fig. 9. a) MPPT application using the PV array block from Matlab as PV generator b) MPPT application using the MbZ method as PV generator.

Table 9. Calculation time for reconstructing I-V curves in sparse case of Table 8.

| Hour-Range | NRI_{recon} ms | MbZ_{Matlab} ms | MbZ_{Matlab} ms |
|------------|----------------|-----------------|-----------------|
| 7-8        | 2.8993         | 23.5784         | 4.8053          |
| 8-9        | 4.2121         | 23.6776         | 5.02008         |
| 9-10       | 4.448482       | 24.4141         | 5.1276          |
| 10-11      | 4.152294       | 24.7924         | 5.1941          |

Table 10. Parameters of the boost converter and P&O algorithm.

| Parameter       | Value                  |
|-----------------|------------------------|
| $C_{IN}$        | $1.9007 \times 10^{-6}$ F |
| $L$             | 0.0136 H               |
| $C$             | $5.0091 \times 10^{-6}$ F |
| $R_{LOAD}$      | 163.5544 Ω            |
| $V_{IN}$        | 580.80 V               |
| $P_{SC}$        | 4640.6 W              |
| Switching Frequency | 20 kHz             |
| $D$             | 0.33                   |
| $\Delta D(P&O)$ | 0.0067                 |
| $\Delta T(P&O)$ | 8 ms                   |

and the proposed MbZ method (red dotted square) to represent the PV generator, the first used as reference case. In order to connect the MbZ output to the boost converter circuit, an ideal controlled source is required. Moreover, an algebraic loop breaking is developed using a unit delay. The parameters of the designed boost converter and P&O algorithm are shown in Table 10.

In Fig. 10, the results of duty cycle $D$ and the string voltage $V_{ST}$ are presented. The blue and red lines correspond to the set-up when the PV generator is implemented by the PV array block and MbZ, respectively. At the beginning, the PV installation is operating in homogeneous conditions at 1000 W/m² and 25 °C. At 0.3 s a shadowing over the modules M1-M8 is simulated by an incident irradiance of 500 W/m². It is observed that in both states, stable and transient, the MbZ method presents a similar response to the one obtained when the PV array block model is used. In this example, the MAPE of the string voltage is 0.0319 %, and the one of string power is 0.1892 %. Therefore, these results confirm that MbZ is suitable for simulating PV strings into real-time applications such as MPPT systems.
6. Conclusions

In this paper a fast simulation method of mismatched PV strings is proposed, which is called Model by Zone MbZ. In the proposed approach a commutation of the model representing modules in a PV installation is done. Taking advantage that a PV module exhibits opposite sensitivity of current to voltage variations at the left and right side of MPP, it is proposed to use the SDM or linear approximation model when the module is working on those zones, respectively. Moreover, modules bypassed are represented by the bypass diode voltage. The proposed commutation of models avoids to use any iterative algorithm to determine the string current, so saving computation time. Moreover, a digital hardware optimization by coding-hand style is developed to implement the proposed MbZ in FPGA systems. The proposed optimization consists in three main factors. Firstly, terms in the equations that remain constant at different operating points are identified and grouped, so they do not need to be re-calculated. Secondly, some equations are divided in different terms which can be operated in parallel with other terms. Finally, the use of a hand-code style instead of high-level synthesis tools to design the digital hardware architecture allows to create dedicated functions. Those all optimizations converge into save time and resources. The proposed MbZ method required 28% of the available resources in the medium level FPGA system Intel Arria 10AS066K3F40E2SG FPGA clocked at 50 MHz. Besides, the computation time to obtain one point of the I-V curve was 2.8 μs, which is lower than one reported in literature in similar conditions of mismatching and size of PV strings. Finally, the accuracy suits real time applications and long-term power production analyses, because MAPE is lower than 0.1892 % and 1.6 %, respectively. An extension of the proposed hardware architecture to simulate arrays in real-time is being considered for future work.

Declarations

Author contribution statement

Alejandro Carabali-Isajar, Martha Lucia Orozco-Gutierrez, Jose Alex Restrepo: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper. Edinson Franco-Mejia, Giovanni Spagnuolo: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This work has been supported by Universidad del Valle, Cali, (Colombia) under the project CI 1036. Moreover, the authors gratefully acknowledge the financial support provided by the Colombia Scientific Program within the framework of the call Ecosistema Cientifico (contract number: FP44842-218-2018).

Data availability statement

The data have been used in a research so those will be used in next publications.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.
References

[1] REN21, Renewables 2021 global status report, 2021.

[2] G. Petrone, G. Spagnuolo, M. Vitelli, Analytical model of mismatched photovoltaic fields by means of Lambert W-function, Sol. Energy Mater. Sol. Cells 91 (18) (2007) 1652–1657.

[3] G. Petrone, C.A. Ramos-Paja, Modeling of photovoltaic fields in mismatched conditions for energy yield evaluations, Electr. Power Syst. Res. 81 (4) (2011) 1003–1013.

[4] M.L. Orozco-Gutierrez, J.M. Ramirez-Scarpetta, G. Spagnuolo, C.A. Ramos-Paja, A technique for mismatched pv array simulation, Renew. Energy 55 (2013) 417–427.

[5] M.Z. Hossain, N.A. Rahim, Jeyraj Selvaraj, Recent progress and development on power DC-DC converter topology, control, design and applications: a review, Renew. Sustain. Energy Rev. 81 (2018) 205–230 (July 2017).

[6] J. Prasanth Ram, N. Rajasekar, Manufumi Miyatake, Design and overview of maximum power point tracking techniques in wind and solar photovoltaic systems: a review, Renew. Sustain. Energy Rev. 73 (2017) 1138–1159 (March 2016).

[7] Razman Ayop, Chee Wei Tan, A comprehensive review on photovoltaic emulator, Renew. Sustain. Energy Rev. 80 (2017) 430–452.

[8] A.S. Vijay, Suryanarayana Doodla, Mukul C. Chandorkar, Real-time testing approaches for microgrids, IEEE J. Emerg. Sel. Top. Power Electron. 5 (3) (2017) 1356–1376.

[9] I. Moussa, A. Bouallegue, A. Khedher, Development of a low cost pv simulator based on fpga technology, in: 2017 International Conference on Green Energy Conversion Systems (GECS), March 2017, pp. 1–6.

[10] A. Gutierrez, M. Bressan, J.F. Jimenez, C. Alonso, Development of real-time supervision HIL emulator of shaded PV systems, in: 2017 6th International Conference on Renewable Energy Research and Applications, ICREA 2017, 2017, pp. 499-504.

[11] Antonio Guerino, Eric Momannson, Giovanni Spagnuolo, Soc-based embedded real-time simulation of mismatched photovoltaic strings, in: ELECTRIMACS 2019 ENGI-NEERING: Modelling and computational simulation for analysis and optimisation in electrical power engineering, Math. Comput. Simul. 184 (2021) 267–281.

[12] M.L. Orozco-Gutierrez, B. Ospina-Agudelo, C.A. Ramos-Paja, G. Petrone, G. Spagnuolo, Real-time simulation of mismatched PV arrays, in: SPEEDAM 2018 - Proceedings: International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2018, pp. 130–135.

[13] T.V. Shibu Raj, Ann Mary Joshua, Real time simulation of KY Boost converter on miniature full spectrum simulator, in: 2016 International Conference on Next Generation Intelligent Systems, ICNGIS 2016, (2), 2017, pp. 4–9.

[14] Jee Hoon Jung, Shehab Ahmed, Real-time simulation model development of single crystalline photovoltaic panels using fast computation methods, Sol. Energy 86 (6) (2012) 1826–1837.

[15] Giovanni Petrone, Carlos Andres Ramos-Paja, Giovanni Spagnuolo, Photovoltaic Sources Modeling, Wiley IEEE Press, 2017.

[16] Efstratios Batzelis, Georgios Anagnostou, B.C. Fal, Computation of the Lambert W function in photovoltaic modeling, in: ELECTRIMACS 2019, in: Lecture Notes in Electrical Engineering, April 2019.

[17] Rajasekar Selvamuthukumar, Rajesh Gupta, Rapid prototyping of power electronics converters for photovoltaic system application using Xilinx system generator, IET Power Electron. 7 (9) (2014) 2269–2278.

[18] Efthimos Koutrolis, Kostas Kalaitzakis, Vasileios Tzitzilonis, Development of an fpga-based system for real-time simulation of photovoltaic modules, in: Mixed-Technology Testing Rapid System Prototyping, Microelectron. J. 40 (7) (2009) 1094–1102.

[19] Alberto Sanchez, Elias Todorovich, Angel de Castro, Impact of the hardened floating-point cores on HIL technology, Electr. Power Syst. Res. 165 (July 2018) 53–59.

[20] Darko Veberir, Lambert w function for applications in physics, Comput. Phys. Commun. 183 (12) (2012) 2622–2628.