Model of Series-Parallel Photovoltaic Arrays Using Double-Diode Model and Parallel Computing

Juan David Bastidas-Rodríguez 1,†, Carlos Andrés Ramos-Paja 2,† and Sergio Ignacio Serna-Garcés 3,* †

Abstract: Several applications require to estimate the power production of photovoltaic (PV) systems under partial shading conditions. For example, dynamic reconfiguration of the array connections is needed to maximize the power production under partial shading conditions, which requires estimating the power generated by the PV array in several possible configurations. Therefore, a fast and accurate modeling technique is needed to perform those calculations in practical times and with low estimation errors. To address those kinds of problems, this paper proposes a modeling approach based on the double-diode model to provide high accuracy at low voltage and low irradiance conditions, which are important for partial-shading analysis. Moreover, the proposed modeling technique is designed to be implemented in parallel processing devices; thus, the calculation time is much shorter in comparison with classical serial solutions. The proposed model is tested in terms of accuracy and speed, obtaining satisfactory results. Finally, the applicability of the parallel model in reconfiguration applications is demonstrated using an application example.

Keywords: photovoltaic array; series-parallel; double-diode model; parallel computing; implicit model

1. Introduction

Photovoltaic (PV) power generation is a growing option for providing electricity, where large PV fields are more common every year due to multiple reasons such as the need of reducing costs [1,2], fossil fuel shortage, greenhouse effect, the integration of generation systems to buildings, and the deployment of stand-alone systems for remote/rural areas. The energy harvest from PV fields depends on several factors [3], some intrinsic (panel degradation) and other extrinsic (solar irradiance, partial shading), where the fast estimation of the power production is crucial for control decisions like dynamic reconfiguration in the presence of partial shading [4].

Determine the power produced by a PV array requires the calculation of the voltage (V) and current (I) of the modules, which is needed to reconstruct the current vs. voltage (I–V) curves of the PV system; those curves can be later used to interpolate the power production at a given operation point [4]. Alternatively, a mathematical model of each PV section can be obtained using an Equivalent Electrical Circuit (EEC), where the single-diode and double-diode models are the most widely adopted [5]. The main difference between those models concerns the current produced by the recombination mechanism, which is only considered by the double-diode model. Comparing those models using criteria like root mean square error (RMSE), normalized RMSE, absolute error, relative error (RE), and mean RE, demonstrate the higher accuracy provided by the double-diode model. On the other hand, the number of parameters to be estimated for the single-diode model is lower, hence it is a simpler model [6]. However, the recombination current, which is
only considered by the double-diode model, occurs in the depletion region (a low voltage region); this characteristic allows the double-diode model to provide better reproduction of the PV module behavior at low voltages [7], which is particularly important under partial shading conditions.

Several works highlight the differences between the single-diode and double-diode models. For example, in [8–20] different optimization methods are used to calculate the parameters of the models, such as FBIA (Forensic-Based Investigation Algorithm), GBO (Gradient-based optimizer), COA (Coyote Optimization Algorithm), CSO (Cat Swarm Optimization), BHCS (Biogeography-based Heterogeneous Cuckoo Search), PSO (Particle Swarm Optimization), GA (Genetic Algorithm), MLSHADE (Multi-strategy Success-History based Adaptive Differential Evolution), and IJAYA (Improved Jaya Algorithm); those works put into evidence the higher accuracy provided by the double-diode model over the single-diode one. Moreover, the double-diode model is also more accurate for different types of PV modules and under different operating conditions, such as low irradiance conditions and temperature variations [17,19,21,22].

The calculation of the power generated by a PV array requires information about the model and the interconnection topology of the PV modules. Regardless of the adopted model (single-diode or double-diode), the mathematical expression describing the I-V curve of the module is a transcendent equation, which is commonly solved using numerical methods. On the other hand, the modules-interconnection topologies are diverse, but the series-parallel structure is the most widely adopted for both small (urban) and large-scale installations [23–25].

There are several numerical solutions for the transcendental equations, where one simple method concerns the use of the Lambert-W function to transform those equations into explicit expressions [26]; however, the Lambert-W function is not available in several programming languages, and its accurate implementation requires complex numerical methods. Another strategy adopted to solve transcendental equations is the use of recursive numerical methods, where the most common options are the Newton-Raphson method, the bisection method, and the false position method [3,27,28]. However, the previous methods only seek one solution of the transcendental equation in a given range; hence, those methods are not recommended to solve systems with more than one solution. Other methods that have been used, due to their implementation in specialized software like Matlab, are the trust-region-dogleg [23,29], trust-region and Levenberg-Marquardt. However, only the trust-region-dogleg algorithm is designed to solve nonlinear equations, while the other ones are mainly aimed at minimizing the squares of a function.

The implicit equation of the single-diode model has been used to evaluate the power production of series-parallel PV arrays under mismatching conditions [29]. That method avoids evaluating the Lambert-W function, which decreases the calculation time, but keeps the relatively low accuracy of the single-diode model. A similar approach is reported in [23,29], both adopting the implicit expression of the PV current, but in this case, based on the double-diode model. In those works, the equation system representing the series-parallel PV array is solved using the trust-region-dogleg algorithm, and it is tested under non-homogeneous conditions, but only considering a series (classical) model execution. Similarly, the work reported in [3] proposes a methodology to solve the equation system of a series-parallel PV array based on the double-diode model. As in the previous case, this method is tested under non-homogeneous conditions, but the Newton-Raphson method used for the solution is not parallelized.

The Lambert-W function is a mathematical definition that has been used in the last years to solve the transcendental equation of PV models. For example, the work reported in [30] uses the Lambert-W function to estimate the power of a PV panel in a power system with a DC/DC converter, battery, load, and MPPT controller. In this case, the PV model is based on a single equation, and the authors do not provide the calculation analysis. Another approach is presented in [31], which uses the bisection method and a modified version of the Lambert-W function to speed up the calculation time by, approximately, a
factor of 2 concerning other methods [32–34]; however, this solution is only tested with the fifth-order Schröder’s equation. A different approach is reported in [35], which uses an explicit expression for the double-diode model based on the Lambert-W function, and it is applied to solve a series-parallel PV array, including the action of the bypass diode. This solution provides an accurate solution, but the calculation algorithm is not concurrent.

A different solution was presented in [36], which proposes a discrete representation based on the Lambert-W function to calculate the power vs. voltage (P-V) and I-V curves of a PV array in different configurations and under homogeneous and non-homogeneous conditions. This solution, based on the single-diode model, is evaluated point-to-point to reach the MPPT, analyzing both regions of the diode conduction (negative and positive voltage) using the Lambert-W function [37,38]. The accuracy of the model is validated under partial shading, but there is not an analysis of performance and speed.

The difficulty of dealing with transcendent equations has been faced by transforming those expressions; for example, the authors of [39] propose a power-law model to calculate the I-V curve under real operating conditions for different types of PV modules, which avoids the use of the single-diode model transcendental equation. Although some parameters of this model lack physical meaning, the resulting expression has an analytical solution, high accuracy, and low computational burden. Another approach, reported in [40], proposes an explicit model based on the sine function, which only has a shape parameter. This model is aimed at constructing the I-V and P-V curves for the PV cells under different temperature and solar irradiance conditions, but the model is not tested under mismatched conditions. Similarly, the authors of [41] present a transformation of the transcendental equation of the PV cell model into a non-transcendental expression that reduces the computation time without impacting the accuracy, but this is applied only to the single diode model.

Other modern approaches to solve transcendental equations include artificial intelligence and advanced numerical methods. For example, in [42] is proposed a multilayer neural network to solve a transcendental equation; this method is tested in some problems modeled by transcendent expressions such as the diode equation. However, the authors do not provide a performance evaluation or a comparative analysis against other solutions. Similarly, Ref. [43] proposes some approximate analytical solutions for a transcendental equation similar to the PV model, where the goal is to obtain a simpler equation with an acceptable numerical evaluation. Despite some cases are evaluated, it is not confirmed that such an approximation improves the accuracy or speed in comparison with other solutions adopted to calculate the PV model.

To take advantage of the concurrent execution provided by FPGAs, the work in [44] uses an SoC formed by an ARM processor and an FPGA based on the Artix-7 logic, all in a single chip, for the simulation of a mismatched PV string. This solution is based on the system of equations reported in [45], which is formed by the transcendental equations derived from the single-diode model of the PV modules. Such a technique requires constructing a Jacobian matrix representing the nonlinear equations system, which can be symbolically inverted [45–47]; hence, the numerical method to solve the system is reduced to algebraic matrix operations. The authors split the simulation algorithm into three parts: the model update, the guess solution, and the Newton-Raphson numerical method. To perform a comparative analysis, the authors implemented three software versions. First, the complete solution runs on the ARM processor, which is the baseline; second, using the SIMD feature (Single-Instruction Multiple-Data) of the NEON coprocessor integrated into the ARM processor, the instructions are parallelized to calculate the currents and the differential resistances necessary to obtain the inverse of the Jacobian matrix; third, the numerical solution of the Lambert-W function [32] was described in VHDL to be implemented on the FPGA. The first two versions were software-only solutions, while the third one is a hardware/software hybrid solution. The second software version improved the execution time by 68% concerning the baseline (first solution), and the third solution improves the execution time by an additional 3%. In this work, the parallelization of the algorithm is
the main element used to improve the execution time, since the performance of the hybrid solution depends mainly on the communication protocol between the hardware and the software stages. Finally, it is worth noting that the work uses the single-diode model and an approximation of the Lambert-W function, which prevents a complete parallelization of the system solution.

This paper proposes a solution aimed at providing high accuracy and speed in the calculation of the power production of a PV array since many PV installations are based on multiple strings. First, this new solution is based on the more accurate double-diode model of the PV modules, thus improving the accuracy over solutions based on the single-diode model for mismatched (partial-shading) conditions. Second, the proposed solution is focused on performing a parallel calculation of the strings, thus providing a fast calculation over solutions based on classical (series) calculation algorithms. Finally, the use of this parallel and accurate solution is tested under a real-world condition, which considers the reconfiguration of a partially-shaded PV array. The paper is organized in the following way: the PV array model considering the double-diode model for the PV modules is presented in Section 2. Section 3 reports the results and discusses the different test conditions. Finally, Section 4 presents the conclusions of the work.

2. Proposed Model

The general structure of the most widely used PV array configuration, named series-parallel (SP), is illustrated in Figure 1. The array is formed by $M$ columns or strings, and each string is formed by a set of modules and a blocking diode connected in series. In turn, each module is formed by one or more series-connected submodules, where each submodule is formed by $N_s$ cells connected in series and a protection diode connected in anti-parallel, named bypass diode; such a submodule structure is depicted in Figure 2, where the double diode model represents the $N_s$ series-connected cells. Hence, each string of a SP array can be analyzed as a set of $N$ submodules and one blocking diode connected in series. It is worth noting that the array voltage ($V_{arr}$ in Figure 1) is usually fixed by the power converter where the array is connected to; therefore, it is assumed as a known value.
Classical double-diode model

Bypass diode

Figure 2. Equivalent circuit of the double-diode model used to represent each submodule.

This section introduces the mathematical descriptions of the main elements that form the PV array (submodules and blocking diodes) and the system of non-linear and implicit equations that describe each string, as proposed in [23]. Moreover, this section also presents the procedure to calculate the array current by using parallel computing.

2.1. Submodule and Blocking Diodes Model

The submodule is the main element of the PV array and it is represented by using the equivalent electric circuit illustrated in Figure 2, where the current source ($I_{ph}$) represents the current generated by the photovoltaic effect and the diodes $D_1$ and $D_2$ represent the cells diffusion and recombination currents, respectively. Furthermore, the resistors $R_s$ and $R_h$ represent the ohmic losses and leakage currents, respectively, and the bypass diode (BD) limits the power dissipation in the submodule cells when they are under partial shading conditions [48].

The equation that relates the submodule's current ($I_{sm}$) and voltage ($V_{sm}$) is shown in (1), which is a nonlinear and implicit equation. In (1) the parameters $I_{s1}$ and $\eta_1$ are the inverse saturation current and ideality factor of diode $D_1$, respectively; while $I_{s2}$, $\eta_2$, $I_{sbd}$ and $\eta_{bd}$ correspond the same parameters for diodes $D_2$ and BD. Moreover, $V_{td}$ is the thermal voltage of the PV cells and it is defined as $V_{td} = k \cdot T / q$, where $k$ is the Boltzmann constant, $q$ is the electron charge, $T$ is the cells temperature. Additionally, $I_{bd}$ is the current through the diode BD, as shown in (2), and its thermal voltage is defined as $V_{tbd} = k \cdot T_{bd} / q$, where $T_{bd}$ is the bypass diode temperature. It is worth noting that the submodule’s parameters depend on the irradiance and temperature, which is discussed in detail in [49].

\[ f(V_{sm}, I_{sm}) = -I_{sm} + I_{ph} - I_{s1} \cdot \left( e^{\frac{V_{sm} + (I_{sm} - I_{bd}) \cdot R_s}{N_s \cdot \eta_1 \cdot V_{td}}} - 1 \right) - I_{s2} \cdot \left( e^{\frac{V_{sm} + (I_{sm} - I_{bd}) \cdot R_s}{N_s \cdot \eta_2 \cdot V_{td}}} - 1 \right) - V_{sm} + (I_{sm} - I_{bd}) \cdot R_s + I_{bd} = 0 \]  

\[ I_{bd} = I_{sbd} \cdot \left( e^{\frac{V_{sm}}{N_s \cdot \eta_{bd}}} - 1 \right) \]  

The other key element in the SP arrays is the blocking diode, which prevent opposite current flows through the string coming from the other strings of the array. The relation between the blocking diode current ($I_{blk}$) and voltage ($V_{blk}$) is introduced in (3), where $I_{sblk}$ is the inverse saturation current, $\eta_{blk}$ is the ideality factor, and $V_{tblk}$ is the thermal voltage. Furthermore, $V_{tblk}$ is defined as $V_{tblk} = k \cdot T_{blk} / q$, where $T_{blk}$ is the blocking diode temperature.

\[ -I_{blk} + I_{sblk} \cdot \left( e^{\frac{V_{blk}}{\eta_{blk} \cdot V_{tblk}}} - 1 \right) = 0 \]  

2.2. Calculation of the String Current and Voltages

In a SP array formed by $M$ strings connected in parallel, where each string has $N$ submodules, i.e., an $M \times N$ array, each string is formed by $N + 1$ elements ($N$ submodules and one blocking diode). Therefore, there are $N + 2$ unknown electrical variables that
correspond to: the voltages of the $N$ submodules ($V_{sm1}, \ldots, V_{smN}$), the blocking diode voltage, and the string current ($I_{str}$), which is shared by all the elements in the string.

To calculate the $N + 2$ unknowns, it is necessary to define the system of $N + 2$ nonlinear and implicit equations given in (4) [23], where the first $N$ equations are obtained by applying (1) for each submodule, the $N + 1$ equation is obtained by using (3), and the last equation is obtained by applying KVL to the string including the array voltage. In expression (4), $V$ is a vector formed by $N + 1$ voltages, where the first $N$ elements correspond to the submodules voltages from top to bottom (i.e., $V(i) = V_{smi}, \forall \ i \in [1 \ldots N]$) and the last element is the blocking diode voltage with the opposite polarity (i.e., $V(N + 1) = -V_{blk}$). Moreover, $I_{str}$ is the string current and $V_{arr}$ is the array voltage, which is shared by all the strings and it is assumed known. The system of equations $F(V, I_{str})$ can be solved by using a numerical method or an optimization algorithm to obtain all the electrical unknowns in the string [50,51]. In this paper the Trust-Region Dogleg optimization algorithm [51] is used to solve $F(V, I_{str})$ for each string, but any other suitable method can be adopted.

$$F(V, I_{str}) = \begin{bmatrix} f_1(V(1), I_{str}) \\ f_2(V(2), I_{str}) \\ \vdots \\ f_N(V(N), I_{str}) \\ -I_{str} + I_{blk} \cdot \left(e^{\left(V_{blk} - V_{arr}\right)} - 1\right) \\ V(1) + \cdots + V(N + 1) - V_{arr} \end{bmatrix} = \vec{0} \tag{4}$$

2.3. Calculation of the Array Current Using Parallel Computing

To calculate the array current it is necessary to solve (4) for each string of the array. At that point, all the strings currents are known and the array current can be calculated by using (5).

$$I_{arr} = \sum_{k=1}^{M} I_{str,k} \tag{5}$$

It is worth noting that the system of nonlinear equations of each string does not depend on the electrical variables of the other strings; hence, all the strings in an $M \times N$ array can be solved independently. This particular characteristic is used in this paper to reduce the model solution time by using parallel computing, since each string can be solved by a different core or processing worker. This type of parallel computation can be performed using synchronous or asynchronous algorithms [52], where the workers in synchronous solutions wait for some synchronization condition, while the workers in asynchronous solutions can be used for other task when they finish the initial work. As it was previously described, the array current is calculated as the sum of all the strings currents; hence, that condition is an inherent synchronization point for the parallel calculation of the array current. Therefore, the solution proposed in this paper is a synchronous parallel algorithm.

The flow chart of the proposed model to solve SP arrays using parallel computing is shown in Figure 3, where the input data are: the array voltage ($V_{arr}$), $N$, $M$, and the model parameters of all the submodules and blocking diodes. Those parameters are organized in eleven $N \times M$ matrices ($M_{iphr}, M_{is1}, M_{is2}, M_{ip1}, M_{ip2}, M_{Vibd}, M_{Rhr}, M_{Rhr}, M_{Ibd}, M_{Vibd}$, and $M_{Vibd}$), for the DDM parameters, and three $1 \times M$ matrices for the blocking diode parameters ($M_{Ibd}, M_{Vibd}, M_{Vibd}$). Moreover, $N_w$ is the number of available workers and $S_f$ and $S_l$ are auxiliary variables that define the first and last string to be evaluated, respectively.
Figure 3. Flow chart of the solution of an $M \times N$ Series-Parallel array using parallel computing.

The first step is to send the data of $N_w$ strings to the available workers. Then, each worker constructs the vector $V$ and solves the system of nonlinear and implicit equations of the corresponding string. Once the string is solved, the current is saved to calculate $I_{arr}$ later. When the $N_w$ workers have solved the $N_w$ strings, it is necessary to verify if all the strings has been solved; otherwise, the process is repeated until the $M$ strings are solved. Finally, the array current is calculated by using (5).

At this point all the array is solved for a given value of $V_{arr}$, which can be used for different model based applications. For example, to generate the array I-V and P-V curves it is necessary to perform a voltage sweep of the array voltage from 0 V to the array open-circuit voltage and save the values of $I_{arr}$. Moreover, the proposed model can also be used to perform dynamic simulations of SP arrays in conjunction with power converters and maximum power point tracking techniques.

3. Result and Discussion

The proposed model is validated using experiments and detailed simulations, and those results are contrasted with the model proposed in [23], which is denominated Se-
rial model. The details of the experimental results are introduced in the corresponding Section 3.4; hence, the following paragraphs provide information that apply for all the simulation scenarios.

This section includes, first, three evaluation scenarios to show the performance of the proposed model in terms of the errors in the I-V curve reproduction, the improvement in the calculation time for SP arrays with different number of strings, and an application example (model-based reconfiguration analysis). Moreover, all the simulations of the proposed model, from here on denominated Parallel model, are performed by using only two processing workers (i.e., \( N_w = 2 \)) to consider the worst case scenario for a parallel computing application. However, higher number of workers can be used depending on the computational platform.

The Trina Solar TSM-PD05 270 W module is used as a reference to construct the arrays considered in this section, whose main electrical characteristics in standard test conditions (STC) are: short-circuit current 9.22 A, open-circuit voltage 37.9 V, Maximum Power Point (MPP) current 8.73 A, MPP voltage 30.9 V, shot-circuit current temperature coefficient 0.05 %/K, open-circuit voltage temperature coefficient −0.32 %/K, number of cells 60, and number of submodules 3; therefore, each submodule has 20 cells (i.e., \( N_s = 20 \)). From those characteristics, and considering an irradiance of 1 kW/m\(^2\) (\( G = 1 \) kW/m\(^2\)) and a cell temperature of 55 °C (\( T = 328.15 \) K), it is possible to calculate the following DDM parameters, of each submodule, by using the procedure proposed in [53]: \( I_{\text{ph}} = 9.3583 \) A, \( I_{s1} = 18.846 \) μA, \( I_{s2} = 16.316 \) μA, \( \eta_1 = 1 \), \( \eta_2 = 2 \), \( R_s = 0.1002 \) Ω, \( R_{b} = 307.487 \) Ω. Moreover, the parameters of both bypass and blocking diodes models are calculated from the GF3045T bypass diode datasheet information, obtaining \( I_{\text{bd}} = I_{\text{blk}} = 1 \) μA, and \( \eta_{sbd} = \eta_{sblk} = 0.2694 \).

For sake of simplicity, it is assumed that all the submodules and blocking diodes have the same parameters, and also it is considered that the cells, bypass diodes, and blocking diodes operate at almost the same temperature (i.e., \( T \approx T_{\text{bd}} \approx T_{\text{blk}} \) [47,54]). In any case, under experimental conditions, each parameter can be calculated independently using the procedure reported in [55,56]. Finally, the partial shading conditions are defined by using the matrix \( M_{1\text{ph}} \), since \( I_{\text{ph}} \) is directly proportional to the irradiance reaching each submodule [57].

### 3.1. Error Evaluation under Different Partial Shading Conditions

This analysis considers a \( 15 \times 2 \) (i.e., \( N = 15 \) and \( M = 2 \)) array with two partial shading conditions (C1 and C2) to evaluate the errors in the current prediction of both the Series and Parallel models regarding the array equivalent circuit implemented in Simulink. Condition C1 represents a shading profile with two irradiance levels (\( G = 1 \) kW/m\(^2\) and \( G = 0.8 \) kW/m\(^2\)), while C2 represents a random shading condition with submodule irradiance varying from \( G = 0.0286 \) kW/m\(^2\) to \( G = 0.7765 \) kW/m\(^2\).

The I-V and P-V curves of the SP array, for operating conditions C1 and C2, are introduced in Figure 4, where the Simulink results are plotted in red lines, while Serial and Parallel models are plotted in magenta and blue discontinuous lines, respectively. As expected, Serial and Parallel models provide the same results for both operating conditions, since the blue dashed line is over the magenta dashed line.

The errors in the current calculation of the Series and Parallel models are shown in Figure 5. As expected, the errors obtained with the two models are exactly the same, since both models solve the same systems of equations with the same algorithm; hence, both models provide the same Normalized Sum of Squared Errors (NSSE) for both operating conditions: 0.0139% for C1 and 0.0032% for C2. In the left plot of Figure 5, it can be observed that the largest errors for condition C1 are close to the open-circuit voltage (169 V) and the inflection point (71 V), where the errors are +0.91 A (+93%) and −29.2 mA (−0.17%), respectively. Around those points the derivative of the current with respect to the voltage increases; therefore, the small differences between the equivalent array circuit and the models are translated into increments in the current errors. Additionally, it is important to mention that the high percentage in the current error close to the open-circuit voltage is
expected since in this region of the I-V curve the currents are close to 0 A. Nevertheless, those current errors do not considerably affect the I-V curve reproduction, as observed in Figure 4. The increment of the current errors around the inflection points of the I-V curve is also evidenced in the right plot of Figure 5 for condition C2. In such a plot each increment in the magnitude of the current error corresponds to an inflection point in the I-V curve; however, those errors do not affect significantly the I-V reproduction. Moreover, for condition C2 the errors move from $-0.103 \, \text{A} (-0.75\%)$ close to 16 V to $+0.253 \, \text{A} (6.53\%)$ close to 127 V; while the largest percentage error is again close to the open-circuit voltage (163 V) with $0.160 \, \text{A} (74.6\%)$.

![Figure 4](image-url)

**Figure 4.** I-V (left) and P-V (right) curves of a $15 \times 2$ array under two different partial shading conditions.

![Figure 5](image-url)

**Figure 5.** Current errors in A and % for a $15 \times 2$ array under two different partial shading conditions. (Left): current errors for condition C1. (Right): current errors for condition C2.

### 3.2. Calculation Time for Different Number of Strings

The main advantage of the Parallel model is the reduction of the calculation time without sacrificing the solution accuracy. Hence, the analysis performed in this section evaluates the calculation time of both Series and Parallel models to generate the I-V curve for a SP array with fifteen rows ($N = 15$), and the number of columns varying from one to twenty (i.e., $M = [1 \cdots 20]$). All the I-V curves are generated with voltage steps of 2 V in $V_{arr}$, and each I-V curve is calculated six times to evaluate the average computation time.
time and standard deviation for each array. In addition, the shading profiles for all the strings are randomly generated to represent a complex partial shading condition with a high number of inflection points in the I-V curves.

Figure 6 shows the calculation time and standard deviation of the I-V curves for both Serial and Parallel models considering SP arrays from $15 \times 1$ to $15 \times 20$. It is worth noting that those results are obtained with two processing workers (i.e., $N_w = 2$) for the Parallel model, which is the worst case scenario regarding the number of workers. Moreover, the exact mean values and standard deviations are presented in Table 1 as well as the average time reduction obtained with the Parallel model and its standard deviation.

![Figure 6. Calculation time scalability analysis for an SP array with 15 rows ($N = 15$) and columns varying from 1 to 20 ($M = [1 \cdots 20]$).](image)

Table 1. Average calculation time and standard deviation (std) for an SP array with 15 rows ($N = 15$) and columns varying from 1 to 20 ($M = [1 \cdots 20]$).

<table>
<thead>
<tr>
<th>Columns (M)</th>
<th>Serial Model</th>
<th>Parallel Model</th>
<th>Average Time Reduction (%)</th>
<th>Std (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Time (s)</td>
<td>Std (s)</td>
<td>Average Time (s)</td>
<td>Std (s)</td>
</tr>
<tr>
<td>1</td>
<td>4.950</td>
<td>1.485</td>
<td>3.870</td>
<td>0.914</td>
</tr>
<tr>
<td>2</td>
<td>6.513</td>
<td>0.632</td>
<td>4.655</td>
<td>0.650</td>
</tr>
<tr>
<td>3</td>
<td>8.015</td>
<td>0.074</td>
<td>6.257</td>
<td>0.435</td>
</tr>
<tr>
<td>4</td>
<td>9.986</td>
<td>0.324</td>
<td>6.249</td>
<td>0.067</td>
</tr>
<tr>
<td>5</td>
<td>12.556</td>
<td>0.927</td>
<td>8.500</td>
<td>0.241</td>
</tr>
<tr>
<td>6</td>
<td>14.251</td>
<td>0.529</td>
<td>8.604</td>
<td>0.013</td>
</tr>
<tr>
<td>7</td>
<td>16.389</td>
<td>0.603</td>
<td>10.530</td>
<td>0.034</td>
</tr>
<tr>
<td>8</td>
<td>17.800</td>
<td>0.441</td>
<td>10.791</td>
<td>0.045</td>
</tr>
<tr>
<td>9</td>
<td>19.204</td>
<td>0.805</td>
<td>12.718</td>
<td>0.046</td>
</tr>
<tr>
<td>10</td>
<td>20.169</td>
<td>0.062</td>
<td>13.261</td>
<td>0.030</td>
</tr>
<tr>
<td>11</td>
<td>22.010</td>
<td>0.252</td>
<td>15.108</td>
<td>0.027</td>
</tr>
<tr>
<td>12</td>
<td>23.708</td>
<td>0.062</td>
<td>15.228</td>
<td>0.046</td>
</tr>
<tr>
<td>13</td>
<td>25.686</td>
<td>0.067</td>
<td>17.323</td>
<td>0.043</td>
</tr>
<tr>
<td>14</td>
<td>27.164</td>
<td>0.132</td>
<td>16.942</td>
<td>0.112</td>
</tr>
<tr>
<td>15</td>
<td>29.289</td>
<td>0.854</td>
<td>18.795</td>
<td>0.088</td>
</tr>
<tr>
<td>16</td>
<td>30.667</td>
<td>0.045</td>
<td>19.149</td>
<td>0.029</td>
</tr>
<tr>
<td>17</td>
<td>32.717</td>
<td>0.065</td>
<td>21.193</td>
<td>0.030</td>
</tr>
<tr>
<td>18</td>
<td>34.619</td>
<td>0.075</td>
<td>21.463</td>
<td>0.064</td>
</tr>
<tr>
<td>19</td>
<td>36.427</td>
<td>0.131</td>
<td>23.297</td>
<td>0.038</td>
</tr>
<tr>
<td>20</td>
<td>38.301</td>
<td>0.100</td>
<td>23.444</td>
<td>0.042</td>
</tr>
</tbody>
</table>
For arrays with from 1 to 3 the calculation times are close but there is a reduction below between 2.5% and 28.0% with the Parallel model. Then, the calculation time of both models increases linearly regarding the number of strings in the array. However, the calculation time of the Serial model increases with a higher slope (1.73 s/string) than the one of the Parallel model (1.058 s/string). Hence, in general, the proposed model reduces the calculation time between 20.6% and 39.5% with respect to the Serial model with an average standard deviation of 2.13%. Furthermore, the standard deviations of the calculation times of both models confirm their repeatability.

3.3. Application Example: Exhaustive Model-Based Reconfiguration

The reduction in the calculation time provided by the Parallel model is useful in model-based reconfiguration systems. In those applications, the I-V curves must be calculated as fast as possible to reduce the reconfiguration time (period), since the shorter the reconfiguration period, the higher the power generation under partial shading conditions [58,59]. This is due to the continuous change in the shading profiles generated by the movements of the sun in the sky along the day and year.

The reconfigurable array considered in this application example is illustrated in Figure 7, which is a 15 × 4 array where the first two rows are reconfigurable (i.e., 2 × 4) and the rest of the array is fixed (i.e., 13 × 4). The eight reconfigurable submodules can be connected to any string through the reconfiguration matrix, which is implemented with switches (e.g., relays or MOSFETs) and controlled by the reconfiguration algorithm. This type of systems with a fixed and a reconfigurable parts have been previously reported in the literature because, sometimes, only part of the array is affected by partial shading conditions [60].

![Figure 7. Reconfiguration system used in the application example.](image-url)

In this example, it is assumed that the reconfiguration algorithm evaluates all the possible configurations, i.e., it performs an exhaustive search. For the array explained before, it is possible to construct 105 different configurations with the eight reconfigurable...
submodules and the reconfiguration matrix. For each configuration the model generates the P-V curve, with voltage steps of 2 V, to identify the global maximum power point (GMPP). Then, the GMPP of each configuration is saved to determine the one that provides the highest GMPP. Once the best configuration is identified, the reconfiguration algorithm sets the reconfiguration matrix to implement such a best configuration into the PV array.

The analysis is performed for three different shading profiles defined through the $M_{Iph}$ matrix. The values of $M_{Iph}$ for the reconfigurable part are reported in Table 2, while the values of $M_{Iph}$ for the fixed part of the array are defined as follows: 5.6150 A for rows 3 to 9 (i.e., $M_{Iph}(a, b) = 5.6150 \forall a \in [3 \cdots 9] \land b \in [1 \cdots 4]$) and 2.8075 A for rows 10 to 15 (i.e., $M_{Iph}(c, b) = 2.8075 \forall c \in [10 \cdots 15] \land b \in [1 \cdots 4]$).

Table 2. $I_{ph}$ currents of the reconfigurable submodules in the array (the first two rows).

<table>
<thead>
<tr>
<th>Shading Profile</th>
<th>M = 1</th>
<th>M = 2</th>
<th>M = 3</th>
<th>M = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1799</td>
<td>7.1662</td>
<td>5.9916</td>
<td>3.1579</td>
</tr>
<tr>
<td></td>
<td>7.2665</td>
<td>3.6340</td>
<td>1.0624</td>
<td>6.8555</td>
</tr>
<tr>
<td>2</td>
<td>4.5745</td>
<td>0.7131</td>
<td>3.5509</td>
<td>1.5638</td>
</tr>
<tr>
<td></td>
<td>5.0860</td>
<td>5.1287</td>
<td>0.5475</td>
<td>3.0708</td>
</tr>
<tr>
<td>3</td>
<td>3.5842</td>
<td>0.5899</td>
<td>3.5831</td>
<td>2.9958</td>
</tr>
<tr>
<td></td>
<td>3.6119</td>
<td>3.6333</td>
<td>1.8170</td>
<td>0.5312</td>
</tr>
</tbody>
</table>

The 105 possible configuration were evaluated six times for each shading profile, with both the Serial and Parallel models, in order to calculate the average processing time values and the standard deviations shown in Table 3. The average time required by the Serial and Parallel models to evaluate the 105 configurations are 719.96 s (or 11.99 min) and 518.58 s (or 8.64 min), respectively, which implies a time reduction between 26.6% and 29.3% (with standard deviations lower than 0.23%) in the reconfiguration algorithm period when the Parallel model is used. Such a reduction indicates that the reconfiguration system using the Parallel model is able to enhance the array power production under mismatching conditions. Once again, it is important to highlight that these results were obtained with two processing workers ($N_w = 2$); hence, the calculation time can be further reduced by using as many workers as strings in the array (i.e., $N_w = M$).

Table 3. Average reconfiguration times and standard deviations (std) to evaluate 105 configuration for an SP array with 15 rows ($N = 15$) and 4 columns ($M = 4$).

<table>
<thead>
<tr>
<th>Profile</th>
<th>Serial Model</th>
<th>Parallel Model</th>
<th>Power Increment (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Time (s)</td>
<td>Std (s)</td>
<td>Average Time (s)</td>
</tr>
<tr>
<td>1</td>
<td>733.045</td>
<td>1.236</td>
<td>517.957</td>
</tr>
<tr>
<td>2</td>
<td>715.455</td>
<td>0.654</td>
<td>515.705</td>
</tr>
<tr>
<td>3</td>
<td>711.399</td>
<td>0.307</td>
<td>522.080</td>
</tr>
</tbody>
</table>

In addition, in this application example the improvement in the maximum power extracted, comparing the best and worst configurations, varies from 7.39% for shading profile 1 to 4.36% for shading profile 3 (see Table 3). This indicates that the increment in the power production obtained by a reconfiguration system depends on the shading profile; however, such increment is translated into a higher energy production, a reduction in the project’s payback time, and a mitigation of the power generation variability of PV generators.
3.4. Experimental Results

The proposed model is also validated with experimental results for a $3 \times 3$ array formed by three modules Trina Solar TSM-PD05 270W connected in parallel, where each module has three series-connected submodules. The I-V curves of each string and the array are measured with a BK Precision electronic load, which is controlled from a PC through serial communication. The parameters of the DDM parameters used for the experimental validation are introduced in Tables 4 and 5, where the first one shows the $I_{ph}$ values for each submodule, while the second table presents the rest of the DDM parameters. It is worth highlighting that the parameters of the three submodules in one string are assumed equal; that is why Table 5 shows one DDM parameter for each string.

Table 4. $I_{ph}$ currents of the experimental $3 \times 3$ array.

<table>
<thead>
<tr>
<th>Row</th>
<th>$M = 1$</th>
<th>$M = 2$</th>
<th>$M = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 1$</td>
<td>3.7249</td>
<td>8.3564</td>
<td>6.2051</td>
</tr>
<tr>
<td>$N = 2$</td>
<td>3.7249</td>
<td>8.3564</td>
<td>6.1083</td>
</tr>
<tr>
<td>$N = 3$</td>
<td>3.7249</td>
<td>8.3564</td>
<td>6.1083</td>
</tr>
</tbody>
</table>

Table 5. Double diode model parameters of the $3 \times 3$ array.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_s$ (mA)</td>
<td>$M = 1$ and $N = [1, 2, 3]$</td>
</tr>
<tr>
<td>$I_{ph}$</td>
<td>0.548</td>
</tr>
<tr>
<td>$R_s$ (Ω)</td>
<td>0.548</td>
</tr>
<tr>
<td>$R_h$ (kΩ)</td>
<td>0.548</td>
</tr>
<tr>
<td>$I_{ph}$ (mA)</td>
<td>0.011</td>
</tr>
<tr>
<td>$V_{td}$ (V)</td>
<td>6.006</td>
</tr>
<tr>
<td>$\eta_{bd} \cdot V_{td}$ (mV)</td>
<td>26.892</td>
</tr>
</tbody>
</table>

Serial and Parallel models are used to reproduce the experimental results. The I-V curves of each string and the array are introduced in the left plot of Figure 8; while the error in the estimation of the array current is presented in the right plot of Figure 8. As in the previous examples, the results obtained with both models are the same, which entails the same errors. Moreover, the error varies from $-0.62$ A ($-36.64\%$) to $+0.41$ A ($+4.79\%$), where the highest values of the errors are obtained close to the open-circuit voltage, as in the simulation results. However, both models correctly reproduce the electrical behavior of the PV array according to the results shown in Figure 8 and the NSSE obtained for both models ($NSSE = 0.012\%$).

Additionally, Serial and Parallel (with $N_{w} = 2$) models are executed six times to validate the repeatability of the results obtaining that the average time required by the Serial and Parallel models are 2.1464 s and 2.0908 s. Hence, in this case, the time improvement obtained with the Parallel model is relatively low ($2.59\%$), which is expected considering the low number of strings and rows in the array.

Finally, the simulation and experimental results introduced in this section show the applicability of the proposed model and the consistent time reduction obtained in different evaluating scenarios.
This paper has presented a parallel modeling strategy for SP arrays, which can reduce the processing time required to calculate the power production of the array without impacting the solution accuracy. This parallel solution is based on the double-diode model, it providing a higher accuracy in comparison with the classical single-diode model at low voltage and low irradiance conditions, which are important for partial shading analysis. Therefore, this solution is very useful for reconfiguration systems, since the parallel model will enable a fast modification of the array connections to improve the power production, and such a reconfiguration will be based on accurate predictions.

The previous claims were tested in several scenarios. The first test demonstrated the high accuracy of the parallel model since the prediction errors were very low in comparison with a commercial solution based on Simulink. The second test demonstrated that the proposed parallel model significantly improves the calculation speed over the classical serial model, which in this case reaches 40%; however, the calculation speed can be further improved by increasing the number of workers used to parallelize the calculation process. The third test demonstrated the applicability of the parallel model to reconfiguration systems, where the calculation speed was significantly improved, thus enabling the reduction of the reconfiguration period. Therefore, the PV array can be reconfigured more times during the day, which enables the optimization of the array connections to more partial shading conditions; hence, more power can be generated from the same reconfigurable array. The final test, based on experiments made on commercial PV panels, confirmed the model accuracy for real-world applications.

Another possible application for this parallel solution concerns the estimation of the potential power production of PV plants, which can be subjected to partial shading during some hours of the day (caused by seasons or unavoidable obstacles). This is critical for the economic viability analysis of PV plants, since partial shading could significantly decrease the power production; hence, the lifetime production of the PV plant is a decisive factor to consider. However, estimating the power production of a partially-shaded PV array requires a lot of calculations (and time), thus the proposed parallel solution could significantly reduce the processing time required. Such a fast calculation could enable to test much more possible installation sites and architectures since each test will take less time.

The main disadvantage of the parallel model is the requirement of multiple cores or workers, which is not needed to implement the classical serial model. However, processors
available inside commercial computers have multiple cores (4, 6, 8, and more), thus the parallel model can take advantage of that hardware, which is neglected by the serial model. Another scenario concerns the embedded implementation adopted for real-time applications, which usually provide a single core; in any case, newer SoCs provide multiple cores [44], hence the parallel model can be implemented in embedded applications using those newer devices. Of course, multi-core embedded systems are costly in comparison with single-core devices, but the increased power production provided by the parallel model in reconfiguration systems, under partial shading conditions, could justify the increased cost of the embedded system. Otherwise, in the cases when the power increment is not enough, the serial solution will be more profitable; thus, an economic viability analysis is mandatory.


**Funding:** The APC were covered by the Instituto Tecnológico Metropolitano under the project P21101 “Fortalecimiento y Consolidación del Grupo Automática, Electrónica y Ciencias Computacionales para responder a las necesidades de las industrias 4.0”.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All the data generated in this study are reported in both figures and tables of the paper.

**Acknowledgments:** The authors thank the support of the Instituto Tecnológico Metropolitano, the Universidad Nacional de Colombia, and Minciences (Fondo nacional de financiamiento para la ciencia, la tecnología y la innovación Francisco José de Caldas), under the projects P21101 “Fortalecimiento y Consolidación del Grupo Automática, Electrónica y Ciencias Computacionales para responder a las necesidades de las industrias 4.0” and “Estrategia de transformación del sector energético Colombiano en el horizonte de 2030—Energetica 2030”—“Generación distribuida de energía eléctrica en Colombia a partir de energía solar y eólica” (Code: 58838, Hermes: 38945).

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

**References**


34. Fritsch, F.N.; Shafer, R.E.; Crowley, W.P. Solution of the transcendental equation we w = x. *Commun. ACM* 1973, 16, 123–124. [CrossRef]


