A Rapid Feasibility Checking for Reconfiguration of Mismatched PV Arrays

Dafang Zhao, Fukuhito Ooshita, and Michiko Inoue
Nara Institute of Science and Technology, Japan
Email: {zhao.dafang.yu7, f-ooshita, kounoe}@is.naist.jp

***Abstract*—Power generation efficiency of Photovoltaic (PV) arrays is significantly affected by partial shading and PV cell damage. Partial shading or PV cell damage induces mismatched power generation among PV panels and causes the efficiency loss of power generation. Power generation of mismatched PV arrays can be recovered by re-configuring connection of PV panels. Recently, an efficient and effective reconfiguration method is proposed where it first selects candidates of configurations with approximation and then find the best one with precise power simulation. However, we found that some of configuration candidates are not able to be realized and the method does not show any systematic way to identify such a feasibility. In this paper, we propose an algorithm to rapidly check feasibility that a given configuration candidate can be actually formed by given PV panels. The experimental results demonstrate that the proposed method rapidly checks the feasibility with a small error rate (this part will be written with actual values).**

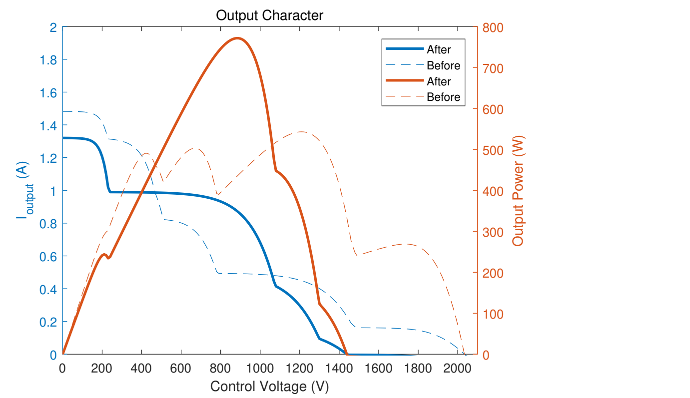


Fig. 1: Power generation before and after reconfiguration

I. Introduction

As fossil fuel depletion and environmental pollution become more serious, green and renewable energy has become necessary for a sustainable society and environment. Photovoltaic (PV) systems receive significant attention since the sun has unlimited energy and PV arrays can be easily scaled up. However, due to the nature of a PV cell, which is a basic component of a PV array, PV arrays are sensitive to partial shading and PV cell damage. PV cells could not uniformly generate power when PV cells experience different irradiances or some of cells have physical defects, and such a mismatched condition might accelerate heating and aging of PV cells and cause further damaging. To prevent PV cells from damaging, bypass diodes are usually placed in PV arrays and they are turned on in mismatched conditions. However, it causes a significant loss of power generation.

PV arrays are hierarchically constructed such that a group of PV cells form a PV module, a group of PV modules form a PV panel, and a group of PV panels form a PV array. Power generation of mismatched PV arrays can be recovered by reconfiguring connection of these components. Fig.1 shows an example of power generations of a PV array with a partial shading. We distributed shading cells non-uniformly to a PV array with 3×4 PV panels and applied power simulation. Before reconfiguration, it has two peaks in power generation, while, if we reconfigure connection among PV panels, it can generate more power.

Though several reconfiguration methods have been proposed, most works consider reconfiguration in PV cell or PV module level [?], [1]–[4]. However, cell level reconfiguration requires a significantly high computation time and a large number of switches for reconfiguration. In addition, they require a special PV panels with capability of switching, and cannot be applied to the system constructed with standard PV panels. From a practical view, reconfiguration of connection among PV panels is a realistic solution since a PV panel is manufactured as a physical one panel with two terminals and PV panels can be flexibly interconnected.

Reconfiguration in PV panel level has been investigated [?], [5], [6]. A PV array reconfiguration using genetic algorithm (GA) was proposed in [6]. Though it can give a new configuration, computing cost is significantly high and the algorithm cannot generate the best configuration precisely. Hu et al. also addressed PV panel reconfiguration where they formulate a nonlinear integer programming problem to optimize power generation by reconfiguration [?]. Orozco-Gutierrez et al. proposed an efficient and effective reconfiguration method [5] where it first selects candidates of configurations with approximation and then finds the best one with precise power simulation. However, the candidates are specified in a PV module level though PV modules could not be fully reconfigured. Actually, we found that some of configuration candidates are not able to be realized. However, the paper [5] does not show any systematic way to identify such a feasibility.

In this paper, we propose an algorithm to rapidly check feasibility that a given configuration candidate is actually formed by given PV panels. The proposed method can efficiently check the feasibility while identifying most feasible cases accurately. The experimental results demonstrate the effectiveness of the proposed method where ...

The remaining of this paper is organized as follows. Section II introduces a PV array which is a targeting PV system in this paper. Section III gives related works. Section IV defines a feasibility, and Sections V and VI give the details of algorithm and evaluation. Section VII shows the experimental results. Conclusions are provided in the last section.

II. Photovoltaic array

A PV system or PV array is composed of PV panels that have two terminals of plus and minus and can be interconnected. There are two common connection styles for PV arrays, *series-parallel* array and *total-cross-tied* array. In a series-parallel array, PV panels are connected in series, and multiple series connections are connected in parallel. In a total-cross-tied array, parallel connections of PV panels are connected in series. In this paper, we focus on series-parallel arrays, however the basic idea of the proposed method can be applied also to total-cross-tied arrays. Hereafter, we simply call a series-parallel array a *PV array*.

Figure 2 shows an example of a PV array. A PV array is a parallel connection of (PV) strings where a string is a series connection of PV panels. A PV panel is a series connection of PV modules, and a PV module is a parallel connection of a series of PV cells and a bypass diode. A typical PV panel is composed of three PV modules each of which has 12-24 PV cells. A PV module can generate power for a given voltage according to its I-V characteristics as shown in Fig.3(a). The I-V characteristics is affected by irradiance level and physical damage of PV cells. Figure 3(a) also shows a typical degradation of a I-V characteristics where generated current is reduced with some ratio while keeping the voltage range. When a PV panel has a partial shade, that is its PV modules have different irradiance levels and hence different I-V characteristics, the PV panels might have multiple peaks (maximum power points, MPPs) in power generation as shown in Fig.3(b).

To find out accurate I-V characteristics of a PV panel, we need a time consuming power simulation. However, we can roughly understand I-V characteristics of a PV panel (and also a string) as follows. When a control voltage is low, PV modules with high irradiance level are active while PV modules with low irradiance level are inactive with turning on their bypass diodes. In this case, a high current can flow in the PV panel. When control voltage is high, some of bypass diodes become turning off and the corresponding PV modules become active. In this case, the current level is reduced and generated power sometimes increases and sometimes decreases. The generated power at MPPs can be roughly estimated. In Fig.3(b), one PV panel has three PV modules with different irradiance levels. The peak currents for these modules are 5A, 3A, and 1.5A, respectively. Voltages at MPPs are roughly 20V, 40V, 60V in Fig.3(b), those are roughly multiples of a voltage of MPP for one PV module (around 20V in this case). At the first MPP (MPP1), only one PV module with a peak current of 5A is active at a control voltage of 20V, so the generated power is 100W. At the second MPP (MPP2), two PV modules with peak currents of 5A and 3A are active at a control voltage of 40V. In this case, the current level of the PV panel is 3A since two PV panels are connected in series and they have to have the same current level, and the generated power is 120W. At the third MPP (MPP3), the current level of the PV panel is 1.5A at a control voltage of 60V, and the generated power is 90W.

In a PV array, we have two constraints. PV panels in the same string have the same current level, while all the strings have the same control voltage. When considering reconfiguration of PV panel connection, we should find out the best configuration while considering these constraints.

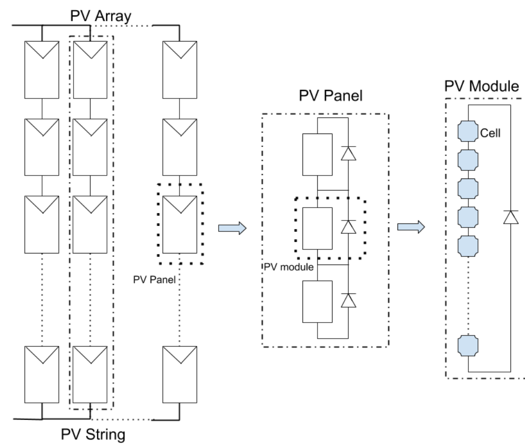
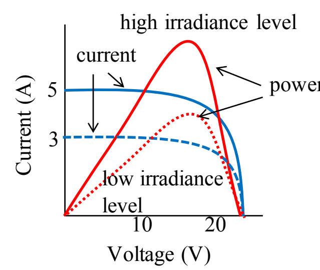
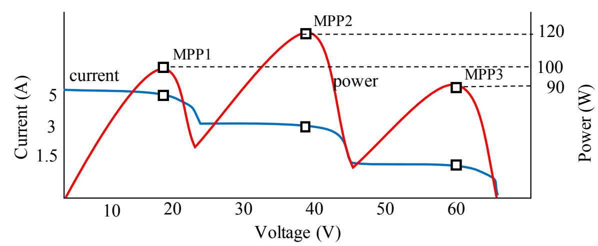


Fig. 2: PV array, string, module and panel



(a) PV module



(b) PV panel

Fig. 3: I-V characteristics

TABLE I: Extracted currents of MPPs

	Panels			
	$P1$	$P2$	$P3$	$P4$
MPP1	3.10A	3.09A	0.52A	2.47A
MPP2	2.98A	2.55A	0.50A	1.53A
MPP3	1.55A	2.48A	0.46A	0.48A

TABLE II: Approximated currents of MPPs

	Panels			
	$P1$	$P2$	$P3$	$P4$
MPP1	3A	3A	0.5A	2.5A
MPP2	3A	2.5A	0.5A	1.5A
MPP3	1.5A	2.5A	0.5A	0.5A

III. OROZCO-GUIERREZ'S METHOD

In this section, we briefly introduce a method proposed by Orozco-Guierrez et al. [5] as the most related work to this paper. Orozco-Guierrez et al. proposed an efficient reconfiguration algorithm for mismatched PV arrays. The method utilizes information of MPPs of every PV panel that are extracted using an online monitoring [8] and a power estimation [9]. Then currents and voltages of MPPs are approximated and grouped into a small number of classes so that the number of possible combinations, or a search space, is reduced. For example, Table I shows extracted currents for 4 panels each of which has 3 modules, and they are approximated into four current levels as shown in Table II. As mentioned in Section II, if some panel has m MPPs with current level of I or larger, m modules can be active with current level of I. Table III shows the maximum number of active modules for each current level.

Then power values are approximated for possible configurations. Table IV shows an example to form a PV array with two strings where control voltages are approximated as multiples of 20V (the number of active modules per string times 20V). The method has a simple feasibility check as follows.

(Feasibility 1) For the n-th highest current level I_n, the number of active modules in a system is $N(I_n)/n$ or less.

Where, $N(I)$ denotes the total number of active modules for a current level I. For example, consider a candidate current pair (3A. 2.5A). In this case, the highest and the second highest current levels are 3A and 2.5A, respectively, and the total number of active modules for 3A and 2.5A are 3 and 6, respectively (see Table III). Therefore, the number of active modules in the first string is 3 or less, and the number of active modules in the second string is also $6/2 = 3$ or less. Consequently, the number of active modules per string is at most 3 for a candidate current pair (3A. 2.5A). Table IV has values in the cells when Feasibility 1 is satisfied.

The method finds the largest power value from possible candidate configurations (330W for current pair (3A, 2.5A) with three active modules in this case) and evaluates the case more precisely along with the cases with close values to the best value since the analysis are given with approximated values. In the case of Table IV, 330W, 320W and 300W are selected for further evaluation.

TABLE III: Maximum number of active modules

	Panels				
	$P1$	$P2$	$P3$	$P4$	total
3A	2	1	0	0	3
2.5A	2	3	0	1	6
1.5A	3	3	0	2	8
0.5A	3	3	3	3	12

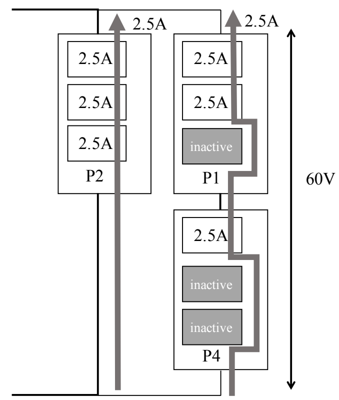


Fig. 4: Feasible configuration

IV. FEASIBILITY

In the method [5], to execute a power simulation for a PV array, we need to assign panels into strings to realize a candidate configuration. For example, a candidate configuration for a current pair (2.5A, 2.5A) with 3 active modules is realized by assign panel P2 to the first string and panels P1 and P4 to the second string as shown in Fig.4. However, we could not find feasible assignment for a current pair (3A, 2.5A) with 3 active modules or a current pair (2.5A, 1.5A) with 4 active modules though they are expected to generate higher powers 330W or 320W. That is, the condition Feasibility 1 is a necessary but not sufficient to check the actual feasibility.

Now we will define the feasibility. For a PV array A with s strings, let $M_{A,i}(I)$ denote the number of modules that can be active with a current level I in the i-th string. Let $Q = (Q_1, Q_2, \ldots, Q_s)$ be a sequence of currents required for strings. A sequence of currents Q is feasible with m modules if and only if it is possible to form a PV array A such that

$$M_{A,i}(Q_i) \geq m \tag{1}$$

holds for each string $i (1 \leq i \leq s)$.

V. AN ALGORITHM TO IDENTIFY FEASIBILITY

A. Feasibility Problem

In this section, we propose an algorithm to identify feasibility. First we will formulate the feasibility problem that identify the feasibility for given current sequence and the number of active modules per string as follows.

Definition 1 (Feasibility Problem):

Input: information of MPPs (approximated values), a current sequence Q, and the number m of active modules per string.

Output: Whether Q with m modules is feasible or not.

TABLE IV: Approximated power

# modules per string	currents sequence for strings (A) (3,3)	(3,2.5)	(3,1.5)	(3,0.5)	(2.5,2.5)	(2.5,1.5)	(2.5,0.5)	(1.5,1.5)	(1.5,0.5)	(0.5,0.5)
1	120W	110W	90W	70W	100W	80W	60W	60W	40W	20W
2	-	220W	180W	140W	200W	160W	120W	120W	80W	40W
3	-	**330W**	270W	210W	**300W**	240W	180W	180W	120W	60W
4	-	-	-	-	-	**320W**	240W	240W	160W	80W
5	-	-	-	-	-	-	**300W**	-	200W	100W
6	-	-	-	-	-	-	-	-	240W	120W

B. Outline of the algorithm

In the proposed algorithm, we try to configure a PV array and identify the feasibility. For a given current sequence $Q = (Q_1, Q_2, \ldots, Q_s)(Q_i \geq Q_j \text{ if } i \leq j)$, we will assign panels to strings in the order of $Q_1, Q_2, \ldots, Q_s$. When selecting a panel to a string, we will consider how much we lose an opportunity that the panel is more effectively used for other current level. For example, if we select Panel 1 in Table III for a string with a current level 3A, two modules can be active while the remaining one module becomes inactive. The remaining one module still is not active if $P1$ is used for a current level 2,5A, while it can be active if it is used for current level 1.5A or 0.5A. We will consider that when selecting $P1$ for a current level 3A, we don't have any loss for a current level 2.5A but lose one module for 1.5A and 0.5A. The algorithm select panels so as to minimize such losses.

Let $M_p(I)$ denote the number of modules that can be active at current level I in a PV panel p. For a given current sequence $Q = (Q_1, Q_2, \ldots, Q_s)(Q_i \geq Q_j \text{ if } i \leq j)$, a loss of selecting a panel p at the $k-1$-th string for the k-th string is defined as $Loss(p, k) = M_p(Q_k) - M_p(Q_{k-1})$.

To assign panels for the n-th string, the following steps are applied. Let m be the required number of active modules.

1) Sort unselected PV panels in a lexicographically ascending order of $Loss(p, n+1)$, $Loss(p, n+2)$,. . ., $Loss(p, s)$ and $M_p(Q_n)$.
2) Select PV panels until selecting m or more active modules for a current level Q_n.
3) Cancel redundant PV panels so that the number of active modules at a current level Q_n is minimized.
4) Swap selected PV panels with unselected PV panels so that the number of selected PV panels is minimized.

Before explaining more details, we first show an example. Table V shows how to select PV panels for the first string where the required number m of active modules is 5. Panels are sorted by $Loss$ and $M_p(Q_1)$ (the number of active modules for the first string) in the order of $P1$, $P2$, …, $P7$. In the example, panels $P2$ to $P6$ are at the same loss level $(1, 0)$. PV panels are selected for each loss level. First, $P1$ is selected, then panels at the second loss level are selected in the order of $P2$, $P3$, …. After selecting until $P5$, the number of active modules exceed $m(= 5)$. In this case, if there is some redundant panels, these panels are canceled. In the example, $P4$ is canceled. Finally, if some of selected PV panels can be swapped with less number of unselected panels, they are swapped. In the example, $P2$ and $P3$ are swapped with $P6$. Consequently, three panels $P1$, $P5$ and $P6$ are selected.

TABLE V: An example of the proposed method ($m = 5$)

	Panels $P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$
$M_p(Q_1)$	1	1	1	1	2	2	2
$M_p(Q_2)$	1	2	2	2	3	3	2
$M_p(Q_3)$	1	2	2	2	3	3	3
$Loss(p, 2)$	0	1	1	1	1	1	1
$Loss(p, 3)$	0	0	0	0	0	0	1
step 2	√	√	√	√	√		
step 3	√	√	√		√		
step 4	√				√	√	

TABLE VI: Swap rule

The numbers of active modules selected panels	unselected panels
1, 1, 1	3
1, 1	2
1, 2	3

C. Algorithm

Step 1 sorts unselected panels (with modules that can be active for the target current), and Step 2 selects panels one by one in the sorted order until reaching the required number of active modules.

Step 3 optimizes the number of active modules. In Step 3, if the number of active modules exceeds the required number m, redundant panels are canceled if exist. A panel is *redundant* if the number of active modules is still m or more even if the panel is canceled. Redundant panels are searched at loss level. Starting with a loss level where the last panel is selected, we will check the number of active modules $M_p(Q_n)$ for each panel p where Q_n is a required current level for the current string. Redundant panels are selected and canceled from panels with less number of $M_p(Q_n)$ in the same loss level. If there still exceed m after canceling, we will go to the previous loss level and try to cancel panels. This procedure is repeated until selecting an exact m active modules or checking all the loss levels.

Step 4 optimizes the number of selected panels. In Step 4, we will swap selected panels and an unselected panel in the same loss level to reduce the number of selected panels. There are 3 swap cases as shown in Table VI. Algorithm 1 shows a pseudo code of the proposed algorithm.

Algorithm 1: Feasibility Check

Input: $Q = Q_1, Q_2, \ldots, Q_s$: required current for each string; m: The minimum number of active modules per string; M_p: The number of active modules in each panel p; s: Number of strings;
Output: Feasibility Result;

```
for each string k(2 ≤ k ≤ s) and panel p do
    Loss(p, k) = M_p(Q_k) − M_p(Q_{k−1})
end
for j = 1 to s do
    if ∑ M_p(Q_j) for unselected panes < m then
        return : Feasibility = NO
    end
    PS = a set of unselected panels with M_p(Q_j) > 0;
    // step 1
    sort PS in a lexicographically ascending order of
     Loss(p, j+1), Loss(p, j+2), ..., Loss(p, s),
     M_p(Q_j);
    // step 2
    select panels in the order in PS until ∑ M_p(Q_j)
     for selected panels ≥ m;
    //step 3
    group selected panels LV_1, LV_2,..., LV_h with loss
     level (LV_1 is the lowest level);
    for i = h to 1 do
        find and cancel redundant panels in LV_i in the
         order in PS
    end
    //step 4
    for i = h to 1 do
        apply swap rules for selected panels in LV_i and
         an unselected panel in the same loss level
    end
end
return : Feasibility = YES
```

D. Evaluation

Although the pilot example refers to three strings only, the extension of the method to any number of parallel connected strings is the same.

To evaluate the performance of proposed algorithm, we compare the result with exhaustive searching algorithm for 100 random shadow distributed PV array. For each PV array, contains 6 - 24 PV panels connected into 3 - 6 PV strings. For each PV panels, it will have 3 - 8 working current value.

The computational time required for finding optimal configuration and accuracy of feasibility judgment in the proposed algorithm and exhaustive search algorithm are compared in Table VII. As shown in Table VII, proposed algorithm achieved less computing time compare with exhaustive search algorithm and with high accuracy.

VI. Simulation

In this section, we consider using 3 different size PV array, which are 3×4 array, 60×4 array and 40×15 array. The whole process though SPICE simulation with the configuration result by proposed algorithm. The PV cell's parameters are shown in Table VIII.

TABLE VII

COMPARISON BETWEEN EXISTED METHODS AND PROPOSED ALGORITHM

	Proposed Algorithm	Exhaustive search Algorithm
Number of PV Array	-	
Number of MPP candidades	-	
Feasible Candidates	-	-
Infeasible Candidates	-	-
Error Rate	0.89%	0%
Ave. times per Array	0.157s	4370s

TABLE VIII: PV Cell Parameters

Parameter	Value
Short-Circuit Current	0.66A
MPP-Current	0.58A
Open-Circuit Voltage	280V
MPP-Voltage	215V
Max Power Output	124.7W

A. 3×4 Array

In a 3×4 array, there are 12 panels connected in parallel for 4 strings, for each string contains 3 PV panels. The shadow conditions for PV array are not uniformly, as shown in Fig. **??**. Without reconfiguration, PV array output I-V characteristic as shown in Fig. **??**. The global MPP as how many Watt on how many voltage. After proposed algorithm based reconfiguration, this PV array achieved how many Watt on how many voltage for global MPP. The output power has increased by how many %

B. 60×4 Array

For medium size 60×4 array, on standard test condition the power output are how many KW without any shadow. Fig. **??** presents the shadow distribution amount the PV array which power output are how may KW. After reconfiguration, panel arrangement as illustrated in Fig. **??**. With reconfiguration, power generation increased by how many %.

C. 40×15 Array

As a large scale PV array. Although the algorithm proposed in [5] are hard to converge when the configuration candidates are widely distributed. The algorithm we proposed still calculate the feasible configuration result very fast. 60 configuration candidates though the feasibility judgment algorithm in 269.6s with 20 feasible configuration as shown in Table **??**.

VII. Conclusions

Acknowledgment

Reference

[1] D. Nguyen and B. Lehman, "An adaptive solar photovoltaic array using model-based reconfiguration algorithm," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2644–2654, 2008.

[2] Y. Wang, X. Lin, Y. Kim, N. Chang, and M. Pedram, "Architecture and control algorithms for combating partial shading in photovoltaic systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 917–930, 2014.

[3] J. P. Storey, P. R. Wilson, and D. Bagnall, "Improved optimization strategy for irradiance equalization in dynamic photovoltaic arrays," *IEEE transactions on power electronics*, vol. 28, no. 6, pp. 2946–2956, 2013.

[4] J. Storey, P. R. Wilson, and D. Bagnall, "The optimized-string dynamic photovoltaic array," *IEEE Transactions on Power Electronics*, vol. 29, no. 4, pp. 1768–1776, 2014.

[5] M. Orozco-Gutierrez, G. Spagnuolo, J. Ramirez-Scarpetta, G. Petrone, and C. Ramos-Paja, "Optimized configuration of mismatched photovoltaic arrays," *IEEE J. Photovolt*, vol. 6, no. 5, pp. 1210–1220, 2016.

[6] P. L. Carotenuto, A. Della Cioppa, A. Marcelli, and G. Spagnuolo, "An evolutionary approach to the dynamical reconfiguration of photovoltaic fields," *Neurocomputing*, vol. 170, pp. 393–405, 2015.

[7] F. Iraji, E. Farjah, and T. Ghanbari, "Optimisation method to find the best switch set topology for reconfiguration of photovoltaic panels," *IET Renewable Power Generation*, vol. 12, no. 3, pp. 374–379, 2017.

[8] P. L. Carotenuto, P. Manganiello, G. Petrone, and G. Spagnuolo, "Online recording a pv module fingerprint," *IEEE Journal of Photovoltaics*, vol. 4, no. 2, pp. 659–668, 2014.

[9] M. Orozco-Gutierrez, J. Ramirez-Scarpetta, G. Spagnuolo, G. Petrone, and C. Ramos-Paja, "Fast estimation of mpps in mismatched pv arrays based on lossless model," in *Clean Electrical Power (ICCEP), 2015 International Conference on*, pp. 435–440, IEEE, 2015.