

Accurate Modeling of Partially Shaded PV Arrays

Bennet Meyers* Mark Mikofski†

*Department of Electrical Engineering, Stanford University, Palo Alto, CA, 94305, USA

†SunPower Corporation, Richmond, CA 94804, USA

Abstract—Partial shading is the condition of nearby objects casting shade onto part of a photovoltaic (PV) array, causing the PV modules to receive non-uniform irradiance. Non-uniform shading causes electrical mismatch between elements within the array, resulting in a non-linear reduction in energy capture. Accurately modeling mismatch conditions is a particularly difficult problem due to the large number of parameters needed to fully define a PV system and its operating state. Furthermore, the large number of possible system state conditions make the models computationally complex. Previous work on this topic has addressed these difficulties by simplifying the system representation, reducing the number of parameters used, and limiting the domain space to simple irradiance patterns that may not be representative of real shade conditions. In this paper, we review common modeling approaches to address this problem and provide an overview of PV equivalent circuit theory. We then present PVMismatch—free open-source software written in Python by the authors for simulating full PV system current-voltage curves. Finally, we demonstrate an improvement over common practice for modeling this type of behavior, illustrating that modeling mismatch behavior at the PV cell rather than PV module level provides more accurate results—up to 30% less over-prediction with respect to module level estimates—and better insight into system behavior.

Index Terms—photovoltaic systems, modeling, partial shade modeling, electrical mismatch

I. INTRODUCTION

Partial shading of photovoltaic arrays is well-known to cause significant reduction in system performance. According to the California Energy Commission, “Shading of photovoltaic systems, even partial shading of arrays, can be the most important cause of failure to achieve high system performance” [1]. A significant body of research has been developed around modeling the performance of partially shaded PV systems [2]–[7]. These models are being used by researchers and industry professionals to evaluate novel power electronics solutions, the feasibility of mitigating shade losses through non-standard wiring topologies, and the impact of shade on yearly energy estimates. Typically, researchers utilize an equivalent circuit model to describe the I-V characteristics of individual PV modules within the array. This allows researchers to predict the power lost to current mismatch between series-connected modules and voltage mismatch between parallel-connected strings.

However, this approach ignores electrical mismatch that occurs between PV cells within the modules and does not address the impact of bypass diodes. This paper will show that there can be significant errors introduced by ignoring these effects. Using a module-level equivalent circuit model does not adequately differentiate between the performance under

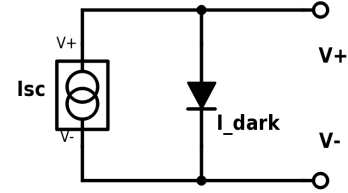


Fig. 1. The simplest equivalent circuit model: a current source in parallel with an ideal diode.

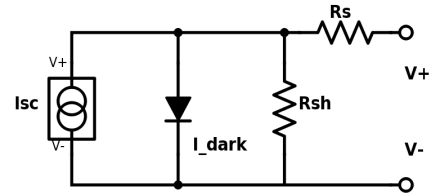


Fig. 2. A one-diode equivalent circuit model with lossy components.

different shade geometries or the impacts of new module-level designs such as cross-tied cells and different diode configurations. Additionally, using a module-level approach tends to underestimate the loss in power overall.

Previous studies have also been limited to simple distributions of irradiance within an array, such as 3 to 5 irradiance levels oriented in simple rows and columns (see [3]). A more complex study such as [4] relies simply on random patterns of irradiance to generate a dataset of 30 scenarios.

This paper demonstrates the open-source software package PVMismatch, developed by the authors. The theoretical basis of the software is presented. Using this software, the impact of different shading geometries on a representative system is evaluated, using both the cell-level model and a module-level model, and the results of the two modeling approaches are compared.

II. EQUIVALENT CIRCUIT MODELS

A. One-Diode Models

The current-voltage (IV) response of a PV cell can be approximated by the superposition of a short-circuit photocurrent and a “dark current,” the current response of the cell under an applied voltage and no illumination [8]. In other words, an ideal photovoltaic cell is electrically equivalent to a current generator in parallel with a non-linear resistive element such as a diode, as shown in Fig. 1. When the cell is illuminated, it

produces a photocurrent proportional the light intensity, which is then split between the diode and the load. As the load resistance increases, more of the current flows through the diode, resulting in a larger terminal voltage at the load. This behavior is described by the following equation:

$$I = I_{sc} - I_0 \left(e^{qV/k_B T} - 1 \right) \quad (1)$$

where I is the terminal current of the cell, I_{sc} is the short-circuit current under a given illumination, I_0 is a constant known as the saturation current, q is the elementary charge, k_B is the Boltzmann constant, and V and T are the voltage and temperature of the diode, respectively. The second term in Eqn. 1 is simply the Shockley diode law.

Real cells dissipate power through many different parasitic resistances. These parasitic effects are grouped into two electrically equivalent resistances—series resistance (R_s) and shunt resistance (R_{sh})—as shown in Fig. 2. Series resistance arises from resistance to current flow by the cell material itself and by the connections between the semiconductor material and the cell contacts. Shunt resistance occurs due to leakage of current through the cell, around the edges of the device, and between contacts of different polarity. In an ideal cell, $R_s = 0$ and $R_{sh} = \infty$.

Additionally, the ideal diode behavior described in Eqn. 1 is not realistic; the dark current generally has a weaker dependence on bias [8]. So, an ideality factor n is introduced, with typical values between 1 and 2, where $n = 1$ describes an ideal diode. Taking the parasitic resistances and diode nonideality in to account, Eqn. 1 becomes:

$$I_{cell} = I_{sc} - I_0 \left(e^{q(V_c + IR_s)/nk_B T} - 1 \right) - \frac{V_c + IR_s}{R_{sh}} \quad (2)$$

where I_{cell} and V_c are the terminal current and voltage respectively. The diode voltage is given by $V_d = V_c + IR_s$. The last term (V_d/R_{sh}) is called the shunt current (I_{sh}).

B. Two-Diode Models

Further improvements to this model can be made by taking recombination effects into account. Recombination in the depletion region is a loss factor not considered in the original Shockley equation nor the lossy equivalent circuit model described in Eqn. 2. However, it has been shown that these effects can be modeled by considering a second diode in parallel to the first [9], leading to a two-diode equivalent circuit model, which is shown in Fig. 3.

In most implementations of the 2-diode model, the diode ideality factors, n_1 and n_2 , are set to the values 1 and 2 respectively. This differs from a 1-diode model in which the ideality factor is variable. The diode currents, I_{d1} and I_{d2} , are again given by Shockleys equation:

$$I_{d1} = I_{sat1} \left(e^{qV_d/n_1 k_B T} - 1 \right) \quad (3)$$

$$I_{d2} = I_{sat2} \left(e^{qV_d/n_2 k_B T} - 1 \right) \quad (4)$$

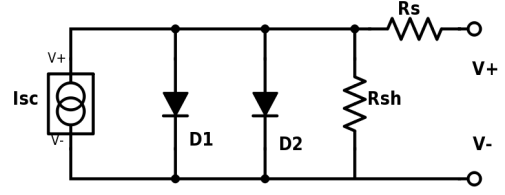


Fig. 3. A two-diode equivalent circuit model with lossy components.

where I_{sat1} and I_{sat2} are the saturation currents for each diode. This allows us to formulate the generic 2-diode model:

$$I_{cell} = I_{sc} - I_{d1} - I_{d2} - I_{sh} \quad (5)$$

C. Additional Model Improvements

For some cell technologies, the saturation current of the first diode shows a strong cubic correlation with temperature [10] as given in Eqn. 6. E_g is the band gap of the PV device material and T_0 is the reference temperature, 25 °C. The saturation current of the second diode is assumed to be constant.

$$I_{sat1} = I_{sat1} \bigg|_{T=T_0} \frac{T^3}{T_0^3} \exp \left[\frac{E_g q}{k_b} (T_0^{-1} - T^{-1}) \right] \quad (6)$$

Additionally, the short circuit current term in Eqn. 5 (I_{sc}) is replaced with the more general photogenerated current (I_{ph}), as shown in Eqn. 7. I_{sc} is a function of the effective irradiance ($E_e = E/E_0$), the temperature coefficient (α), and the difference between the cell and reference temperatures ($\Delta T = T - T_0$). Effective irradiance is defined as the ratio of incident irradiance transmitted to the cell and the reference irradiance, $E_0 = 1000 \text{ W m}^{-2} = 1\text{sun}$.

$$I_{ph} = A_{ph} I_{sc} = A_{ph} E_e I_{sc0} (\alpha \Delta T) \quad (7)$$

The proportionality constant, A_{ph} , can be represented explicitly by solving the circuit in Fig. 3 at the short circuit condition (when $V = 0$).

$$A_{ph} = 1 + \frac{I_{d1} + I_{d2} + I_{sh}}{I_{sc}} \bigg|_{V=0} \quad (8)$$

Finally, a better estimate for I_{sh} is obtained by adding a term representing the reverse breakdown current (I_{rbd}).

$$I_{sh} = \frac{V_d}{R_{sh}} + I_{rbd} \quad (9)$$

The reverse breakdown current is given by an avalanche breakdown expression from [11] which was modified by adding a quadratic term to allow more flexibility in fitting reverse bias. The expression has four parameters, the device specific avalanche breakdown voltage (V_{rbd}), a positive exponent (n_{rbd} , typically set to 4), and two additional coefficients (a_{rbd} and b_{rbd} , typically set to 10^{-4} and zero).

Letting $r_v = V_d / (R_{sh} I_{sc0})$, we have the following equation for the reverse breakdown current:

$$I_{rbd} = I_{sc0} (a_{rbd} r_v + b_{rbd} r_v^2) \left(1 - \frac{V_d}{V_{rbd}}\right)^{-n_{rbd}} \quad (10)$$

III. PVMISMATCH OVERVIEW

PVMismatch is free open source software written in the Python computer language. PVMismatch can be downloaded from the Python Package Index at <https://pypi.python.org/pypi/pvmismatch>. The software can be used to simulate both forward and reverse bias regions of current-voltage (IV) curves for various combinations of photovoltaic cells, modules, and strings to form full PV systems. Cell, module, string, and system constraints are all independent and variable. The basic model building block is the 2-diode equivalent circuit model described in the previous section.

PVMismatch uses an explicit method to calculate the full IV curve simultaneously at each instance [11] instead of an iterative or Lambert W-function method to solve for individual points sequentially. There are two advantages to this method.

- In general for very large datasets, libraries like BLAS (Basic Linear Algebra Subprograms, available at NetLib <http://www.netlib.org/blas/>) can perform linear algebra operations on vectors of data significantly faster than looping.
- The result of the simulation is the entire IV curve instead of select points.

The user specifies the desired number of points in the IV curve, and the points are distributed using two log-space distributions to increase density around the maximum power point and the reverse breakdown voltage. Then Bishop's explicit method is applied in three steps:

- 1) Create a range of diode voltages with user specified resolution.
- 2) Evaluate corresponding cell currents from diode voltages utilizing Eqns. 3–10.
- 3) Evaluate cell voltage from $V_d = V_c + IR_s$.

The cells' IV curves are then combined according to Kirchhoff's circuit laws to form modules, strings, and full PV systems by adding either voltages or currents and interpolating when necessary. Bypass diodes within modules are treated very simply as perfect conductors when the cell substring voltage is less than the bypass diode trigger voltage and perfect insulators otherwise.

PVMismatch is an object oriented program. Cells, modules, and strings are instances of objects that contain the attributes and methods that relate to them. A system contains strings, a string contains modules, and modules contain cells. Memoization of identical cells, modules and strings is used to speed up calculations and save memory. Each simulation initially begins with only a single instance of each cell, module and string given by the initial parameters specified by the user. In addition to the parameters already described, the user can



Fig. 4. A diagram of the cell layout of the 96-cell module modeled in this study. The three colors designate the cells belonging to each of the three bypass diodes.

also specify the module cell configuration, the number of modules per string, and number of strings in the system. Cell configurations can be customized for in-series or cross-tied substrings of cells. Then as the user changes cell parameters such as irradiance and temperature, copies of cells are made as needed. Therefore as systems become more complex (for example, different degradation on every cell), the optimization through memoization becomes less efficient. Typical systems with mostly nearly identical cells will be the most optimized.

IV. METHODOLOGY

For this study, an 8×3 system was simulated in PVMismatch—three strings in parallel, each with eight PV modules. This PVMismatch model was previously validated against field data in [12]. The modules each have 96 cells in series, arranged physically in 8 columns of 12. The modules have three bypass diodes each, grouping the cells in a 24 – 48 – 24 pattern. The physical layout of the module, including the grouping by bypass diodes, is shown in Fig. 4.

The Python library Shapely [13] was used to generate iterative geometric shade patterns to project onto a 2D representation of the array, with the modules arranged in a simple rectangle and the ground cover ratio (GCR) in both dimensions equal to 1. The modules were oriented in “portrait” relative to the orientation of the module strings, as shown in Fig. 5.

Six families of geometries were selected for this study:

- 1) Vertical shade increasing from the left (short edge)
- 2) Horizontal shade increasing from the bottom (long edge)
- 3) Increasing rectangles from the bottom left (rectangles)
- 4) Increasing circles from the bottom left (circles)
- 5) Increasing angular shade from the left (angles sweep short)
- 6) Increasing angular shade from the bottom (angles sweep long)

The general behavior of each set is shown in Fig. 6. 50 iterations of each geometry were generated, for a total of 300

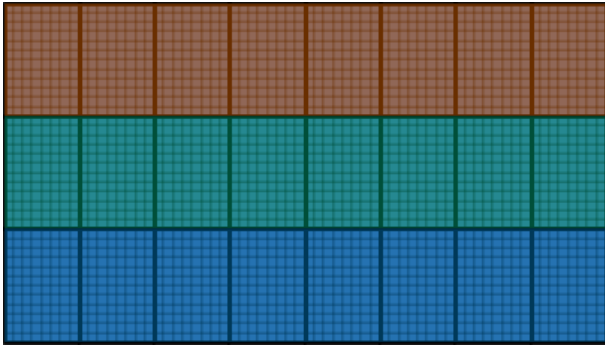


Fig. 5. The 2D layout of the system, as it was modeled for shade projections. The three colors represent the three parallel strings of 8 modules each. Note that the bold rectangles represent modules, while the thin squares represent individual cells.

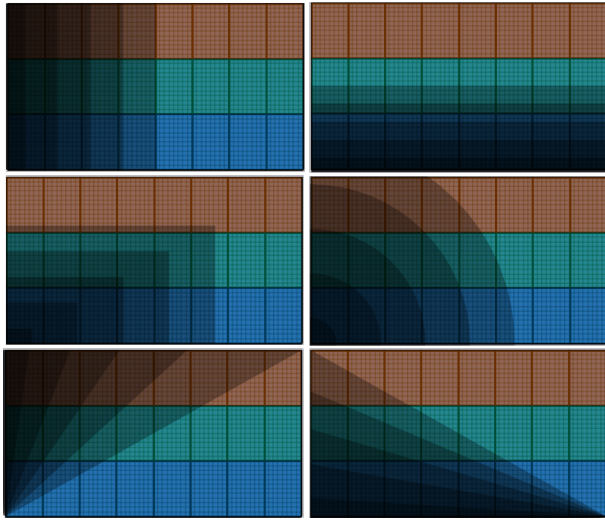


Fig. 6. A graphical representation of the six shade geometry families in this study. Only 5 of the 50 iterations in each set are shown here, for the sake of clarity. The numbering in the text is in row-major order.

unique shade scenarios. Each set of iterations range from 1% shade coverage to 50% shade coverage.

We chose to model the system with an unshaded irradiance of 1000 W m^{-2} and a cell temperature of 25°C . It has been shown previously [12] that the light intensity in the shaded region does not have a strong impact on normalized performance, as long as the shaded irradiance is less than 25% of the unshaded irradiance.

In addition to the shade patterns, all system objects (cells, modules, and strings) are also represented as Shapely geometry objects. The Shapely software can then calculate the intersection between the shade geometries and the system objects, which we use to estimate the irradiance on every individual cell. If entire modules or strings are uniformly shaded, this information is also captured for more efficient modeling in PVMismatch. The irradiance information is packaged into a nested dictionary data structure, which is then passed to PVMismatch for IV and PV curve modeling.

For each shade scenario, system performance was modeled

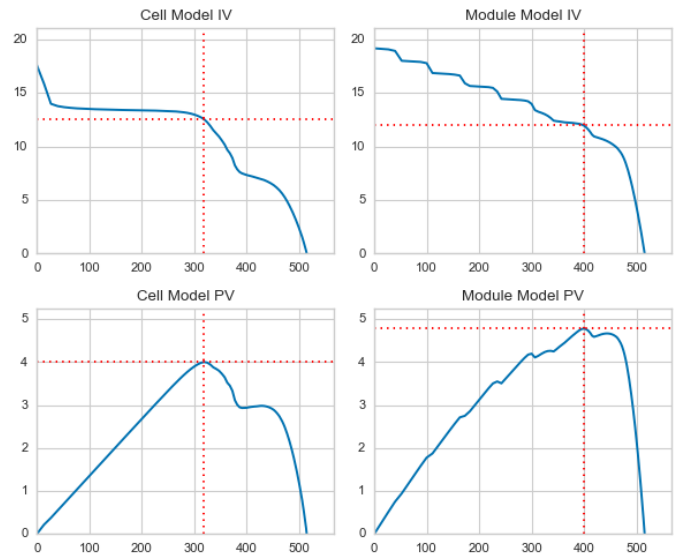


Fig. 7. A comparison of the system IV and PV curves generated by PVMismatch. The plots show a single shade scenario from group #6. The cell-level model predicts approximately 10% less power, relative to P_{mp0} , or 20% less power relative to the module-level model. Red dotted lines show max-power operating point.

two ways. First, the full dictionary of irradiance values was passed to PVMismatch to calculate the system IV and PV curves based on the cell-level irradiance values. Second, average irradiance values were calculated for each module, and PVMismatch was used to calculate the system IV and PV curves based on the average module-level irradiance values. As previously stated, the goal is to compare the two approaches in order to determine the utility of implementing a cell-level equivalent circuit model instead of the much simpler module-level model.

V. RESULTS AND DISCUSSION

Fig. 7 compares the full IV and PV curves for a single shade geometry, modeled using both approaches. In this case, the cell-level model predicts significantly more power loss than the module-level model. The models have some agreement on the operating current (I_{mp}), but predict very different operating voltages (V_{mp}).

A full summary of the simulations described in the previous section are plotted in Figs. 8 and 9. The response variable in both plots is normalized maximum power: the true max power point of the array under that shade condition divided by P_{mp0} (the maximum power of the system under standard test conditions with no shade). The independent variable in this analysis is shade ratio, or the fraction of the array area covered by shade. On both plots, reference lines at $y = 1 - x$ and $y = 1 - 2x$ are included for reference. The first line is the theoretical maximum power output of any system for the given shade ratio. The second line represents the result of assuming a Shade Impact Factor (SIF) of 2, the default value chosen by the California Energy Commission for shade performance analysis (see page C-18) [1].

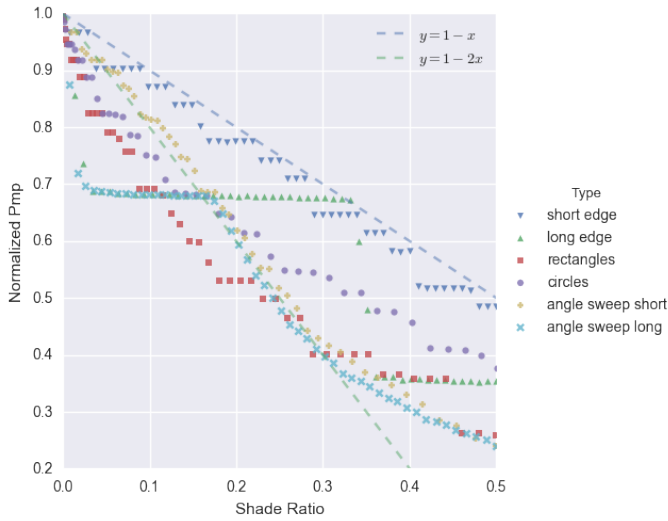


Fig. 8. Plot of normalized power versus shade ratio when utilizing the full cell-level model. The data is labeled by the six shade geometry families.

In Fig. 8, the impact of specific geometry on the performance of the system is clearly observable. At around 10% shade, the system can experience a reduction in power output in the range of 10–30%, simply due to the different geometries.

In Fig. 9, many of the trends lines are significantly smoothed out by the averaging of irradiance on each module. As would be expected, the behavior of individual diode-protected cell substrings is completely missing. Particularly striking is the change in modeled system response to shade set #2 (horizontal shade increasing from the bottom) which exactly follows the $y = 1 - x$ line in the module-level model. This occurs because the model-level model does not capture any mismatch conditions with this geometry; the short-circuit current is reduced on the shaded string in exact proportion to the amount of shade. In reality, a small amount of shade cutting across the diode-protected substrings causes a significant amount of mismatch, such that the entire power of the string is removed after just one or two rows of cells are shaded, as seen in Fig. 8.

Fig. 10 summarizes the differences between the cell-level approach and the module-level approach. This analysis illustrates that module-level modeling over-predicts power for most geometries modeled in this study. However, as seen in the “short edge” data, model-level modeling can also under-predict power in some circumstances. A likely explanation for this is that the bypass diodes tend to be quite helpful in mitigating power loss when the shade is correctly aligned with the substring geometry. The cell-level model captures this behavior, while the module-level model does not.

VI. CONCLUSION

The necessity of correctly modeling sub-module behavior when executing partial shade mismatch studies has been shown. In the set of shade geometries evaluated for this study,

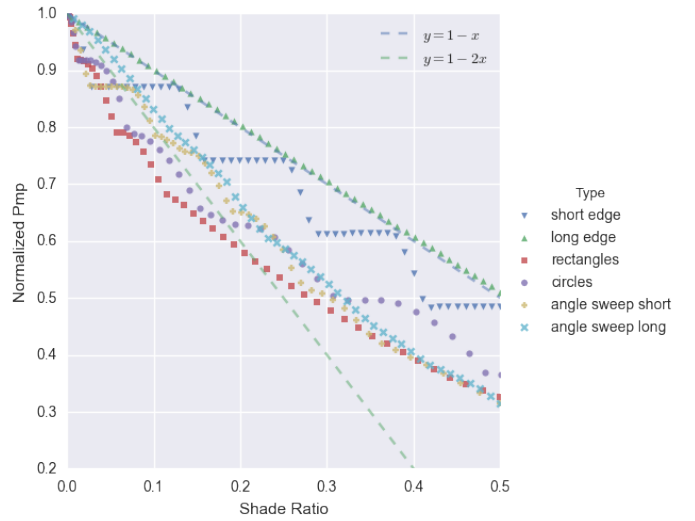


Fig. 9. Plot of normalized power versus shade ratio when utilizing a module-level, average irradiance model. Note the smoothing out of many of the trend lines and the significantly different behavior of the long edge trend (set #2).

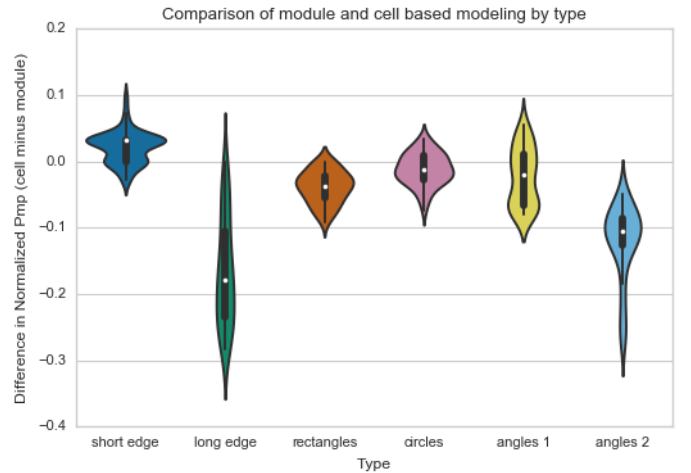


Fig. 10. Comparison of cell-level and module-level modeling with PVMismatch. The y-axis is difference in normalized P_{mp} . A negative value means the cell-level model predicts less power.

the module-level model tended to over-predict system power relative to the cell-level model. However, certain geometries showed the opposite effect. In addition to inaccurately estimating total system power, the operating voltage point of the system is incorrectly estimated by the module-level model.

An open source software solution has been presented to facilitate analysis of partial shade conditions. The techniques and tools presented here may be used by other researchers to facilitate further research on the topic of partially shaded PV arrays. For example, the authors have previously used the techniques presented in this paper to facilitate the generation of large-scale, detailed datasets of PV array performance under various shade conditions for performing feature extraction using machine learning techniques [12].

ACKNOWLEDGMENT

This work is supported by the Grid Integration Systems and Mobility (GISMO) group at SLAC National Accelerator Laboratory, the U.S. Department of Energy SunShot Initiative, and SunPower Corporation. The development of the PVMismatch software would not have been possible without the patient guidance of Akira Terao.

REFERENCES

- [1] G. W. Pennington, P. Saxton, S. Neidich, S. Taheri, F. Nasim, and J. Folkman, "Guidelines for California's Solar Electric Incentive Programs, Fifth Edition," California Energy Commission, Tech. Rep., 2013.
- [2] S. Malathy and R. Ramaprabha, "Comprehensive analysis on the role of array size and configuration on energy yield of photovoltaic systems under shaded conditions," *Renewable and Sustainable Energy Reviews*, vol. 49, pp. 672–679, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.rser.2015.04.165>
- [3] E. Karatepe, M. Boztepe, and M. Çolak, "Development of a suitable model for characterizing photovoltaic arrays with shaded solar cells," *Solar Energy*, vol. 81, no. 8, pp. 977–992, 2007.
- [4] G. Velasco-Quesada, F. Guinjoan-Gispert, R. Piqué-López, M. Román-Lumbreras, and A. Conesa-Roca, "Electrical PV array reconfiguration strategy for energy extraction improvement in grid-connected PV systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4319–4331, 2009.
- [5] N. Belhaouas, M.-S. A. Cheikh, P. Agathoklis, M.-R. Oularbi, B. Amrouche, K. Sedraoui, and N. Djilali, "PV array power output maximization under partial shading using new shifted PV array arrangements," *Applied Energy*, vol. 187, pp. 326–337, 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0306261916316233>
- [6] C. A. Ramos-Paja, J. D. Bastidas, A. J. Saavedra-Montes, F. Guinjoan-Gispert, and M. Goetz, "Mathematical model of total cross-tied photovoltaic arrays in mismatching conditions," *2012 IEEE 4th Colombian Workshop on Circuits and Systems, CWCAS 2012 - Conference Proceedings*, 2012.
- [7] P. Srinivasa Rao, G. Saravana Ilango, and C. Nagamani, "Maximum power from PV arrays using a fixed configuration under different shading conditions," *IEEE Journal of Photovoltaics*, vol. 4, no. 2, pp. 679–686, 2014.
- [8] J. Nelson, *The Physics of Solar Cells*. Imperial College Press, 2003.
- [9] C. T. Sah, R. N. Noyce, and W. Shockley, "Carrier Generation and Recombination in P-N Junctions and P-N Junction Characteristics," *Proceedings of the IRE*, vol. 45, no. 9, pp. 1228–1243, 1957.
- [10] W. De Soto, S. A. Klein, and W. A. Beckman, "Improvement and validation of a model for photovoltaic array performance," *Solar Energy*, vol. 80, no. 1, pp. 78–88, 2006.
- [11] J. W. Bishop, "Computer simulation of the effects of electrical mismatches in photovoltaic cell interconnection circuits," *Solar Cells*, vol. 25, no. 1, pp. 73–89, 1988.
- [12] B. Meyers, M. Mikofski, and M. Anderson, "A Fast Parameterized Model for Predicting PV System Performance under Partial Shade Conditions," *2016 IEEE 43rd Photovoltaic Specialist Conference, PVSC 2016*, no. 1, pp. 3173–3178, 2016.
- [13] S. Gillies, "Shapely 1.2 and 1.3 documentation," 2013. [Online]. Available: <http://toblerity.org/shapely/index.html>