Journal of Computational Science 75 (2024) 102199

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Efficient computation of the photovoltaic single-diode model curve by means of a piecewise linear self-adaptive representation

F. Javier Toledo [a,*], Vicente Galiano [b], Victoria Herranz [a], Jose M. Blanes [c]

[a] Center of Operations Research, Miguel Hernández University of Elche, Spain
[b] Computers Engineering Department, Miguel Hernández University of Elche, Spain
[c] Industrial Electronics Group, Miguel Hernández University of Elche, Spain

ARTICLE INFO

Keywords:
Piecewise linear interpolation
Photovoltaic single-diode model
Data point selection
Mean slope point
Graphical representation of functions
Data reduction

ABSTRACT

The current–voltage curve (I–V curve) associated to the photovoltaic (PV) single-diode model (SDM) is an important tool to analyze the behavior of a PV panel, nevertheless, obtaining it is not easy due to the implicit nature of the SDM equation that requires a lot of computation to solve it accurately. In this paper we provide a simple, accurate and almost instantaneous method to obtain the I–V curve which can be easily programmed, for example, in a microcontroller. The main tool is a recent parametrization of the SDM I–V curve which allows to compute the I–V points explicitly when the slope of the curve is known. Then, an iterative sequence of points based in the mean slope between the points of the previous step is constructed. The new methodology is compared with the most common method and the superiority of our proposal is demonstrated with a large repository of curves. Moreover, using the distribution of points obtained with the new methodology, it is possible to represent, with high precision and speed, other curves such as the power and the curvature functions providing a deeper information of the SDM.

1. Introduction

The representation of a data set or the graph of a function provides important information about its behavior, allowing to find out analytic-geometric properties such as monotonicity, convexity, asymptotic behavior, and, even detect singularities that would be difficult to notice with basic data analytics. For example, a graphical representation of the I–V curve is essential for understanding the performance of the PV module and optimizing its operation. This is why the first thing that is done when one has a set of data or a function is to represent it graphically. In the particular case of graphing functions, a piecewise linear approximation is normally performed from a selection of points in an interval. The simplest approach is to take a uniform partition of the interval, compute the images of the function at each element of the partition, and perform a linear interpolation with the corresponding computed points of the function, providing a piecewise linear approximation whose quality will depend on the size of the interval partition. If the function does not have a very complex expression, the calculation of the points can be more or less fast and precise and the computational cost will not vary much depending on the number of points, but if the expression of the function is complex and it is required a numerical algorithm for the calculation of the points, which happens, for example, if the function is given implicitly, then it is important to reduce the number of calculations as much as possible, and a way to implement this is with a good selection of the points to be calculated.

The selection of points providing efficient piecewise approximations of a curve, is a mathematical problem which has been thoroughly studied in the field of digital image processing [1–5]. It has been shown that some strategies based on the curvature of the curve [6,7], are capable to select a reduced number of points efficiently distributed on the curve that allows to construct a piecewise linear interpolation providing an approximation with an error below a previously fixed tolerance. Nevertheless, it is difficult to find in the literature generic algorithms that are easily implementable in practice and, then, the ad-hoc construction of a distribution for each curve is practically necessary. This is precisely the case of the I–V curve of the PV SDM, in which the curve is implicitly given by an equation whose solutions require numerical calculus. The most usual way to represent the SDM I–V curve is to compute a set of points between the short-circuit point (corresponding to zero voltage) and the open circuit-point (corresponding to zero current) distributed uniformly with respect to the voltage. It is very well-known that the SDM equation can be rewritten with the Lambert W function giving rise to explicit expressions of the current in terms of the voltage and vice versa. Although the Lambert W function also requires numerical calculus, there exist very fast and accurate

* Corresponding author.
E-mail addresses: javier.toledo@umh.es (F.J. Toledo), vgaliano@umh.es (V. Galiano), mavi.herranz@umh.es (V. Herranz), jmblanes@umh.es (J.M. Blanes).

https://doi.org/10.1016/j.jocs.2023.102199
Received 19 September 2023; Received in revised form 15 November 2023; Accepted 6 December 2023
Available online 7 December 2023

algorithms to compute its images [8–11]. However, the shape of the I–V curve with two clearly differentiated parts, one linear and the other exponential, [12,13], means that a uniform distribution of points requires a large number of points for the piecewise linear interpolation to provide an accurate approximation and, furthermore, so that the segments of the representation are not appreciable to the naked eye.

In this work we will provide a distribution of points on the SDM I–V curve that has the following advantages: (i) the points are easily calculable, in fact they will be obtained explicitly except for the extreme ones, (ii) the points are efficiently distributed, in fact they will automatically adapt to the shape of the curve and will provide an efficient piecewise linear approximation. The key tool to do this will be a recent parametrization of the SDM [14] which allows the explicit calculation of the unique mean slope point between two arbitrary points of the I–V curve. A detailed algorithm that has the virtue of supplying the points distribution in order will be described. We will compare our distribution with the uniform one for different PV modules to demonstrate the advantages of our proposal. Moreover, we will use the obtained distribution to represent other curves associated to the SDM which will be previously parametrized.

2. Preliminaries

2.1. The single-diode model

The single-diode model (SDM) equation associated to a solar panel with n_s cells in series is given by

$$I = I_{ph} - I_{sat}\left(e^{\frac{V+IR_s}{a}} - 1\right) - \frac{V + IR_s}{R_{sh}} \tag{1}$$

where I is the panel current measured in Amperes, V is the panel voltage measured in Volts, I_{ph} is the panel photocurrent in Amperes, I_{sat} is the panel diode saturation current in Amperes, R_s is the panel series resistance in Ohms and, R_{sh} is the panel shunt resistance in Ohms. On the other hand, $a = n_s n V_T$ where n is the ideality factor and $V_T = \frac{k}{q}T$ is the so-called thermal voltage, being T the temperature in Kelvin degrees, $k = 1.3806488 \times 10^{-23}$ J/K the Boltzmann's constant and, $q = 1.60217653 \times 10^{-19}$ C the electron charge.

The solutions of Eq. (1) generate the so-called I–V curve of the SDM. The point of the I–V curve corresponding to voltage zero is called short-circuit point, it is denoted by $SCP = (0, I_{sc})$. The point corresponding to current zero is called the open-circuit point and it is denoted by $OCP = (V_{oc}, 0)$.

2.1.1. The SDM in terms of the lambert W function

It is well-known that the current I and the voltage V of the SDM can be alternatively expressed through the Lambert W function as [15]

$$\begin{aligned} I =& \frac{1}{R_{sh}+R_s}\left(R_{sh}\left(I_{ph}+I_{sat}\right) - V\right) \\ & - \frac{a}{R_s} W_0\left(\frac{I_{sat}R_{sh}R_s}{a\left(R_{sh}+R_s\right)} \exp\left(\frac{R_{sh}\left(R_s\left(I_{ph}+I_{sat}\right)+V\right)}{a\left(R_{sh}+R_s\right)}\right)\right) \end{aligned} \tag{2}$$

$$\begin{aligned} V =& R_{sh}\left(I_{ph}+I_{sat}\right) - \left(R_{sh}+R_s\right) I \\ & - aW_0\left(\frac{I_{sat}R_{sh}}{a} \exp\left(\frac{R_{sh}\left(I_{ph}+I_{sat}-I\right)}{a}\right)\right) \end{aligned} \tag{3}$$

where W_0 is the positive branch of the real Lambert W function, that is, the inverse of the function $f(x) = xe^x$ in the interval $[-1, +\infty[$ [16]. In [8] a fast and accurate algorithm was proposed to compute the Lambert W function that is capable of evaluating W_0 in very large arguments. Applying this algorithm to (2) and (3) we straightforwardly obtain, as particular cases, the values of the short-circuit current and the open-circuit voltage which are usually the extreme points of the I–V curve to be computed and plotted.

2.1.2. A parametrization of the SDM

In [14] was proved that any point of the I–V curve can be expressed as

$$\begin{cases} \mathcal{V}(x) = \frac{a(R_s+R_{sh})}{R_{sh}}\left(\ln x - \frac{R_s}{a}\left(\frac{R_{sh}(I_{ph}+I_{sat})}{R_s+R_{sh}} - x\right) - \ln\left(\frac{I_{sat}R_{sh}}{R_{sh}+R_s}\right)\right) \\ \mathcal{I}(x) = \frac{R_{sh}(I_{ph}+I_{sat})-\mathcal{V}(x)}{R_s+R_{sh}} - x \end{cases} \tag{4}$$

for certain parameter $x > 0$. This parameter is indeed the vertical distance from the I–V curve to its oblique asymptote [12,14]

$$x = \frac{I_{sat}R_{sh}}{R_{sh}+R_s} e^{\frac{V+IR_s}{a}}$$

In [14] was also proved that, if the slope I' of the I–V curve at a point is known, the parameter x corresponding to this point is given by

$$x = -\frac{a}{R_{sh}+R_s} \frac{I'\left(R_{sh}+R_s\right)+1}{1+I'R_s} \tag{5}$$

Expressions (4) and (5) will be essential in the development of this work.

3. An efficient self-adaptive representation of the SDM I-V curve

It is well known [12] that Eq. (1) defines I as a function of V which is indefinitely differentiable over the whole real line. The graph of I, that is the I–V curve, is parametrized through (4) as $(\mathcal{V}(x), \mathcal{I}(x))$. Thus, to represent the I–V curve, it will be sufficient to take a set of values for the parameter x and draw the corresponding set of points $(\mathcal{V}(x), \mathcal{I}(x))$ connected by a line segment, giving rise to a piecewise linear function that will interpolate the I–V curve with more or less precision depending on the number of points and their distribution. It is evident that what one wants is to have the best representation with the least amount of points possible, and for this it is essential to achieve an efficient distribution of the points to be drawn, which is generally not a uniform distribution as we will see later. Next we describe how to obtain a distribution of points that, as we will see, is, among other things, computationally efficient.

3.1. A mean slope points sequence

As we have seen in the preliminary section, if we know the slope of the I–V curve at a point, automatically we get its corresponding parameter with (5) and, then, the coordinates of the point with (4). Then, if we take two arbitrary points of the I–V curve, (V_1, I_1) and (V_2, I_2), the mean slope between them is $I'_{12} = \frac{I_2 - I_1}{V_2 - V_1}$ and (5) provides the parameter

$$x_{12} = -\frac{a}{R_{sh}+R_s} \frac{I'_{12}\left(R_{sh}+R_s\right)+1}{1+I'_{12}R_s}$$

which gives the mean slope point $(V_{12}, I_{12}) = (\mathcal{V}(x_{12}), \mathcal{I}(x_{12}))$ computed with (4).

Taking this idea in mind, we can start with the short-circuit and the open-circuit points, compute the mean slope point, and then continue computing the mean slope points between the pairs of consecutive points available in the previous step. This strategy obtains a self-adaptive distribution of points on the I–V curve in the sense that, the greater the curvature of the curve, the greater the accumulation of points. The pseudocode for this strategy is given in algorithm 1. This algorithm provides $2^N + 1$ points of the I–V curve and it has the virtue of directly providing the position of the points that orders them increasingly with respect to the voltage.

In Fig. 1 it is graphically illustrated the operation of Algorithm 1 with N = 2, which generates 5 points, for the I–V curve corresponding to Module 1 of Table 1.

Algorithm 1: Pseudocode MSP sequence

Input: Input $I_{ph}, I_{sat}, a, R_{sh}, R_s, N$ // $a = n_s n V_T$, $2^N + 1$ is the number of points

Output: (V, I)

$$I_{sc} = \frac{R_{sh}\left(I_{ph}+I_{sat}\right)}{R_{sh}+R_s} - \frac{a}{R_s} W_0\left(\frac{I_{sat}R_{sh}R_s}{a\left(R_{sh}+R_s\right)}\exp\left(\frac{R_{sh}R_s\left(I_{ph}+I_{sat}\right)}{a\left(R_{sh}+R_s\right)}\right)\right)$$

$$V_{oc} = R_{sh}\left(I_{ph}+I_{sat}\right) - aW_0\left(\frac{I_{sat}R_{sh}}{a}\exp\left(\frac{R_{sh}\left(I_{ph}+I_{sat}\right)}{a}\right)\right)$$

$V_0 = 0,\ I_0 = I_{sc},\ V_{2^N} = V_{oc},\ I_{2^N} = 0$

for $j = 1$ to N **do**

  for $k = 1$ to 2^{j-1} **do**

$$I'_{(2k-1)2^{N-j}} = \frac{I_{k2^{N-j+1}} - I_{(k-1)2^{N-j+1}}}{V_{k2^{N-j+1}} - V_{(k-1)2^{N-j+1}}}$$

$$x_{(2k-1)2^{N-j}} = -\frac{a}{R_{sh}+R_s}\,\frac{I'_{(2k-1)2^{N-j}}\left(R_{sh}+R_s\right)+1}{1+I'_{(2k-1)2^{N-j}}R_s}$$

$$\left(V_{(2k-1)2^{N-j}}, I_{(2k-1)2^{N-j}}\right) = \left(\mathcal{V}\left(x_{(2k-1)2^{N-j}}\right), \mathcal{I}\left(x_{(2k-1)2^{N-j}}\right)\right)$$

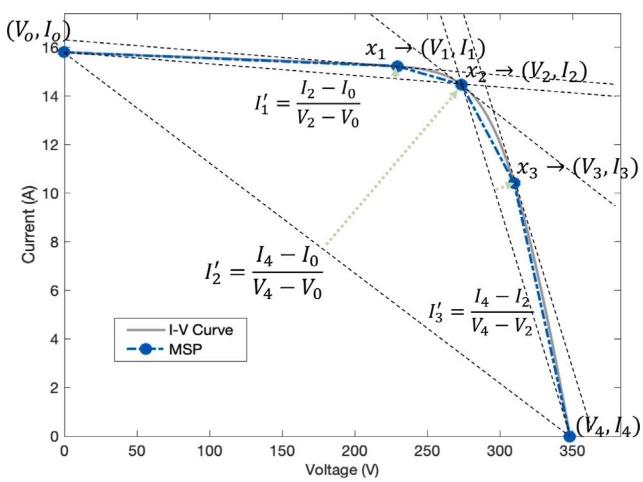


Fig. 1. Graphical illustration of Algorithm 1. Data corresponding to Module 1 of Table 1.

4. Computational experiments

In this section we are going to prove the advantages of our new methodology, let us call Mean Slope Point (MSP) method, compared to one of the options most used by researchers due to its simplicity and accuracy, let us call usual (USU) method.

The USU method consists of representing the SDM model I–V curve between the short-circuit point $\left(0, I_{sc}\right)$ and the open-circuit point $\left(V_{oc}, 0\right)$ taking a uniform voltage distribution in the interval $\left[0, V_{oc}\right]$, and calculating the corresponding currents with the explicit formula (2) using the MATLAB's built-in *lambertw* function. Then, a piecewise linear function is traced across the computed points which, in fact, provides an approximation of the exact I–V curve. Our new proposal, the MSP method, is based in the Mean Slope Point sequence constructed in Algorithm 1, then, the explicitly obtained points are used to trace the corresponding piecewise linear function. As commented previously, we will compute I_{sc} and V_{oc} with the formulas (2) and (3) but using the algorithm proposed in [8] to compute the Lambert W function images, although the *lambertw* function of MATLAB or any other numerical method could be used to compute I_{sc} and V_{oc} without affecting significantly the effectiveness of the MSP method.

To estimate the precision of the USU and MSP methods, for a given voltage V_i, we will take as benchmark the corresponding current value obtained with the formula (2) computed with MATLAB and its function *lambertw*, this current will be denoted by I_i^b (the superindex b comes from benchmark). In this way, the discrepancy between the reference I–V curve and an approximation (in our approach a piecewise linear function) of it will be measured with the root mean square error (RMSE) given by the formula

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(I_i^b - I_i^a\right)^2}$$

where I_i^a (the superindex a comes from approximated) is the current value corresponding to the approximated I–V curve for the voltage V_i. The number of points (distributed uniformly with respect to the voltage in $\left[0, V_{oc}\right]$) taken to compute accurately the RMSE will be $M = 2^{12}+1 = 4097$.

Table 1
Parameters of the 9 theoretical SDM I–V curves.

Module	I_{ph}(A)	I_s (A)	a	R_s (Ω)	R_{sh} (Ω)
1	15.880	7.44E−10	14.670	2.0400	425.2
2	1.032	2.51E−06	1.300	1.2390	744.7
3	3.654	4.00E−21	0.516	2.6900	2329.0
4	0.578	1.34E−10	0.012	0.0127	612.0
5	0.761	3.11E−07	0.039	0.0370	52.9
6	4.802	4.02E−07	0.037	0.5906	1167.0
7	4.942	1.84E−07	1.222	0.2460	387.0
8	2.501	1.13E−07	1.228	0.2283	442.2
9	0.991	5.47E−07	1.398	0.0386	844.4

Remark 1. Since the USU method also makes use of the *lambertw* function, the difference $I_i^b - I_i^a$ in the selected points of the interval $\left[0, V_{oc}\right]$ will be zero, which will entail an advantage of the USU method front the MSP one, nevertheless, we will see that despite this fact, the MSP method is always more accurate than the USU method except in one extreme case with a large number of points in the interval, where the USU method is insignificantly more accurate than the MSP method although significantly slower. But, any case, we must point out that this advantage of the USU method in this extreme case is actually misleading, because machine errors in the calculation of the explicit values obtained with the MSP method are inherent to the procedure and, in this case, they cannot offset the above null differences of the USU method.

In order to obtain the computational time and to make it reliable, we will repeat each experiment 1000 times and take the average time.

The experimentation has been carried out in two settings, a theoretical one with 9 I–V curves used in previous papers [8], that cover the widest range of possible situations and the shapes of the I–V curves are significantly different, and a real context where 1000 curves have been randomly selected from the National Renewable Energy Laboratory (NREL) dataset of the USA [17], specifically from the mSi460A8 panel at the Cocoa location.

4.1. Computational experiments with theoretical I–V curves

The SDM parameters corresponding to the 9 theoretical I–V curves are provided in Table 1.

Fig. 2 illustrates the piecewise linear interpolation functions approximating the I–V curve corresponding to Module 1 (with parameters in Table 1). It is interesting to observe how the points are distributed with the two methods, especially with the MSP method, which accumulates more points in the area of greatest curvature, which is where it is most needed. With very few points it is evident that the MSP method manages to capture the shape of the I–V curve with much better fidelity.

Table 2 provides the RMSE obtained with the USU and the MSP methods for each one of the theoretical I–V curves corresponding of Table 1. The RMSE has been computed for piecewise linear functions computed with $2^k + 1$ points, $k = 2, \ldots, 10$, included the short-circuit

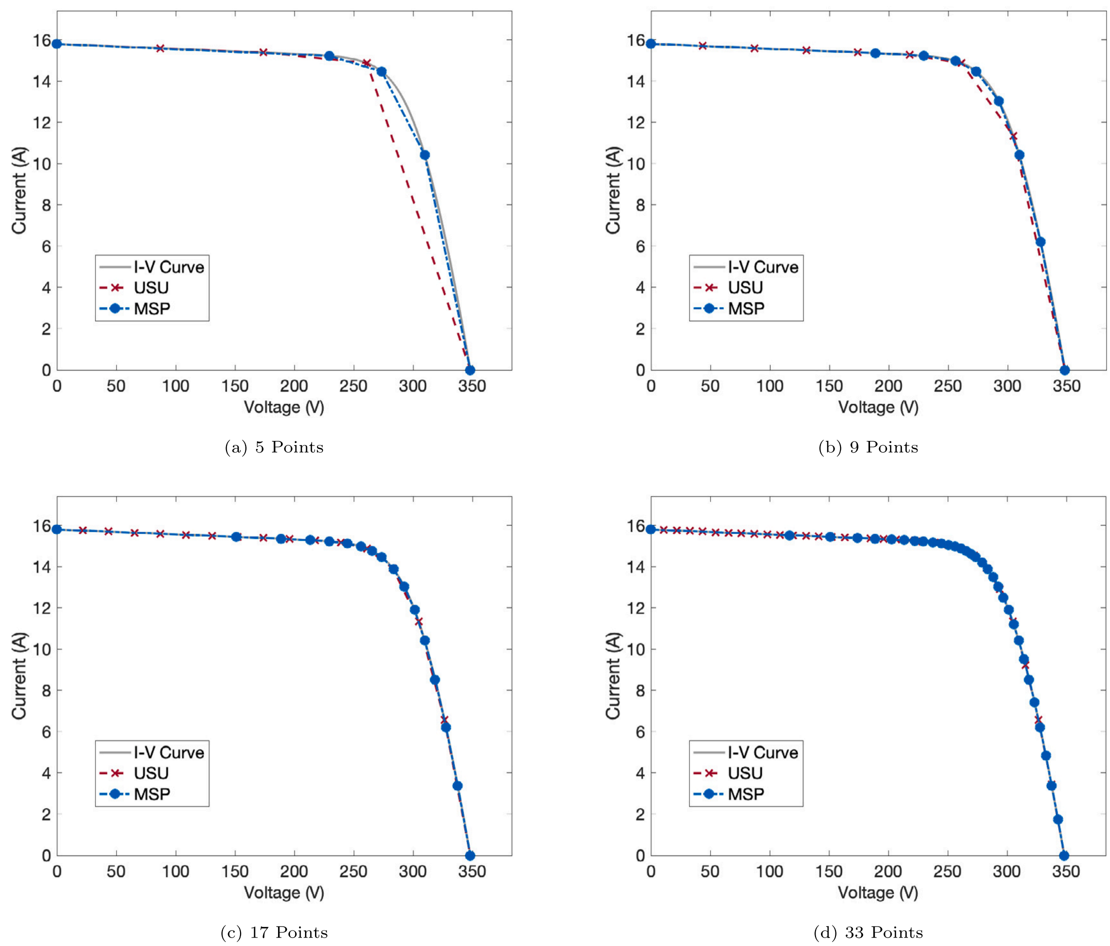


Fig. 2. Piecewise linear approximations of the I–V curve corresponding to Module 1 of Table 1 with MSP and USU methods with four different amount of points.

and the open circuit points. As can be observed, the RMSE is always approximately one order of magnitude smaller with the MSP method, except in the I–V curve 6 in which both methods have errors of the same order of magnitude. In this case, although the MSP method is still slightly better than the USU method for all the sets of points selected for the piecewise linear representations, there is one exception for 1025 points where the USU method break the norm in a negligible amount, but with a logical explanation because, in this case, the I–V curve is practically a line and the points are almost all computed with the *lambertw* function taking deceptive advantage the USU method as commented in Remark 1.

Table 3 shows the computational times needed to obtain each set of points for the piecewise linear functions.

It is evident from Table 1 that the MSP method is much faster than the USU method, but the difference is much clearer if we calculate the percentage of time saved with the MSP method compared to the USU method, indeed, in the worst case, time saving with MSP method reaches 82.6% although is capable of reaching a time saving of 96.9%. Another way of visualizing the difference between the computation times can be seen in Fig. 3, where the time of each method is represented as a function of the number of points used. We only show the graph of the I–V curve corresponding to Module 1 since the graph of the remaining I–V curves is similar.

4.2. Computational experiments with NREL I-V curves

In this subsection we want to validate the superiority of the MSP method with I–V curves measured in real conditions. As previously stated, 1000 curves have been randomly taken from the NREL database [17]. The parameters corresponding to the NREL curves have been computed with the Two-Step Linear Least-Squares (TSLLS) method [13] (online accessible at https://pvmodel.umh.es/) and the average of the RMSE and the computational times measured on the 1000 curves are shown in Table 4.

As can be seen in Table 4, similarly to what happens with the theoretical modules analyzed in the previous subsection, the RMSE and the computational time with the NREL curves are approximately of one lower magnitude order with the MSP method than with the USU method, which confirms the superiority of the MSP method, both for its precision as well as speed, with a large amount of real data.

Remark 2. Apart from linear interpolation, there are many other forms of interpolation of points that are known to be more accurate than linear interpolation, for example, interpolation by cubic splines. Although it is not included in this work, we have verified that using few points, linear interpolation with the MSP distribution remains the most accurate, but as the number of points increases, cubic spline interpolations improve the RMSE of the linear ones, ending up being the best one which uses the distribution obtained with MSP distribution. However, the computing time used for the graph representation of the I–V curve with spline interpolations is significantly higher than with linear interpolations and, in addition, the graphical representation of the I–V curve with few points is distorted because with splines, regions of current growth occur that do not correspond to the behavior of a solar panel. In any case, it is interesting the use of cubic spline interpolation together with MSP distribution of points because this combination is capable of achieving a very small RMSE and this could be interesting in other applications of photovoltaic modeling.

Table 2
RMSE with the I–V curves of Table 1 and the piecewise linear functions obtained with different sets of points.

		5	9	17	33	65	129	257	513	1025
Module 1	USU	1.19E−01	2.66E−02	6.81E−03	1.71E−03	4.27E−04	1.07E−04	2.66E−05	6.55E−06	1.54E−06
	MSP	2.00E−02	4.89E−03	1.22E−03	3.04E−04	7.58E−05	1.89E−05	4.62E−06	1.09E−06	3.08E−07
Module 2	USU	2.74E−04	6.69E−05	1.68E−05	4.19E−06	1.05E−06	2.62E−07	6.52E−08	1.61E−08	3.77E−09
	MSP	7.50E−05	1.84E−05	4.59E−06	1.15E−06	2.86E−07	7.13E−08	1.76E−08	4.21E−09	1.04E−09
Module 3	USU	5.10E−04	1.65E−04	4.00E−05	9.99E−06	2.51E−06	6.27E−07	1.56E−07	3.85E−08	9.04E−09
	MSP	1.51E−04	3.69E−05	9.18E−06	2.29E−06	5.72E−07	1.43E−07	3.52E−08	8.53E−09	2.32E−09
Module 4	USU	5.50E−06	1.78E−06	4.48E−07	1.12E−07	2.81E−08	7.02E−09	1.75E−09	4.31E−10	1.01E−10
	MSP	7.04E−07	1.73E−07	4.30E−08	1.07E−08	2.68E−09	6.63E−10	1.60E−10	3.77E−11	1.70E−11
Module 5	USU	9.83E−06	2.53E−06	6.34E−07	1.59E−07	3.97E−08	9.91E−09	2.47E−09	6.08E−10	1.43E−10
	MSP	2.00E−06	4.92E−07	1.22E−07	3.06E−08	7.63E−09	1.90E−09	4.66E−10	1.10E−10	2.95E−11
Module 6	USU	2.90E−09	7.26E−10	1.82E−10	4.54E−11	1.13E−11	2.83E−12	7.06E−13	1.74E−13	4.09E−14
	MSP	2.87E−09	7.17E−10	1.79E−10	4.48E−11	1.12E−11	2.80E−12	6.96E−13	1.72E−13	4.09E−14
Module 7	USU	2.44E−03	6.07E−04	1.52E−04	3.80E−05	9.50E−06	2.37E−06	5.91E−07	1.46E−07	3.42E−08
	MSP	4.54E−04	1.12E−04	2.78E−05	6.93E−06	1.73E−06	4.30E−07	1.06E−07	2.49E−08	6.92E−09
Module 8	USU	1.55E−03	4.65E−04	1.18E−04	2.96E−05	7.42E−06	1.85E−06	4.62E−07	1.14E−07	2.67E−08
	MSP	2.48E−04	6.10E−05	1.52E−05	3.79E−06	9.46E−07	2.35E−07	5.72E−08	1.34E−08	4.75E−09
Module 9	USU	6.30E−04	2.16E−04	6.01E−05	1.55E−05	3.90E−06	9.75E−07	2.43E−07	5.98E−08	1.41E−08
	MSP	1.04E−04	2.57E−05	6.40E−06	1.60E−06	3.99E−07	9.89E−08	2.40E−08	5.75E−09	2.57E−09

Table 3
Computational times to obtain the sets of points for the piecewise linear functions approximating the I–V curves of Table 1.

		5	9	17	33	65	129	257	513	1025
Module 1	USU	2.37E−05	2.12E−05	2.18E−05	2.62E−05	4.02E−05	3.87E−05	6.02E−05	7.54E−05	1.15E−04
	MSP	1.83E−06	1.41E−06	1.66E−06	1.94E−06	2.75E−06	4.32E−06	7.50E−06	1.11E−05	1.56E−05
Module 2	USU	6.37E−05	2.27E−05	2.25E−05	2.57E−05	3.02E−05	3.86E−05	5.99E−05	7.30E−05	1.12E−04
	MSP	3.19E−06	1.40E−06	1.61E−06	1.98E−06	2.64E−06	4.39E−06	7.48E−06	1.04E−05	1.50E−05
Module 3	USU	2.33E−04	2.46E−05	2.65E−05	3.01E−05	3.69E−05	5.05E−05	7.96E−05	1.11E−04	1.39E−04
	MSP	1.89E−06	1.46E−06	1.66E−06	1.92E−06	2.82E−06	4.18E−06	7.03E−06	1.07E−05	1.55E−05
Module 4	USU	2.43E−05	2.10E−05	2.17E−05	2.62E−05	2.98E−05	3.83E−05	5.82E−05	7.15E−05	1.06E−04
	MSP	1.45E−06	1.49E−06	1.59E−06	1.90E−06	2.91E−06	4.27E−06	7.50E−06	1.17E−05	1.78E−05
Module 5	USU	2.94E−05	2.08E−05	2.21E−05	2.50E−05	2.97E−05	3.83E−05	5.92E−05	7.25E−05	1.10E−04
	MSP	1.55E−06	1.47E−06	1.70E−06	1.99E−06	2.61E−06	4.09E−06	7.51E−06	1.11E−05	1.55E−05
Module 6	USU	7.35E−05	3.01E−05	3.23E−05	3.76E−05	4.73E−05	6.38E−05	6.95E−05	9.62E−05	1.49E−04
	MSP	1.48E−06	1.41E−06	1.62E−06	1.90E−06	2.62E−06	4.33E−06	7.61E−06	1.14E−05	1.54E−05
Module 7	USU	2.91E−05	2.09E−05	2.23E−05	2.45E−05	2.96E−05	3.83E−05	6.21E−05	7.16E−05	1.32E−04
	MSP	1.48E−06	1.51E−06	1.60E−06	2.03E−06	2.70E−06	4.27E−06	7.45E−06	1.05E−05	1.50E−05
Module 8	USU	9.04E−05	2.10E−05	2.20E−05	2.57E−05	3.07E−05	3.89E−05	6.03E−05	6.99E−05	1.04E−04
	MSP	1.71E−06	1.52E−06	1.66E−06	2.05E−06	2.70E−06	4.26E−06	7.40E−06	1.16E−05	1.64E−05
Module 9	USU	4.67E−05	2.42E−05	2.55E−05	2.87E−05	3.29E−05	3.97E−05	5.07E−05	6.62E−05	9.45E−05
	MSP	1.47E−06	1.54E−06	1.70E−06	2.26E−06	2.57E−06	4.48E−06	7.90E−06	1.25E−05	1.60E−05

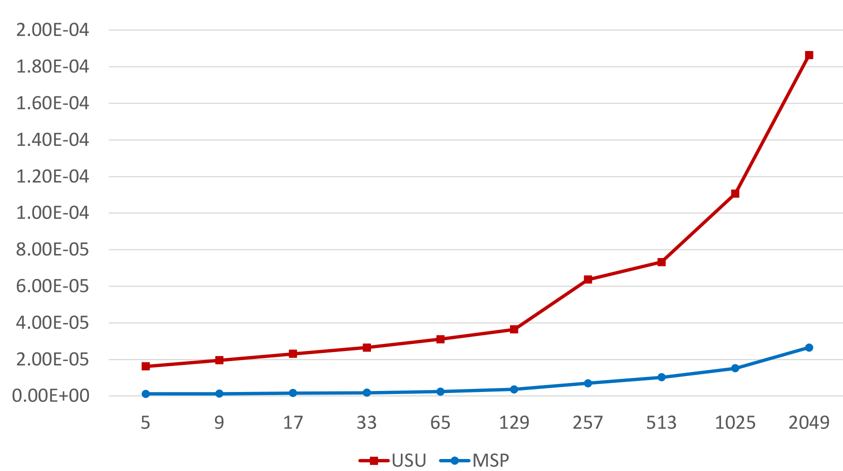


Fig. 3. Computational time (in seconds) behavior of USU and MSP methods in terms of the number of points used. Data corresponding to Module 1 of Table 1.

Table 4
Average of the RMSE and the computing time for different number of points in 1000 I–V curves from NREL.

		5	9	17	33	65	129	257	513	1025
RMSE	USU	2.36E−01	6.52E−02	1.67E−02	4.20E−03	1.05E−03	2.63E−04	6.58E−05	1.64E−05	3.98E−06
	MSP	4.39E−02	1.08E−02	2.69E−03	6.72E−04	1.68E−04	4.20E−05	1.05E−05	2.62E−06	6.55E−07
Time	USU	2.36E−05	2.57E−05	2.67E−05	2.77E−05	3.29E−05	3.65E−05	8.94E−05	1.09E−04	1.52E−04
	MSP	2.10E−06	1.84E−06	1.84E−06	2.18E−06	2.92E−06	5.16E−06	1.09E−05	1.53E−05	2.18E−05

4.3. Graphical representation of other SDM curves

In this section we show how to represent other important SDM curves, namely, the power function, the first and the second derivatives of I, and the curvature of the I–V curve. We must point out that the usual graphic representation of these curves is cumbersome since it requires the previous calculation of the images of other functions, for example, to calculate the curvature it is necessary to previously calculate $I'(V)$ and $I''(V)$ for the selected set of voltages. However, our proposal is practically immediate, we will use a parametrization of these curves, with the same parameter x given in (4), together with the distribution of x obtained for the I–V curve. It must be said that, although it would be ideal to be able to obtain a sequence of mean slope points for each of the curves, we have not found a simple way to calculate them. Nevertheless, the distribution used contains intrinsic information about the curves and the result that is obtained with a moderate amount of points is highly satisfactory by its quality but above all for its simplicity.

Next we provide the parametrizations of other curves associated with the SDM.

- The power function P
  From the parametrization (4), one obtains straightforwardly the parametrization of the power function $P = VI$ as $(\mathcal{V}(x), \mathcal{P}(x))$ where $\mathcal{P}(x) = \mathcal{V}(x)\mathcal{I}(x)$.
- The first derivative function I'
  From the expression (5) of parameter x as a function of the slope I' of the I–V curve at the point (V, I), one obtains I' as a function of x as

$$I' = -\frac{a + x\left(R_{sh} + R_s\right)}{\left(a + xR_s\right)\left(R_{sh} + R_s\right)} = \mathcal{I}p1(x) \tag{6}$$

  It leads to the parametrization $(\mathcal{V}(x), \mathcal{I}p1(x))$ of the graph of the first derivative function I'.
- The second derivative function I''
  It is also easy to write the second derivative of I in terms of x [14]

$$I'' = -\frac{xR_{sh}^2 a}{\left(a + xR_s\right)^3\left(R_{sh} + R_s\right)^2} = \mathcal{I}p2(x) \tag{7}$$

  which gives the parametrization $(\mathcal{V}(x), \mathcal{I}p2(x))$ of the graph of the second derivative function I''.
- The curvature function k
  By using the expressions of I' and I'' in terms of x, we can directly obtain the curvature $k = \frac{|I''|}{\left(1+(I')^2\right)^{3/2}}$ as a function of x.
  A simplified expression of this function [14] is

$$k = \frac{\delta^2 x}{\left(\alpha + 2\beta x + \gamma x^2\right)^{3/2}} = \mathcal{K}(x)$$

  where

$$\alpha = 1 + \frac{1}{\left(R_{sh} + R_s\right)^2}, \ \beta = \frac{R_s}{a} + \frac{1}{a\left(R_{sh} + R_s\right)}$$
$$\gamma = \frac{1 + R_s^2}{a^2}, \ \delta = \frac{1}{a} - \frac{R_s}{a\left(R_{sh} + R_s\right)} \tag{8}$$

  Therefore, a parametrization of the graph of the curvature function is given by $(\mathcal{V}(x), \mathcal{K}(x))$.

The graph of the curves parametrized above associated to Module 1 can be visualized in Fig. 4. Only 65 points have been used for the graphic representations.

Just to quantitatively illustrate the goodness of the distribution obtained with MSP to represent the other curves related to the SDM, in Table 5 we show the RMSE obtained for the power function with the USU and MSP distributions. It can be seen, similarly as it happened with the I–V curve, that in almost all modules and for any amount of points used, the error obtained with MSP is an order of magnitude smaller than the one obtained with USU, which confirms what we said at the beginning of this section about the intrinsic information contained in the MSP point distribution.

5. Conclusions

In this work we obtain a distribution of points on the I–V curve of the SDM that has the following properties: (i) all the points, except the extreme ones, are calculated explicitly and, therefore, the computation time in comparison with numerical methods is almost negligible, (ii) the points are efficiently distributed since they give rise to a piecewise linear approximation that minimizes the RMSE, in addition, they are obtained automatically without user intervention. With this distribution, the number of points to be used can be considerably reduced and, therefore, cheapen the computational cost. The key tool used to achieve this has been a recent parametrization of the SDM [14] that allows the explicit calculation of the unique mean slope point between two arbitrary points on the I–V curve. A detailed algorithm has been described that has the virtue of ordering the points of the distribution in real time. We have compared our distribution with the uniform one for different PV modules that supports the advantages of our proposal. In addition, we have seen that the distribution obtained can be used to represent other parametrized curves associated with the SDM, such as the power curve. Finally, it could be interesting to obtain efficient piecewise linear approximations by interpolation, to use the idea of constructing a sequence of points based on the mean slope to other curves, particularly concave or convex curves, in which it is possible to easily calculate the mean slope point between two arbitrary points.

Declaration of competing interest

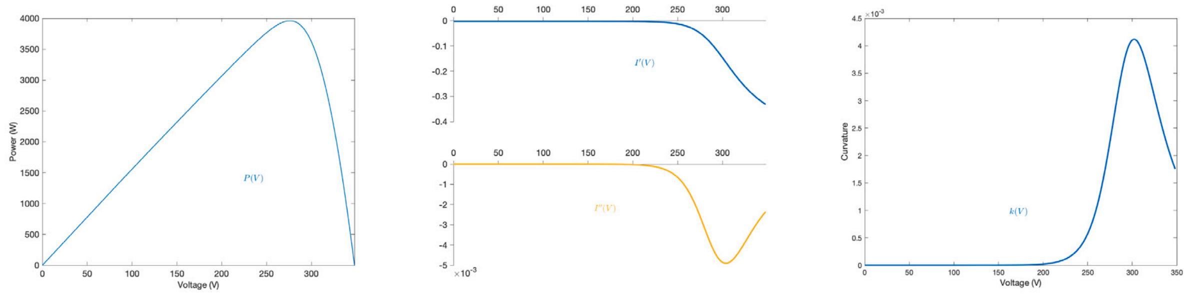
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Some date are available in the paper and the large repository is available on request.

Acknowledgments

Dr. F. Javier Toledo, Dr. V. Galiano, Dr. Victoria Herranz and Dr. José M. Blanes have received funding from grant TED2021-130025B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. Dr. F. Javier Toledo's work has received funding from the Ministerio de Ciencia e Innovación of Spain (PID2022-136399NB-C22), the Government of the Valencian Community, Spain (PROMETEO/2021/063) and the European Union



(a) Power function   (b) First and second derivative functions   (c) Curvature function

Fig. 4. Power, first and second derivative and curvature functions corresponding to Module 1 of Table 1.

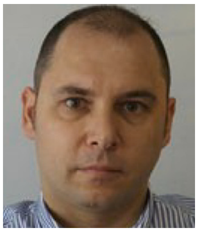
Table 5
RMSE with the power curves of Table 1 and the piecewise linear functions obtained with different sets of points.

		5	9	17	33	65	129	257	513	1025
Module 1	USU	5.44E+02	1.28E+02	3.25E+01	8.13E+00	2.03E+00	5.08E−01	1.27E−01	3.18E−02	7.93E−03
	MSP	9.90E+01	2.65E+01	8.80E+00	3.70E+00	1.63E+00	6.57E−01	2.28E−01	6.81E−02	1.83E−02
Module 2	USU	1.33E+00	3.36E−01	8.45E−02	2.11E−02	5.29E−03	1.32E−03	3.31E−04	8.26E−05	2.06E−05
	MSP	3.60E−01	8.95E−02	2.24E−02	5.64E−03	1.42E−03	3.55E−04	8.87E−05	2.22E−05	5.53E−06
Module 3	USU	2.51E+00	6.48E−01	1.58E−01	3.95E−02	9.90E−03	2.47E−03	6.19E−04	1.55E−04	3.86E−05
	MSP	2.35E+00	6.65E−01	1.72E−01	4.33E−02	1.09E−02	2.79E−03	7.53E−04	2.21E−04	6.71E−05
Module 4	USU	2.35E−02	7.91E−03	2.03E−03	5.11E−04	1.28E−04	3.20E−05	8.00E−06	2.00E−06	4.99E−07
	MSP	3.09E−03	7.61E−04	1.90E−04	4.74E−05	1.19E−05	2.98E−06	7.47E−07	1.88E−07	4.70E−08
Module 5	USU	4.49E−02	1.21E−02	3.06E−03	7.66E−04	1.92E−04	4.79E−05	1.20E−05	3.00E−06	7.48E−07
	MSP	9.20E−03	2.28E−03	5.71E−04	1.44E−04	3.63E−05	9.15E−06	2.29E−06	5.74E−07	1.44E−07
Module 6	USU	6.93E−03	1.73E−03	4.33E−04	1.08E−04	2.71E−05	6.77E−06	1.69E−06	4.23E−07	1.06E−07
	MSP	6.99E−03	1.75E−03	4.37E−04	1.09E−04	2.73E−05	6.84E−06	1.71E−06	4.27E−07	1.07E−07
Module 7	USU	1.11E+01	2.90E+00	7.29E−01	1.83E−01	4.57E−02	1.14E−02	2.85E−03	7.13E−04	1.78E−04
	MSP	2.10E+00	5.20E−01	1.30E−01	3.31E−02	8.46E−03	2.16E−03	5.47E−04	1.37E−04	3.44E−05
Module 8	USU	6.79E+00	2.12E+00	5.47E−01	1.38E−01	3.45E−02	8.62E−03	2.16E−03	5.39E−04	1.34E−04
	MSP	1.10E+00	2.74E−01	6.97E−02	1.83E−02	4.93E−03	1.32E−03	3.43E−04	8.71E−05	2.18E−05
Module 9	USU	2.73E+00	9.63E−01	2.72E−01	7.02E−02	1.77E−02	4.43E−03	1.11E−03	2.77E−04	6.92E−05
	MSP	4.55E−01	1.13E−01	2.88E−02	7.42E−03	1.93E−03	4.93E−04	1.25E−04	3.13E−05	7.79E−06

(ERDF, "A way to make Europe"). Dr. V. Galiano's work has received funding from the Valencian Ministry of Innovation, Universities, Science and Digital Society (Generalitat Valenciana), Spain under Grant CIAICO/2021/278 and from Grant PID2021-123627OB-C55 funded by MCIN/AEI/10.13039/501100011033 and, by "ERDF A way of making Europe".

References

[1] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Comput. Graph. Image Process. 1 (1972) 244–256.
[2] Pavlidis, Polygonal approximations by Newton's method, IEEE Trans. Comput. C-26 (8) (1977) 800–807, http://dx.doi.org/10.1109/TC.1977.1674918.
[3] C.M. Williams, An efficient algorithm for the piecewise linear approximation of planar curves, Comput. Graph. Image Process. 8 (2) (1978) 286–293, http://dx.doi.org/10.1016/0146-664X(78)90055-2.
[4] E. Camponogara, L.F. Nazari, Models and algorithms for optimal piecewise-linear function approximation, Math. Probl. Eng. 2015 (2015).
[5] L.R. Emmendorfer, G.P. Dimuro, A point interpolation algorithm resulting from weighted linear regression, J. Comput. Sci. 50 (2021) 101304, http://dx.doi.org/10.1016/j.jocs.2021.101304.
[6] M. Crampin, R. Guifo Guifo, G. Read, Linear approximation of curves with bounded curvature and a data reduction algorithm, Comput. Aided Des. 17 (6) (1985) 257–261, http://dx.doi.org/10.1016/0010-4485(85)90110-1.
[7] B. Hamann, J.-L. Chen, Data point selection for piecewise trilinear approximation, Comput. Aided Geom. Design 11 (5) (1994) 477–489, http://dx.doi.org/10.1016/0167-8396(94)90300-X.
[8] F.J. Toledo, M.V. Herranz, J.M. Blanes, V. Galiano, Quick and accurate strategy for calculating the solutions of the photovoltaic single-diode model equation, IEEE J. Photovolt. 12 (2) (2022) 493–500, http://dx.doi.org/10.1109/JPHOTOV.2021.3132900.
[9] D.A. Barry, P.J. Culligan-Hensley, S.J. Barry, Real values of the W-function, ACM Trans. Math. Software 21 (2) (1995) 161–171, http://dx.doi.org/10.1145/203082.203084.
[10] R. Iacono, J.P. Boyd, New approximations to the principal real-valued branch of the lambert W-function, Adv. Comput. Math. 43 (2017) 1403–1436, http://dx.doi.org/10.1007/s10444-017-9530-3.
[11] E. Batzelis, G. Anagnostou, C. Chakraborty, B. Pal, Computation of the lambert w function in photovoltaic modeling, in: ELECTRIMACS 2019, Vol. 604, 2019, pp. 583–595, http://dx.doi.org/10.1007/978-3-030-37161-6_44.
[12] F. Toledo, J.M. Blanes, Geometric properties of the single-diode photovoltaic model and a new very simple method for parameters extraction, Renew. Energy 72 (2014) 125–133, http://dx.doi.org/10.1016/j.renene.2014.06.032.
[13] F.J. Toledo, J.M. Blanes, V. Galiano, Two-step linear least-squares method for photovoltaic single-diode model parameters extraction, IEEE Trans. Ind. Electron. 65 (8) (2018) 6301–6308, http://dx.doi.org/10.1109/TIE.2018.2793216.
[14] F.J. Toledo, V. Galiano, J.M. Blanes, V. Herranz, E. Batzelis, Photovoltaic single-diode model parametrization. An application to the calculus of the euclidean distance to an I–V curve, Math. Comput. Simulation (2023) http://dx.doi.org/10.1016/j.matcom.2023.01.005.
[15] T.C. Banwell, A. Jayakumar, Exact analytical solution for current flow through diode with series resistance, ELectron. Lett. 36 (2000) 291–292, http://dx.doi.org/10.1049/el:20000301.
[16] R.M. Corless, G.H. Gonnet, D.E. Hare, D.J. Jeffrey, D.E. Knuth, On the lambert W function, Adv. Comput. Math. 5 (1996) 329–359.
[17] B. Marion, A. Anderberg, C. Deline, M. Muller, G. Perrin, J. Rodriguez, S. Rummel, T. Silverman, F. Vignola, S. Barkaszi, Data for validating models for PV module performance, 2021, http://dx.doi.org/10.21948/1811521.



F. Javier Toledo was born in Almoines (Valencia), Spain, in 1973. He received the M.Sc. degree in mathematics from the University of Valencia, Spain, in 1997, and the Ph.D. degree from the Miguel Hernández University of Elche (UMH), Spain, in 2003. He is currently an Associate Professor in the Department of Statistics, Mathematics and Informatics, UMH, and a Researcher in the Institute "Center of Operations Research" (CIO), UMH. His research interests are divided into two lines, photovoltaics in the field of engineering and stability in optimization in the field of mathematics. Some of his most interesting contributions in photovoltaics can be found on the website https://pvmodel.umh.es/.

V. Galiano was born in Elche, Spain, on November 24, 1975. He received the M.Sc. degree in telecommunication engineering from the Polytechnic University of Valencia, Valencia, Spain, in 1999, and the Ph.D. degree in computer science from the University Miguel Hernandez of Elche, Alicante, Spain, in 2007. He is currently an Associate Professor with the Department of Computers Engineering, Miguel Hernandez University in Elche, Alicante, Spain. His main research interests include high-level interfaces design in parallel libraries, high efficiency video coding, parallel simulations for modeling electronic systems and photovoltaics modeling.

V. Herranz was born in Valencia, Spain, in 1974. He received the M. Sc. Degree in mathematics from the University of Valencia, Spain, in 1997, and the Ph.D. degree from the Miguel Hernández University of Elche (UMH), Spain, in 2006. She is currently an Associate Professor in the Department of Statistics, Mathematics and Informatics, UMH, and a Researcher in the Institute "Center of Operations Research" (CIO), UMH. Her main research interests include photovoltaic modeling, coding theory and cryptography.

José M. Blanes was born in Elche, Spain, in 1974. He received the M.Sc. degree in telecommunication engineering from the Universidad Politécnica de Valencia, València, Spain, in 1998, and the Ph.D. degree in industrial technologies from the Miguel Hernández University, Elche, Spain, in 2011. He is currently an Associate Professor in the Department of Materials Science, Optics and Electronics Technology, Miguel Hernández University. His main research interests include space power systems, industrial electronics and photovoltaic modeling.