d/Collection

Effective Modeling of the Photovoltaic Characteristic Curves using Superellipse

Tofopefun Nifise Olayiwola

Supervisor: Prof. Sung-Jin Choi

Department of Electrical, Electronic and Computer Engineering
University of Ulsan

This dissertation is submitted for the degree of
Master of Science

Effective Modeling of the Photovoltaic Characteristic Curves using Superellipse

Tofopefun Nifise Olayiwola

Supervisor: Prof. Sung-Jin Choi

Department of Electrical, Electronic and Computer Engineering
University of Ulsan

This dissertation is submitted for the degree of
Master of Science

공학석사 학위논문

초타원을 이용한 태양광 특성 곡선의 모델링 기법 연구

EFFECTIVE MODELING OF THE PHOTOVOLTAIC CHARACTERISTIC CURVES USING SUPERELLIPSE

울산대학교 대학원

전기전자컴퓨터공학과

TOFOPEFUN NIFISE OLAYIWOLA

초타원을 이용한 태양광 특성 곡선의 모델링 기법 연구

Effective Modeling of the Photovoltaic Characteristic Curves using Superellipse

지도 최성진

이 논문을 공학석사 학위논문으로 제출함

2023 년 08 월

울산대학교 대학원

전기전자컴퓨터공학과

TOFOPEFUN NIFISE OLAYIWOLA

TOFOPEFUN NIFISE OLAYIWOLA 의 공학석사학위

논문을 인준함

심사위원장	전태원	
심사위원	최성진	(인)
심사위원	박재현	

울산대학교 대학원

전기전자컴퓨터공학과

2023 년 08 월

UNIVERSITY OF ULSAN

Effective Modeling of the Photovoltaic Characteristic Curves using Superellipse

by

TOFOPEFUN NIFISE OLAYIWOLA

Supervisor: Professor SUNG-JIN CHOI

A dissertation submitted in partial fulfillment of

the requirements for the degree of Master of Science

in the

Department of Electrical, Electronic and Computer Engineering

University of Ulsan

August 2023

This certifies that the dissertation of TOFOPEFUN NIFISE OLAYIWOLA is approved by:

Committee Chair: **Professor Tae-Won Chun**
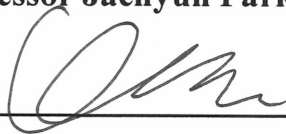
Signature:

Committee Member: **Professor Sung-Jin Choi**

Signature:

Committee Member: **Professor Jaehyun Park**

Signature:

Department of Electrical, Electronic, and Computer Engineering

University of Ulsan

August 2023

I dedicate this thesis to my loving parents.

(Mo ya iwe afọwọkọ yii si awọn obi mi)

Acknowledgements

Tofopefun Nifise Olayiwola

Ulsan, South Korea

August 2023

Abstract

Solar energy remains a clean, reliable, and affordable alternative to fossil fuel energy for achieving carbon neutrality. Thus, photovoltaic (PV) panels are often used for the conversion of this abundant light energy into electrical energy in the form of DC currents for the end user. To achieve an effective performance evaluation of photovoltaic (PV) panels in real-time, the accurate representation of its characteristic curves especially at maximum power point (MPP) is essential.

In recent years, equivalent circuit models which are often derived from the conversion behavior of PV panels have been one of the approaches used in its modeling, simulation, and analysis. However, since the basic I-V characteristic equation is both implicit and nonlinear, achieving full-range enumeration is quite tedious. Thus, this thesis puts forward a novel, easy-to-fit empirical model for the modeling and analysis of PV panels. Based on the unique similarities between the graphical characteristics of the typical I-V curve and the geometric shape of a superellipse, an explicit mathematical equation describing an accurate approximation of the PV characteristic curve under both uniform and non-uniform conditions is established.

In this thesis, a step-by-step procedure for deriving the novel PV model multi-dimensional equation and parameter extraction procedures are extensively illustrated and discussed. Performance evaluation using three different criteria including the IEC EN 50530 standard

show that the newly proposed model maintained low absolute errors with the vicinity of MPP, and across the full range of the reconstructed PV characteristic curves irrespective of the PV panel specifications, cell material, and ambient condition.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Overview

The rapid growth in the integration of renewable energy sources such as photovoltaic (PV) has contributed to cleaner, and eco-friendlier environment. In its simplest form, solar cells can be regarded as a tool for harnessing the abundant light energy from the sun and transforming it into electrical energy for the end consumer [1].

However, since the rated voltage of a single solar cell ranges between $0.5 - 0.8$ V, the maximum power point (MPP) achievable is usually very low. As a result, solar cells are often connected in series and parallel connections (see Fig. 1.1) to form a PV panel. In large PV plants, to reach the specified power level, PV panels are generally connected in either in series and or parallel connections to boost the voltage of the module which in turn increases the overall output power.

In real applications as shown in Fig. 1.2, PV installations are usually located very far away from their control center and are often expected to operate maximally under varying

Cell

Module
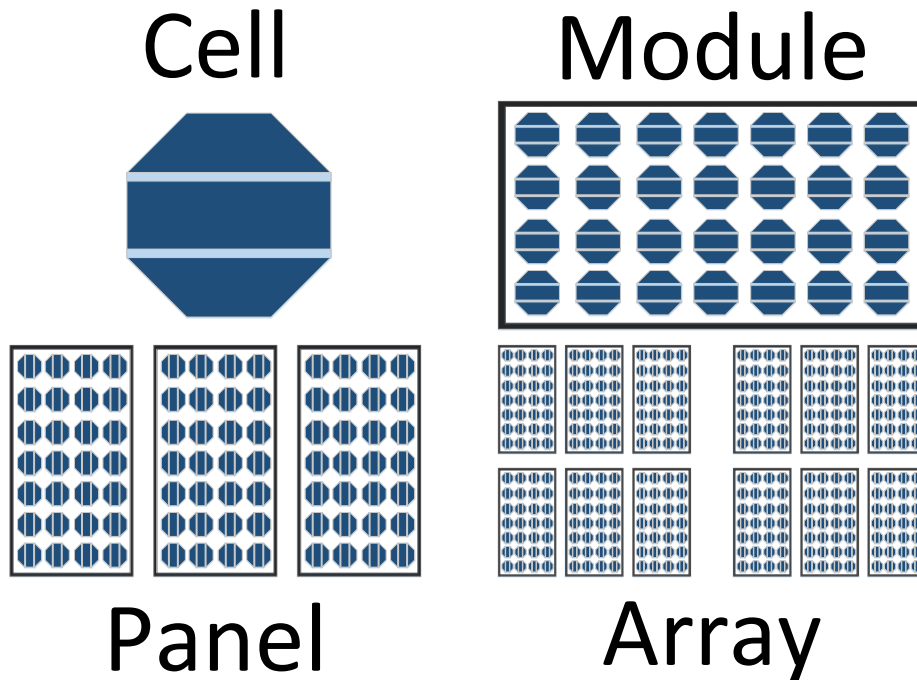
Panel

Array

Fig. 1.1 Classification of PV component — cell, panel, string, array.

ambient conditions [2]. As a result, the real-time performance analysis, and optimization of these PV installations under various ambient conditions is quite a challenge.

To address this challenge, alternative solutions in the form of equivalent models have been proposed in literature for the effective modeling, simulation, and study of both the static and dynamic behavior of PV panels. Generally, these equivalent models which are based on the conversion principle of the PV panel can be classified as either single-diode, double-diode, or triple-diode model (see Fig. 1.3).

Nonetheless, due to the mathematical complexities of the resulting characteristic equations, and the required number of parameters, the implementation and full understanding of the behavior of the PV panel is still a major challenge even in commercial available software environments such as PSIM, PLECS, MATLAB/Simulink, etc.
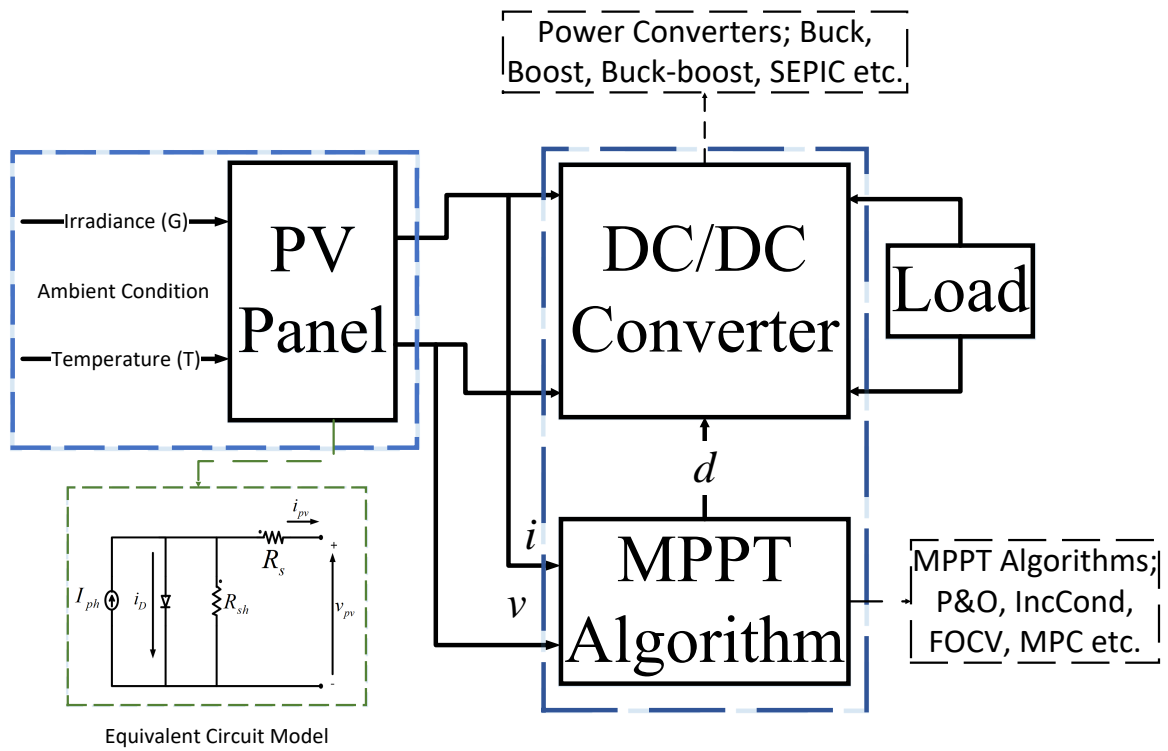
Fig. 1.2 Basic block diagram describing the general operation of PV installations.

(a)

(b)

(c)
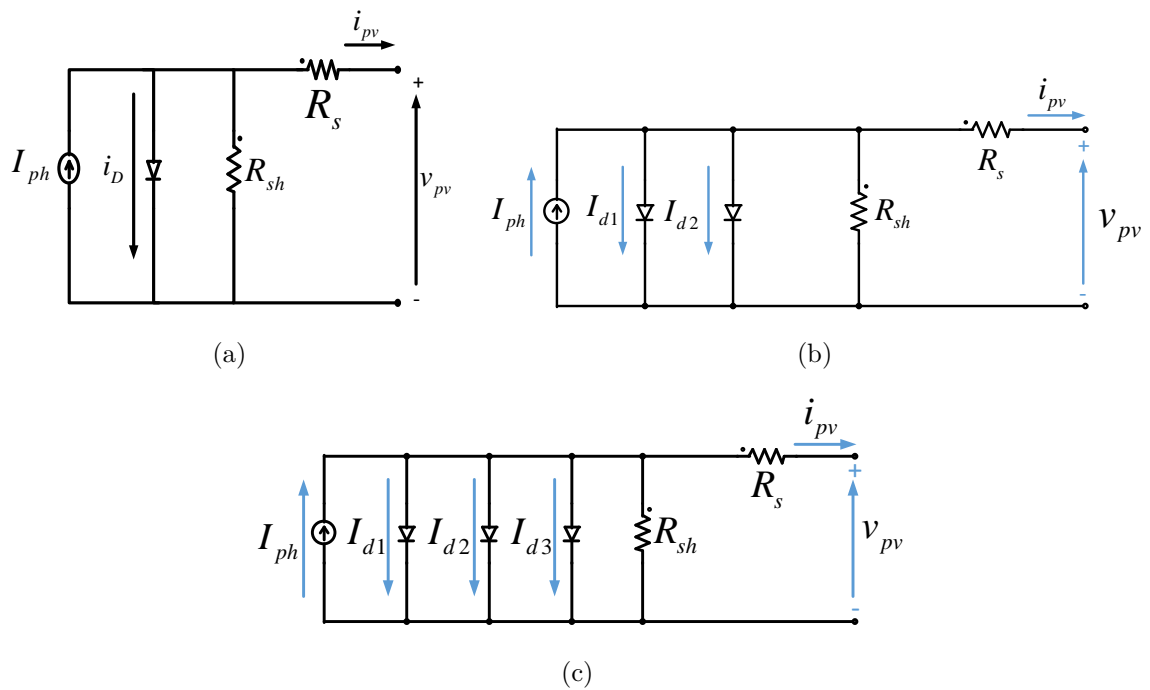
Fig. 1.3 Equivalent circuit-based representation of the photovoltaic (PV) panels (a) single-diode (b) double-diode (c) triple-diode.

1.2 Objectives and contributions

Simple and accurate modeling of the characteristic curves of PV panels is essential for its effective performance evaluation. Hence, the objectives of this thesis include

(A) Empirical modeling of PV panels – In this thesis, a novel empirical model based on the theoretical and mathematical similarities between the graphical curves of the I–V curve and the geometric shapes of the superellipse is proposed. As a result of this transformation, the explicit equation, and corresponding fitting parameters describing the superellipse model loses its direct physical meaning with the conventional electrical circuit parameters.

(B) Multidimensional PVM equation — Based on the theoretical finding, an explicit PVM simultaneous equations describing the full-range enumeration of the PV characteristic curves which are valid under both uniform and varying ambient conditions are derived.

(C) Parameter convergence — Due to non-convergence, infinite iterations, and numerical stability issues, special attention is required in obtaining the solutions to the multidimensional equations. Hence, in this thesis, three distinct optimization algorithms are utilized in obtaining the fitting parameters of the superellipse model for 6 PV panels with different cell materials at varying ambient conditions.

(D) Model accuracy — To evaluate the superiority of the superellipse model, four different creiteria were utilized. While at MPP, the accuracy of the superellipse model is evaluated in accordance with the IEC EN 50530 standard, the errors across the full range of the reconstructed PV characteristic curves are also computed using well known mathematical equations.

1.3   Thesis organization

The structure and brief overview of this thesis are as follows

Chapter 1 (Introduction) This chapter gives a brief overview of the need for PV modeling, the objective, and the main contributions of this thesis.

Chapter 2 (Conventional circuit-based modeling of the PV characteristic curves) This chapter gives a detailed review of the conventional single-diode model for PV panels and its approximate PV model equations. In addition, the advantages and limitations of the conventional PVM equations were also identified.

Chapter 3 (Superellipse-based modeling of the PV characteristic curve) In this chapter, the superellipse model is introduced as an easy-to-fit alternative model for PV panels. Also included in this chapter is the step-by-step derivation of explicit equations describing full-range approximation of the PV characteristic curves and its multidimensional equations whose roots are always the fitting parameters of the superellipse model.

Chapter 4 (Parameter convergence of the superellipse model) In this chapter, three different optimization algorithms were utilized to extract the parameters of the superellipse model at varying ambient condition. The main goal of this chapter is to identify the most suitable optimization algorithm with high numerical stability and low parametric distortion.

Chapter 5 (Accuracy evaluation of the superellipse model) In this chapter, a comprehensive performance evaluation and accuracy of the superellipse model is carried out.

By utilizing the IEC EN 50530 standard and other evaluation criteria, the superiority of the proposed model is confirmed.

Chapter 6 (Conclusion and future work) This chapter gives a summary of the thesis and suggests possible future work.

Chapter 2

Conventional circuit-based modeling of the PV characteristic curves

2.1 Introduction

The characteristic curves describing the behavior of PV panels under standard test condition (STC) conditions are usually readily available in most manufacturer's datasheet. These curves are plots of either the output current against the rated output voltage, or a plot of the output power against its rated voltage i.e. I–V and P–V curves.

Over the years, numerous empirical models have been proposed in literature for the effective remodeling and approximation these PV characteristic curves. By taking advantage of the conversion principle describing a typical PV panel, equivalent circuit-based models have been derived as one of the approach used evaluating the performance of these complex (or nonlinear) systems.

Based on the internal structure of its circuit representation i.e. number of fitting parameters, and diode components, these equivalent circuit-based models can be classified as either single-diode, double-diode or triple-diode models. It therefore becomes imperative that the complexity of the mathematical equation describing the behavior of panel would increase drastically with the number of electrical components (i.e. fitting parameters).

As such, most researchers, and technicians in both academia and industry rely heavily on the single-diode model for implementation in most power electronic software environments as it contains the fewest number of electrical parameters, and simpler I–V characteristic equation.

2.2 Single-diode model

2.2.1 Enumeration under STC

By applying the circuit analysis to Fig. 2.1a, the basic characteristic equations describing the typical I–V curve as outlined in the datasheet can therefore be expressed as

$$i_{pv} = I_{ph} - I_s \cdot \left[e^{\left(\frac{v_{pv} + i_{pv} R_s}{A N V_t} \right)} - 1 \right] - \frac{v_{pv} + i_{pv} R_s}{R_{sh}} \tag{2.1}$$

where i_{pv} is the PV output current (A), v_{pv} is the PV output voltage (V), I_{ph} is the photovoltaic current (A), I_s the saturation current of the diode (A), V_t is the thermal voltage (V), A is the ideality factor, while R_s, R_{sh} and N are the series resistance (Ω), parallel resistance (Ω), and number of cells in a series string inside the panel respectively.
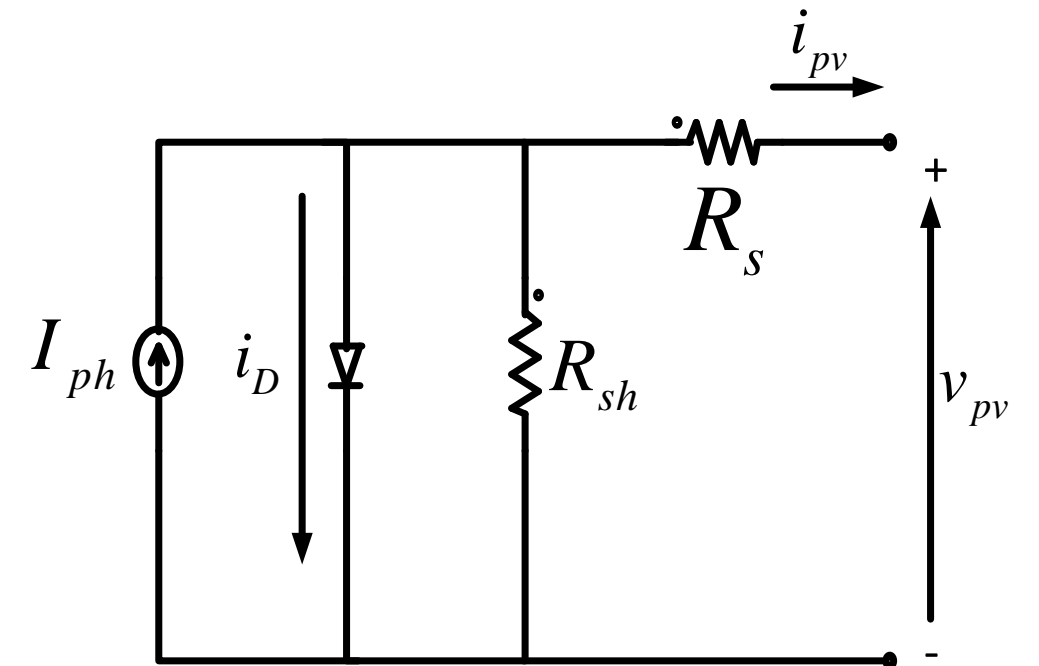
(a)
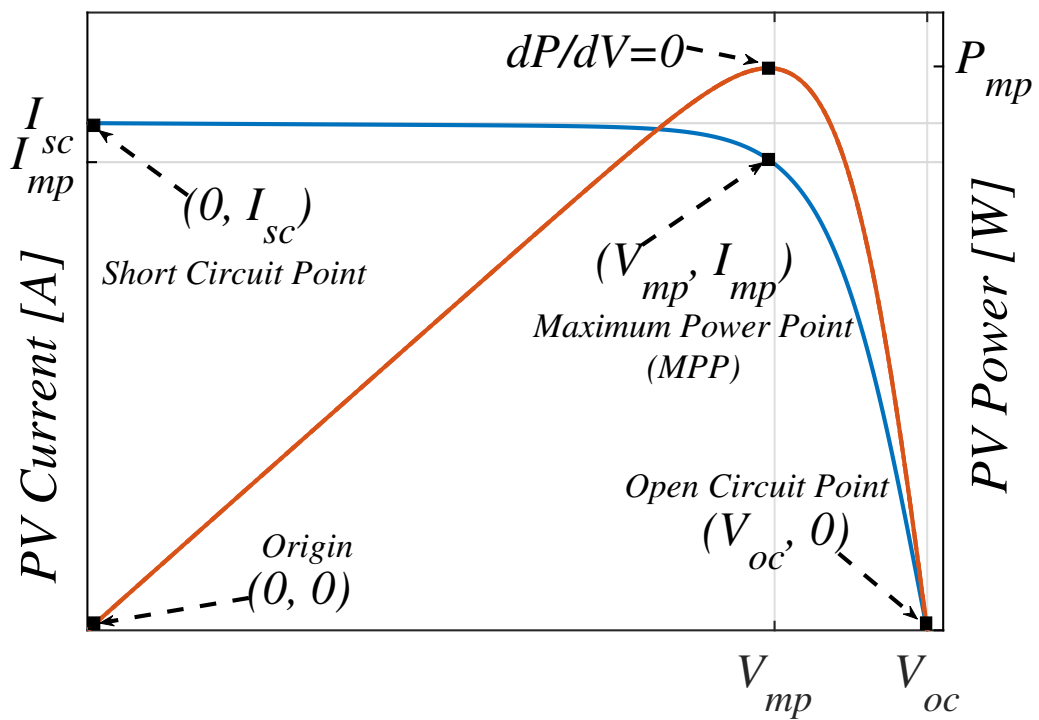
(b)

Fig. 2.1 Equivalent electrical circuit of the single-diode model and its corresponding PV characteristic curves under STC as obtained from a typical PV panel (a) single-diode model (b) key points of the I–V and P–V curves.

9

Accordingly, I_{ph} as the name appears is heavily dependent on the ambient condition i.e. irradiance G, and temperature T of the photovoltaic panel. As such, by carrying out further circuit-analysis, I_{ph} can be expressed mathematically as

$$I_{ph} = \frac{G}{G_n}\left(I_{scn} + \beta_I(T - T_n)\right) \tag{2.2}$$

where all the terms in (2.2) are defined such that I_{scn} is the short-circuit current at STC $(T_n - 298.14K, G_n = 1000W/m^2)$ while β_I is the temperature coefficient of I_{sc}.

Similarly, the saturation current in (2.1) can also be further expressed mathematically as

$$I_s = \frac{I_{scn} + \beta_I(T - T_n)}{e^{\frac{Vocn + \beta_V(T - Tn)}{AV_t}} - 1} \tag{2.3}$$

where V_{ocn} is the open-circuit voltage at STC and β_V is the temperature coefficient of V_{oc}.

By obtaining the numerical solutions to equations to (2.1), the regeneration of the I–V curve across its full-range from short-circuit I_{sc} to open-circuit V_{oc} can therefore be achieved as shown in Fig. 2.1b. Aside from these two points, a typical I–V curve regardless of ambient conditions would always include the voltage at maximum power point V_{mp} and the current at maximum power point I_{mp}.

2.2.2 Effect of varying ambient conditions

However, in basic form (2.1) doesn't take into account the effects of varying ambient conditions of the PV panel as a slow or fast transient change in ambient conditions leads to corresponding changes in the key points of the I–V curve.
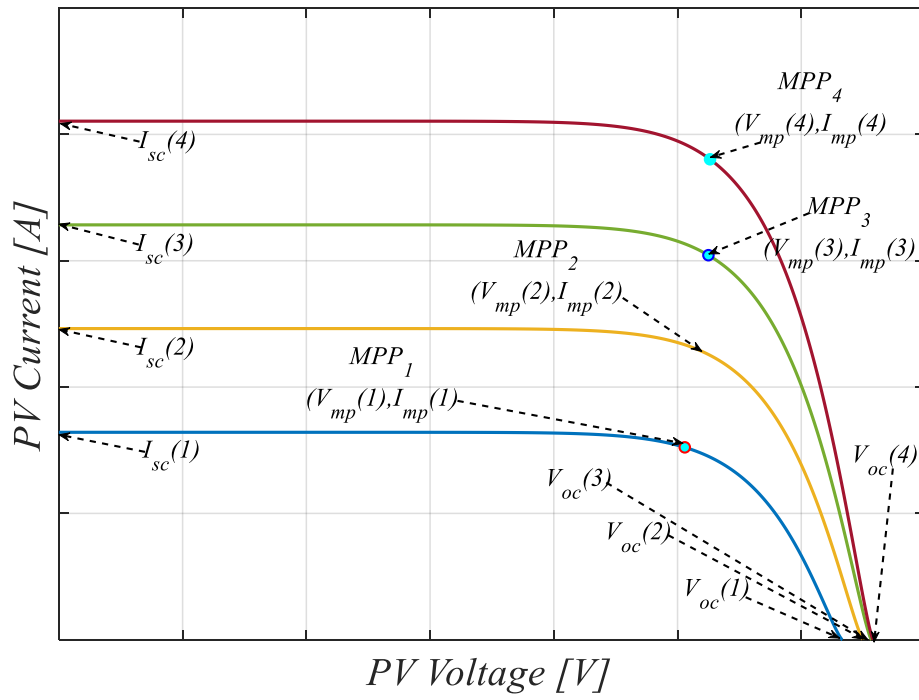
(a)
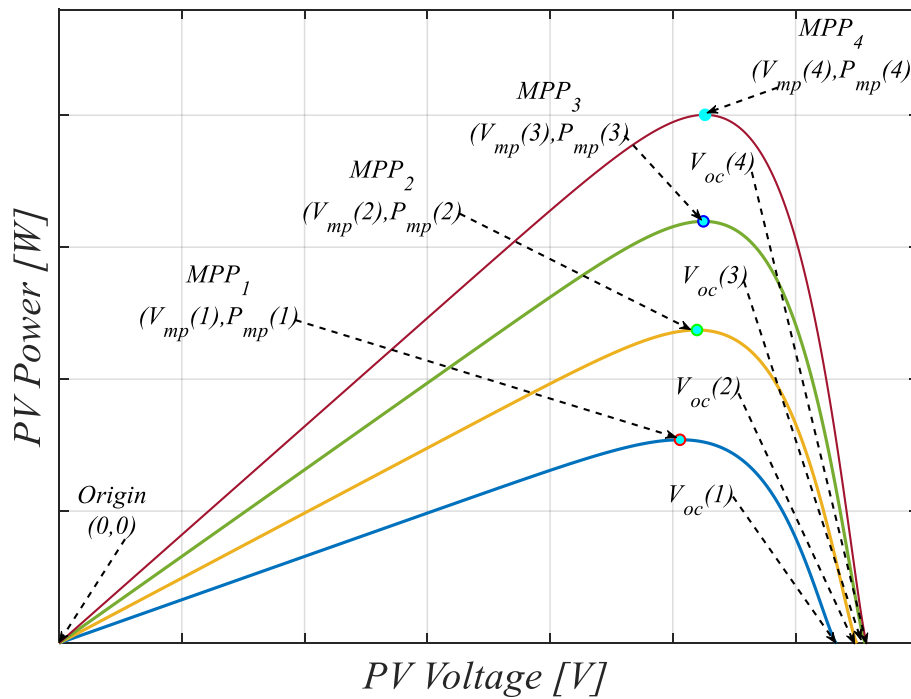
(b)

Fig. 2.2 Reconstruction of the PV characteristic curves under varying irradiance condition (a) I—V curve (b) P—V curve.
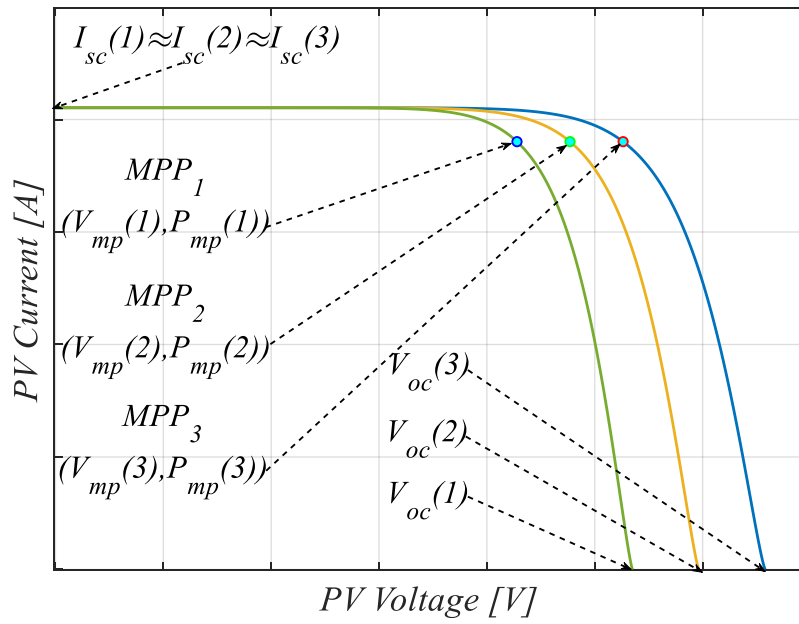
(a)
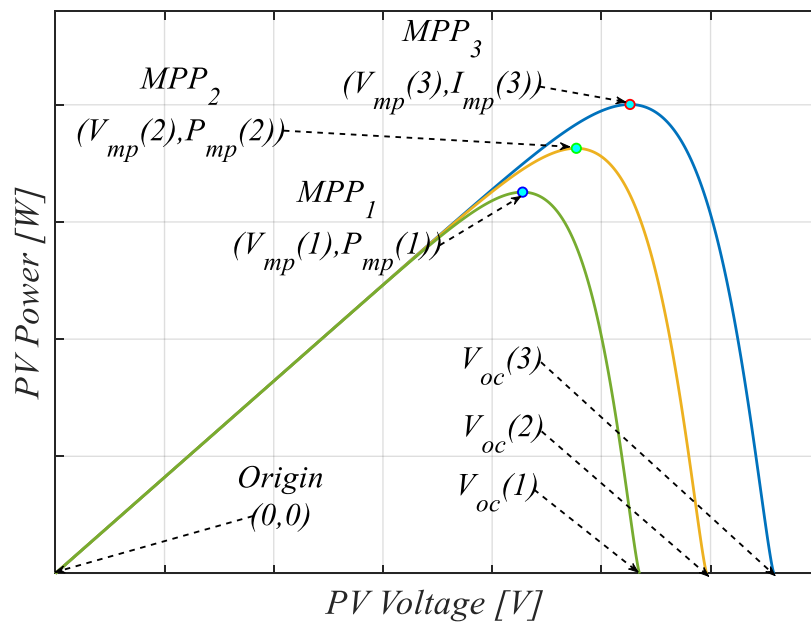
(b)

Fig. 2.3 Reconstruction of the PV characteristic curves under varying temperature condition (a) I—V curve (b) P—V curve.

Taking into account these changes, researchers have clearly defined mathematical equations for estimating the key point values of the I–V curve under this conditions (see Fig. 2.3). As derived in [3], the equations can be expressed as

$$I_{sc} = I_{scn}\frac{G}{G_n}\left[1 + \beta_I(T - T_n)\right] \tag{2.4a}$$

$$V_{oc} = V_{ocn} + NA_n\frac{kT}{q}ln\left(\frac{G}{G_n}\right) + \beta_V(T - T_n) \tag{2.4b}$$

$$I_{mp} = I_{mpn}\frac{G}{G_n}\left[1 + \beta_{I,mp}(T - T_n)\right] \tag{2.4c}$$

$$V_{mp} = V_{mpn} + NA_n\frac{kT}{q}ln\left(\frac{G}{G_n}\right) + \beta_{V,mp}(T - T_n) \tag{2.4d}$$

where the quantities with the subscript n denotes its values under STC. While A_n determines the squareness of the I–V curve, β_{Imp} and β_{Vmp} can be easily approximated as [3]

$$\beta_{I,mp} \cong \beta_I, \beta_{V,mp} \cong \beta_V. \tag{2.5}$$

2.3   Approximate PVM equations

However, due to the exponential term, (2.1) is inherently nonlinear and implicit. As such, obtaining the numerical solutions for the full-range enumeration of the PV curves are quite tedious and cumbersome. To address this challenge, substitute or approximate PV model (PVM) equations have been successfully proposed and implemented in literature. These approximate PVM equations are basically series of interdependent or interconnected equations that are used to transform or parameterize the exponent term in (2.1).

Thus, based on the structure, PVM equations can be easily classified as either iteration-based or analytical-based PVM equations. As the name suggests, iteration-based PVM are

derivative equations of (2.1) which are usually obtained after applying datasheet constraints under specified conditions. On the other hand, analytical-based PVM equations are usually series of explicit interconnected equations obtained after transformation of the exponent in (2.1) into separate domains for performance analysis.

2.3.1 Iteration-based PVM equations

Examples of some the most widely cited PVM equations that fall under the iteration-based PVM equations are given in [4]. By separating the fitting parameters in (2.1) into dependent and independent variables, an approximate PVM equation known as the reduced two-parameter model was obtained at utilized for approximating the PV characteristic curves [5, 6].

[7] also suggested that by replacing the conventional $dI/dV = -1/R_p$ equation with the $dP/dI = 0$, a simplified PVM equation can be easily obtained. Besides, the five fitting parameters of the single-diode model were also obtained using algebraic equations after the introduction of two additional *"thermal"* parameters. These parameters as proposed in [8] were obtained as solutions to the solar cell temperature coefficients. [9] also proposed utilizing the remarkable points of the I–V curve and utilizing a specialized optimization algorithm for the reconstruction of the I–V curve without estimating the electrical fitting parameters with the exemption of the diode ideality factor A_n.

The initialization conditions which are often selected by trial-and-error, and mathematical complexity of the iterative algorithms all contribute to the non-convergence and high computational speed often associated with this method. As such, iteration-based PVM equations are not the preferred approach employed in the modeling and approximation of PV systems in most commercially available power electronics software environments.

2.3.2 Analytical-based PVM equations

As explained in Section (2.3), by the decoupling or parameterization the exponent in (2.1), several explicit equations describing the I–V curve can be obtained. Due to the easiness in substituting the $W(x)$ function into (2.1), approximations using the Lambert-W function remains one of the most popular methods in this category. In essence, these method expresses the I–V characteristic equation in asymptotic formula [10, 11] thereby establishing an explicit mathematical relationship between (2.1) and either branches of the $W(x)$. Nonetheless, the basic Lambert-W method (also known as Haley's method) doesn't express $W(x)$ as simple elementary equations. Hence, obtaining the data point solutions to these equations are still quite challenging.

To address this challenge, several improvements or modifications have been suggested and successfully applied in the in literature. These improvements involve applying series expansion formulas to the asymptotic equation describing the I–V curve (2.6) [4] such as hermite-padé interpolation [12], exact closed-form using maple software [13], hybrid calculation formula [14], simple approximate formula [15], barry's analytical approximation formula [16, 17], marine predator algorithm [18], hybrid analytical [19], hessian function [17].

$$
W(x) = L_1 + \frac{L_2}{L_1} + \frac{L_2(-2 + L_2)}{2L_1^2} + \frac{L_2(6 - 9L_2 + 2L_2^2)}{6L_1^3} + \frac{L_2(-12 + 36L_2 - 22L_2^2 + 3L_2^2)}{12L_1^4}
$$
$$
+ \frac{L_2(60 - 300L_2 + 350L_2^2 - 125L_2^3 + 12L_2^4)}{60L_1^5} + O\left[\left(\frac{L_2}{L_1}\right)^6\right] \tag{2.6}
$$

where $L_1 = ln(x),\ L_2 = ln(ln(x))$

2.3.2.1 Hybrid explicit expansion

To obtain the numerical solutions and cope with the unsuitability of the asymptotic formula [14] proposed approximating (2.6) using the hybrid formula which expands the asymptotic expression (7 terms) using special series expansion techniques into

$$i_{pv} = \left(\frac{R_{sh}(I_{ph} + I_s) - v_{pv}}{R_s + R_{sh}} \right) - \left(\frac{a}{R_s} \right) W \left[\frac{R_s R_{sh} I_s}{a(R_s + R_{sh})} exp \left(\frac{R_s R_{sh}(I_{ph} + I_s) + R_{sh} v_{pv}}{a(R_s + R_{sh})} \right) \right]$$
(2.7)

where $a = ANV_t$,

$$W_1 = u + \left(\frac{u}{1+u} \right) p + \left[\frac{u}{2(1+u)^3} \right] p^2 - \left[\frac{u(6u^2 - 8u + 1)}{24(1+u)^7} \right] p^4 \left[\frac{u}{2(1+u)^3} \right] p^2$$
$$- \left[\frac{u(24u^3 - 58u^2 + 22u - 1)}{120(1+u)^9} \right] p^5$$
(2.8)

$$W_2 = L_1 + \frac{L_2}{L_1} + \frac{L_2(-2 + L_2)}{2L_1^2} + \frac{L_2(6 - 9L_2 + 2L_2^2)}{6L_1^3} + \frac{L_2(-12 + 36L_2 - 22L_2^2 + 3L_2^2)}{12L_1^4}$$
$$+ \frac{L_2(60 - 300L_2 + 350L_2^2 - 125L_2^3 + 12L_2^4)}{60L_1^5}$$
(2.9)

and $u = x/e$, $p = 1 - (x/e)$. By utilizing this specialized formula, the relative error was observed to be lower than 0.1% for the entire real non-negative argument range [14].

2.3.2.2 Barry analytical expansion

Similar to [14], [16] proposed another special series expansion techniques based on [17] such that

$$W(x) = (1+\varepsilon)In\left[\cfrac{\left(\cfrac{6}{5}\right)x}{In\left(\cfrac{\left(\cfrac{12}{5}\right)x}{In(1+\left(\cfrac{12}{5}\right)x)}\right)}\right] - \varepsilon In\left[\cfrac{2x}{In(1+2x)}\right] \qquad (2.10)$$

where $\epsilon = 0.4586887$ is a constant. For the values of $x \geq 0$, the approximate error using this formula was claimed to be 0.196%.

2.3.2.3   Winitzki approximation

A simpler series expansion based on [20] has also been utilized in literature for approximating $W(x)$ such that [15]

$$W(x) = In(x)\left[1 - \frac{In(In(x))}{In(x)+1}\right]. \qquad (2.11)$$

reported to yield errors less than 1.5% for $x \geq 2$ [15]. At smaller arguments, the calculation error is very large and (2.11) is not suitable for application.

2.3.2.4   3 point model

By utilizing the datasheet information (i.e. I_{SC} and V_{OC}) and the valid assumption that for a PV module $(exp((V_{OC} - I_{SC}R_s)/V_t) >> 1)$ [21, 22] transformed (2.1) into

$$i_{pv} = I_{sc} - \left(\frac{v_{pv}}{R_{sh}+R_s}\right) - \left(I_{sc} - \frac{V_{oc}}{R_{sh}+R_s}\right) \times exp\left(\frac{v_{pv} - V_{oc}+i_{pv}R_s}{V_t}\right). \qquad (2.12)$$

With a high R_{sh} $(500 - 1000\Omega)$ for the silicon modules, the assumptions that $I_{sc} >> v_{pv}/(R_{sh} + R_s)$ and $I_{sc} >> V_{oc}/(R_{sh} + R_s)$, in (2.12) are valid. In comparison with $(V - V_{oc})$ in (2.12), the term $i_{pv}R_s$ can be neglected such that

$$i_{pv} = I_{sc} - I_{sc} \cdot exp\left(\frac{v_{pv}}{V_{oc}} - 1\right)^{\left(\frac{V_{oc}}{V_t}\right)}. \tag{2.13}$$

For any module, the condition that $-1 < (v_{pv}/V_{oc}) - 1 < 0$ is always true. Therefore, $exp((v_{pv}/V_{oc}) - 1) \approx 1 + (v_{pv}/V_{oc}) - 1$, approximation is also valid in (2.13). Thus, (2.13) can therefore be simplified as

$$i_{pv} = I_{sc} - I_{sc}\left(\frac{v_{pv}}{V_{oc}}\right)^{\left(\frac{V_{oc}}{V_t}\right)} \tag{2.14}$$

Thus, (2.14) can be represented in a simple polynomial model as

$$i_{pv} = c + a \cdot (v_{pv})^b \tag{2.15}$$

where $a = \frac{I_{sc}}{(V_{oc})^b}$, $b = \frac{V_{oc}}{V_t}$, and $c = I_{sc}$ are the model coefficients. It is important to emphasize that these coefficients often change with variations in the PV operating conditions G and T. (2.15) therefore represents a new explicit expression for computing the I–V curve and the three coefficients are obtained on-line using only three (I, V) coordinates measurements.

2.3.2.5 Padé approximant

Since the I–V characteristic equation contains an exponential function, the padé approximant can also be utilized to represent the exponential term in (2.1). Many experiments

have shown that the precision, computation speed is best and fastest when $m+n$ is a constant and when $m=n$ in

$$[m/n](x) = \frac{P_m(x)}{Q_n(x)} \tag{2.16}$$

where $P_m(x) = \sum_{i=0}^m P_i x^i$, $Q_n(x) \sum_{j=0}^n Q_j x^j$ are the polynomials of the degree $m, n \epsilon N$ [23]. According to PV cell material, different padé approximant model can be selected to express $exp(\frac{i_{pv}R_s}{A_n})$. The padé approximant $[m/n]_{exp(z)}$ of the exponential function $exp(z)$ is as the follows:

$$[m/n]_{exp(z)} = \frac{\sum_{i=0}^n (2n-i)!\,n!}{(2n)!\,i!\,(n-1)!} z^i \Big/ \frac{\sum_{i=0}^n (2n-i)!\,n!}{(2n)!\,i!\,(n-1)!} (-z^i) \tag{2.17}$$

where $z = i_{pv}Rs/A_n$. By substituting (2.1) into (2.17), and then solving it, a new explicit representation of I—V is obtained. Nonetheless, for a high degree equation, an analytical root is difficult to obtain. Therefore, for simplicity and easy computation, $m=n\leq 4$ of $[m/n]_{exp(z)}$ may be set in practical application. If padé approximants $[2/2]_{exp(z)}$ is used to represent $exp(i_{pv}Rs/A_n)$, i.e., $m=n=2$ in (2.17), $exp(i_{pv}Rs/A_n)$ can be given by

$$[2/2]_{exp(z)} = \frac{12+6z_z^2}{12-6z+z^2}. \tag{2.18}$$

According to (2.18), (2.1) can therefore be written as

$$i_{pv} = I_{ph} - I_s \cdot exp\left(\frac{v_{pv}}{A_n}\right) \cdot \frac{12+6z+z^2}{12-6z+z^2} - 1 - \frac{v_{pv}-i_{pv}R_s}{R_{sh}}. \tag{2.19}$$

Let $a_1 = A_n R_{sh} + A_n R_s$, $b = -6 A_n R_{sh} - I_{ph} R_s R_{sh} - I_s R_s R_{sh} + v_{pv} R_s + I_s R_s R_{sh} exp(v_{pv}/A_n) - 6 A_n R_s$, $c = 12 A_n R_{sh} + 6 I_{ph} R_s R_{sh} + 6 I_s R_s R_{sh} - 6 v_{pv} R_s + 6 I_s R_s R_{sh} exp(v_{pv}/A_n) + 12 a R_s$, $d = -12 I_{ph} R_s R_{sh} - 12 I_s R_s R_{sh} + 12 v_{pv} R_s + 12 I_s R_s R_{sh} exp(v_{pv}/A_n)$, (2.19) can be rewritten in the following manner:

$$a_1 z^2 + b z^2 + c z + d = 0. \tag{2.20}$$

According to Shengjin's formula in Fan [24, 25], we can therefore take $A = b^2 - 3 a_1 c$, $B = bc - 9 a_1 d$, $C = c^2 - 3bd$. Thus, according to (2.1) and (2.19), we can obtain $D = B^2 - 4AC > 0$. Since z is a positive real, the solution of (2.20) should be given by

$$z = \frac{-b - \sqrt[3]{Y_1} - \sqrt[3]{Y_2}}{3 a_1} \tag{2.21}$$

where $Y_1 = Ab + \frac{3 a_1 \left(-B + \sqrt{B^2 - 4AC} \right)}{2}$, $Y_2 = Ab + \frac{3 a_1 \left(-B - \sqrt{B^2 - 4AC} \right)}{2}$. Hence, the approximation of i_{pv} in (2.19) can be given by

$$I_P = a \frac{-b - \sqrt[3]{Y_1} - \sqrt[3]{Y_2}}{3 a_1 R_s} \tag{2.22}$$

where I_P is the new approximation of the current i_{pv}.

2.3.2.6 Taylor's series expansion

To decouple the exponential term in (2.1), we can also employ the Taylor's series expansion to express $exp(\frac{i_{pv} R_s}{A_n})$ such that

$$e^z \approx 1 + z + \frac{1}{2!} z^2 + \frac{1}{3!} z^3 \tag{2.23}$$

where $z = \frac{i_{pv} R_s}{A_n}$.

Accordingly, (2.23) can be rewritten as

$$a_1 z^3 + b z^2 + c z + d = 0 \qquad (2.24)$$

where $a_1 = \frac{1}{6} I_s exp(\frac{v_{pv}}{A_n})$, $b = \frac{1}{2} I_s exp(\frac{v_{pv}}{A_n})$, $c = I_s exp(\frac{v_{pv}}{A_n}) + \frac{A_n}{R_s} + \frac{A_n}{R_{sh}}$, $d = I_s exp(\frac{v_{pv}}{A_n}) + \frac{v_{pv}}{R_s} + I_L + I_{ph}$.

Thus, to obtain z we solve (2.24) according to Shengjin's formula in Van Zeghbroeck, and then the estimation current $I_T = \frac{A_n z}{R_s}$. According to Shengjin's formula [24, 25], let $A = b^2 - 3a_1 c$, $B = bc - 9a_1 d$, $C = c^2 - 3bd$. According to (2.1) we can obtain $I_s \cdot exp(v_{pv}/a) + (v_{pv}/R_{sh}) - I_{ph} - I_s \leq 0$ and then $D = B^2 - 4AC > 0$ according to (2.24). Since z is positive and real, the solution to (2.24) can be obtained using

$$z = \frac{-b - \sqrt[3]{Y_1} - \sqrt[3]{Y_2}}{3a_1} \qquad (2.25)$$

where $Y_1 = Ab + \frac{3a_1(-B + \sqrt{B^2 - 4AC})}{2}$, $Y_2 = Ab + \frac{3a_1(-B - \sqrt{B^2 - 4AC})}{2}$. So, we have

$$I_T = A_n \frac{-b - \sqrt[3]{Y_1} - \sqrt[3]{Y_2}}{3a_1 R_s} \qquad (2.26)$$

Hence, (2.26) becomes the an explicit PVM equation for the computation of the I–V curve.

2.3.2.7   Two-port network expansion

Since (2.1) is implicit and transcendental in nature, the curve computation can performed in two ways:

(A) An iterative solution for i_{pv} with known v_{pv} values ($0 \leq v_{pv} \leq V_{oc}$)

(B) An iterative solution for v_{pv} with known i_{pv} values ($0 \leq i_{pv} \leq I_{sc}$)

Generally, the diode voltage ($V_d = v_{pv} + i_{pv}R_s$) variation in the SC to OC regions is such that $I_{sc}R_s \leq V_d \leq V_{oc}$. By taking advantage of the independent variation in V_d, [26] proposed new explicit equations for computation of the I–V curve.
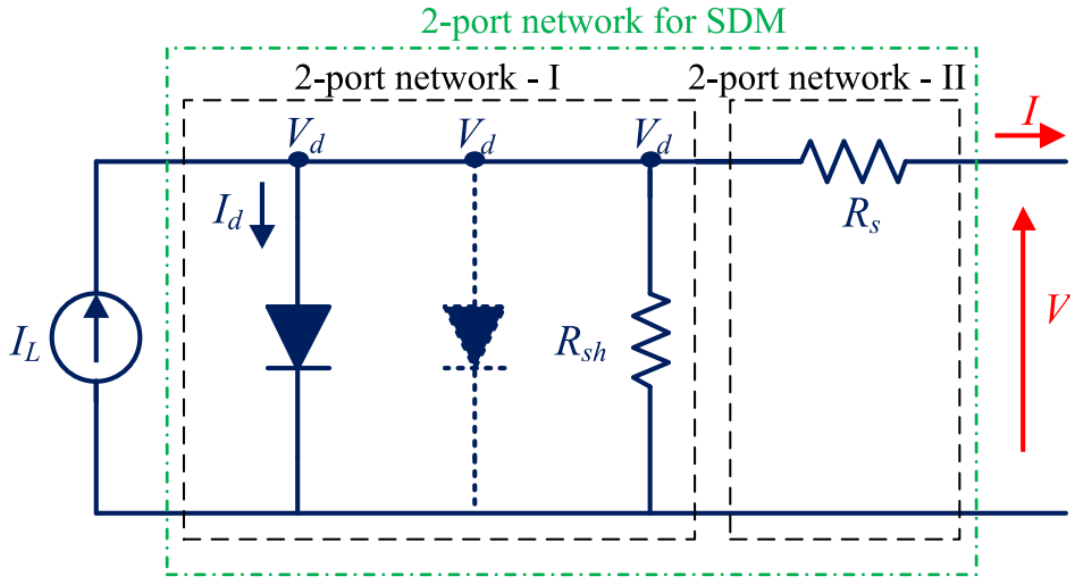
Fig. 2.4 The single-diode model (SDM)/double-diode model (DDM) equivalent circuit with two 2-port networks [26].

The 2-port network-I as proposed by [26] includes I_{ph} current source, the diode and R_{sh} into (2.1). Since the currents through the diode and R_{sh} are governed by V_d, the output current i_{pv}, from the proposed network has been expressed as

$$i_{pv} = I_{pv} - I_s\left[exp\left(\frac{V_d}{V_t}\right) - 1\right] - \frac{V_d}{R_{sh}} = f(V_d). \tag{2.27}$$

(2.27) is therefore an explicit function of V_d and thus the i_{pv} computation is explicit. The 2-port network-II contains only R_s. Now, V_d and i_{pv} are the inputs to this network and compute the output voltage (v_{pv}) as following

$$i_{pv} = I_{pv} - I_s\left[exp\left(\frac{V_d}{V_t}\right) - 1\right] - \frac{V_d}{R_{sh}} = f(V_d) \tag{2.28}$$

(2.28) gives an explicit function for V_d with the i_{pv} computation being explicit, and the v_{pv} computation is always explicit. Consequently, (2.27) and (2.28) are two simultaneous equations representing the 2-port networks-I and II, respectively. This approach first computes i_{pv} and subsequently v_{pv}, and is designated as I-approach.

In an optional approach, the output current from the 2-port network-I is the same as the input current to the 2-port network-II, i.e. i_{pv}. The equivalence in currents is achieved by the substitution of (2.27) in (2.28) and the following expression for v_{pv} is generated

$$i_{pv} = I_{pv} - I_s\left[exp\left(\frac{V_d}{V_t}\right) - 1\right] - \frac{V_d}{R_{sh}} = f(V_d) \tag{2.29}$$

Hence, (2.29) gives an explicit function of V_d while the v_{pv} computation is explicit. Thus, the proposed 2-port network-II expresses i_{pv} as

$$i_{pv} = \frac{V_d - v_{pv}}{R_s} = f(V_d, v_{pv}) \tag{2.30}$$

2.3.2.8   Two-parameter model

[27, 28] also proposed expressing (2.1) as a function of the voltage such that

$$i_{pv} = I_{sc} \cdot \left[1 - C_1 \cdot \left(exp(\frac{v_{pv}}{C_2 V_{oc}}) - 1 \right) \right] \tag{2.31a}$$

$$C_1 = \left(1 - \frac{I_{mp}}{I_{sc}} \right) \cdot exp\left(\frac{-V_{mp}}{C_2 V_{oc}} \right) \tag{2.31b}$$

$$C_2 = \frac{\left(\frac{V_{mp}}{V_{oc}} - 1 \right)}{In\left(1 - \frac{I_{mp}}{I_{sc}} \right)}. \tag{2.31c}$$

with the coefficients C_1 and C_2 being dependent on the following module parameters

(A) Short circuit current I_{sc}

(B) open circuit voltage V_{oc}

(C) maximum power point voltage V_{mp}

(D) maximum power point current I_{mp}

Although these analytical-based PVM equations do not preserve the physical meaning of the photovoltaic conversion process, they have been utilized for the accurate modeling and approximation of the PV characteristic curves.

Regardless, the mathematical complexity, and required number of fitting parameters (or information) are all still a major limitations hindering the full understanding of the behavior of PV panels. With the increasing wide-range application of PV panels, simple and easy-to-fit empirical models are essential for the effective performance evaluation of these nonlinear systems.

Chapter 3

Superellipse-based modeling of the PV characteristic curves

3.1 Introduction

To address the challenges as identified in Chapter 2, a novel superellipse-based PV model is proposed for the representation of the behavior of the PV characteristic curves. Due to the unique similarities between the graphical characteristics of a typical I–V curve at STC and the geometric shapes of a superellipse, a novel empirical model describing the behavior of PV panels is obtained.

The structure of this chapter is as follows. In Chapter 3.2, the theoretical background of the basic superellipse, its variations, and its relationship with the typical I–V curve under STC are discussed extensively. By taking into account the effects of varying ambient conditions on the I–V curve, explicit equations describing the reconstruction PV curves are subsequently well-explained in Chapter 3.3. To obtain the optimum fitting parameters of the superellipse

model, a generalized step-by-step procedure for PV panels is outlined in Chapter 3.4.1. Finally, in Chapter 3.4.2, numerical techniques applicable for extracting the parameters from this multidimensional equation using unconstrained optimization all explained in detail.

3.2   Proposed model

In its simplest form, a superellipse is a geometric curve that always retains x and y intercepts irrespective of any variation or distortion in its overall shape. The single fitting parameter constraints of the single-shaped superellipse as shown in Fig. 3.1a greatly reduce the degree of freedom and flexibility of its reconstructed geometric curves [29, 2].

On the hand, a double-shaped superellipse with two fitting parameters as shown in Fig. 3.1b retains its fixed axes, with a higher degree of flexibility. Along these geometric curves, the implicit equation describing any point $P(x,y)$ can therefore be expressed mathematically as

$$\left(\frac{x}{A}\right)^m + \left(\frac{y}{B}\right)^n = 1 \tag{3.1}$$

where A is the $x-intercept$ value, B is the $y-intercept$ value, and m and n are the optimum fitting parameters of the superellipse. A detailed description of the major differences between the single-shaped and the double-shaped superellipse is given in Appendix A.

By the direct substitution of A and B as the V_{oc} and I_{sc} of a typical I–V curve, a novel implicit equation describing any point along the I–V curve can therefore be approximated as

$$\left(\frac{v}{V_{oc}}\right)^m + \left(\frac{i}{I_{sc}}\right)^n = 1. \tag{3.2}$$

where i and v are the output current and output voltage of the superellipse model respectively.
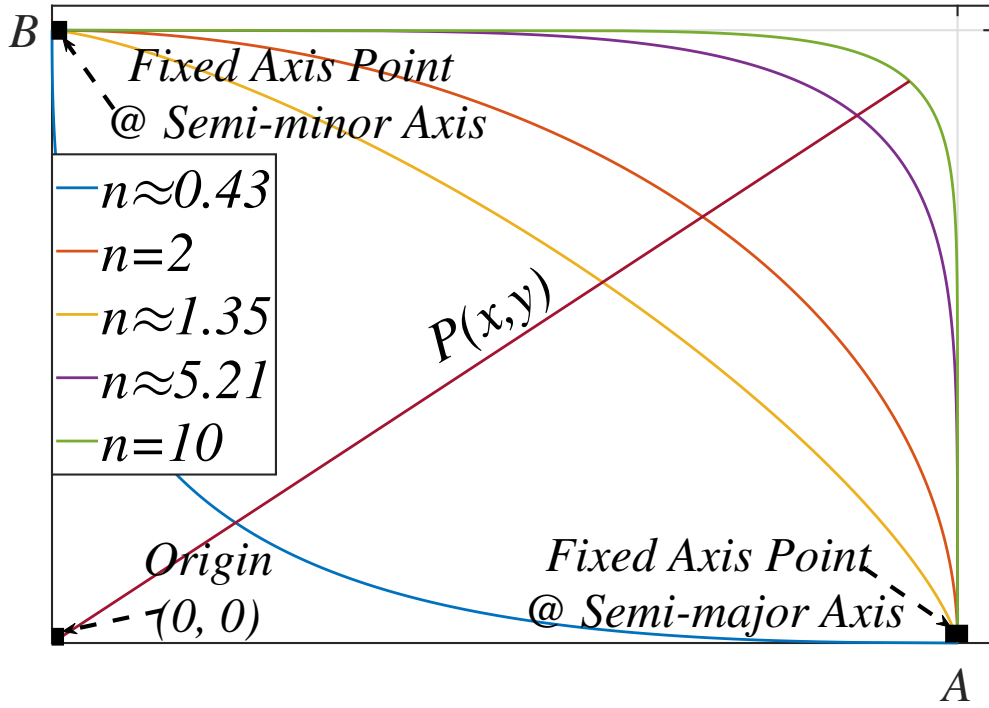
(a)
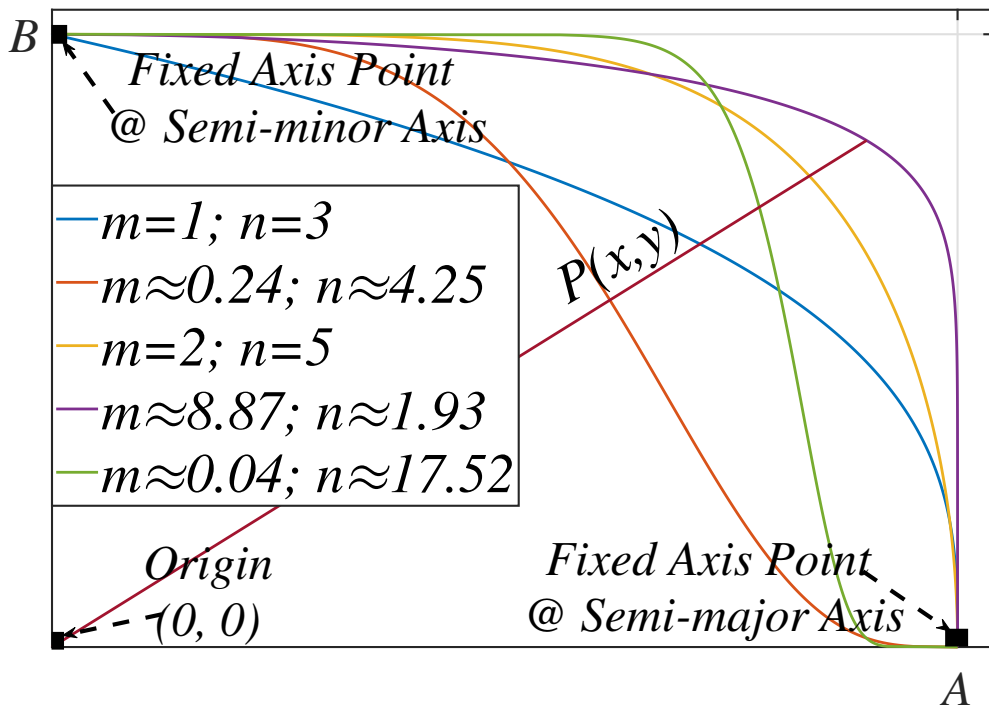
(b)

Fig. 3.1 Superellipse with varying parameter values (a) single-shaped superellipse (b) double-shaped superellipse.

By making i subject of the formula, an explicit equation describing the full-range range reconstruction of the I–V curve under STC similar to (2.1) can therefore be written as

$$i = I_{sc}\left[1 - \left(\frac{v}{V_{oc}}\right)^m\right]^{\frac{1}{n}}.$$

(3.3)

The MATLAB code for the implementation of this novel explicit equation is given in Appendix B.

3.3 Consideration of varying ambient condition

As explained in Chapter 2.2.2, the superellipse model are also dependent on the varying ambient conditions of the PV panel. As such, the slightest variations in environmental conditions leads to corresponding changes in the V_{oc} and I_{sc} of (3.3).

To estimate these fixed axes values, many researchers have studied various approximations of the I–V curve using datasheet constraints [30]. By applying this equation and constraints, the short circuit current I_{sc}^* and open circuit voltage V_{oc}^* of the superellipse model under varying ambient conditions can therefore be estimated as

$$I_{sc}^* = I_{scn}\frac{G}{G_n}$$

(3.4a)

$$V_{oc}^* = V_{ocn} + NA_n\frac{kT}{q}ln\left(\frac{G}{G_n}\right) + \beta_V(T - T_n).$$

(3.4b)

However, (3.4b) is still heavily dependent on the accurate estimation of A_n which generally ranges between $1 < A_n < 2$. Different techniques with varying levels of mathematical complexities have been proposed in literature to estimate these values under varying ambient conditions [31–33].

According to the physics describing the conversion behavior of the PV panels, the typical I–V curve under any ambient condition has been defined as a superposition of its diode characteristic curve under the same condition [34]. Therefore, A_n can be simplify considered as the reciprocal slope factor of the voltage ratio of the I–V curve, which in practice characterizes the Shockley-Read-Hall recombination of the diode.

Thus, in this thesis, a simple and quick mathematical equation for obtaining A_n in (3.4b) for any PV panel can therefore be expressed as

$$A_n = \frac{V_{oc}}{V_{mp}}.$$
(3.5)

Afterwards, by combining (3.4) and (3.5) into (3.3), an updated explicit equation for expressing the full-range approximation of the I–V curve under varying ambient conditions can therefore be expressed as

$$i^* = I_{sc}^* \left[1 - \left(\frac{v^*}{V_{oc}^*} \right)^m \right]^{\frac{1}{n}}.$$
(3.6)

where i^* and v^* become the output current and voltage of the superellipse model under varying ambient condition. In Appendix C, a detailed study of the effect of variations in MPP values on the proposed empirical model is examined.

3.4   Parameter extraction

3.4.1   Finding optimum fitting parameters

Based on the theoretical facts establishing the proposed empirical model in Chapter 3.2, we can therefore assume that the superellipse model will exhibit both the datasheet constraints

describing a typical I–V curve and its subsequent mathematical properties. Optimum fitting parameters of the superellipse model are crucial for achieving a high model accuracy, especially within the vicinity of MPP. Thus, the datasheet constraints [3] are applied to the basic explicit equation (3.3).

(A) Constraint 1: I–V curve enumeration starts from the open-circuit voltage point $(V_{oc}, 0)$ and ends at the short-circuit point $(0, I_{sc})$. Therefore, the fixed axes point of the superellipse model at its voltage and current source segments are always the I_{sc} and V_{oc} of the I–V curve respectively.

(B) Constraint 2: The enumeration I–V curve must always pass through its MPP. Thus, by substituting the MPP values at STC as specified in any manufacturer's datasheet into (3.3), an explicit equation describing the accurate and exact MPP of the superellipse model can therefore be expressed as

$$I_{mp} = I_{sc} \left[1 - \left(\frac{V_{mp}}{V_{oc}} \right)^m \right]^{\frac{1}{n}}. \tag{3.7}$$

(C) Constraints 3: At MPP, the slope of the P–V curve is null. In accordance with the single-diode model, the instantaneous power p of the superellipse model can therefore be defined as

$$p = i \cdot v. \tag{3.8}$$

To meet this constraint, we can therefore differentiate (3.3) such that

$$\left. \frac{dp}{dv} \right|_{v=V_{mp}} = i \cdot \left(\frac{dv}{dv} \right) + v \cdot \left(\frac{di}{dv} \right) \Bigg|_{i=I_{mp}, v=V_{mp}} = 0, \tag{3.9}$$

then, we obtain

$$I_{mp} = \frac{mI_{sc}}{n} \left(\frac{V_{mp}}{V_{oc}} \right)^m \left(\frac{I_{mp}}{I_{sc}} \right)^{1-n}. \tag{3.10}$$

By combining Constraints (B) and (C), a multidimensional equation describing the novel empirical PV model can therefore be expressed as

$$I_{mp} = I_{sc} \left[1 - \left(\frac{V_{mp}}{V_{oc}} \right)^m \right]^{\frac{1}{n}} \tag{3.11a}$$

$$I_{mp} = \frac{mI_{sc}}{n} \left(\frac{V_{mp}}{V_{oc}} \right)^m \left(\frac{I_{mp}}{I_{sc}} \right)^{1-n}. \tag{3.11b}$$

Thus, (3.11) subsequently creates the set of necessary and sufficient conditions that must always be obeyed to obtain the optimum fitting parameters of the superellipse model. Based on the mathematical properties of the superellipse model, the fitting parameters of the superellipse should be regarded as unchanging and invariant. Nonetheless, a comprehensive assessment of this assumption will be evaluated in Chapter 4.

3.4.2   Unconstrained optimization algorithms

Multidimensional equations such as (3.11) are not new in engineering. Over the years, several numerical techniques have been proposed in literature [35] for obtaining solutions to multivariable equation. Basically, numerical techniques are genetic algorithms that search for the solutions to multivariable equations by constantly moving along a specified search direction during an iteration. As such, depending on the approach employed in determining the search vector, several optimization algorithms have been proposed in literature.

Table 3.1 Classification of the numerical techniques for unconstrained optimization

Zero-order (Non-gradient)	First-order (Gradient)	Second-order
Scan and Zoom	Steepest Descent	Pattern Search
Random Walk	Conjugate Gradient	Newton-Raphson (NR)
Pattern Search	Davidon-Fletcher-Powell (DFP)	Levenberg-Marquardt
Powell's Method (PM)	Broydon-Fletcher-Goldfarb-Shanno (BFGS)	

Based on the number of derivative of the objective function required for establishing the search direction during iteration, the numerical techniques can be classified as either zero-order, first-order, and second-order methods (see Table 3.1) [35].

3.4.2.1   Zero order numerical techniques

As the name suggests, the zero-order methods require no derivative of the cost function in determining or establishing its search vector during iteration. Thus, only changes in the cost function or design parameters can terminate the iteration loop to result in parameter convergence. Examples of numerical techniques under this category are Scan and Zoom, Random Walk, Pattern Search, and Powell's method. Due to its quadratic convergence property, Powell's method remains one of the most popular techniques adopted in minimizing any cost function. This quadratic convergence property ensures that *"For any quadratic problem with n-variables, parameter convergence would always be achieved in less than or equal to n Powell cycles"*. The step-by-step procedure describing Powell's method is given below

Based on **A1**, it can be observed that in determining its search direction, Powell's method would always utilize the history of its previous search directions. As such, by combining (3.11) using conventional algebraic equations, a novel cost function describing the superellipse

---

Algorithm 1 Powell method

Step 1: Choose $\boldsymbol{X}_1, \boldsymbol{N}_c$ (number of cycles)
$\quad\quad f_c(1) = f(\boldsymbol{X}_1); \boldsymbol{X_c(1)} = \boldsymbol{X}_1$
$\quad\quad \varepsilon_1, \varepsilon_2 :$ tolerance for stopping criteria
$\quad\quad$**Set** $j = 1$ (Initialize Powell cycle count)
$\quad\quad$**For** $i = 1, n$
$\quad\quad \boldsymbol{S}_i = \hat{\boldsymbol{e}}_i$ (univariate step)
Step 2: For each cycle of j
$\quad\quad$For $i = 1, n$
$\quad\quad$**If** $(j \geq 2)\boldsymbol{S}_i \leftarrow \boldsymbol{S}_{i+1}$ (Powell shift)
$\quad\quad \boldsymbol{X}_{i+1} = \boldsymbol{X}_i + \boldsymbol{\alpha}_i \boldsymbol{S}_i$
$\quad\quad \boldsymbol{\alpha}_i$ is determined by **minimizing** $f(\boldsymbol{X}_{i+1})$
$\quad\quad$end of For loop
$\quad\quad \boldsymbol{S}_j^p = \boldsymbol{S}_{i+1} = \sum_{i=1}^n \boldsymbol{\alpha}_i * \boldsymbol{S}_i = \boldsymbol{X}_{n+1} - \boldsymbol{X}_1$ (Pattern step)
$\quad\quad \boldsymbol{X}_j^p = \boldsymbol{X}_{n+1} + \boldsymbol{\alpha}_j * \boldsymbol{S}_j^p$
$\quad\quad \boldsymbol{X}_c(j+1) \leftarrow \boldsymbol{X}_j^p; f_c(j+1) = f(\boldsymbol{X}_j^p)$ (store cycle values)
Step 3: $\Delta f = f_c(j+1) - f_c(j); \Delta X = X_c(j+1) - X_c(j)$
$\quad\quad$**If** $|\Delta f| \leq \varepsilon_1$; **stop**
$\quad\quad$**If** $\boldsymbol{\Delta X}^T \boldsymbol{\Delta X} \leq \varepsilon_2$; **stop**
$\quad\quad$**If** $j - N_c$; **stop**
$\quad\quad \boldsymbol{X}_1 \leftarrow \boldsymbol{X}_j$
$\quad\quad j \leftarrow j + 1$
$\quad\quad$**Go to Step 2**

---

model can therefore be obtained as

$$p(m,n) = I_{mp} - I_{sc}\left[1 - \left(\frac{V_{mp}}{V_{oc}}\right)^m\right]^{\frac{1}{n}} \quad (3.12a)$$

$$q(m,n) = I_{mp} - \frac{mI_{sc}}{n}\left(\frac{V_{mp}}{V_{oc}}\right)^m\left(\frac{I_{mp}}{I_{sc}}\right)^{1-n} \quad (3.12b)$$

$$fc(m,n) = (p(m,n))^2 + (q(m,n))^2 \quad (3.12c)$$

$$fc(m,n) = \left((I_{mp} - I_{sc}\left[1 - \left(\frac{V_{mp}}{V_{oc}}\right)^m\right]^{\frac{1}{n}}\right)^2 + \left(I_{mp} - \frac{mI_{sc}}{n}\left(\frac{V_{mp}}{V_{oc}}\right)^m\left(\frac{I_{mp}}{I_{sc}}\right)^{1-n}\right)^2 \quad (3.12d)$$

where $\boldsymbol{X}_1 = [m, n]$ in **A1** and the minimum of $fc(m,n)$ becomes the optimum fitting parameters of the superellipse model. The MATLAB source code for implementing the PM is given in Appendix D.

3.4.2.2 First-order numerical techniques

While the search direction of zero-order methods is not dependent on the derivative of the cost function, the search direction of first-order methods is usually constructed around the gradient of the objective function. As such parameter convergence or local optimums would only be obtained if the Karush-Kuhn-Tucker (KKT) conditions for unconstrained problems are obeyed [36].

Numerical techniques under this category include Steepest descent [37], Conjugate gradient also known as Fletcher-Reeves [38], Davidon-Fletcher-Powell (DFP) [39, 40], Brydon-Fletcher-Goldfarb-Shanno (BIFGS) [40]. Thus, the DFP method is considered an improvement of the conventional conjugate method.

Due to its inherent quadratic convergence property, the DFP method can also be considered a Newton-like or second-order method where the search direction is usually an $n \times n$ matrix containing the history of previous iterations. A step-by-step procedure for computing the local optimum of any objective function can therefore be expressed as shown in **A2**. Similar to Powell's method as explained in Chapter 3.4.2.1, the DFP method (**A2**) adopts the same novel objective function (3.12d) for the superellipse model. Appendix E gives the MATLAB code for implementing this algorithm.

3.4.2.3 Second-order numerical techniques

According to literature, the Newton-Raphson method remains one of the most popular numerical techniques for obtaining solutions to multidimensional equations. The fast convergence, and relative ease of this method make it of particular interest in this thesis as one of the algorithms for the parameter extraction of the superellipse model.

Algorithm 2 Davidon-Fletcher-Powell (DFP)

Step 1: Choose $\boldsymbol{X}_1, [\boldsymbol{A}_1]$ (Initial metric), $\boldsymbol{N}$

$\varepsilon_1, \varepsilon_2, \varepsilon_3$ **:** (tolerance for stopping criteria)

Set $i = 1$ (Initialize iteration counter)

Step 2: $\boldsymbol{S}_i = -[\boldsymbol{A}_i]\boldsymbol{\Delta f}(\boldsymbol{X}_i)$

$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i + \boldsymbol{\alpha}_i \boldsymbol{S}_i; \ \boldsymbol{\Delta X} = \boldsymbol{\alpha}_i \boldsymbol{S}_i$

α_i is determined by **Minimizing** $f(\boldsymbol{X}_{i+1})$

Step 3: **If** $\boldsymbol{\Delta f}(\boldsymbol{X}_{i+1})^T \boldsymbol{\Delta f}(\boldsymbol{X}_{i+1}) \leq \varepsilon_3$**; converged**

If $|f(X_{i+1}) - f(X_i)| \leq \varepsilon_1$**; stop** (function not decreasing)

If $\boldsymbol{\Delta X}^T \boldsymbol{\Delta X} \leq \varepsilon_2$**; stop** (design not changing)

If $i + 1 = N$**; stop** (iteration limit)

Else

$\boldsymbol{Y} = \boldsymbol{\Delta f}(\boldsymbol{X}_{i+1}) - \boldsymbol{\Delta f}(\boldsymbol{X}_i)$

$\boldsymbol{Z} = [\boldsymbol{A}_i]\boldsymbol{Y}$

$[\boldsymbol{B}] = \frac{\Delta X \Delta X^T}{\Delta X^T Y}$

$[\boldsymbol{C}] = -\frac{ZZ^T}{Y^T Z}$

$[\boldsymbol{A}_{i+1}] = [\boldsymbol{A}_i] + [\boldsymbol{B}] + [\boldsymbol{C}]$

$j \leftarrow i + 1$

Go to Step 2

To apply **A3**, we can write (3.11) as composite functions such that

$$
\begin{cases}
p(m,n) = I_{mp} - I_{sc}\left[1 - \left(\dfrac{V_{mp}}{V_{oc}}\right)^m\right]^{\frac{1}{n}} = 0 & (3.13) \\[3ex]
q(m,n) = I_{mp} - \dfrac{mI_{sc}}{n}\left(\dfrac{V_{mp}}{V_{oc}}\right)^m \left(\dfrac{I_{mp}}{I_{sc}}\right)^{1-n} = 0 & (3.14)
\end{cases}
$$

where p and q become functions with two independent variables.

Next, we apply the first-order Taylor's series expansion about an initial point (m_0, n_0) such that

$$
\begin{aligned}
p(m_k, n_k) \cong p(m_{k-1}, n_{k-1}) &+ (m_k - m_{k-1})\left.\frac{\partial p}{\partial m}\right|_{(m_{k-1}, n_{k-1})} \\
&+ (n_k - n_{k-1})\left.\frac{\partial p}{\partial n}\right|_{(m_{k-1}, n_{k-1})} = 0
\end{aligned}
\tag{3.15}
$$

Algorithm 3 Newton-Raphson Method

Step 1: Choose $\boldsymbol{X}_1, \boldsymbol{N}$
$\varepsilon_1, \varepsilon_2, \varepsilon_3$ **:** (tolerance for stopping criteria)
Set $i = 1$ (Initialize iteration counter)

Step 2: The search direction is obtained as a solution to
$\boldsymbol{H}(\boldsymbol{X}_i)\boldsymbol{S}_i = -\boldsymbol{\Delta f}(\boldsymbol{X}_i); [\boldsymbol{H}]$ is the Hessian
$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i + \boldsymbol{\alpha}_i\boldsymbol{S}_i; \; \boldsymbol{\Delta X} = \boldsymbol{\alpha}_i\boldsymbol{S}_i$
α_i is determined by **Minimizing** $\boldsymbol{f}(\boldsymbol{X}_{i+1})$

Step 3: **If** $\boldsymbol{\Delta f}(\boldsymbol{X}_{i+1})^T\boldsymbol{\Delta f}(\boldsymbol{X}_{i+1}) \leq \varepsilon_3$**; converged**
If $|\boldsymbol{f}(\boldsymbol{X}_{i+1}) - \boldsymbol{f}(\boldsymbol{X}_i)| \leq \varepsilon_1$**; stop** (function not decreasing)
If $\boldsymbol{\Delta X}^T\boldsymbol{\Delta X} \leq \varepsilon_2$**; stop** (design not changing)
If $i + 1 = N$**; stop** (iteration limit)
Else $i \leftarrow i + 1$
Go to Step 2

$$q(m_k, n_k) \cong q(m_{k-1}, n_{k-1}) + (m_k - m_{k-1})\frac{\partial q}{\partial m}\bigg|_{(m_{k-1}, n_{k-1})}$$
$$+ (n_k - n_{k-1})\frac{\partial q}{\partial n}\bigg|_{(m_{k-1}, n_{k-1})} = 0. \tag{3.16}$$

By re-arranging this equation into its matrix form, the values for m_k, n_k that meets the

KKT conditions

$$p(m_k, n_k) = q(m_k, n_k) = 0 \tag{3.17}$$

can therefore be extracted mathematically using

$$\begin{bmatrix} m_k \\ n_k \end{bmatrix} = \begin{bmatrix} m_{k-1} \\ n_{k-1} \end{bmatrix} - \begin{bmatrix} \frac{\partial p}{\partial m} & \frac{\partial p}{\partial n} \\ \frac{\partial q}{\partial m} & \frac{\partial q}{\partial n} \end{bmatrix}^{-1}\bigg|_{(m_{k-1}, n_{k-1})} \begin{bmatrix} p(m_{k-1}, n_{k-1}) \\ q(m_{k-1}, n_{k-1}) \end{bmatrix} \tag{3.18}$$

where $k = 1, 2, 3\ldots$ is the number of iterations and $\boldsymbol{X}_1 = [m, n]$ in **A2** and **A3**. This algorithm

can therefore be easily in MATLAB using the .m file as defined in Appendix F.

In subsequent Chapters, the robustness, parameter convergence, and model accuracy of

the newly proposed empirical model will be subsequently examined using these three different

optimization algorithms (see Fig. 3.2).
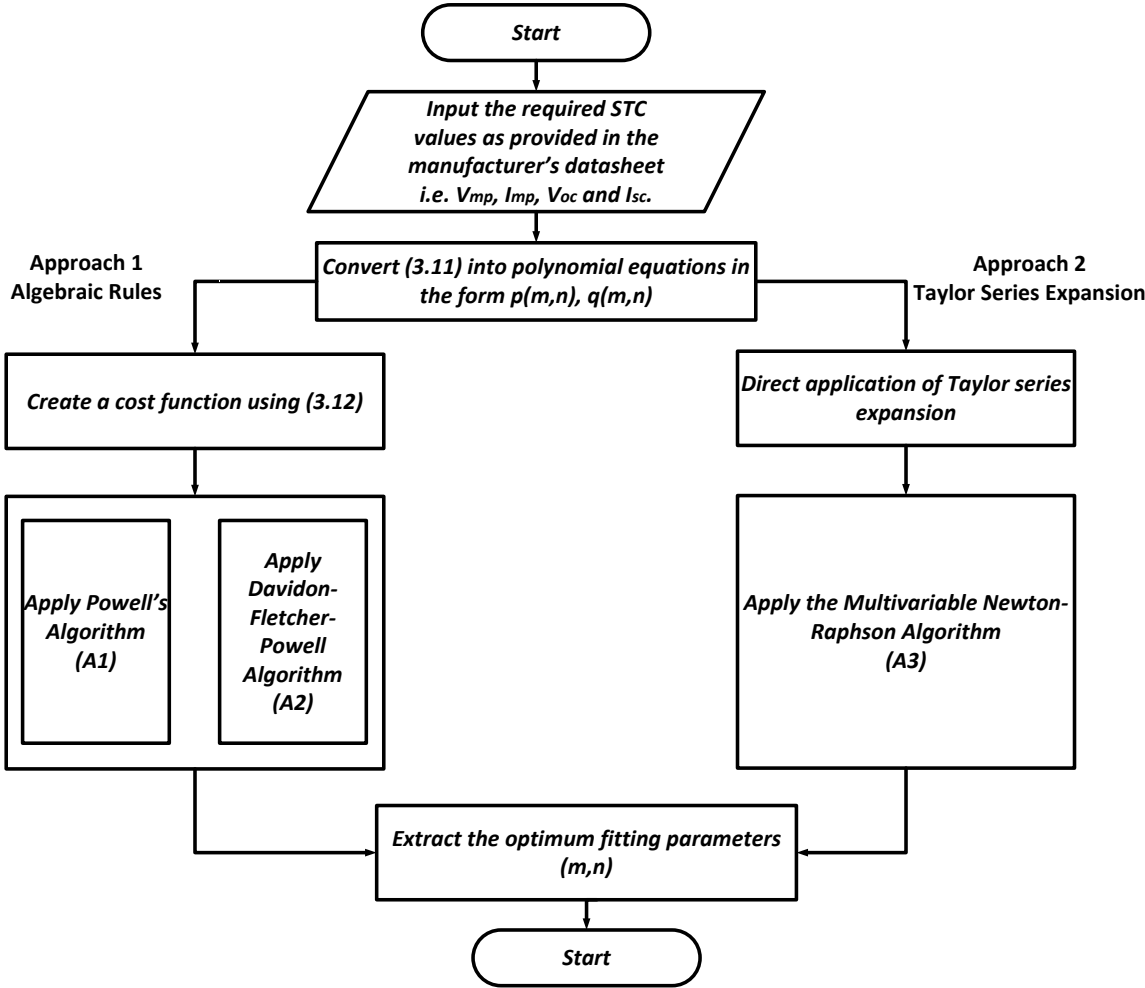
Fig. 3.2 A flowchart describing the various approaches used in extracting the fitting parameters of the superellipse in this thesis.

Chapter 4

Parameter convergence of the superellipse model

4.1 Initialization and termination condition

As described in Chapter 3.4.1, the simultaneous equation describing the superellipse model is inherently nonlinear. To obtain the accurate fitting parameters, close attention must be paid in choosing the most suitable optimization algorithm, its initial values, and terminating condition. In practice, the fitting parameters of a standard ellipse as

$$m = n = 2. \tag{4.1}$$

However, choosing (4.1) as the start value wouldn't reflect the unique theoretical and mathematical relationship of the proposed model in Chapter 3.2. Consequently, if we take into account the unique relationship established by the voltage and current ratios of the typical I–V curve at STC, a simplified mathematical equation for determining the initialization

values for the three optimization algorithm can therefore be expressed as

$$m_0 = \frac{V_{mp}}{V_{oc}} \tag{4.2a}$$

$$n_0 = \frac{I_{mp}}{I_{sc}}. \tag{4.2b}$$

$$\boldsymbol{X}_1 = [m_0, n_0]. \tag{4.2c}$$

(4.2) effectively eliminates the trial-and-error often associated with most optimization algorithms. Furthermore, to prevent non-convergence, non-ending iterative solutions to 3.12d and 3.18, termination conditions are introduced into the three algorithms such that

$$\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 1 \times 10^{-6}. \tag{4.3}$$

4.2 Convergence at STC

By directly substituting the key points values from the 6 PV panels in Table 4.1 into (3.18) and (3.12d), the optimum fitting parameters of the superellipse model are easily extracted. As shown in Table 4.2, it can be observed that the iteration count required for the parameter extraction of the superellipse model are not constant and are heavily dependent on choice of the optimization algorithm and the cell material of the PV panels. However, to determine the most suitable optimization algorithm, the parameter extraction of the proposed empirical model should also be evaluated at varying ambient condition.

Table 4.1 PV panel specifications used in this thesis.

Cell Material	PV Panel	V_{mp} (V)	I_{mp} (A)	V_{oc} (V)	I_{sc} (A)	β_I (A/°C)	β_V (mV/°C)
Multicrystalline	KC200GT	26.30	7.61	32.90	8.21	3.18×10^{-3}	-1.23×10^{-1}
Multicrystalline	CS6P-230P	29.6	7.78	36.80	8.34	65.00×10^{-3}	-3.40×10^{-1}
Monocrystalline	CS6X-305M	36.60	8.33	45.20	8.84	60.00×10^{-3}	-3.5×10^{-1}
CIGS Thin-film	Q.SMART UF L100	69.40	1.44	91.80	1.63	50.00×10^{-3}	-4.20×10^{-1}
Hybrid Thin-film	U-EA110	54.00	2.04	71.00	2.50	56.00×10^{-3}	-3.90×10^{-1}
Ultra-thin amorphous	VBHN330SA16	58.00	5.70	69.70	6.07	3.34×10^{-3}	-1.60×10^{-1}

Table 4.2 Optimum fitting parameters of the superellipse model under STC using three different optimization algorithms.

Cell Material	PV Panel	Optimization Algorithm	Model Parameters		
			m	n	Iteration
Multicrystalline	KC200GT	DFP	13.0287	0.7416	19
		PM	12.8150	0.7690	12
		NR	12.7941	0.7734	10
Multicrystalline	CS6P-230P	DFP	13.7090	0.7286	19
		PM	14.0260	0.6950	7
		NR	14.0435	0.6926	10
Monocrystalline	CS6X-305M	DFP	15.5409	0.5988	19
		PM	16.6430	0.5120	14
		NR	16.6510	0.5174	10
CIGS Thin-film	Q.SMART UF L100	DFP	7.6831	1.0134	10
		PM	7.6070	1.0300	16
		NR	7.5611	1.0372	9
Hybrid Thin-film	U-EA110	DFP	3.9754	2.0187	6
		PM	4.0920	1.9860	15
		NR	3.8084	2.0840	8
Ultra-thin Amorphous	VBHN330SA16	DFP	15.5710	0.9453	19
		PM	15.3770	0.9670	12
		NR	15.4235	0.9630	10

4.3 Convergence under varying ambient conditions

Thus to further verify the numerical stability of the superellipse model, the three optimization algorithms are utilized to extract the fitting parameters of the superellipse are various ambient conditions as provided by most manufacturer's data sheet (see Appendix G). In comparison with the other algorithms, the DFP algorithm achieves high numerical stability with almost constant parameter values across all ambient condition (see Table 4.3).

As shown in the simulation results (see Figs. 4.1 – 4.6), little or no distortion in parameter values are observed either at varying irradiance or temperature condition.

Although the PM achieves high numerical stability for the hybrid thin-film solar cell (i.e. U-EA110) as shown in Table 4.4, significant distortions in the extracted parameter were observed at varying ambient condition as well detailed in Figs. 4.7 – 4.12.

Similar to PM, the extracted fitting parameters using the NR are very unstable (Table 4.5), even though these two algorithms are stable at STC. Thus, based on the simulation results in Figs. 4.13 – 4.18, DFP algorithm is highly recommended for the extracting the fitting parameters of the superellipse model irrespective of the cell material, specification, and ambient conditions of the PV panel.

Table 4.3 Parameter extraction of the superellipse model for the 6 different PV panels under varying ambient conditions using the DFP algorithm.

Cell Material	PV Panel	Ambient Condition		Model Parameters	
		Irradiance	Temperature	m	n
Multicrystalline	KC200GT	200	25	12.3059	0.6786
		400	25	12.3840	0.7335
		600	25	12.5157	0.7580
		800	25	13.0748	0.7176
		900	25	12.9343	0.7176
		1000	25	13.0287	0.7416
		1100	25	12.5829	0.8040
		1000	35	12.5829	0.8040
		1000	50	12.5829	0.8040
		1000	60	12.5829	0.8040
		1000	75	12.5829	0.2030
Multicrystalline	CS6P-230P	200	25	13.0432	0.5847
		400	25	13.1885	0.6584
		600	25	13.2103	0.7107
		800	25	13.2489	0.7471
		900	25	13.3396	0.7543
		1000	25	12.5281	0.8113
		1100	25	14.2919	0.6763
		1000	35	12.9077	0.5782
		1000	50	11.4128	0.4036
		1000	60	10.3851	0.3003
		1000	75	10.2626	0.0969
Monocrystalline	CS6X-305M	200	25	14.0778	0.5459
		400	25	14.3750	0.5997
		600	25	14.5275	0.6341
		800	25	14.6127	0.6613
		900	25	14.6418	0.6731
		1000	25	15.5409	0.5988
		1100	25	14.6847	0.6942
		1000	35	13.9965	0.5624
		1000	50	13.9965	0.5624
		1000	60	11.7710	0.3354
		1000	75	11.5432	0.1549
CIGS Thin-film	Q.SMART UF L100	200	25	11.5432	0.1549
		400	25	11.5432	0.1549
		600	25	7.6369	0.9827
		800	25	7.5741	1.0168
		900	25	7.5651	1.0279
		1000	25	7.6831	1.0134
		1100	25	7.5651	1.0279
		1000	35	7.6922	0.8831
		1000	50	7.7634	0.6911
		1000	60	8.0371	0.5418
		1000	75	7.7812	0.4207
Hybrid Thin-film	U-EA110	200	25	4.1215	1.6656
		400	25	4.0615	1.8158
		600	25	4.0239	1.9051
		800	25	3.9967	1.9689
		900	25	3.9855	1.9952
		1000	25	3.9754	2.0187
		1100	25	3.9678	2.0398
		1000	35	4.0836	1.7773
		1000	50	4.2407	1.4279
		1000	60	4.3409	1.2056
		1000	75	4.5067	0.8862
Ultra-thin Amorphous	VBHN330SA16	200	25	14.9880	0.7590
		400	25	15.2246	0.8392
		600	25	15.4672	0.8756
		800	25	15.4672	0.8756
		900	25	15.4672	0.8756
		1000	25	15.5710	0.9453
		1100	25	15.3407	0.9885
		1000	35	15.3407	0.9885
		1000	50	15.3407	0.9885
		1000	60	15.0121	0.7633
		1000	75	15.0121	0.7633

(a)

(b)

Fig. 4.1 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the KC200GT PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.2 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the CS6P-230P PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.3 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the CS6X-305M PV panel — (a) varying irradiance (b) varying temperature.

45

(a)

(b)

Fig. 4.4 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the UF L100 PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.5 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the U-EA110 PV panel — (a) varying irradiance (b) varying temperature.

Fig. 4.6 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using DFP algorithm for the VBHN330SA16 PV panel — (a) varying irradiance (b) varying temperature.

Table 4.4 Parameter extraction of the superellipse model for the 6 different PV panels under varying ambient conditions using the PM algorithm.

Cell Material	PV Panel	Ambient Condition		Model Parameters	
		Irradiance	Temperature	m	n
		200	25	27.6890	0.0370
		400	25	30.6350	0.0250
		600	25	17.2180	0.3450
		800	25	14.3830	0.5840
		900	25	14.0150	0.6290
Multicrystalline	KC200GT	1000	25	12.8150	0.7690
		1100	25	13.3750	0.7150
		1000	35	13.8630	0.5670
		1000	50	14.3140	0.4120
		1000	60	14.8220	0.3100
		1000	75	15.5870	0.1910
		200	25	17.3600	0.0010
		400	25	17.3600	0.0010
		600	25	17.3600	0.0010
		800	25	17.3600	0.0010
		900	25	17.3600	0.0010
Multicrystalline	CS6P-230P	1000	25	14.0260	0.6950
		1100	25	17.3600	0.0010
		1000	35	12.9077	0.5782
		1000	50	11.4128	0.4036
		1000	60	10.3851	0.3003
		1000	75	10.2626	0.0969
		200	25	49.5010	0.0010
		400	25	45.3600	0.0020
		600	25	20.3580	0.2470
		800	25	27.1920	0.0820
		900	25	24.6810	0.1320
Monocrystalline	CS6X-305M	1000	25	16.6430	0.5120
		1100	25	14.7230	0.6960
		1000	35	24.1600	0.0920
		1000	50	25.1380	0.0290
		1000	60	19.4160	0.0550
		1000	75	24.7030	0.0030
		200	25	6.3840	1.7780
		400	25	25.2530	0.0170
		600	25	27.0850	0.0120
		800	25	20.5150	0.0630
		900	25	14.6450	0.2470
CIGS Thin-film	Q.SMART UF L100	1000	25	7.6070	1.0300
		1100	25	9.7650	0.7040
		1000	35	13.4140	0.2630
		1000	50	16.6570	0.0780
		1000	60	12.2960	0.1830
		1000	75	19.1480	0.0130
		200	25	4.8460	1.5260
		400	25	4.3930	1.7690
		600	25	4.2110	1.8880
		800	25	4.1200	1.9620
		900	25	4.0760	1.9890
Hybrid Thin-film	U-EA110	1000	25	4.0920	1.9860
		1100	25	4.0240	2.0390
		1000	35	4.1940	1.7670
		1000	50	4.4240	1.3950
		1000	60	4.5650	1.1660
		1000	75	4.7630	0.8410
		200	25	52.9030	0.0010
		400	25	45.0450	0.0070
		600	25	39.3940	0.0230
		800	25	32.6930	0.0750
		900	25	20.2780	0.4920
Ultra-thin Amorphous	VBHN330SA16	1000	25	15.3770	0.9670
		1100	25	18.1660	0.6860
		1000	35	21.7280	0.3700
		1000	50	14.7390	0.8630
		1000	60	21.7900	0.2740
		1000	75	26.2430	0.1060
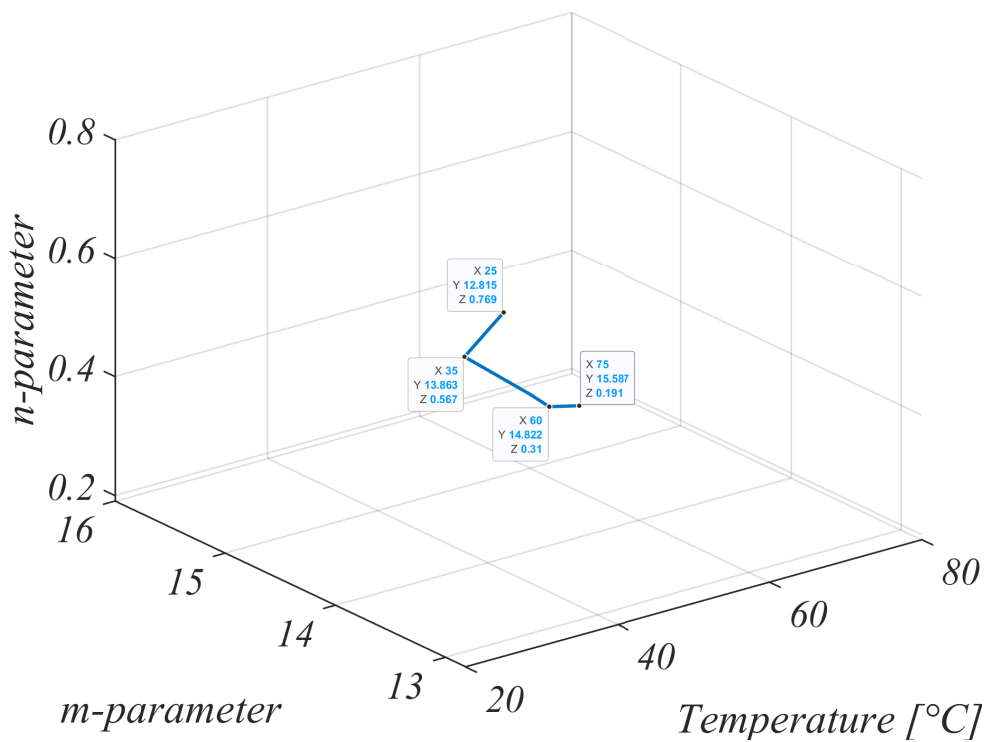
Fig. 4.7 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the KC200GT PV panel — (a) varying irradiance (b) varying temperature.
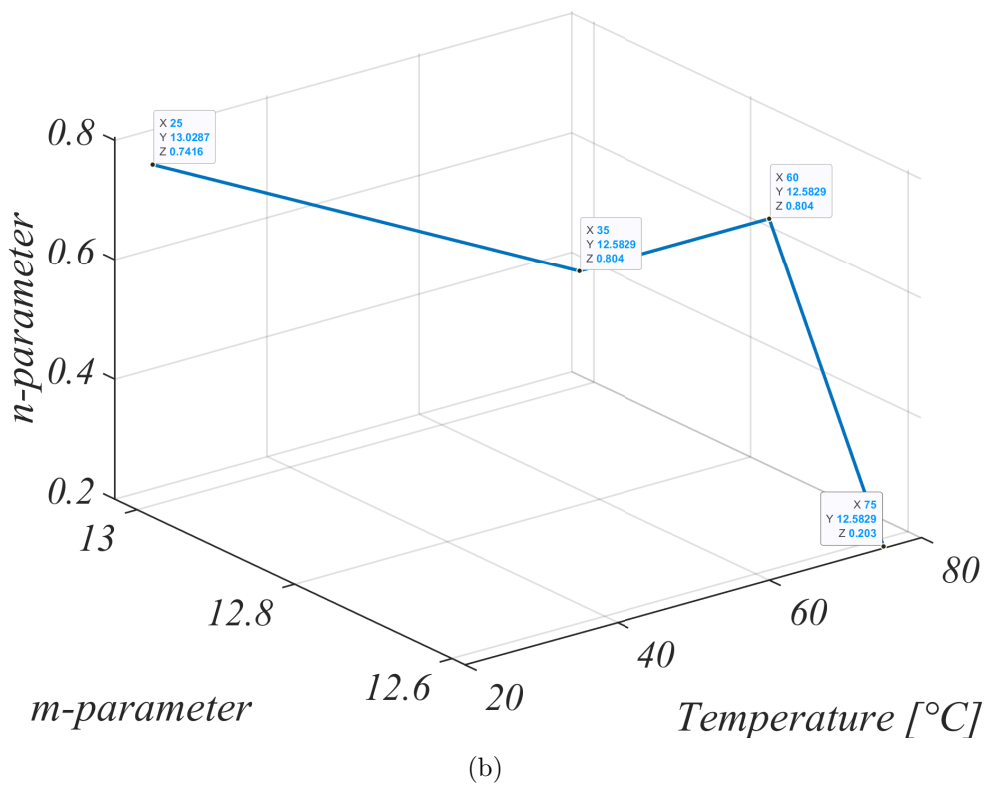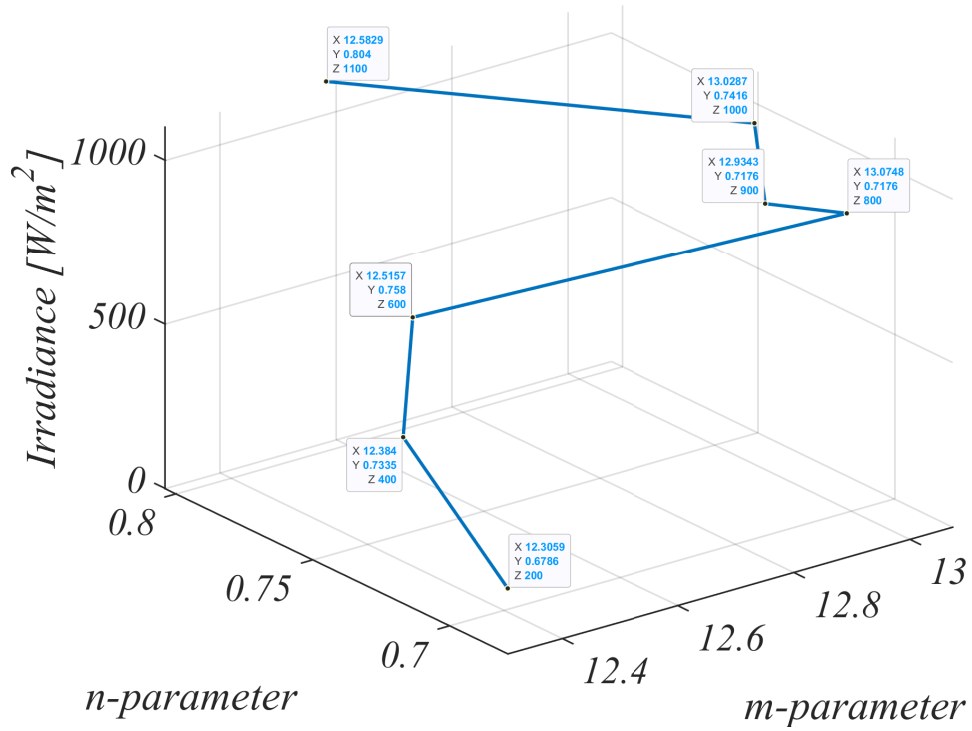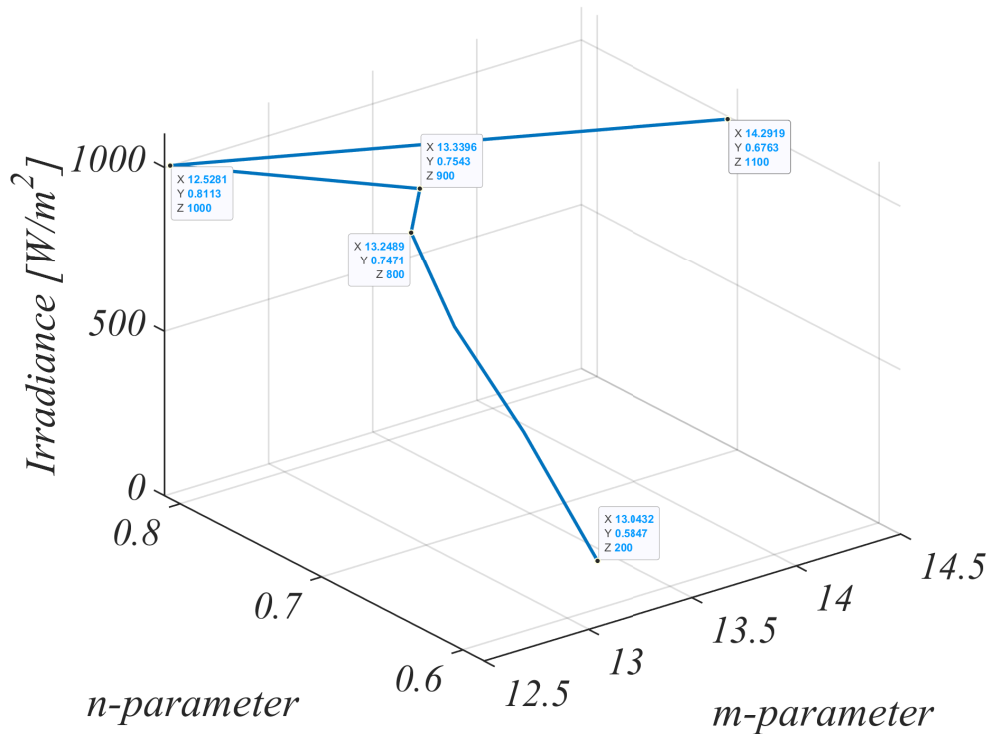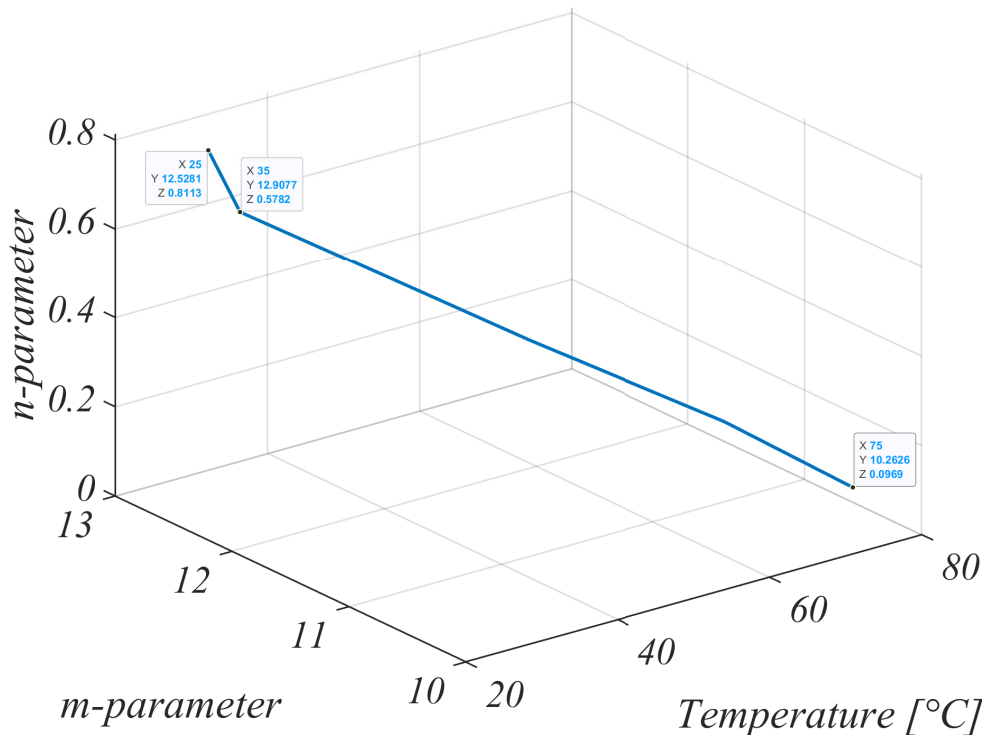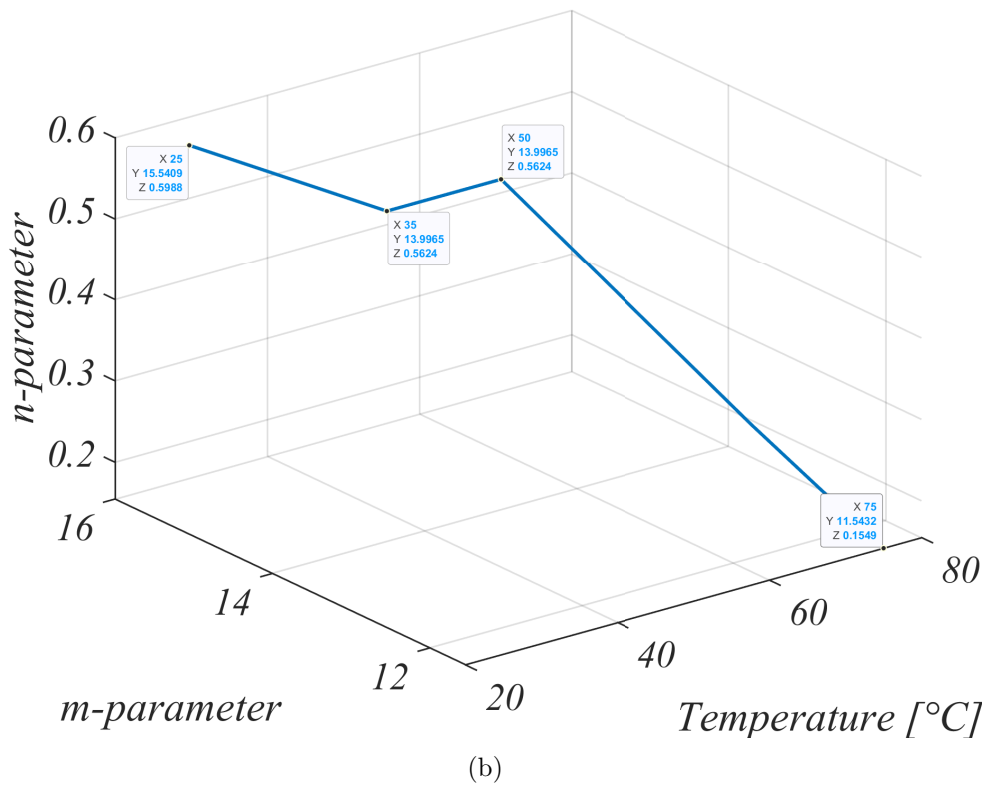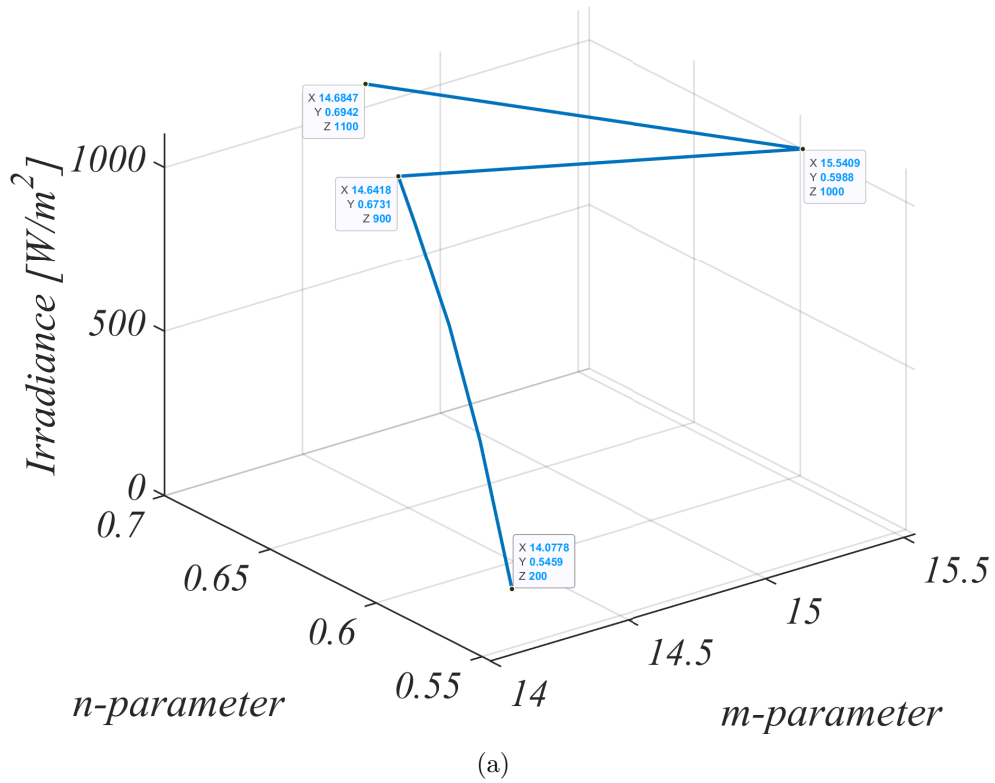
(a)

(b)

Fig. 4.8 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the CS6P-230P PV panel — (a) varying irradiance (b) varying temperature.

51

Table 4.5 Parameter extraction of the superellipse model for the 6 different PV panels under varying ambient conditions using the NR algorithm.

Cell Material	PV Panel	Ambient Condition		Model Parameters	
		Irradiance	Temperature	m	n
Multicrystalline	KC200GT	200	25	2.7740	7.5260
		400	25	1.7790	0.3060
		600	25	-32.7670	-6.1950
		800	25	14.3980	0.5820
		900	25	-32.7670	-6.1950
		1000	25	12.7941	0.7734
		1100	25	-32.7670	-6.1950
		1000	35	13.8440	0.5690
		1000	50	118.5700	-0.0020
		1000	60	14.6830	0.3190
		1000	75	15.3170	0.2030
Multicrystalline	CS6P-230P	200	25	13.0210	0.6010
		400	25	191.8190	62.4330
		600	25	230.6190	63.8990
		800	25	4.4720	14.5220
		900	25	49.6640	26.0130
		1000	25	14.0435	0.6926
		1100	25	1.1240	-6.2320
		1000	35	16.6270	0.3000
		1000	50	99.9990	-0.0020
		1000	60	92.7400	0.0010
		1000	75	74.9010	0.0120
Monocrystalline	CS6X-305M	200	25	3.9890	17.3520
		400	25	-8.2660	-0.0180
		600	25	1.1810	-5.0230
		800	25	4.0080	-5.9550
		900	25	4.0690	-6.2830
		1000	25	16.6510	0.5174
		1100	25	4.1650	-6.8630
		1000	35	-80.2700	-10.4370
		1000	50	-2.5350	-0.0230
		1000	60	73.8550	245.1940
		1000	75	87.3780	0.0150
CIGS Thin-film	Q.SMART UF L100	200	25	6.4650	1.7490
		400	25	88.6320	-0.0010
		600	25	-2839.3850	0.0030
		800	25	-3.4020	51.2080
		900	25	95.0810	-0.0020
		1000	25	7.5611	1.0372
		1100	25	9.9200	0.6820
		1000	35	-122.7050	-262.8810
		1000	50	87.2540	-0.0010
		1000	60	79.4540	-0.0010
		1000	75	70.6740	-0.0010
Hybrid Thin-film	U-EA110	200	25	4.7360	1.5670
		400	25	4.3710	1.7760
		600	25	4.2050	1.8870
		800	25	4.1110	1.9600
		900	25	4.0790	1.9890
		1000	25	3.8084	2.0840
		1100	25	4.0310	2.0360
		1000	35	4.1970	1.7640
		1000	50	4.4220	1.3970
		1000	60	4.5670	1.1640
		1000	75	4.7590	0.8420
Ultra-thin Amorphous	VBHN330SA16	200	25	133.3810	-0.0020
		400	25	4.9010	16.3890
		600	25	5.1450	16.1760
		800	25	5.2780	16.0930
		900	25	5.3270	16.0690
		1000	25	15.4235	0.9630
		1100	25	5.4050	16.0390
		1000	35	5.2330	16.0490
		1000	50	5.0310	16.0450
		1000	60	-908.8850	-0.0010
		1000	75	3.9840	-5.6690

(a)

(b)

Fig. 4.9 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the CS6X-305M PV panel — (a) varying irradiance (b) varying temperature.
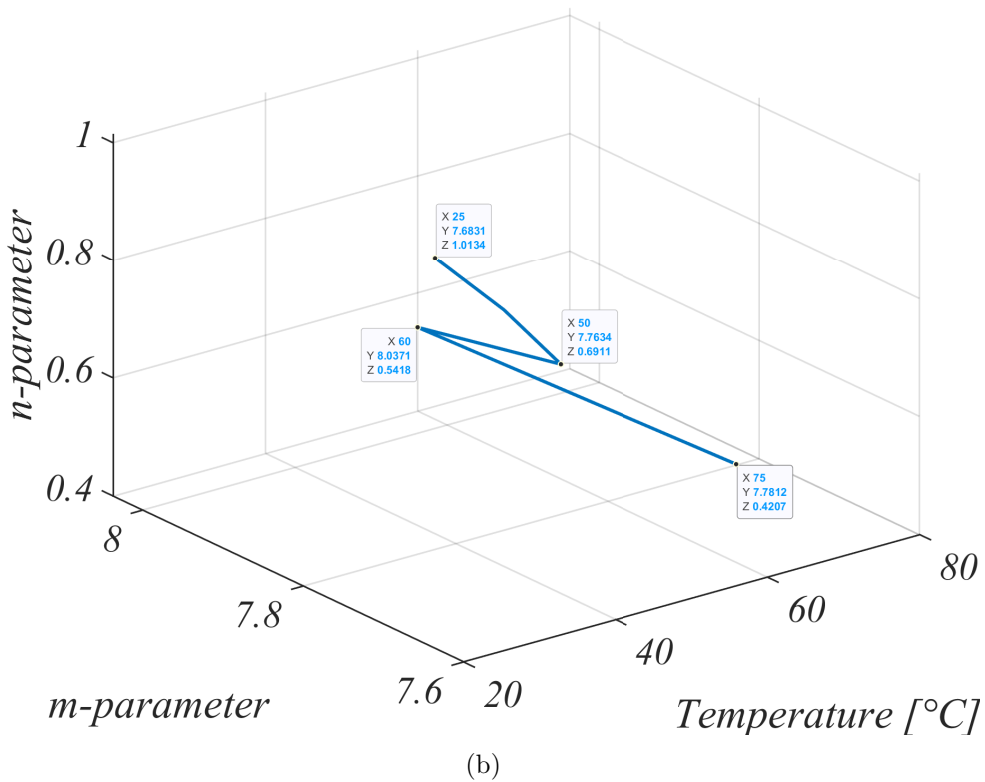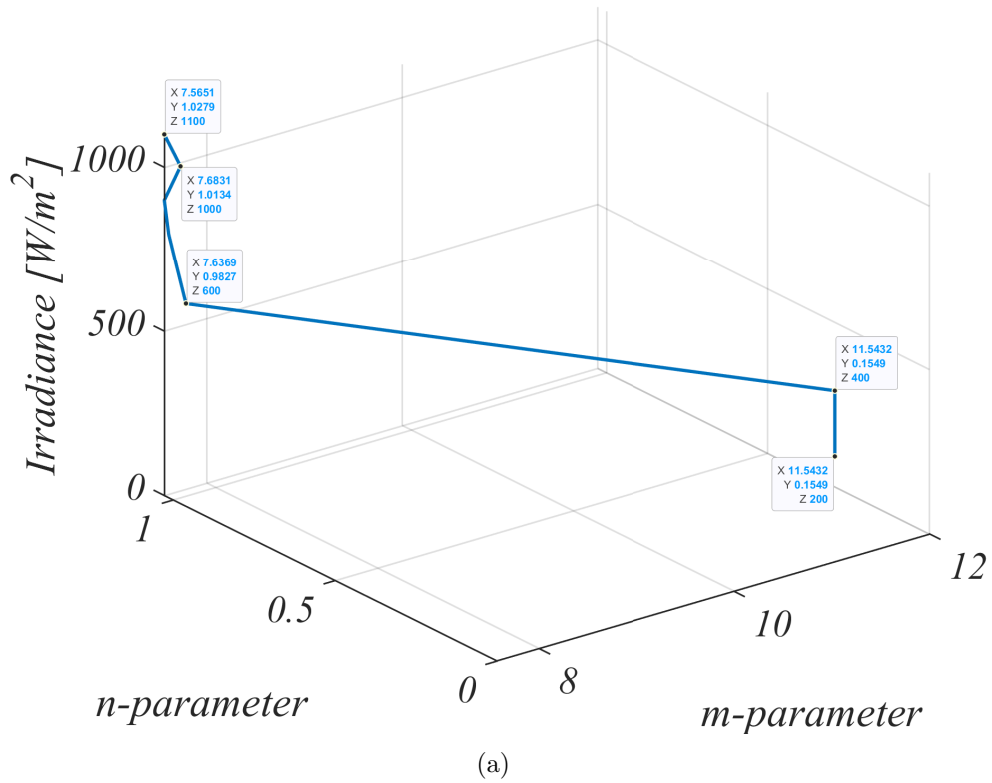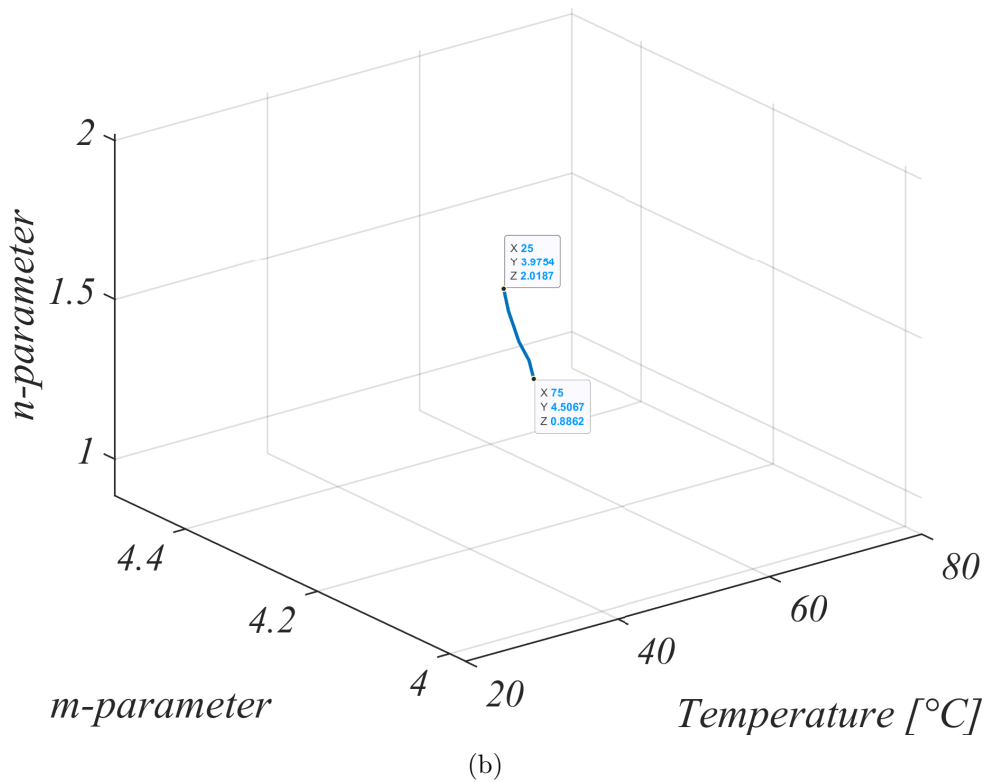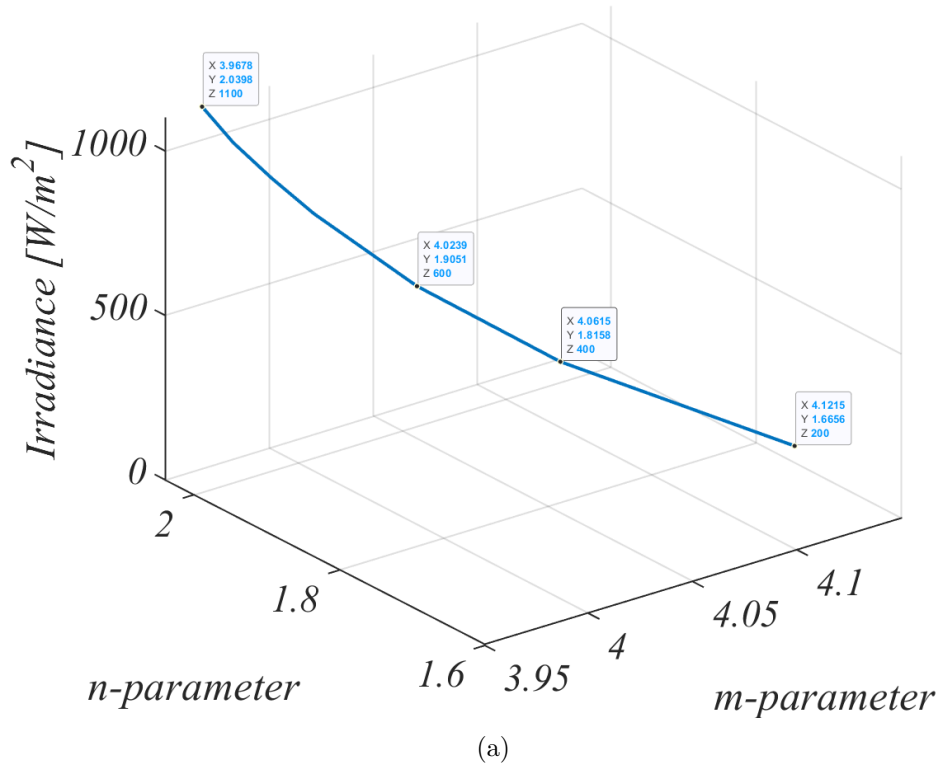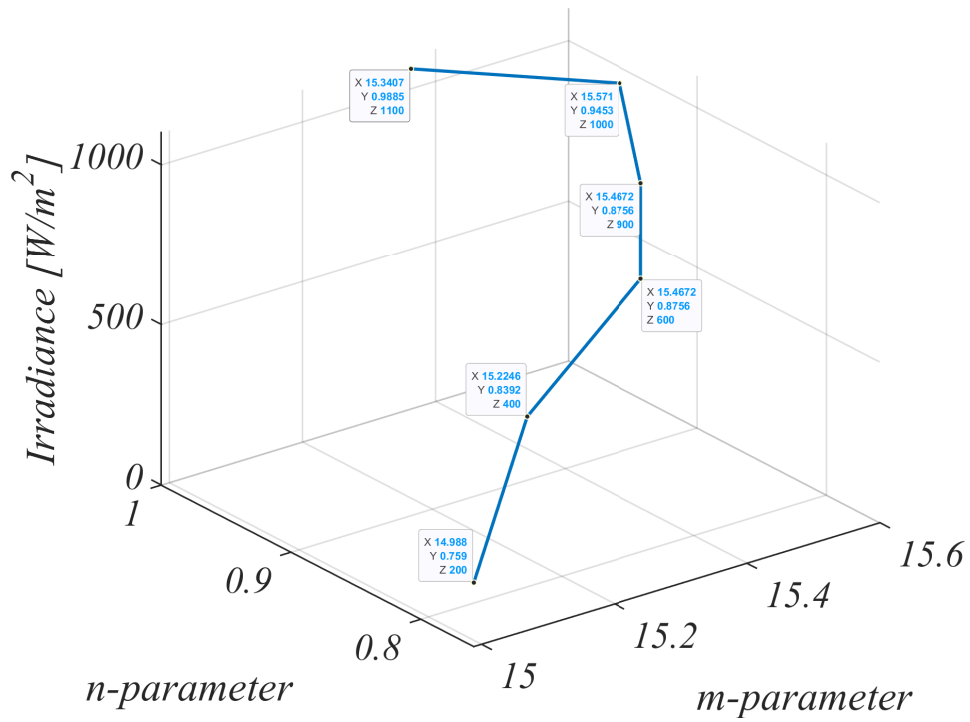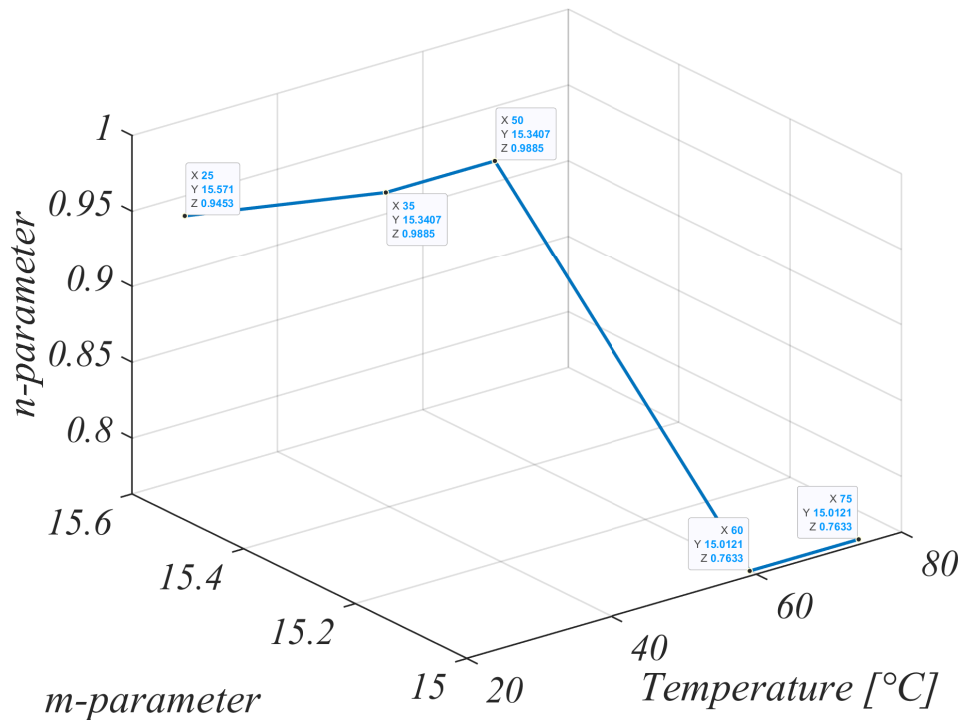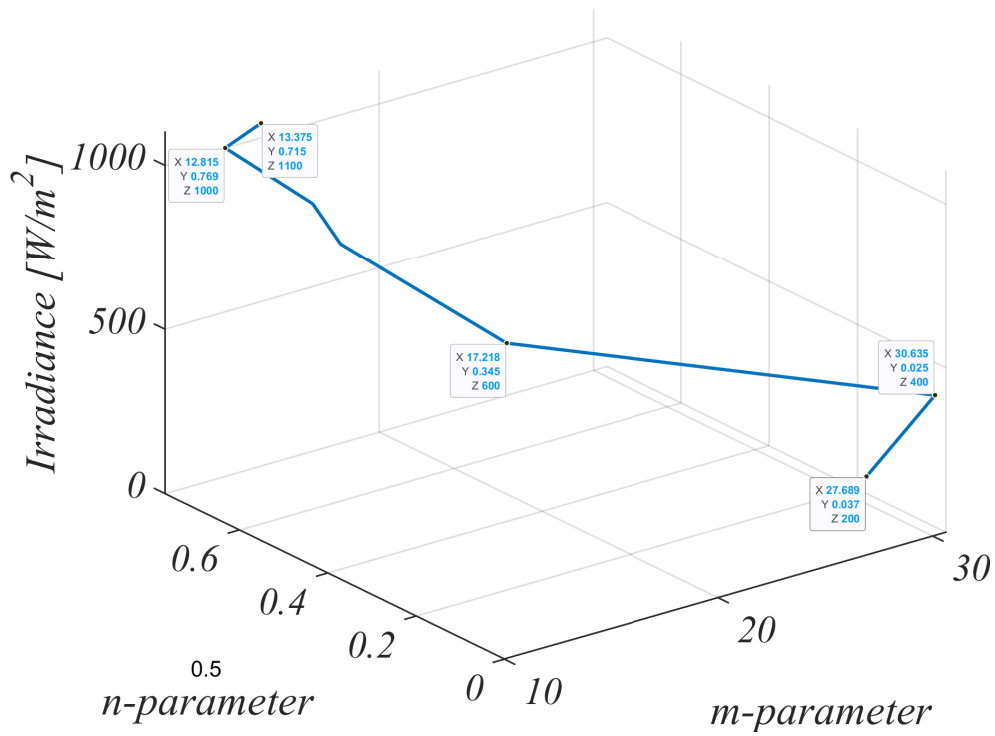
(a)

(b)

Fig. 4.10 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the UF L100 PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.11 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the U-EA110 PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.12 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using PM algorithm for the VBHN330SA16 PV panel — (a) varying irradiance (b) varying temperature.

Fig. 4.13 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the KC200GT PV panel — (a) varying irradiance (b) varying temperature.

Fig. 4.14 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the CS6P-230P PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.15 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the CS6X-305M PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.16 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the UF L100 PV panel — (a) varying irradiance (b) varying temperature.

(a)

(b)

Fig. 4.17 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the U-EA110 PV panel — (a) varying irradiance (b) varying temperature.
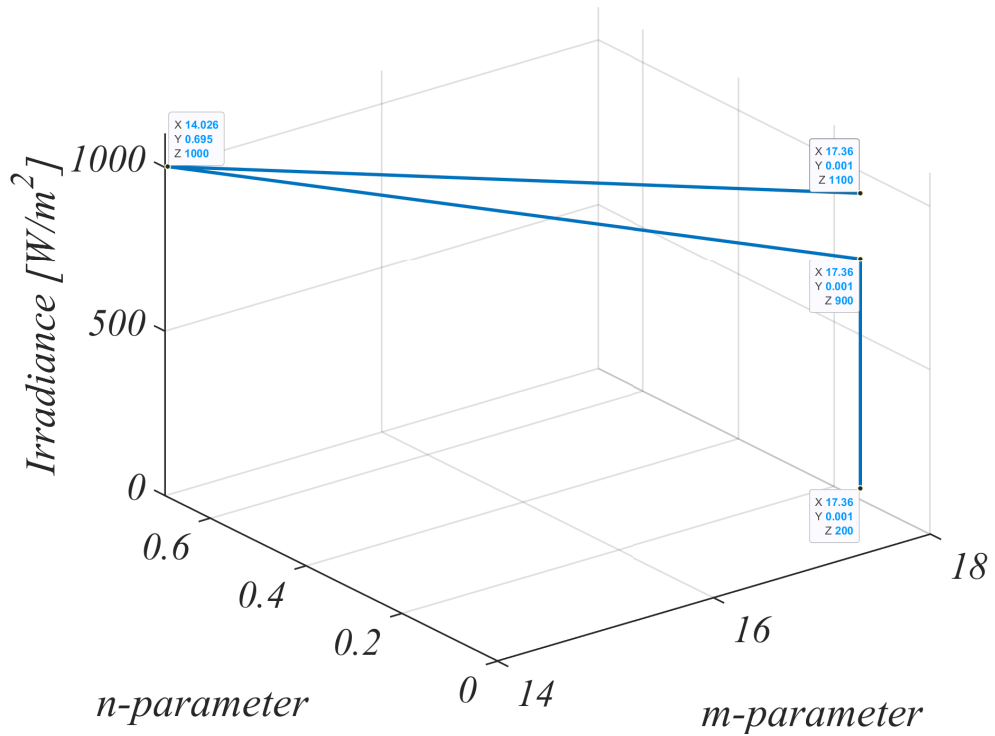
(a)

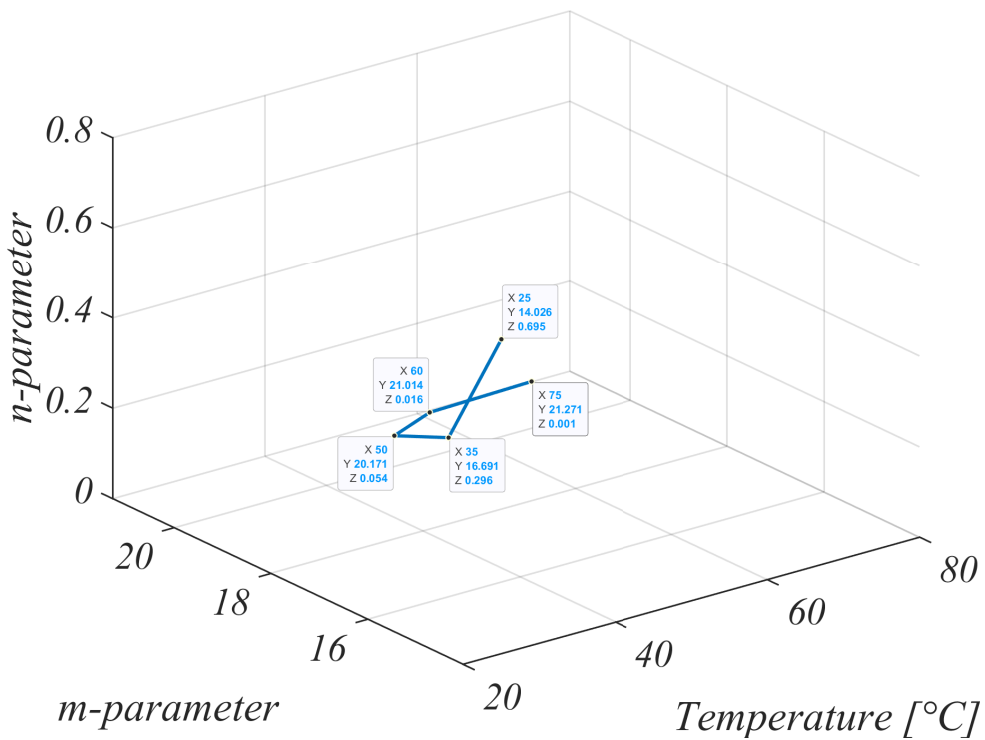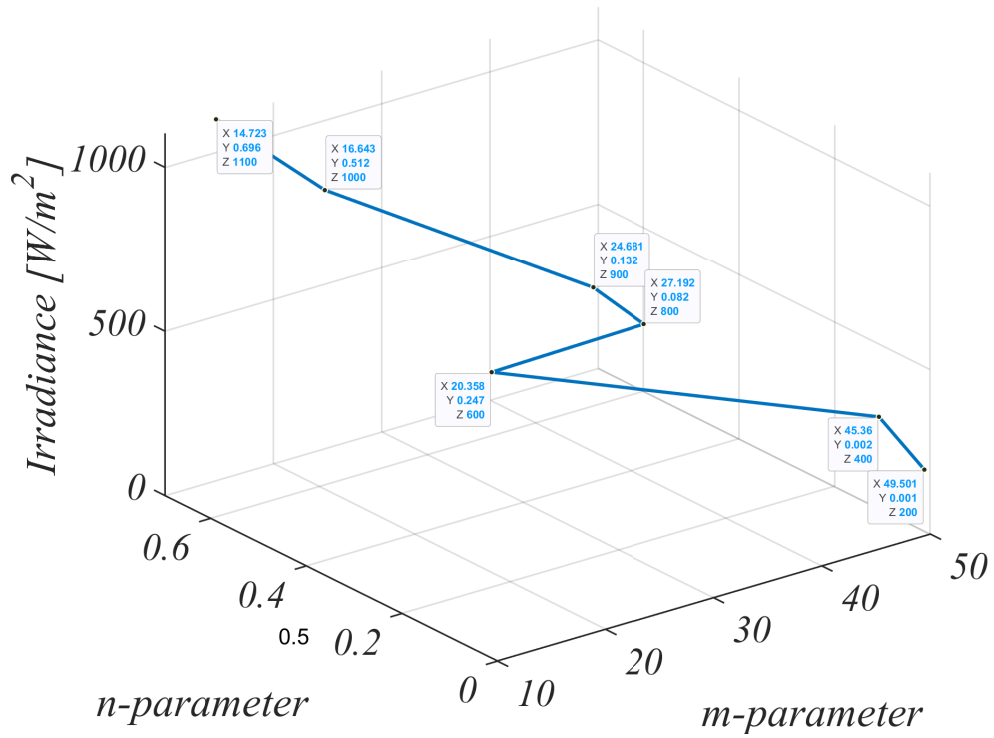(b)

Fig. 4.18 3–D plot of the effect of varying ambient condition on the parameter extraction of the superellipse model using NR algorithm for the VBHN330SA16 PV panel — (a) varying irradiance (b) varying temperature.

Chapter 5

Accuracy evaluation of the superellipse model

5.1 Criteria for evaluating model accuracy

Several methods have been utilized in literature for evaluating the reconstructed PV characteristic curves across its full range and its MPP. In the thesis, the reconstructed curves would be evaluated using three different criteria – absolute error (AE), relative error (RE), and the IEC EN 50530 standard.

The IEC EN 50530 standard has published by the European Committee for Electrotechnical Standardization (CENELEC) provides a set of requirements and guidelines for the design, qualification and approval of PV generators (see Appendix H). To meet its reliability and durability requirements, the absolute current and power errors of the any approximate PV characteristic curve within the vicinity of $\pm 10\%$ of the PV panel's V_{mp}, should always be

less than or equal to 1%. In accordance with this standard, the mathematical equation for

computing these absolute errors can therefore be expressed as [41]

$$\varepsilon_I(\%) = \frac{1}{0.2V_{mp}} \int_{V_{mp}\pm10\%} \left| \frac{i_s(v) - i_r(v)}{i_r(v)} \right| dv \times 100 \tag{5.1a}$$

$$\varepsilon_P(\%) = \frac{1}{0.2V_{mp}} \int_{V_{mp}\pm10\%} \left| \frac{p_s(v) - p_r(v)}{p_r(v)} \right| dv \times 100. \tag{5.1b}$$

where the subscript s represents the measured values of the approximate curves, and r

denotes the data values from the reference model. To compute these integral values, the

Trapezoidal rule [42, 43] is used such that

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f(a+ih) + f(b)\right] \tag{5.2}$$

where $f(x)$ is $\frac{i_s(v)-i_r(v)}{i_r(v)}$ in (5.1a) and $\frac{p_s(v)-p_r(v)}{p_r(v)}$ in (5.1b), $a = -10\%$ of V_{mp}, $b = +10\%$ of

V_{mp}, $h = \frac{b-a}{2n}$ and n is the number of segments within the integral. An example MATLAB

code for computing these absolute error at MPP using the IEC EN 50530 standard for the

KC200GT PV panel with $n = 50$ is given in Appendix I.

In addition, the mathematical expression for computing the *AE* and *RE* values across

the full range of PV characteristic curves can also be expressed mathematically as [44]

$$AE = \left| X_r - X_s \right| \tag{5.3a}$$

$$RE = abs\left(\frac{X_r - X_s}{X_r}\right) \tag{5.3b}$$

where X represents the data points across the PV characteristic curves.

Table 5.1 Classification of the 14 PVM equations used in this thesis.

Proposed Model	Conventional Single-diode Model	Approximate Single-diode Model
Superellipse	Simulink circuit-based	Padé approximant
	Simulink mathematical-based	Newton-Raphson
		MATLAB (Haley's)
		Hybrid Explicit Expansion
		Fixed Point Iteration
		Barry Analytical Expansion
		Wintzki Approximation
		3-point model
		Two-port network Expansion
		Two-parameter model
		Taylor's series Expansion
		Bezier curve

In keeping with most research works in literature, the performance analysis of the newly proposed empirical model is carried out using MATLAB/Simulink in an 11th Gen Intel(R) Core(TM) i9–11900K CPU. As such, by utilizing the manufacturer's datasheet (see Appendix G) as the reference, the PV characteristic curves of the superellipse model and 14 other conventional PVM equations are evaluated (see Table 5.1).

First, the accuracy of the 14 PVM equations including the proposed empirical model are evaluated at STC by the IEC EN 50530 standard and across the full range of the reconstructed curves for the KC200GT PV panel in Chapter 5.2. Next, the superellipse model are then employed to reconstruct the PV characteristic curves for the other 5 PV panels, with their accuracy being evaluated at MPP and across the full range in Chapter 5.3. Finally, in Chapter 5.4, by reconstructing the PV characteristic curves under varying ambient conditions, the superiority of the superellipse model are further verified.

5.2   Comparison with other PVM equations at STC

5.2.1   Comparison at MPP using the IEC EN 50530 standard

To validate the superiority of the superellipse model over the conventional PVM equations, the KC200GT PV panel is utilized. By direct substitution of its fitting parameters, as obtained using DFP from Table 4.2 into (3.3), the data points required for the full range reconstruction of the I–V curve can be easily obtained.

Although most recent PVM equations in literature maintain the IIEC EN 50530 standard at MPP, the absolute errors in the older techniques – Newton-Raphson [45], MATLAB [46, 47], Winitzki approximation [12, 15], 3 point model [21] are extremely low as shown in Fig. 5.1. Due to the robustness of the multidimensional equation in (3.11) and the befitting optimization algorithm, the superellipse model achieves a very high model accuracy which approaches 0 % as shown in Table 5.2.

Other techniques such as the Barry analytical expansion [17], Taylor's series expansion [48], Padé approximant [23, 25] and the Simulink-based techniques [49, 50] are also accurate as they also maintain 1% threshold as specified by the IEC EN 50530 standard (see Fig. 5.2). As shown by the 3–D plot of these four PVM equations, the superellipse model has the closest values with the MPP values with that experimental data curves in Fig. 5.3.

5.2.2   Comparison across the full range of the PV characteristic curves

By applying (5.3), the absolute and relative error across the full range of the 14 reconstructed PV characteristic curves are computed as shown in Figs. 5.4 and 5.5. As expected, it the error margins specifically across the I–V characteristic curves are significantly low. To further validate this statements, the mean error (ME), mean square error (MSE),
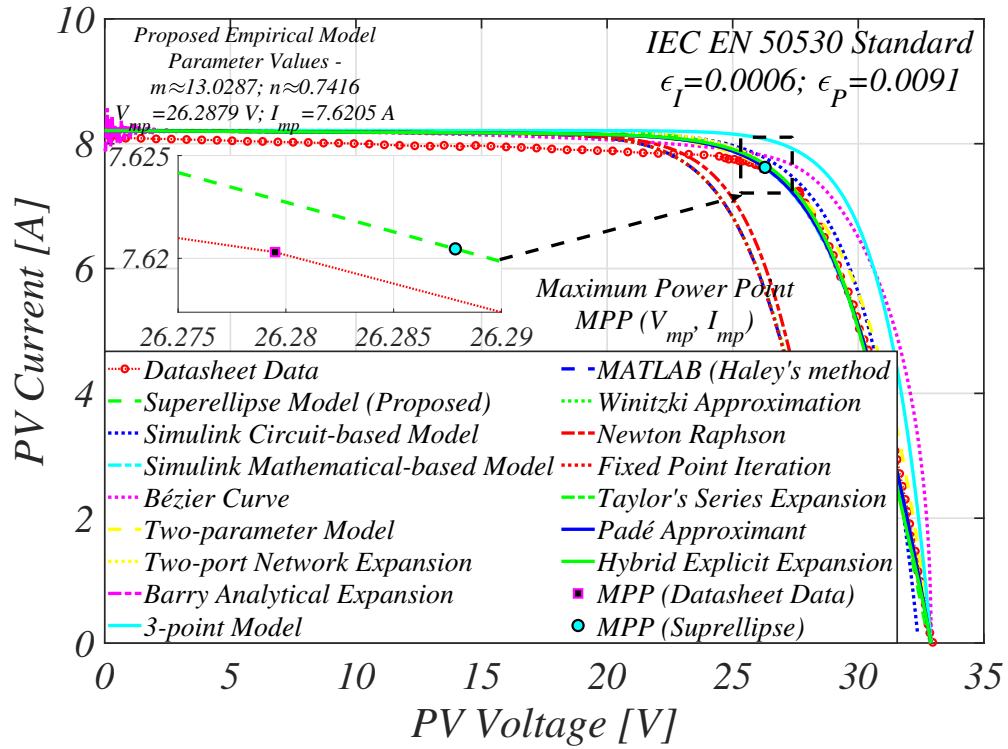
(a)
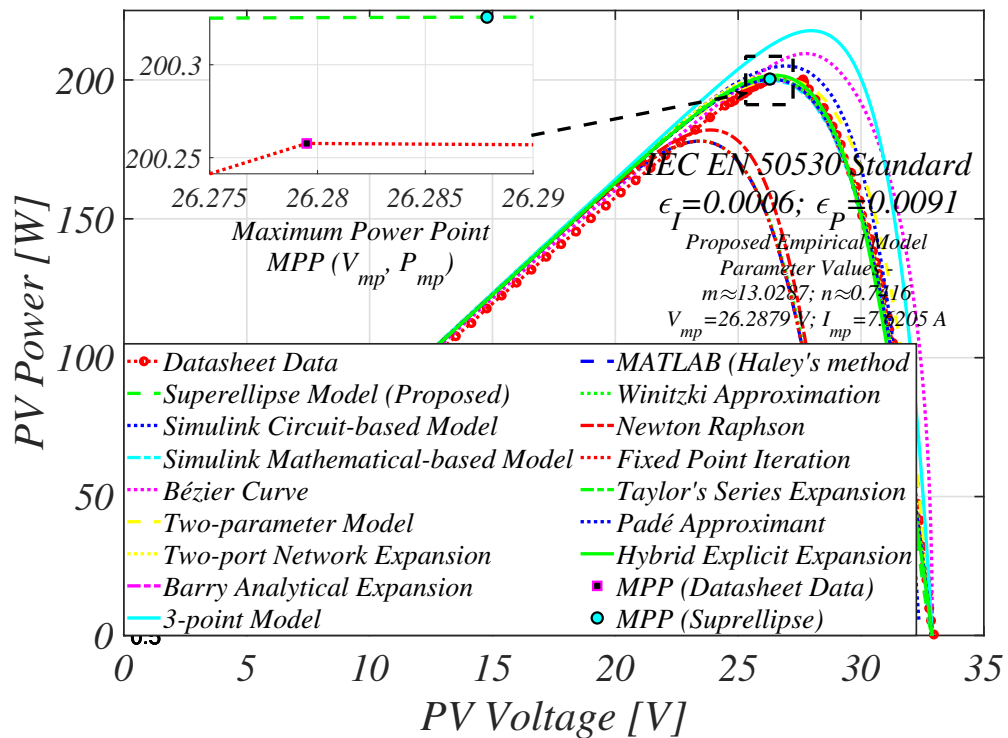
(b)

Fig. 5.1 Approximation of the PV characteristic curves using the proposed model and 14 conventional PVM equations using the KC200GT PV panel (a) I–V curve (b) P–V curve.
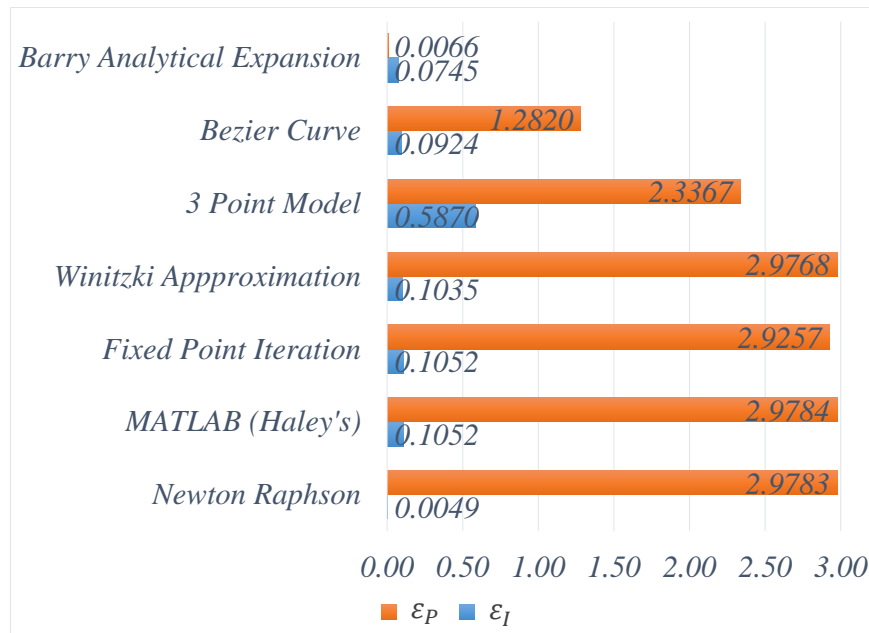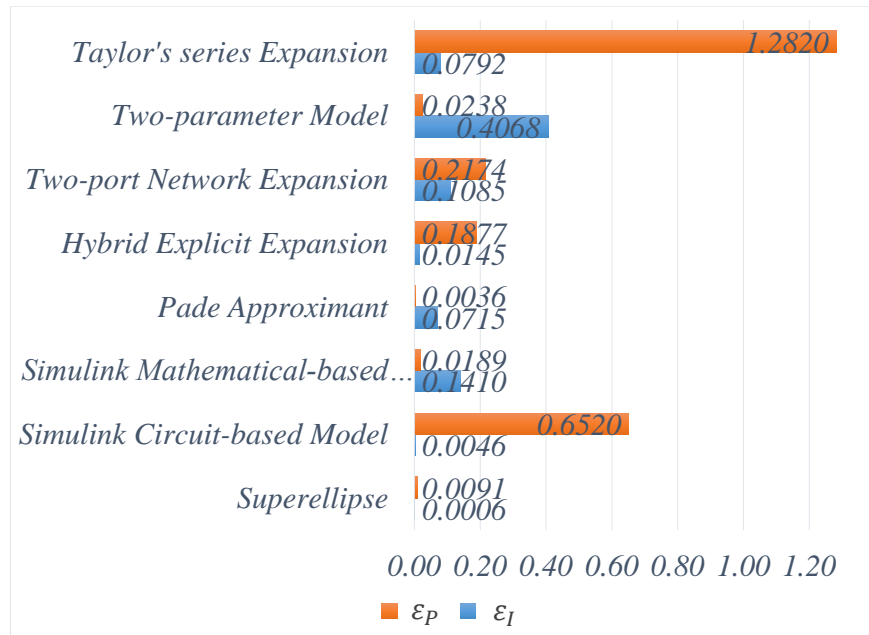
Fig. 5.2 Accuracy evaluation of the 15 different PVM equations in accordance with the IEC EN 50530 standard for KC200GT PV panel.

(a)

(b)
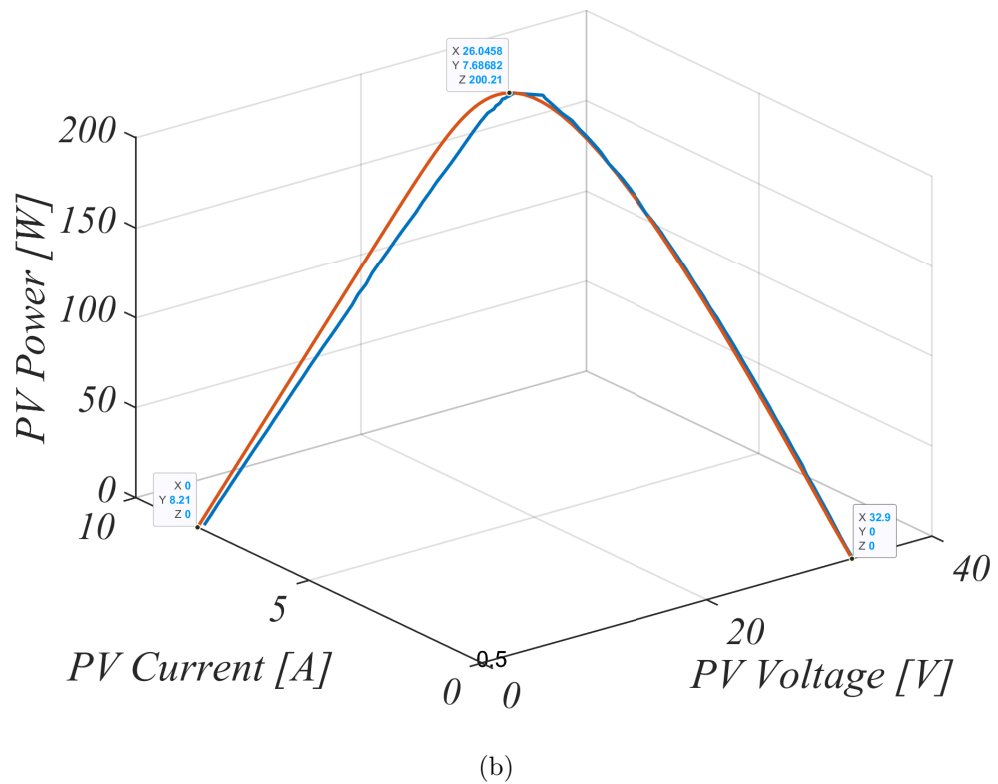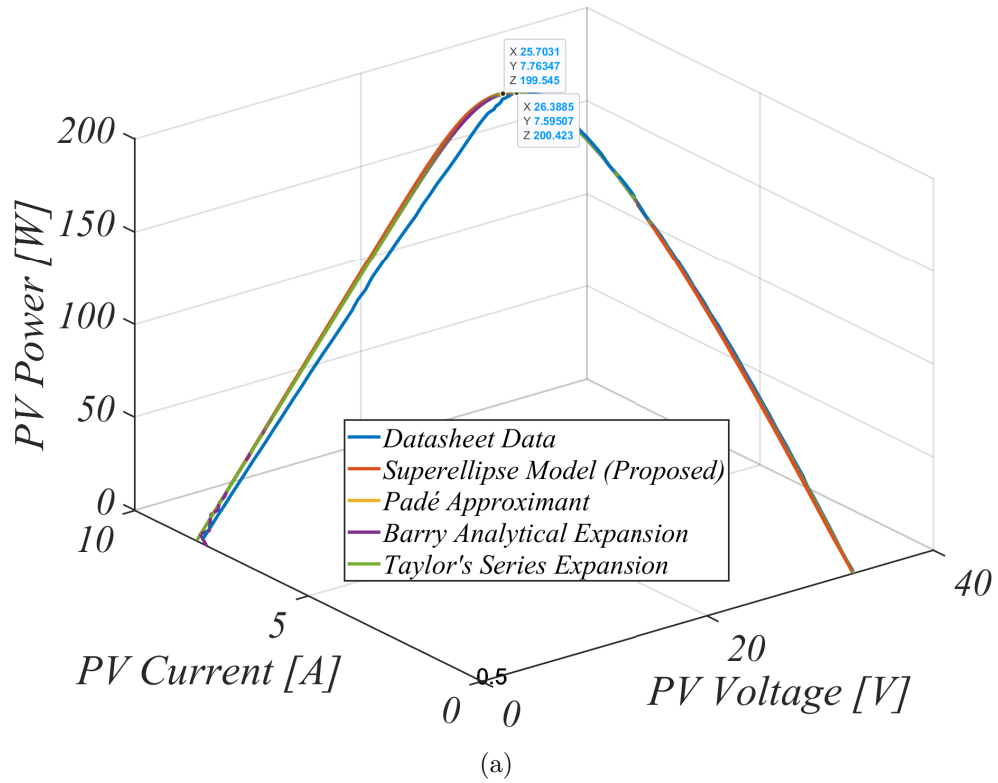
Fig. 5.3 3–D plot of the approximate PV characteristic curves for KC200GT PV panel (a) 4 different PVM equations (b) superellipse model.

Table 5.2 Comparision of 15 different PVM equations for KC200GT PV panel within the vicinity of MPP.

Method	Reference	V_{mp} (V)	I_{mp} (A)	P_{mp} (W)	ε_I (%)	ε_P (%)
Datasheet Data		26.2795	7.6203	200.2577		
Superellipse Model (Proposed)		26.2879	7.6205	200.3260	0.0006	0.0091
Simulink Circuit-based Model	[49]	26.9240	7.6190	205.1341	0.0045	0.6520
Simulink Mathematical-based Model	[50]	26.4000	7.5802	200.1165	0.1392	0.0189
Padé Approximant	[23, 25]	26.3463	7.5999	200.2306	0.0705	0.0036
Newton Raphson	[45]	23.8764	7.6217	181.9785	0.0048	2.9783
MATLAB (Haley's)	[46, 47]	23.4482	7.5903	177.9804	0.1034	2.9784
Hybrid Explicit Expansion	[14]	26.4781	7.6162	201.6613	0.0143	0.1877
Fixed Point Iteration	[51]	23.4482	7.5904	177.9807	0.1033	2.9257
Barry Analytical Expansion	[17]	26.3463	7.5991	200.2083	0.0732	0.0066
Winitzki Approximation	[12, 15]	23.4482	7.5908	177.9917	0.1021	2.9768
3 Point Model	[21]	27.9601	7.7874	217.7352	0.5766	2.3367
Two-Port Network Expansion	[26]	26.3860	7.6512	201.8837	0.1065	0.2174
Two-Parameter Model	[28, 52]	26.7086	7.5045	200.4354	0.3996	0.0238
Taylor's Series Expansion	[48]	26.3793	7.5977	200.4230	0.0781	0.0221
Bézier curve	[44]	27.4431	7.6466	209.8464	0.0908	1.2820

Table 5.3 Comparison of the error across the full range of the reconstructed PV characteristic curves for the KC200GT PV panel.

Method	I—V Curve			P—V Curve		
	ME	MSE	$RMSE$	ME	MSE	$RMSE$
Superellipse Model (Proposed)	1.4470	4.4112	2.1003	53.3742	3.8054×10^3	61.6880
Padé Approximant	1.4384	4.4038	2.0985	53.9231	3.8635×10^3	62.1568
Barry Analytical Expansion	1.4370	4.4000	2.0976	53.9073	3.8604×10^3	62.1321
Taylor's Series Expansion	4.4179	4.4179	2.1019	53.9715	3.8731×10^3	62.2342

Table 5.4 Accuracy of the proposed method for 6 different PV panels in accordance with the IEC EN 50530 standard.

PV Panel	Datasheet Data			Model Parameters		Superellipse Approximation			IEC EN50530 Standards	
	V_{mp}	I_{mp}	P_{mp}	m	n	V_{mp}	I_{mp}	P_{mp}	ε_I	ε_P
KC200GT	26.2795	7.6203	200.2577	13.0287	0.7416	26.2879	7.6205	200.3256	0.0006	0.0091
CS6P-230P	29.5194	7.7717	229.4159	13.7090	0.7286	29.5928	7.7688	229.8993	0.0114	0.0635
CS6X-305M	36.6685	8.3195	305.0623	16.6430	0.5120	36.6015	8.3321	304.9664	0.0564	0.0117
Q.SMART UF L100	70.1776	1.4073	98.7609	7.5611	1.0372	69.3784	1.4404	99.9358	1.6365	0.8405
U-EA110	54.7698	2.0200	110.0350	3.8084	2.0840	54.0140	2.0287	109.5765	0.2061	0.5259
VBHN330SA16	58.6183	5.6738	332.5882	15.4235	0.9630	58.6183	5.6738	332.5882	0.2641	0.3507

and root mean square error ($RSME$) (see Appendix J)of 4 PVM equations in Fig. 5.3a were also calculated. It can be observed that the superellipse model achieves the lowest error margins as shown in Fig. 5.6 and outlined in Table 5.3.

5.3   Application in different PV panels

5.3.1   Comparison at MPP using the IEC EN 50530 standard

Similarly, based on the explicit equation describing the superellipse model under uniform conditions (3.3), and utilizing the extracted parameters for the DFP algorithm, the appropriate optimization algorithm, the reconstructed PV characteristic curves for five other PV panels are examined as shown in Figs. 5.8 — 5.12. The model accuracy of these approximate PV curves was subsequently evaluated by the IEC EN 50530 standard as shown in Table 5.4 while keeping the datasheet data as a reference. Irrespective of PV panel cell material and specifications, the superellipse model is very accurate thereby maintaining the 1% absolute error threshold within the vicinity of MPP as shown in Fig. 5.7.
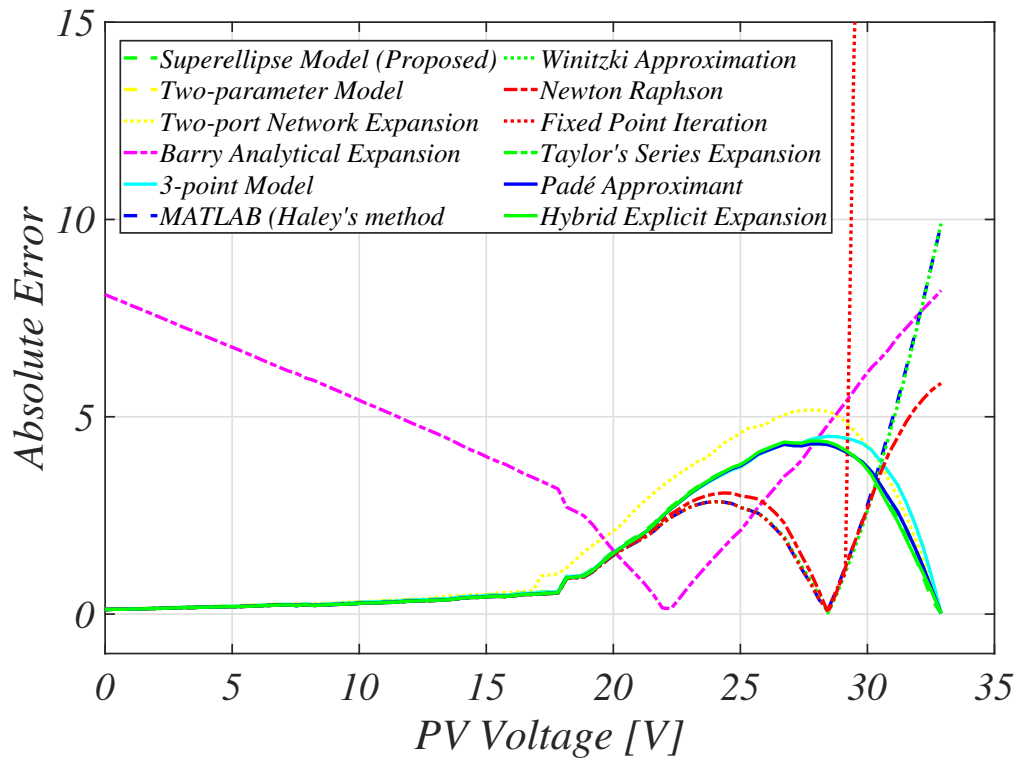
(a)
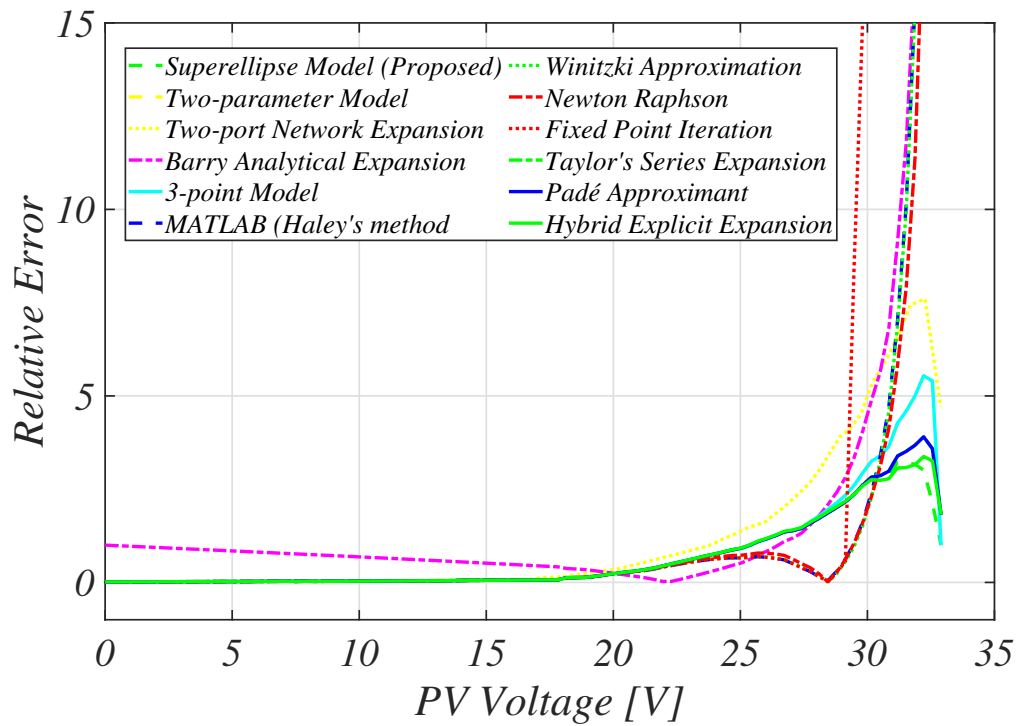
(b)

Fig. 5.4 A plot of the error across the full range of the reconstructed I—V curve using the KC200GT PV panel datasheet as reference (a) absolute error (b) relative error.
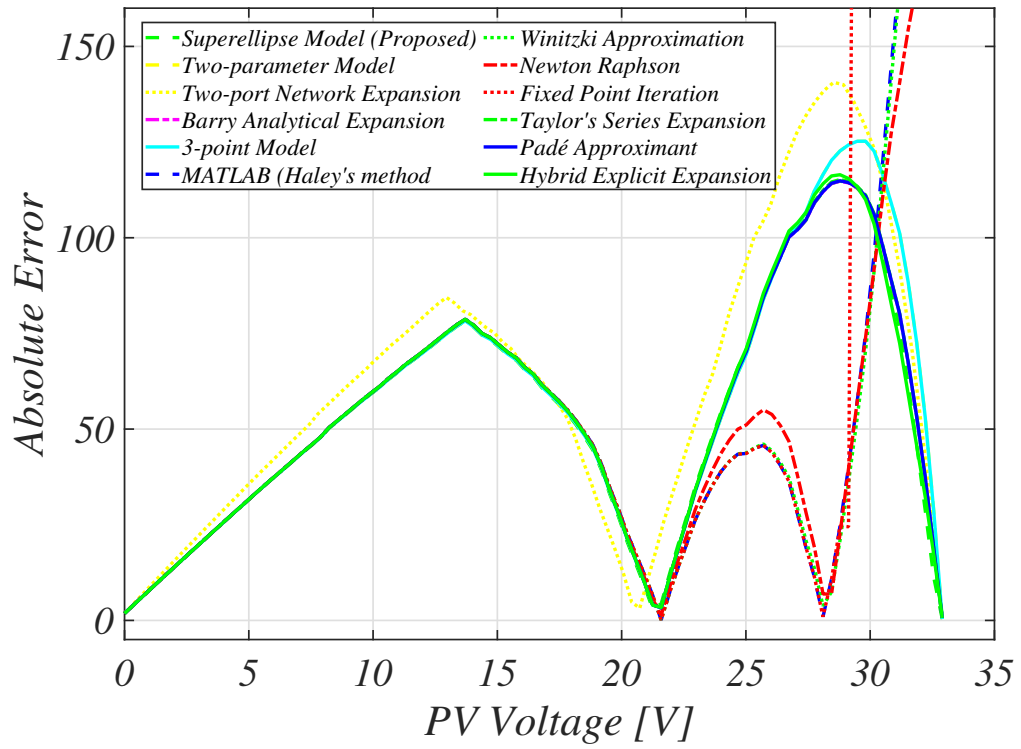
(a)
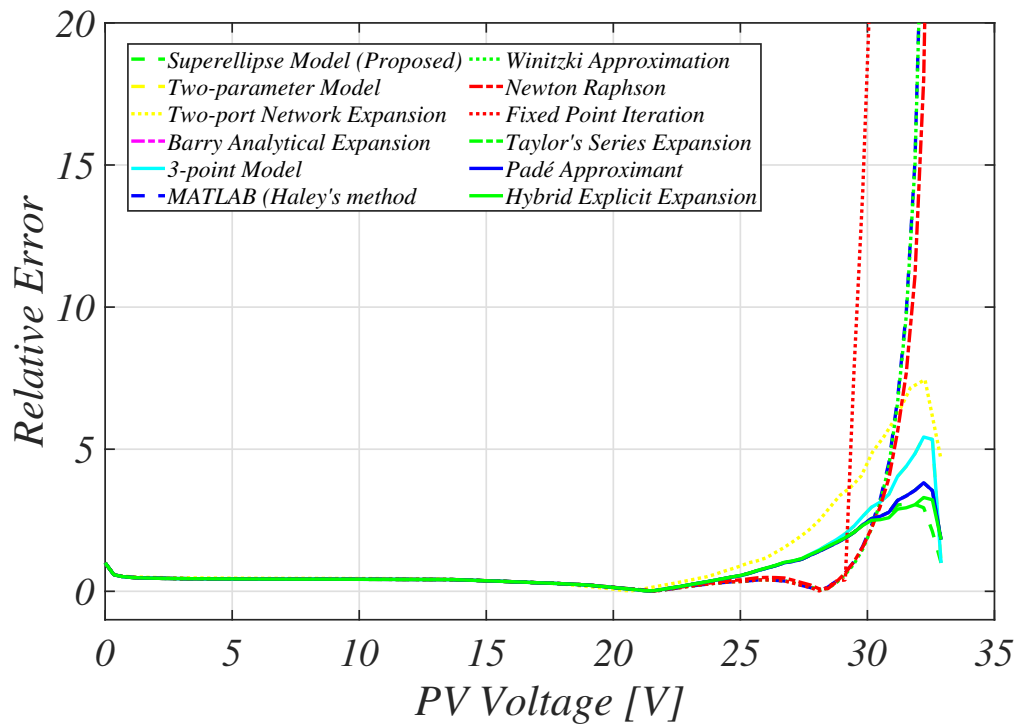
(b)

Fig. 5.5 A plot of the error across the full range of the reconstructed P—V curve using the KC200GT PV panel datasheet as reference (a) absolute error (b) relative error.

73

(a)

(b)
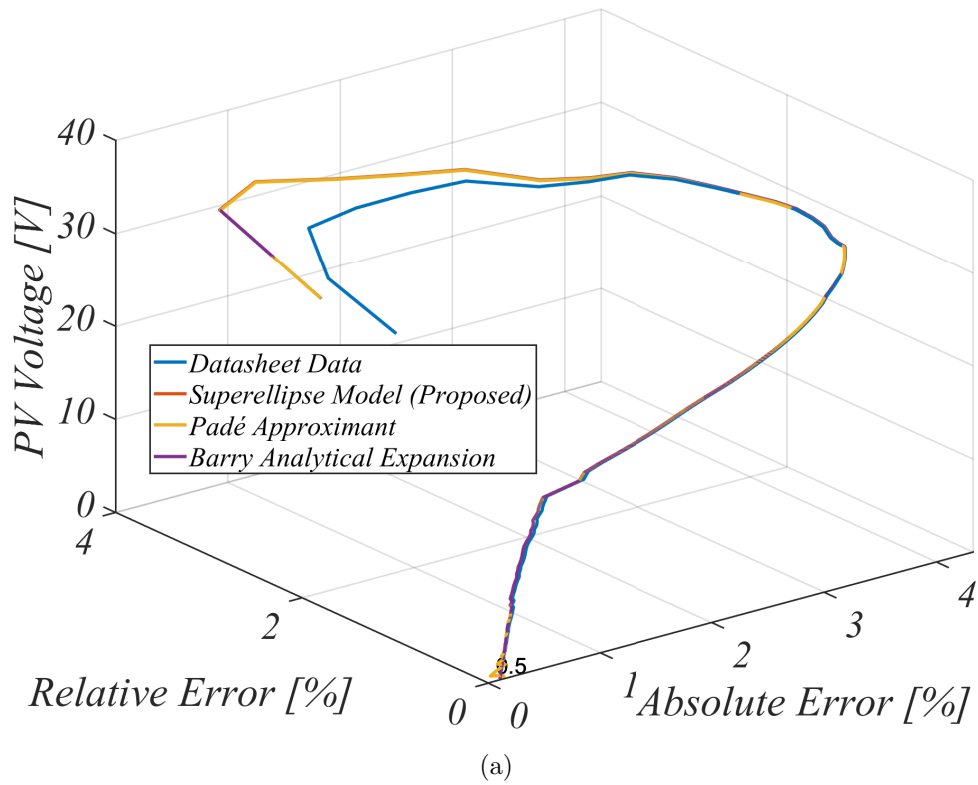
Fig. 5.6 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curves using the experimental data for the KC200GT PV panel as reference (a) I–V curve (b) P–V curve.

74
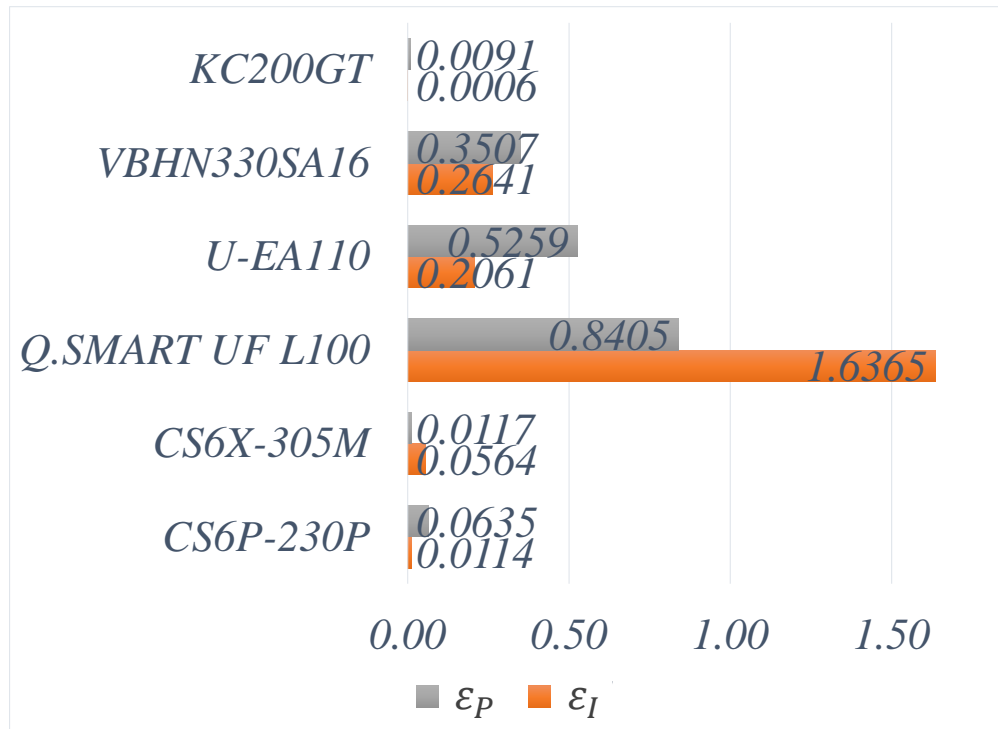
Fig. 5.7 Accuracy within the vicinity of MPP for 6 different PV panels at STC in accordance with the IEC EN 50530 standard.

5.3.2 Comparison across the full range of the PV characteristic curves

Furthermore, by using (5.3), the absolute and relative error are also computed for the reconstructed PV characteristic curves for the 5 other PV panels (see Fig. 5.8 – 5.12). It can be observed that the error the reconstructed PV curves are all minimal especially for thin-film solar cells as well detailed in Table 5.5.

Table 5.5 Evaluation of the error across the full range of the reconstructed curves at STC.

Method	I—V Curve			P—V Curve		
	ME	MSE	$RMSE$	ME	MSE	$RMSE$
KC200GT	1.4470	4.4112	2.1003	53.3742	3.8054×10^3	61.6880
CS6X-230P	1.8830	7.0122	2.6481	72.4308	7.0006×10^3	83.6693
CS6X-305M	2.0969	8.9672	2.9945	97.8711	1.3050×10^4	114.2365
Q.SMART UFL100	0.1596	0.0512	0.2262	13.1217	272.6155	16.5111
U-EA110	0.2641	0.0871	0.2951	22.9475	775.6017	27.8496
VBHN330SA16	0.8013	2.0380	1.4276	62.1374	6.79681×10^3	82.4427

Table 5.6 Accuracy evaluation of the proposed model under varying ambient conditions using KC200GT PV panel in accordance with the IEC EN 50530 standard.

Ambient Condition (AC)		Datasheet Data (DD)			Superellipse Approximation (SM)			IEC EN 50530 Standards	
$Irradiance$ (W/m^2)	$Temperature$ $(°C)$	V_{mp}	I_{mp}	P_{mp}	V_{mp}	I_{mp}	P_{mp}	ε_I	ε_P
400	25	25.7354	3.0414	78.2716	25.0620	2.9910	74.9810	0.0819	0.9458
600	25	26.4015	4.5592	120.3697	25.3913	4.5666	115.9526	0.0676	0.9594
800	25	26.6890	6.1239	163.4408	25.9182	6.0848	157.7084	0.1130	0.9159
1000	25	26.2795	7.6203	200.2577	26.3134	7.6062	200.1463	0.0081	0.0091
1000	25	26.2795	7.6203	200.2577	26.3134	7.6061	200.1428	0.0498	0.0151
1000	50	23.7096	7.5154	178.1871	23.8540	7.6061	181.4365	0.3225	0.0154
1000	75	20.3147	7.5131	152.6264	21.3946	7.6061	162.7301	0.3300	1.7645

5.4 Accuracy evaluation under varying ambient conditions

5.4.1 Constant STC irradiance – varying temperature condition

5.4.1.1 Comparison at MPP using the IEC EN 50530 standard

Irrespective of the changes in ambient conditions, it has been established and verified that the fitting parameters of the superellipse model should be regarded as unchanged. As such, the optimum values for m and n under STC using the DFP algorithm are regarded as invariant. By applying (3.6), the full-range reconstruction of the KC200GT PV panel characteristic curves under constant STC temperature can therefore be achieved. As expected, the reconstructed I–V and P–V curves shown in Fig. 5.19 maintains high model accuracy at MPP with the datasheet curves (see Table 5.6).

5.4.1.2   Comparison across the full range of the PV characteristic curves

By utilizing (5.3), the absolute and relative error for the reconstructed characteristic curves for the 6 different PV panels can also be easily computed (see Fig. 5.20 – 5.31). It can be observed that regardless of cell material, varying irradiance condition, the relative error across the full range of the reconstructed curves are all within the vicinity of 5% with the exemption of hybrid thin-film PV panels (U-EA100).

5.4.2   Constant STC irradiance – varying temperature condition

Similarly, the accuracy of the superellipse model is also evaluated under varying temperature. Although the proposed model retains high accuracy up to $50°C$, high ε_P is observed in the reconstructed PV characteristic curves at $75°C$ as shown in Fig. 5.32. Due to the constant STC irradiation, the estimated value for I_{sc}^* remains unchanged. Consequently, based on its mathematical equation, the value of I_{mp} will also remain the same as shown Table 5.6.

Regardless, this further adds to the numerous advantages of the superellipse model in achieving an easy-to-fit replica of both the I–V and P–V characteristic curves under both STC and varying ambient conditions.

(a)

(b)
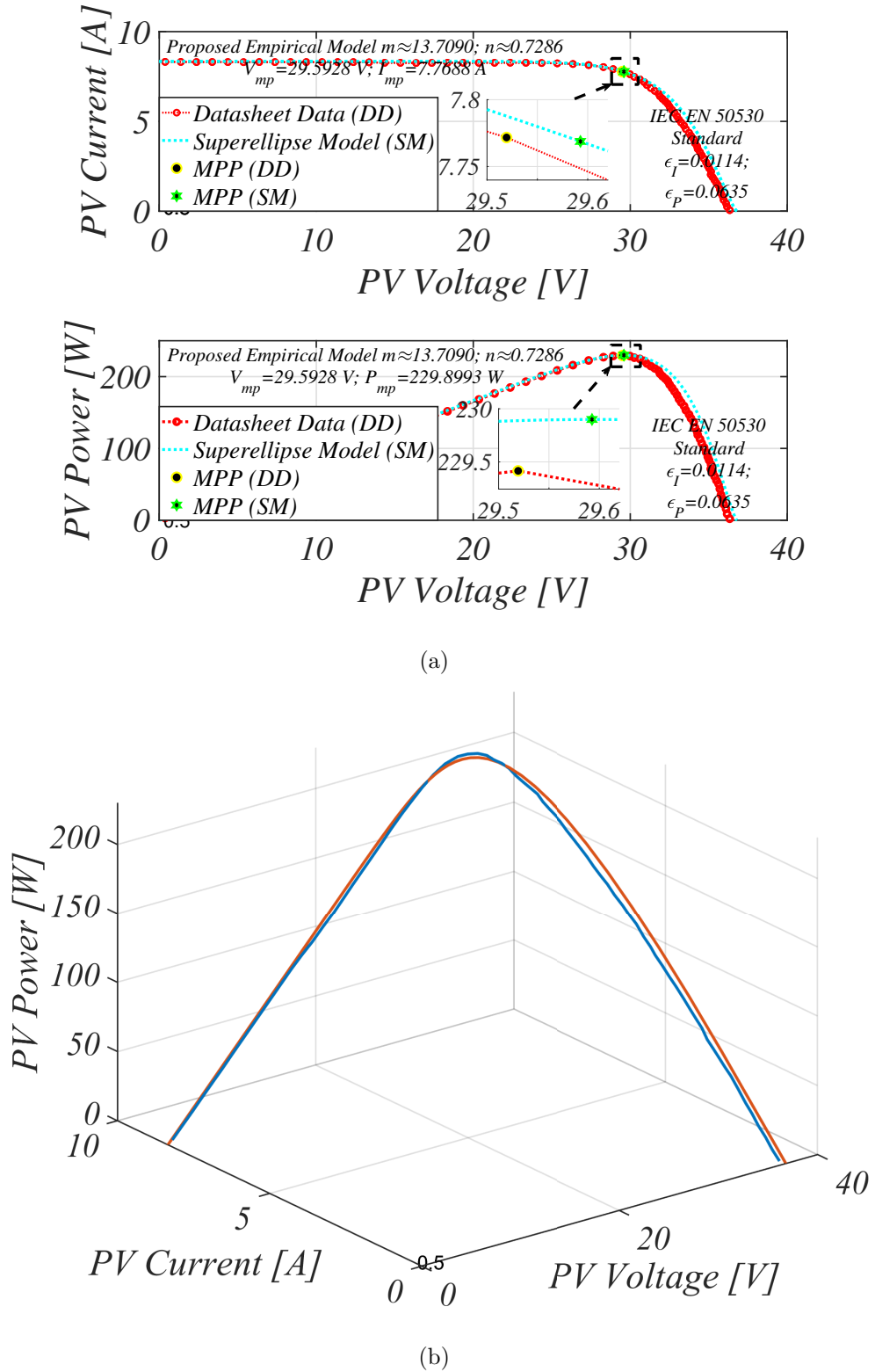
Fig. 5.8 Reconstruction of the PV characteristic curves for the CS6P-230P PV panel using the superellipse model (a) 2–D plot (b) 3–D plot.

(a)

(b)
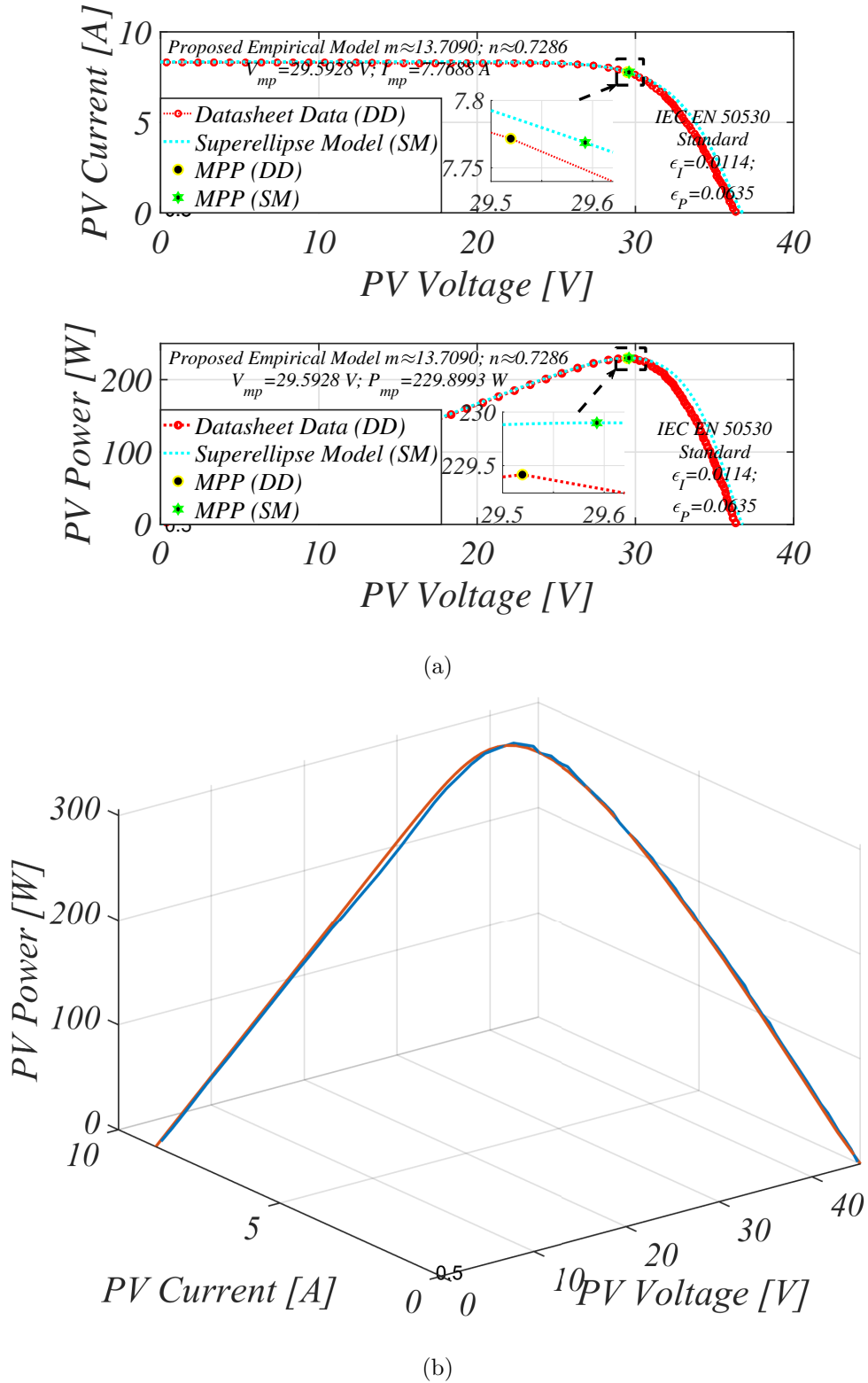
Fig. 5.9 Reconstruction of the PV characteristic curves for the CS6X-305M PV panel using the superellipse model (a) 2–D plot (b) 3–D plot.

(a)

(b)
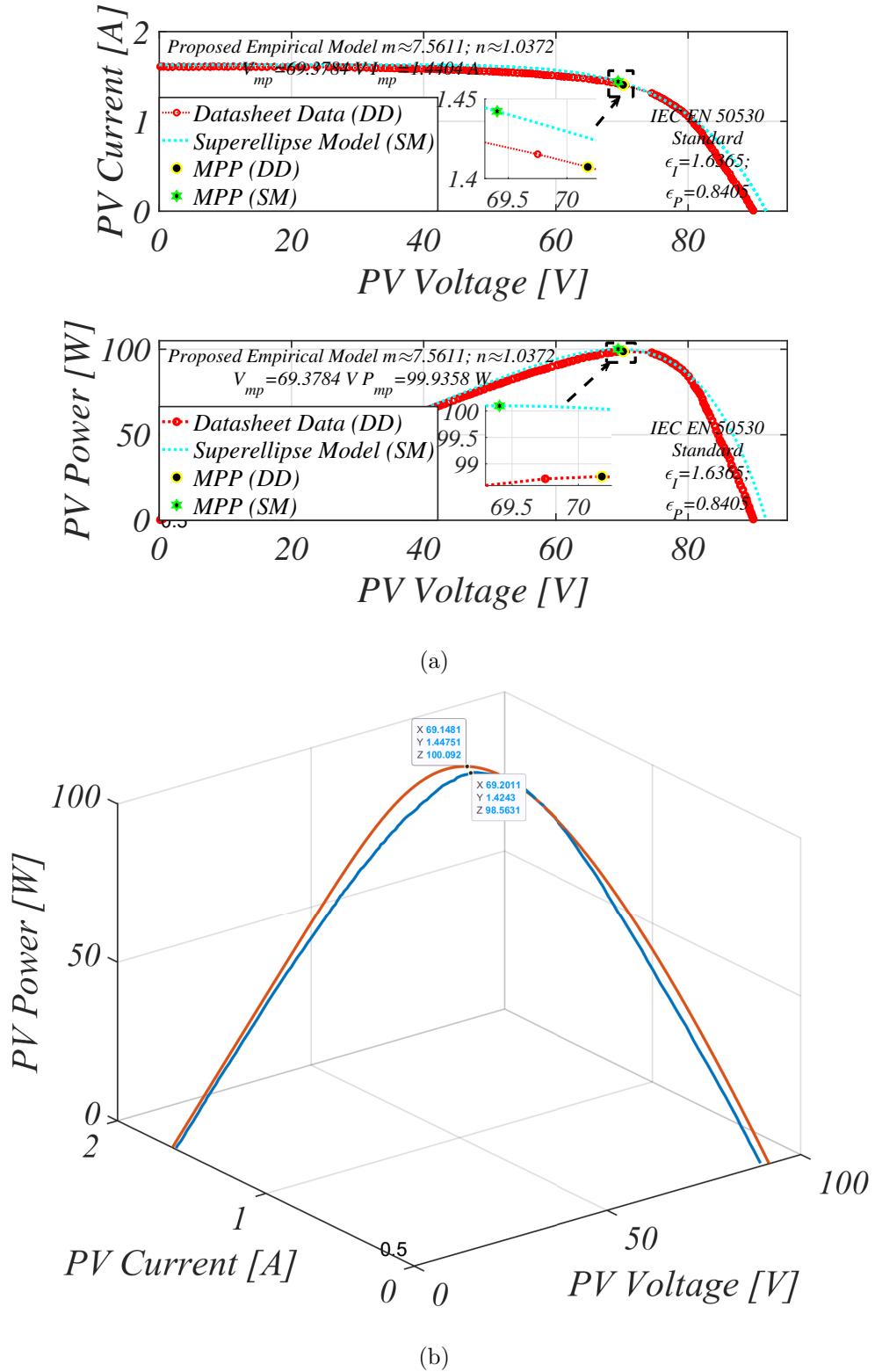
Fig. 5.10 Reconstruction of the PV characteristic curves for the Q.SMART UFL100 PV panel using the superellipse model (a) 2–D plot (b) 3–D plot.

(a)

(b)
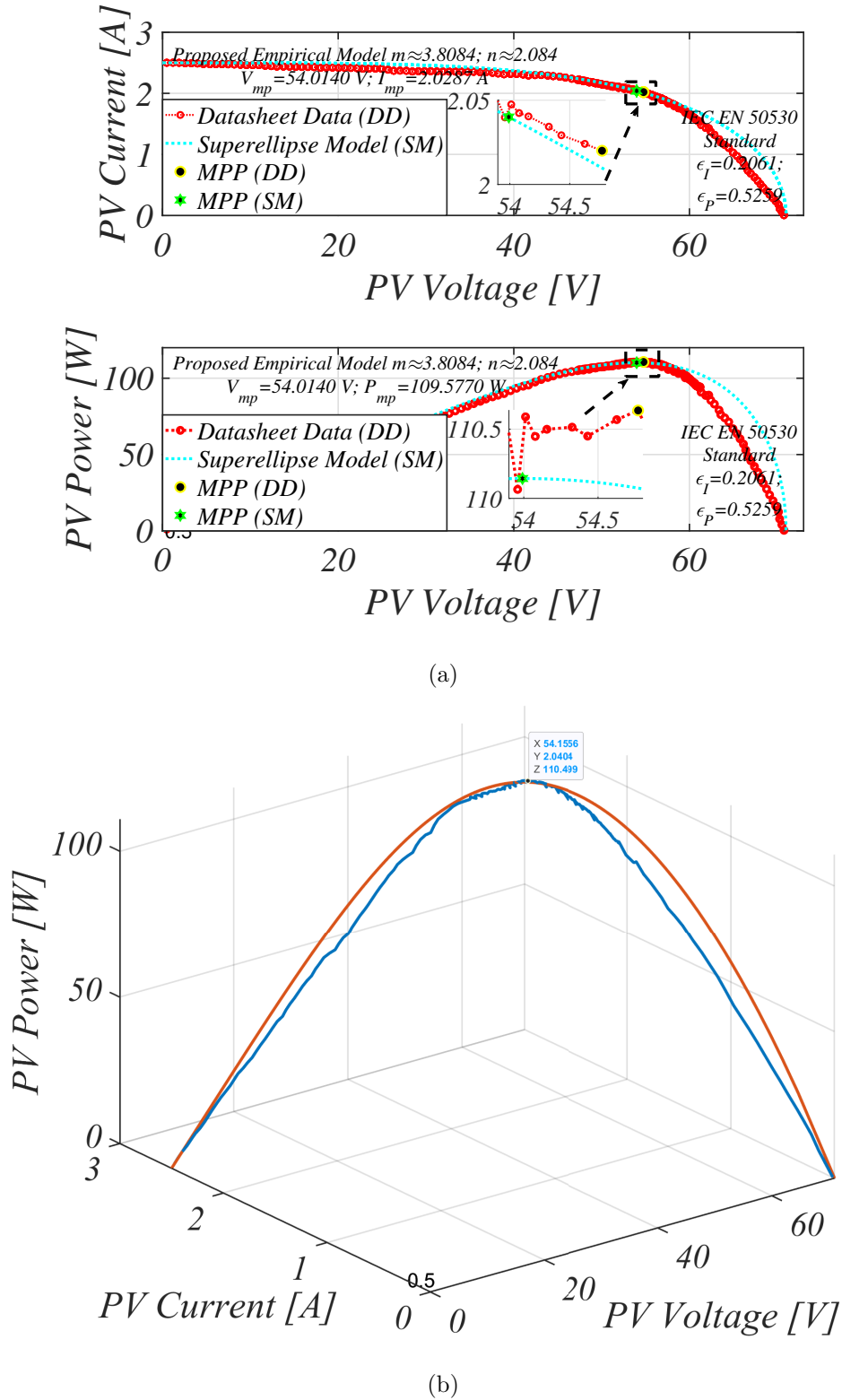
Fig. 5.11 Reconstruction of the PV characteristic curves for the U-EA110 PV panel using the superellipse model (a) 2–D plot (b) 3–D plot.

(a)

(b)
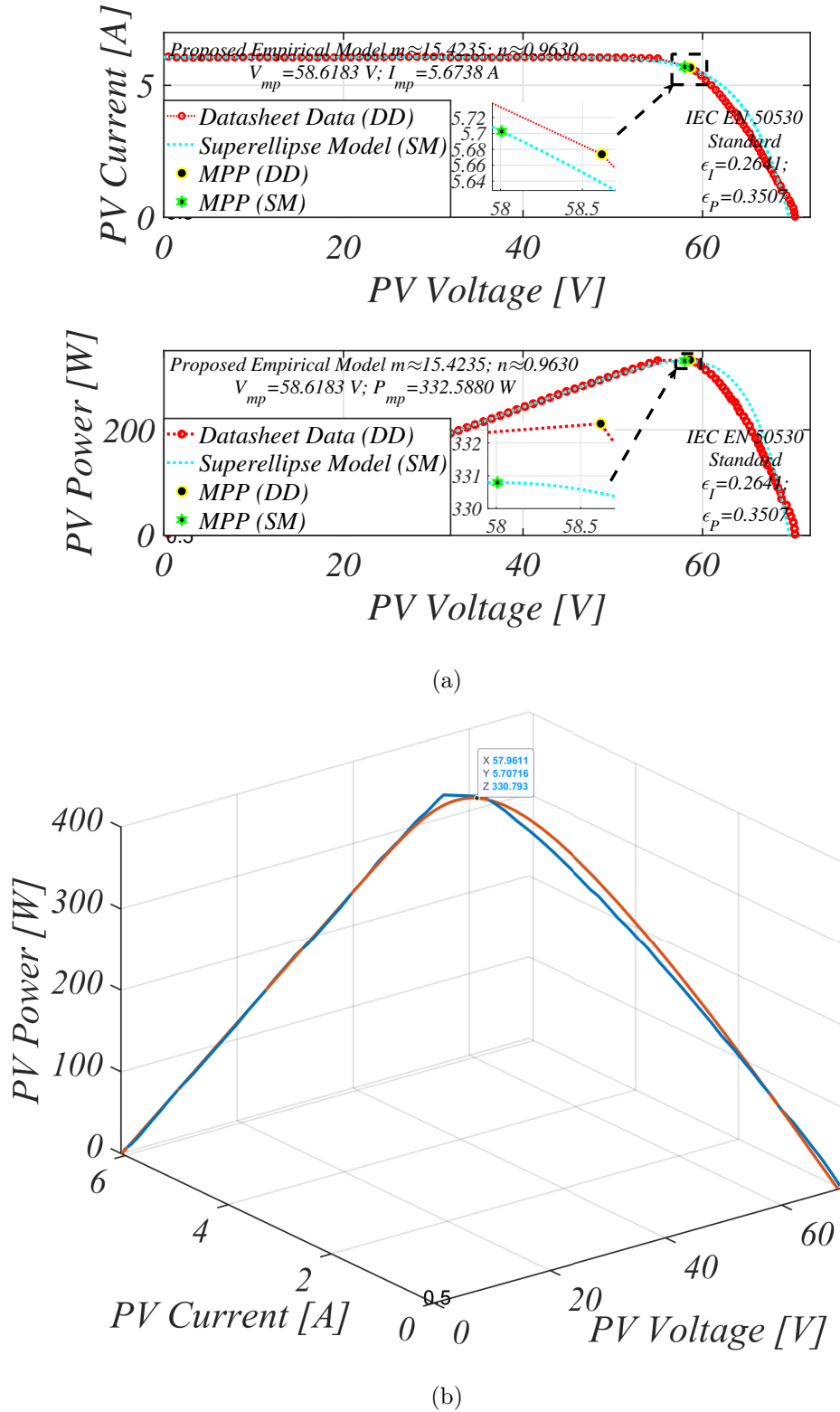
Fig. 5.12 Reconstruction of the PV characteristic curves for the VBHN330SA16 PV panel using the superellipse model (a) 2–D plot (b) 3–D plot.
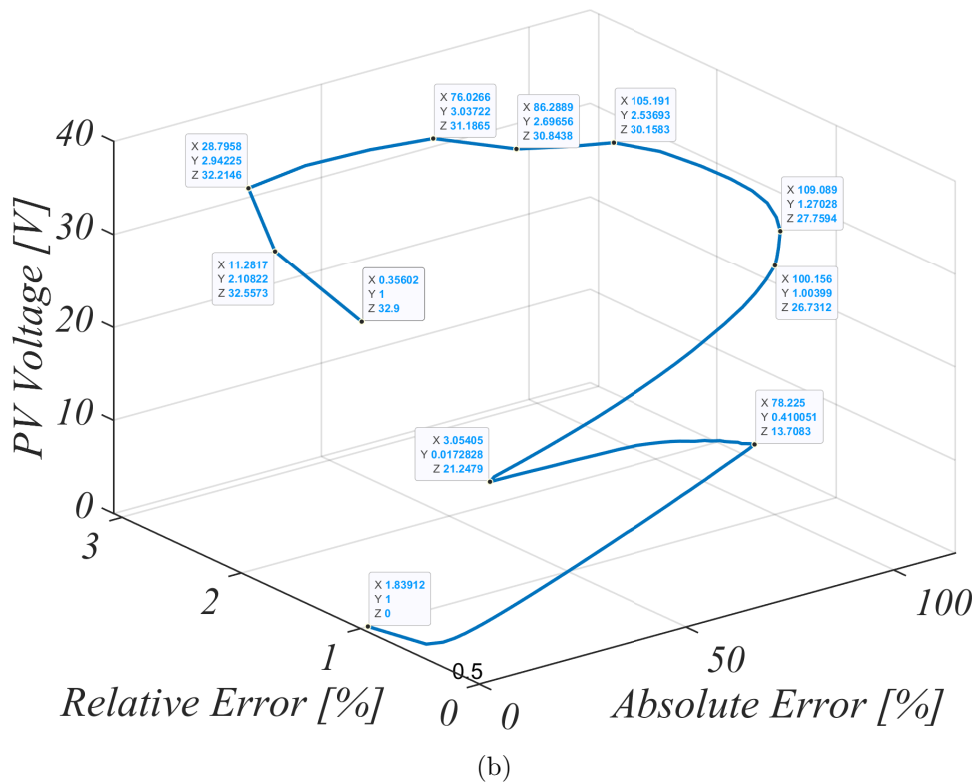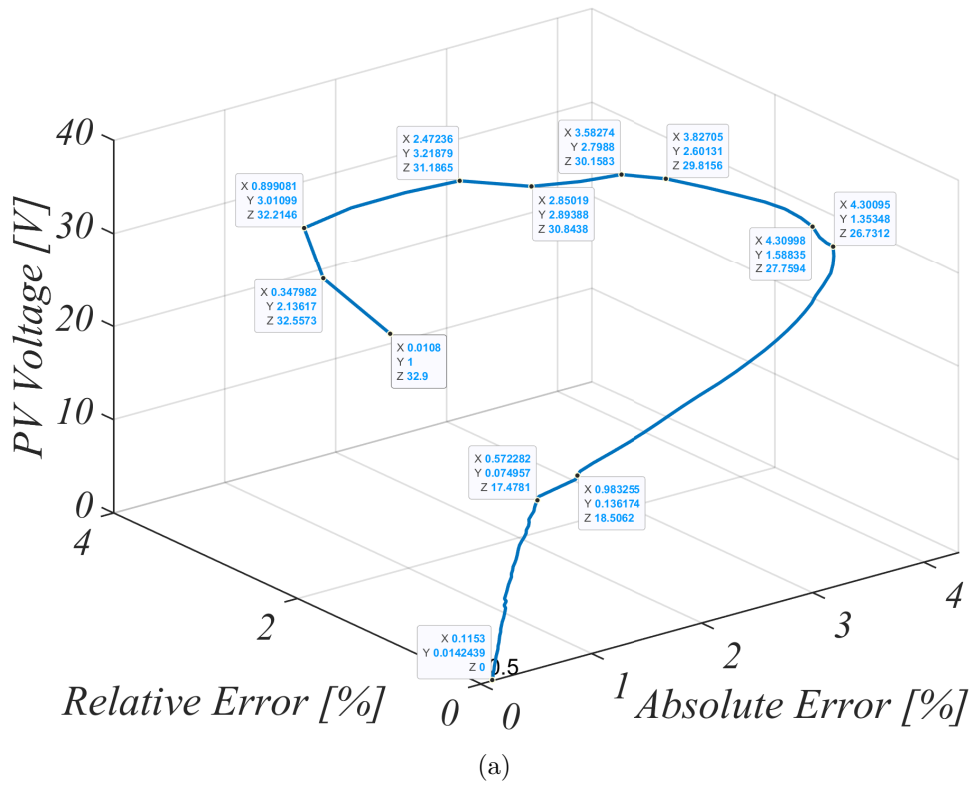
Fig. 5.13 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for KC200GT PV panel (a) I–V curve (b) P–V curve.

(a)

(b)
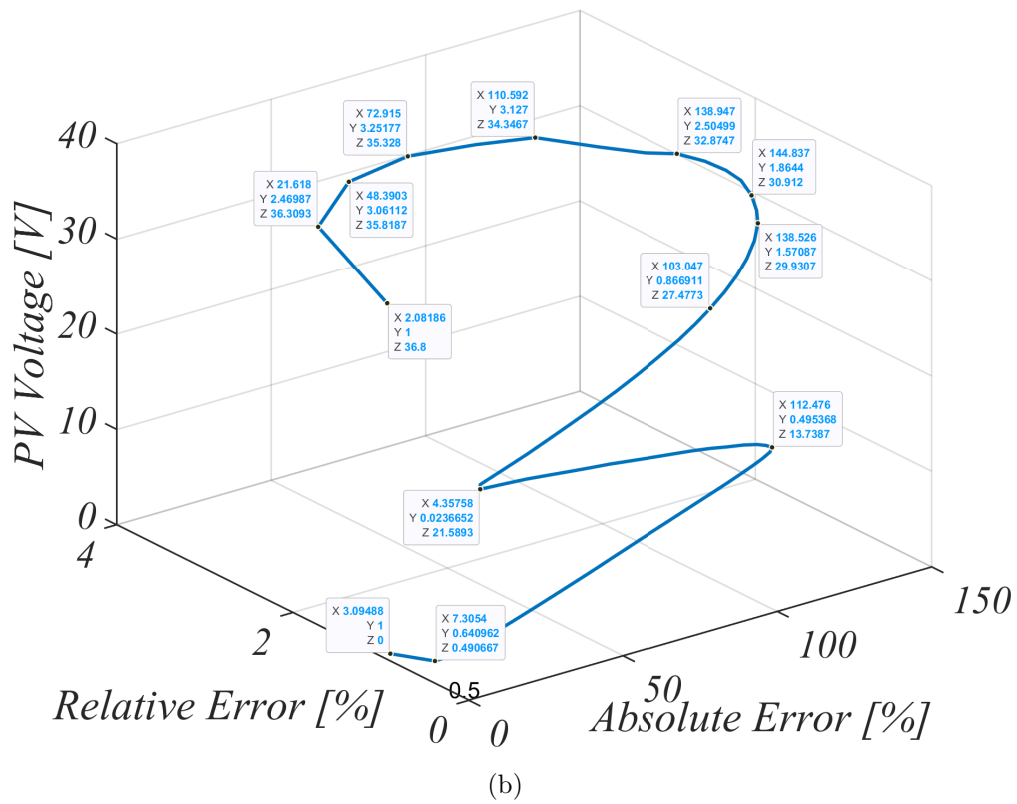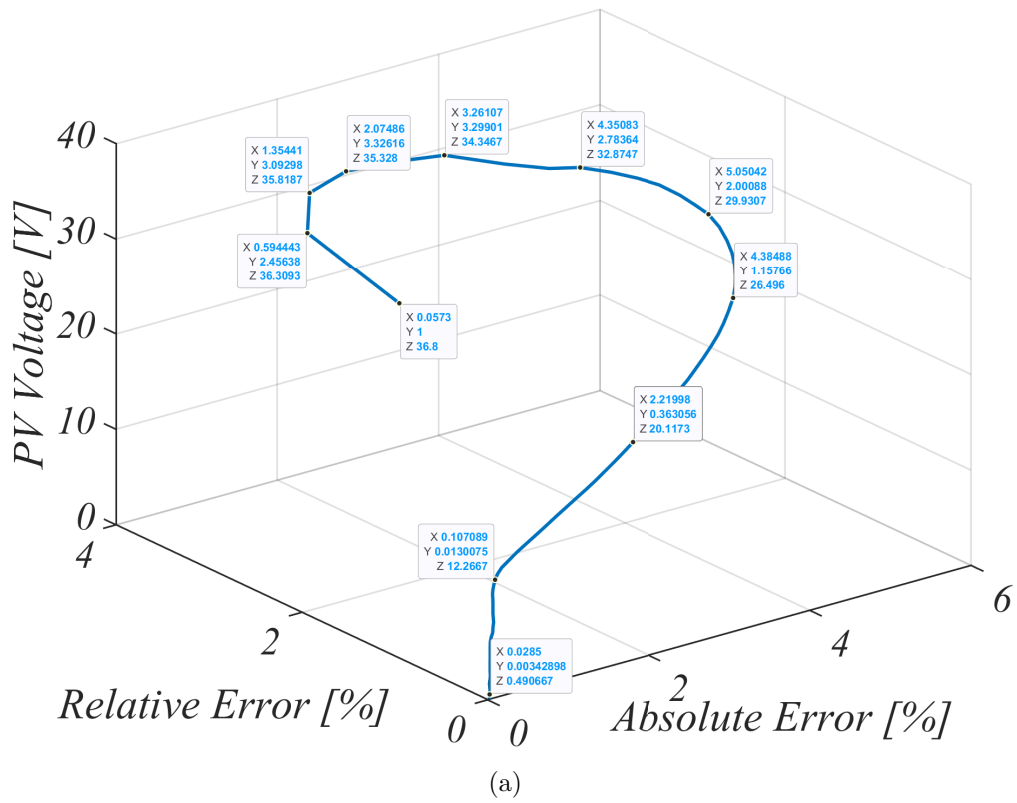
Fig. 5.14 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for CS6X-230P PV panel (a) I–V curve (b) P–V curve.
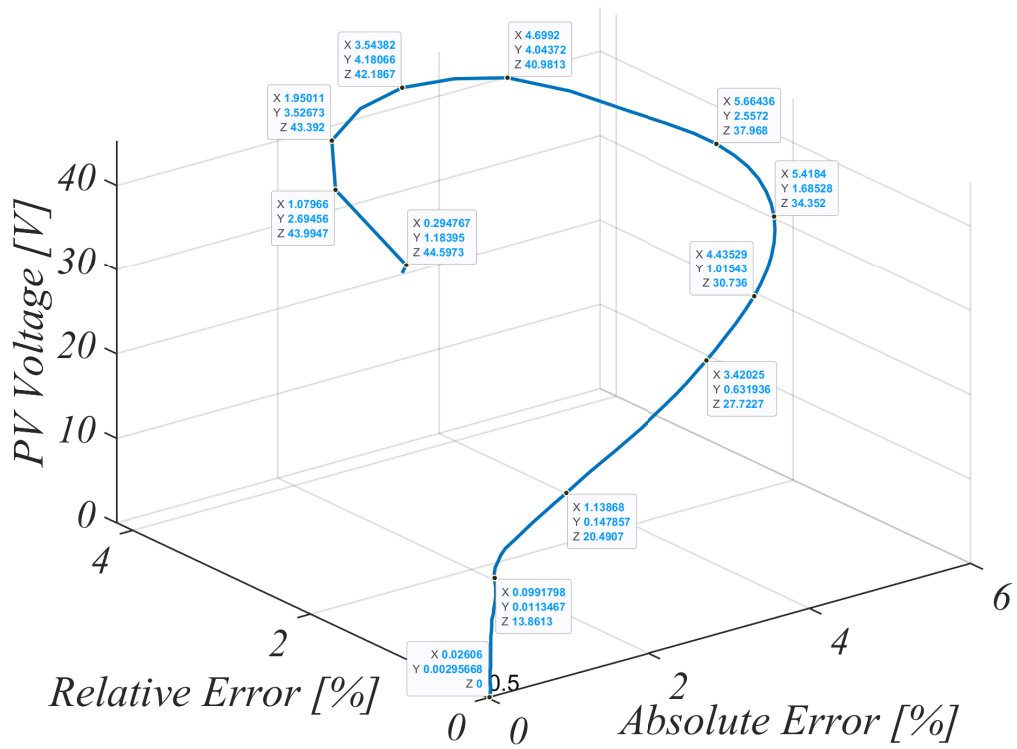
(a)
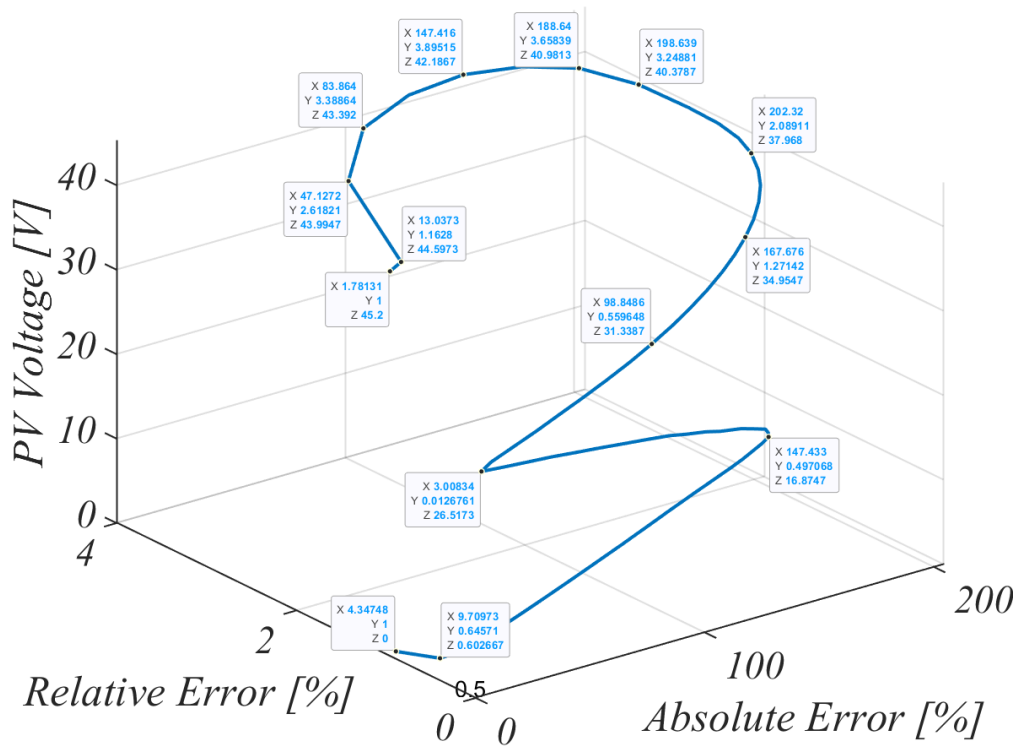
(b)

Fig. 5.15 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for CS6X-305M PV panel (a) I–V curve (b) P–V curve.

(a)

(b)
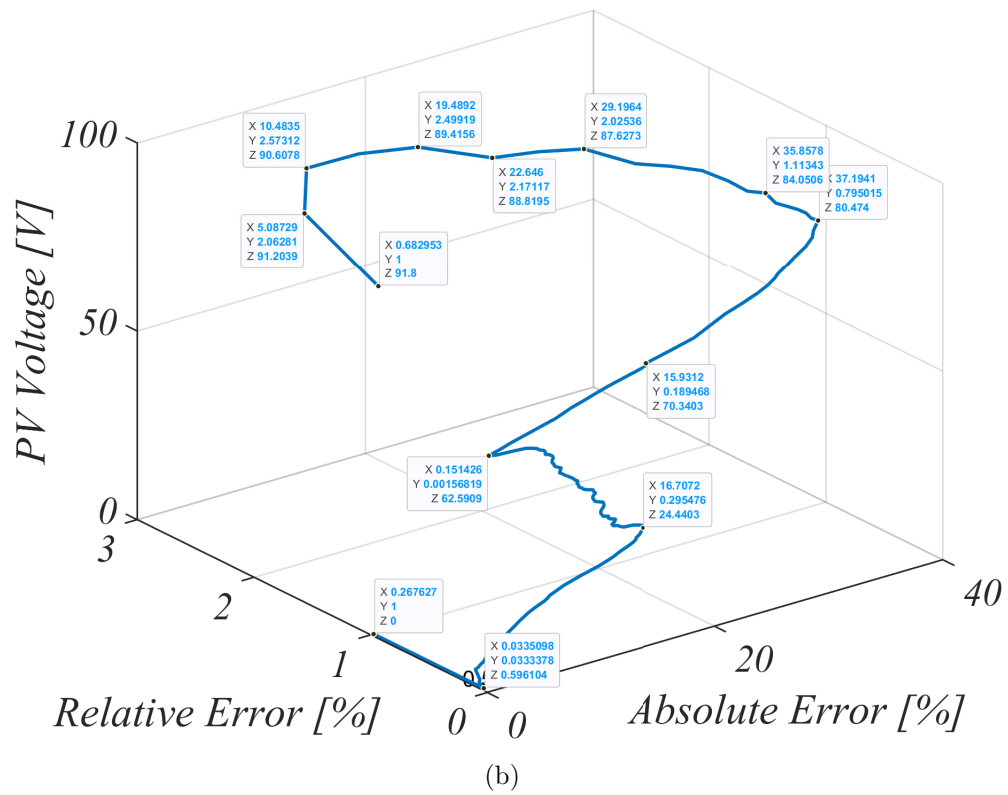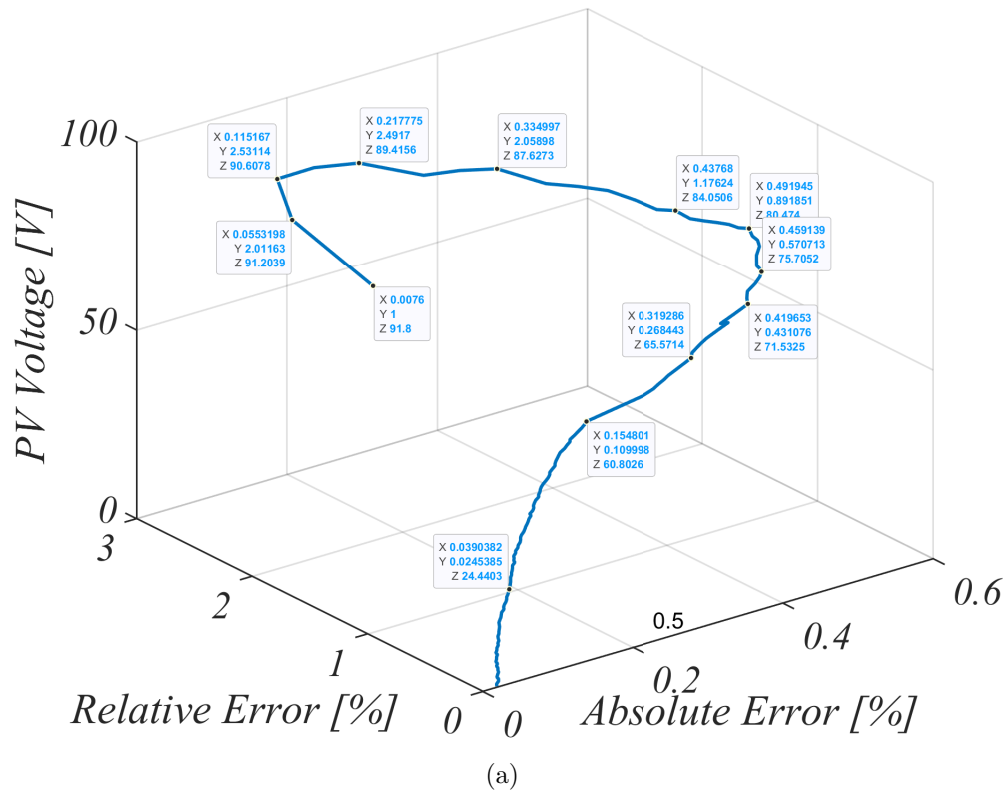
Fig. 5.16 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for Q.SMART UFL100 PV panel (a) I–V curve (b) P–V curve.
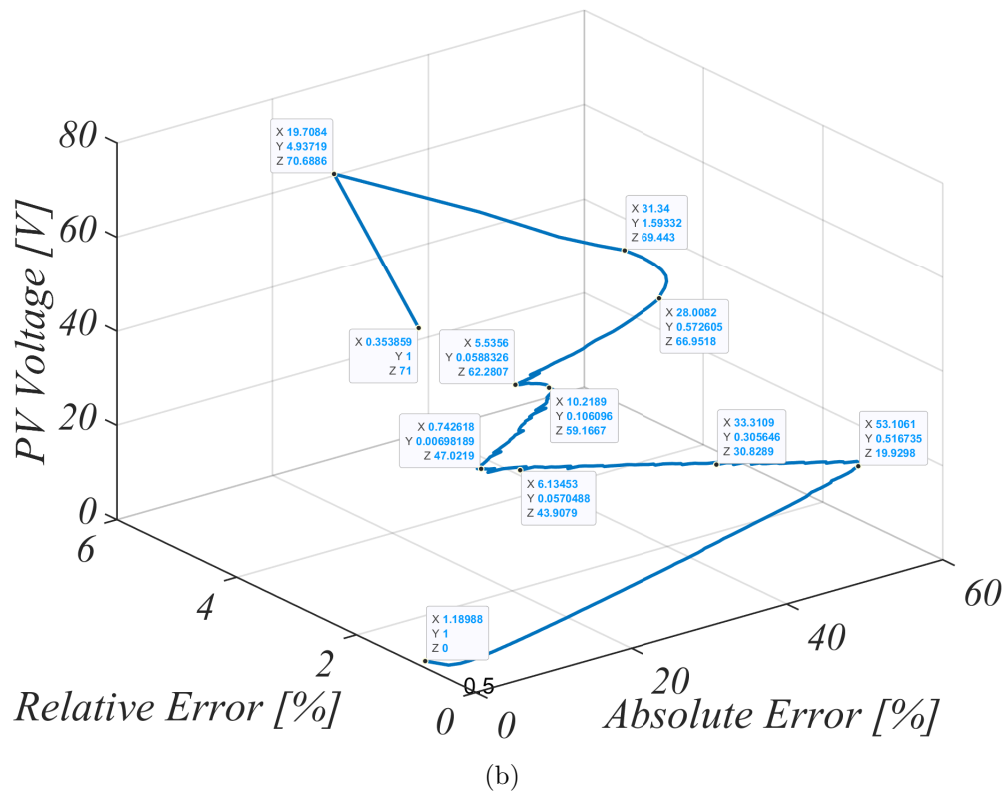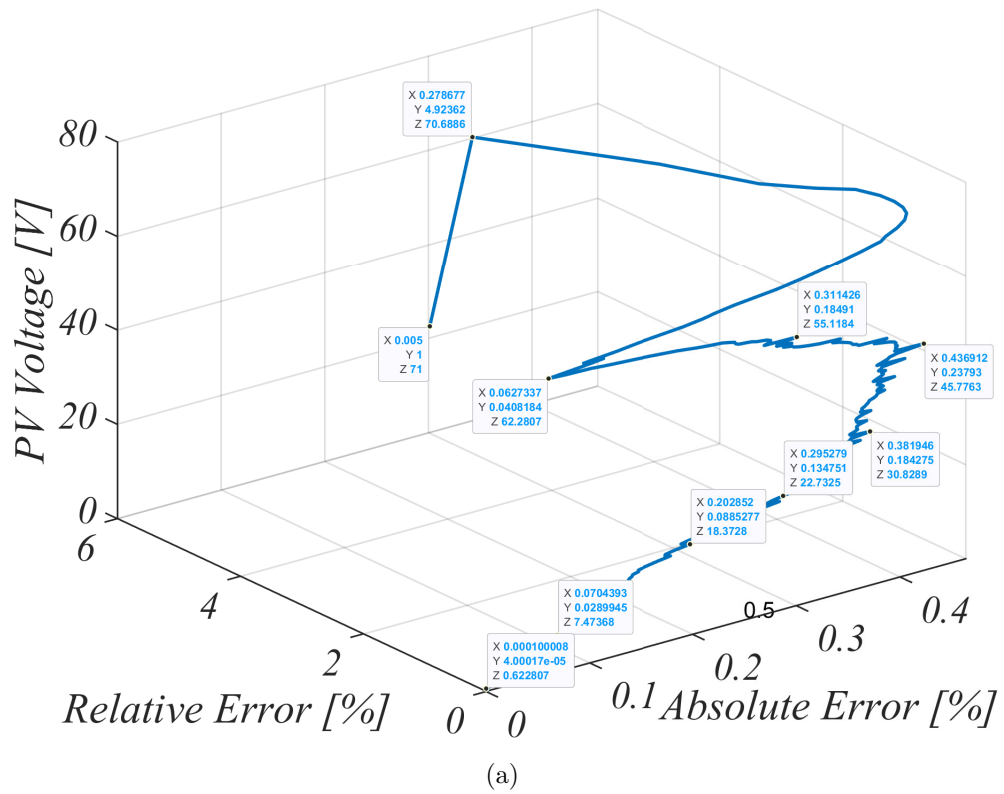
Fig. 5.17 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for U-EA110 PV panel (a) I–V curve (b) P–V curve.

(a)

(b)
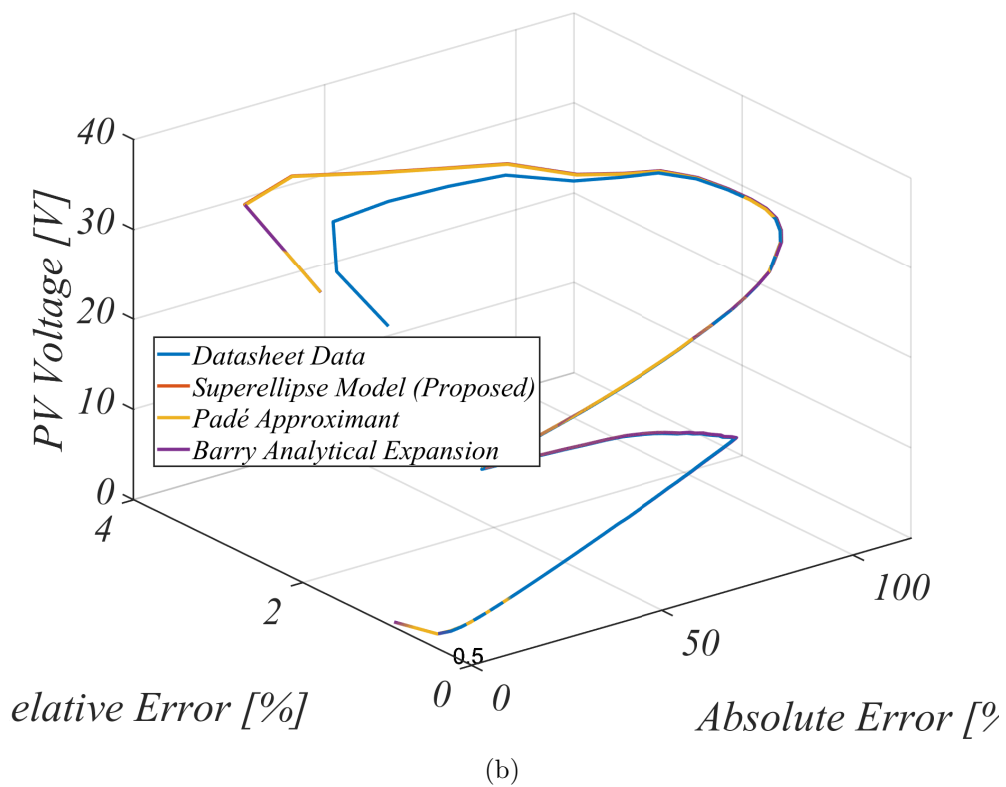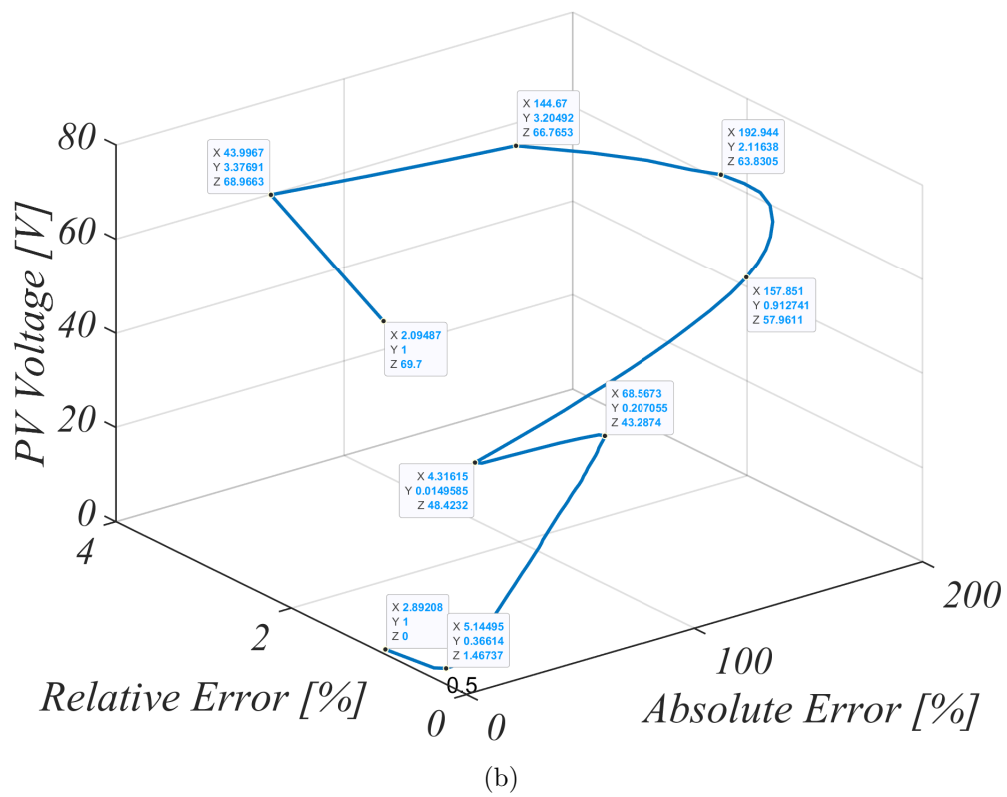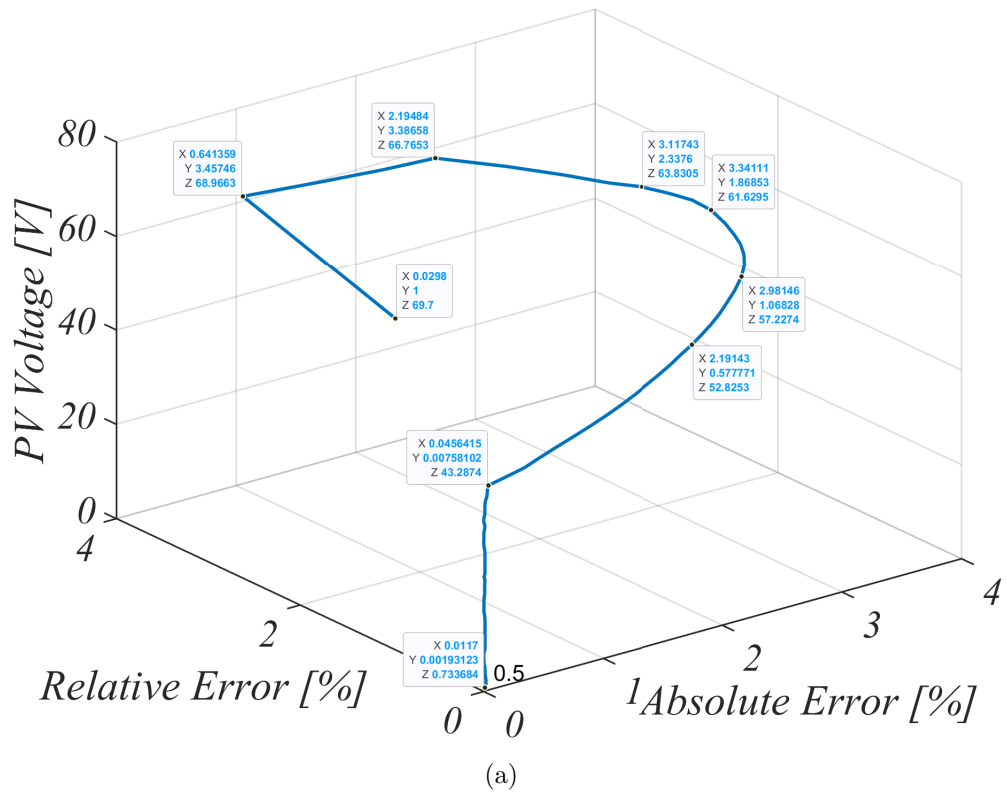
Fig. 5.18 3–D plot of the absolute and relative error across the full range of the reconstructed PV characteristic curve at STC for VBHN330SA16 PV panel (a) I–V curve (b) P–V curve.

(a)

(b)

Fig. 5.19 Accuracy of the reconstructed PV characteristic curves using the KC200GT PV panel under varying irradiance condition (a) PV characteristic curves (b) accuracy at MPP in accordance with the IEC EN 50530 standard.

Fig. 5.20 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for KC200GT PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.21 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for KC200GT PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.22 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for CS6P-230P PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.23 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for CS6P-230P PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.24 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for CS6X–305M PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.25 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for CS6X-305M PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.26 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for Q.SMARTUFL100 PV panel (a) 600 W/m^2 (b) 800 W/m^2 (c) 1000 W/m^2.

Fig. 5.27 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for Q.SMARTUFL100 PV panel (a) 600 W/m^2 (b) 800 W/m^2 (c) 1000 W/m^2.

(a)

(b)

(c)

Fig. 5.28 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for U-EA110 PV panel (a) 600 W/m^2 (b) 800 W/m^2 (c) 1000 W/m^2.

(a)

(b)

(c)

Fig. 5.29 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for U-EA110 PV panel (a) 600 W/m^2 (b) 800 W/m^2 (c) 1000 W/m^2.

Fig. 5.30 3–D plot of the absolute and relative error across the full range of the reconstructed I–V curve under varying irradiance condition for VBHN330SA16 PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

Fig. 5.31 3–D plot of the absolute and relative error across the full range of the reconstructed P–V curve under varying irradiance condition for VBHN330SA16 PV panel (a) 400 W/m^2 (b) 600 W/m^2 (c) 800 W/m^2 (d) 1000 W/m^2.

(a)

(b)

Fig. 5.32 Accuracy of the reconstructed PV characteristic curves using the KC200GT PV panel under varying temperature condition (a) PV characteristic curves (b) accuracy at MPP in accordance with the IEC EN 50530 standard.

Chapter 6

Conclusion and future work

6.1 Conclusion

In this master thesis, a novel superellipse-based empirical model for the accurate and effective modeling and approximation of the characteristic curves for PV panels was proposed. Unlike the conventional single-diode model used widely by both researchers and technicians in this field, which requires five fitting parameters, the superellipse model requires only two fitting parameters i.e. (m, n) at STC and a third parameter A_n under varying ambient condition.

The explicit equation describing the approximate I-V characteristic curve is established as a result of the unique similarities between the geometric shapes of the superellipse and the graphical characteristic of the typical I-V curve as specified in any manufacturer's datasheet. Since the superellipse model is a replica of the manufacturer's PV characteristic curves, all datasheet constraints and mathematical properties describing these curves are also subsequently applied to the explicit equation describing the novel empirical model. As a result,

this ensures the multidimensional equations describing the PV model are also necessary and sufficient conditions are required for extracting its optimum fitting parameters.

Performance evaluation using four different criteria including the IEC EN 50530 standard show that irrespective of PV cell material, ambient conditions, the explicit equation describing the superellipse model can be utilized to achieve the accurate reconstruction of the PV characteristic curves provided that the extracted fitting parameters are optimum.

6.2 Future work

In this thesis, the superellipse-based PV model was introduced as an accurate and easy-to-fit alternative to the conventional single-diode model. Nonetheless, several improvements are still needed such as

(A) Hybrid empirical-physical modeling of PV panels – By combing the simplicity of the superellipse model, and the proper understanding of the electrical parameters of circuit-based models, hybrid models with higher accuracy and faster computational speed can be easily obtained.

(B) Hardware Implementation – This thesis primarily introduced the superellipse model for implementation in commercial software environments such as MATLAB/Simulink etc. A detailed evaluation of these simplified PVM equation in the design of low cost, low power solar array simulators should be carried out.

(C) Partial shading conditions – In this thesis, the model accuracy were mostly evaluated under global shading conditions. Since partial and global shading conditions have

different effects on the PV characteristic curves, the accuracy of the superellipse model under this conditions should be evaluated.

(D) Speed enhancement – Although the superellipse model has relatively high accuracy at MPP, the execution time required for the reconstruction of these curves as quite low. For large scale PV systems, fast reconstruction of the PV characteristic curves is essential. Hence, more study is required for improving the simulation speed.

Appendix A

Single-shaped and double-shaped superellipse

To differentiate between a single-shaped, and double-shaped superellipse as shown in Fig. 3.1, the gradient of these two curves are discussed. At maximum point, the gradients of these two curves can be expressed as

Single-shaped superellipse

$$y = b \left[1 - \left(\frac{x}{a} \right)^n \right]^{\frac{1}{n}} \tag{A.1a}$$

$$\left. \frac{dy}{dx} \right|_{x=max(x)} = -b \left(\frac{y}{b} \right)^{1-n} \left(\frac{x}{a} \right)^{n-1} \tag{A.1b}$$

Double-shaped superellipse

$$y = b \left[1 - \left(\frac{x}{a} \right)^m \right]^{\frac{1}{n}} \tag{A.2a}$$

$$\left. \frac{dy}{dx} \right|_{x=max(x)} = \frac{mb}{n} \left(\frac{y}{b} \right)^{1-n} \left(\frac{x}{a} \right)^{m-1}. \tag{A.2b}$$

106

Since the single-shaped superellipse is smoother at both its semi-major and semi-minor axes, the gradient at any point these curves would always be ≈ -1. On the other hand, depending on the fitting parameters, the gradient of the double-shaped superellipse are not constant (see Fig.).
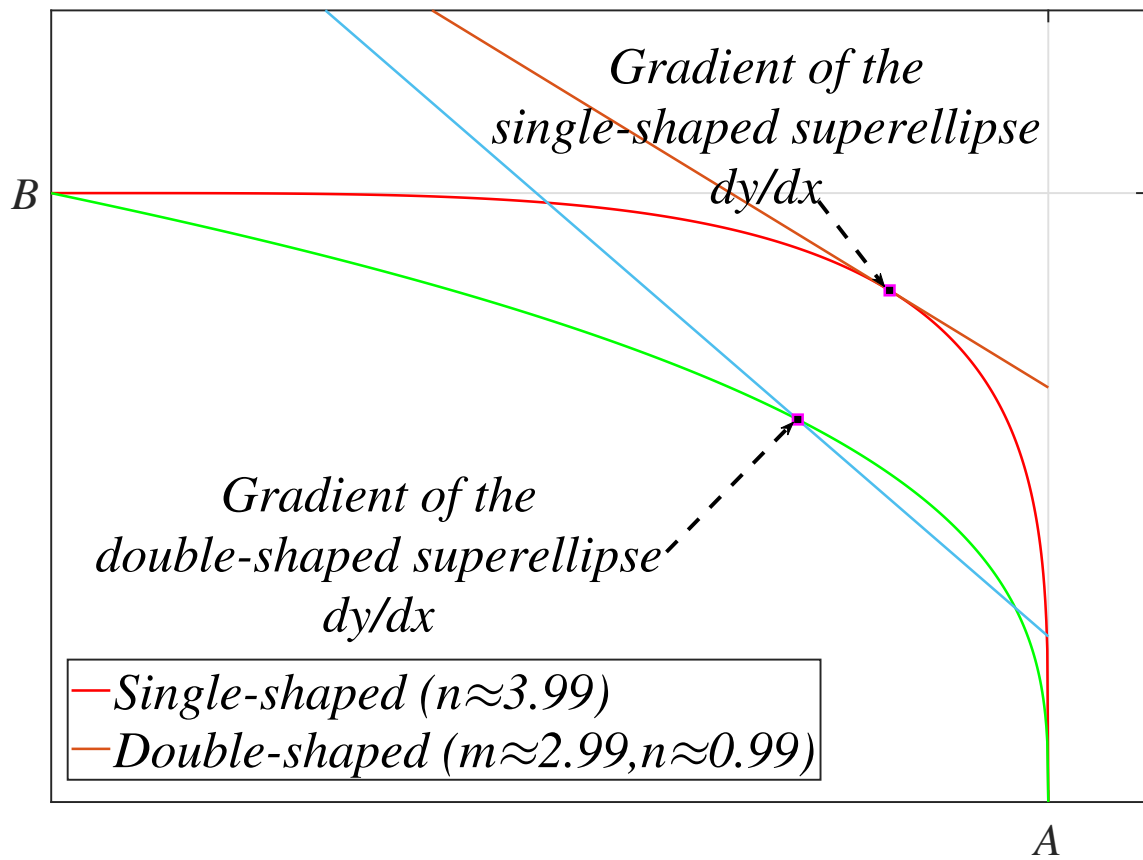
Fig. A.1 Comparison of the gradient of the superellipse where $A = B = 1$.

Appendix B

Implementation of the superellipse model in MATLAB/Simulnk

Based on basic mathematical principles, (3.3) can be easily implemented in MATLAB/Simulink using three different approaches

(A) Approach 1 – By employing the inbuilt $power(p,q)$ function in MATLAB, the data point solutions across the full range of the I–V curve can be easily obtained such that

$$i = I_{sc}\left[1 - \left(\frac{v}{V_{oc}}\right)^m\right]^{\frac{1}{n}} \tag{B.1a}$$

$$i = I_{sc} \cdot power\left[\left(1 - power\left(\frac{v}{V_{oc}}\right)\right), \left(\frac{1}{n}\right)\right] \tag{B.1b}$$

(B) Approach 2 – The $power(p,q)$ in MATLAB are based on the Maclaurin series expansion where

$$power(p,q) = p^q = exp(b \cdot In(a)) \approx 1 + b \cdot In(a) + \frac{(xIn(a))^2}{2!} + \frac{(xIn(a))^3}{3!} + ... \quad \text{(B.2)}$$

Hence, (3.3) can therefore be easily rewritten as

$$i = I_{sc} \cdot exp\left(\frac{1}{n} \cdot z\right) \quad \text{(B.3)}$$

where $z = In\left[1 - exp\left(m \cdot In\left(\frac{1}{n}\right)\right)\right]$.

(C) Approach 3: Also, in its basic form,

$$power(p,q) = p^q \quad \text{(B.4)}$$

As such, by applying (B.4) into (3.3), the I—V curve can be easily reconstructed using

$$i = I_{sc}\left[1 - \left(\frac{v}{V_{oc}}\right)^m\right]^{\frac{1}{n}}. \quad \text{(B.5)}$$

In this thesis, the I–V curve at both STC and varying ambient conditions are reconstructed using Approach 2. A detailed description of the .m file code in MATLAB is given below

```
1        clc
2        clear all
3
```

```matlab
%% Key Points of the I-V curve as provided in the
    datasheet (KC200GT)
Imp = 7.61; %% Peak Maximum Current
Vmp = 26.3; %% Peak Maximum Voltage
Isc = 8.21; %% Short Circuit Current
Voc = 32.9; %% Opem Circuit Voltage
%% Fitting parameters of the superellipse model
m=12.7941; %% m-parameter
n=0.7734; %% n-parameter
%% Explicit equation of the superellipse model -
    Approach 3
v = linspace(0,Voc,1000);
i=[];
i = Isc.*exp((1/n).*log(1-exp(m.*log(v./Voc)))); %%
    Enumerated current
p = v.*i; %% Enumerated power
%% Maximum points of the reconstructed PV
    characteristic curves
maxF = max(p); %% Find max value over all elements.
indexOfFirstMax = find(p == maxF, 1, 'first');  %% Get
    first element that is the max.
% Get the x and y values at that index.
maxP = p(indexOfFirstMax); %% Power at MPP
```

```matlab
22          maxV = v(indexOfFirstMax); %% Voltage at MPP

23          maxI = i(indexOfFirstMax); %% Current at MPP
```

Appendix C

Effects of varying MPP on the superellipse model

As explained in Chapter 3.3, the superellipse model also inherit all the mathematical properties of the typical I—V curve. As such, (3.6) already accounts for the variations in the MPP values which is dependent on the ambient condition of the PV panel as defined in

$$I_{mp}^* = I_{mpn}\frac{G}{G_n} \tag{C.1a}$$

$$V_{mp}^* = V_{mpn} + NA_n\frac{kT}{q}ln\left(\frac{G}{G_n}\right) + \beta_V(T - T_n). \tag{C.1b}$$

where n denotes the values under STC. To validates this fact, the KC200GT PV panel specification is utilized.

```
1        clc

2        clear all
```
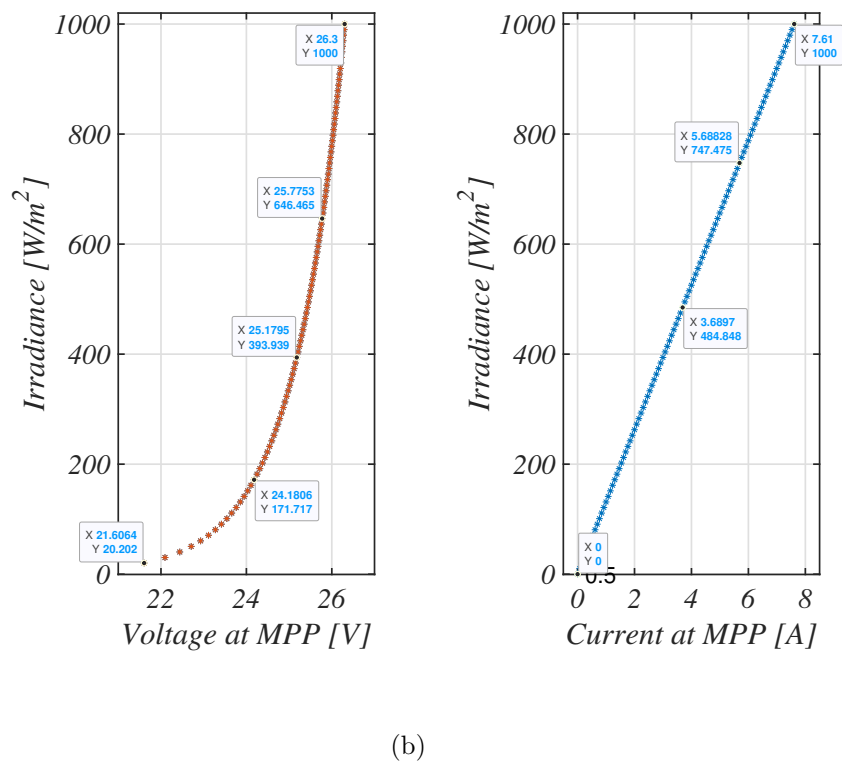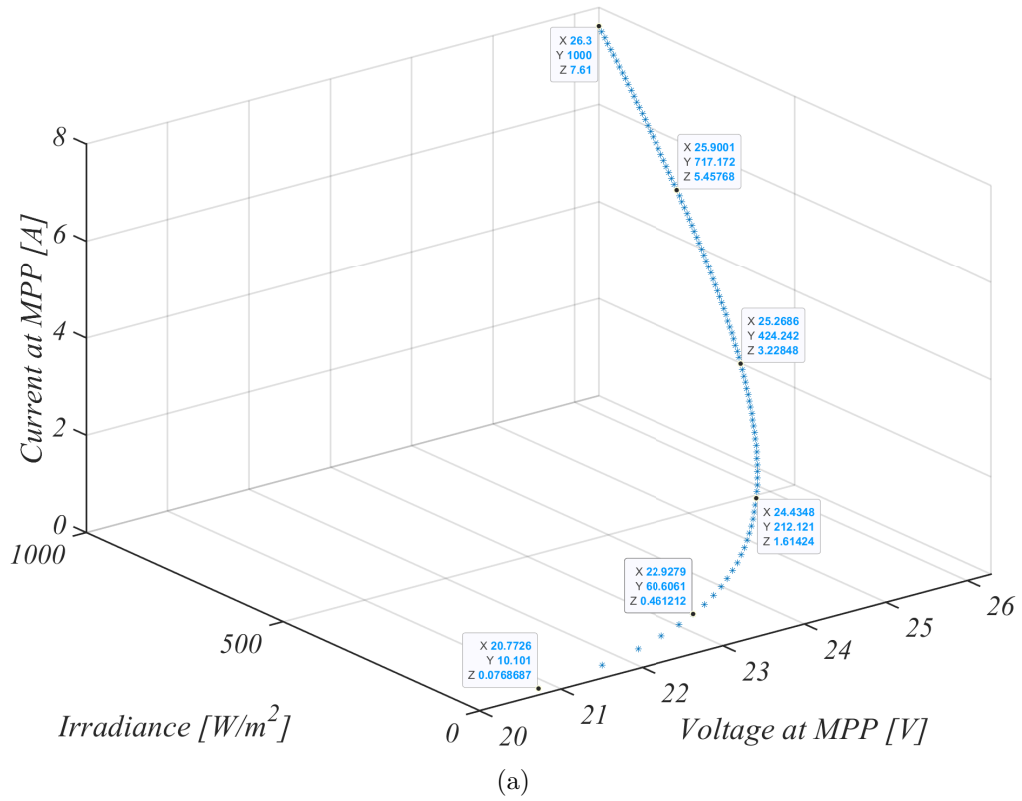
Fig. C.1 Plot of the MPP values (V_{mp}, I_{mp}) under constant STC temperature and varying irradiance conditions (a) 3–D trajectory plot (b) 2–D plot.
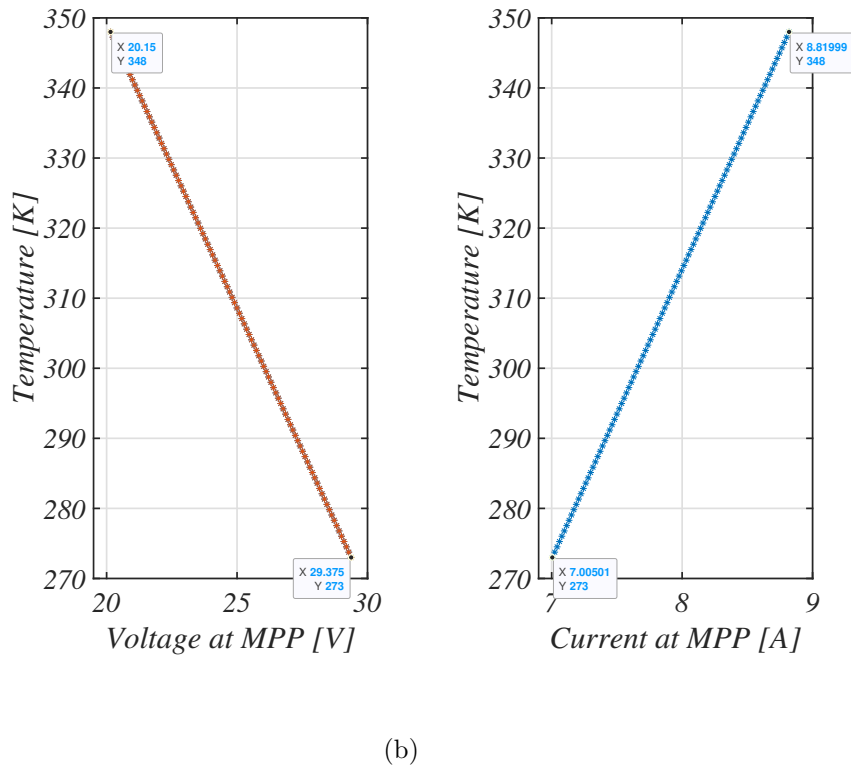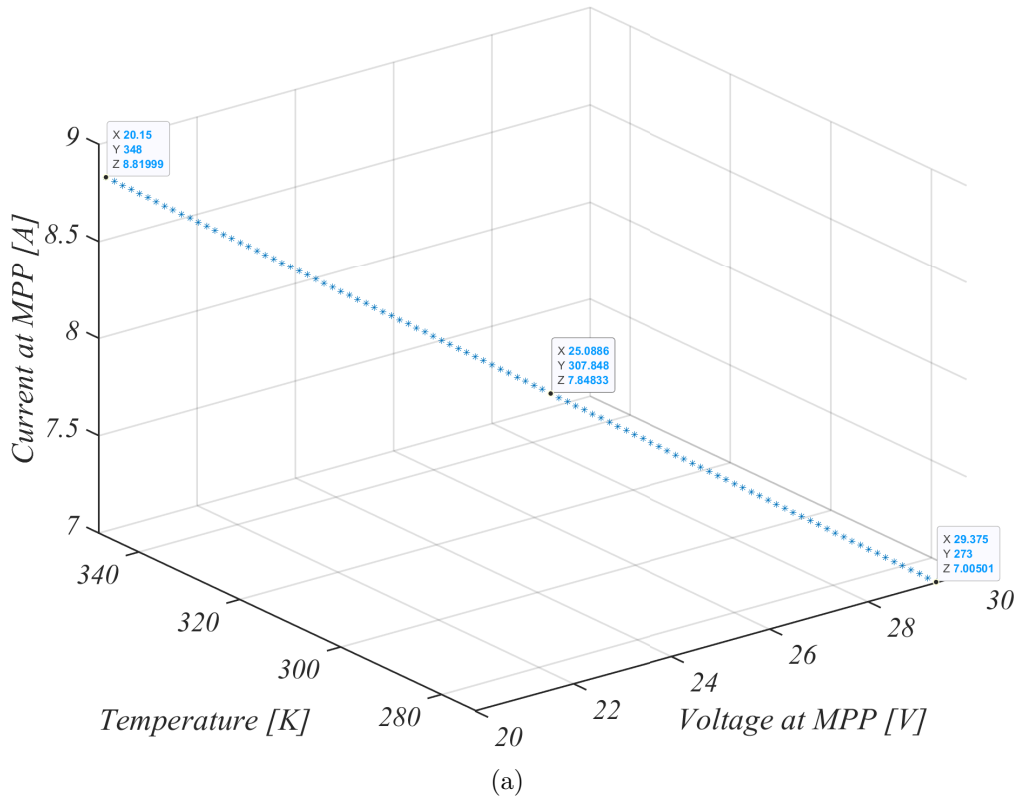
Fig. C.2 Plot of the MPP values (V_{mp}, I_{mp}) under constant STC irradiance and varying temperature conditions (a) 3–D trajectory plot (b) 2–D plot.

114

```matlab
%% KC200GT Datasheet
Imp = 7.61; %% Peak Maximum Current
Vmp = 26.3; %% Peak Maximum Voltage
Isc = 8.21; %% Short Circuit Current
Voc = 32.9; %% Opem Circuit Voltage
A = 1.3; %% Ideality factor
Ns = 36; %% Number of solar cells
k = 1.38e-23; %% Boltzmann Constant
q = 1.6e-19;
Ki = 3.18e-3; %% Current Coefficient
Kv = -1.23e-1; %% Voltage Coefficient
Irr_scn = 1000; %% W/m2
Temp_scn = 25+273; %% Degree celcius
%% Superellipse Parameter
m=12.7941; %% m-parameter
n=0.7734; %%m-parameter
%% Parameters for changing the Key Poiints of the I-V
    Curve
Isc_scn = Isc;
Voc_scn = Voc;
Imp_scn = Imp;
Vmp_scn = Vmp;
```

```matlab
25          A_scn = A;

26          Kimpp = Ki;

27          Kvmpp = Kv;

28          %% Varying irradiance

29          G = linspace(0,Irr_scn,100);

30          Imp_vary = Imp_scn.*(G./Irr_scn);

31          Vmp_vary = Vmp_scn+((Ns*A).*((k*Temp_scn)/q).*log(G./
                Irr_scn))+(Kv*(Temp_scn-Temp_scn));

32          %% Varying temperature

33          G = Irr_scn;

34          T = linspace(273,348,100);

35          Imp_vary_T = Imp_scn.*(G./Irr_scn).*(1+(Ki.*(T-Temp_scn
                )));

36          Vmp_vary_T = Vmp_scn+((Ns*A).*((k*T)/q).*log(G./Irr_scn
                ))+(Kv*(T-Temp_scn));
```

Based on the simulation results above, it can therefore be inferred that the variations in ambient conditions is always accounted for.

Appendix D

Source code for the implementation of the Powell's method in MATLAB

Based on (3.12), the cost function describing the superellipse model can therefore be expressed in .m file as

```matlab
1    function retval = superellipse_cost(x)
2    %%%  KC200GT PV Panel
3    Imp = 7.61; %% Peak Maximum Current
4    Vmp = 26.3; %% Peak Maximum Voltage
5    Isc = 8.21; %% Short Circuit Current
6    Voc = 32.9; %% Open Circuit Voltage
7    retval = (Isc.*(1-(Vmp/Voc)^(x(1)))^(1/(x(2)))-Imp)^2+(
         Imp-(((x(1)*Isc)/x(2)).*((Vmp/Voc)^x(1))*((Imp/Isc)
         ^(1-x(2))))))^2;
```

Hence, the parameters of the superellipse model can be extracted using

```matlab
% n_of_var -> number of design variables
% x = [-3 2] -> starting value of x
% epsilon -> constant used for terminating the
    algorithm
% term -> linearly independent search directions
% falpha_prev -> function value at first/previous
    iteration
% search -> search direction
clear all
clc
%% Variable Definition
n_of_var = 2; %% Number of variables
x = [0.8043 0.9329]; %% Initial guess values
epsilon = 1e-6; %% Error/Tolerance
%% Add-ons 1
falpha_prev = superellipse_cost(x);
fprintf('Initial function value = %7.4f\n ',falpha_prev
    )
fprintf(' No. x-vector f(x) \n')
fprintf('_____\n')
%% Actual codes
for i = 1:n_of_var
for j = 1:n_of_var+1
```

```matlab
21          if (i==j)

22          term(i,j)=1;

23          else

24          term(i,j) = 0;

25          end

26          end

27          end

28          for i = 1: n_of_var

29          search{i} = (term(:,i))';

30          [alpha,falpha] = golden_funct1(x,search{i});

31          x = x + alpha*search{i};

32          end

33          search{i+1} = (term(:,i+1))';

34          for k = 1:200

35          xini = x;

36          i = 1;

37          while i<n_of_var+1

38          [alpha,falpha] = golden_funct1(x,search{i});

39          x = x + alpha*search{i};

40          i = i+1;

41          end

42          if abs(falpha-falpha_prev) < epsilon

43          break;
```

```matlab
44          end
45          search{i} = (x-xini);
46          [alpha,falpha] = golden_funct1(x,search{i});
47          x = x + alpha*search{i};
48          temp = search;
49          for i = 1:n_of_var
50          search{i} = temp{i+1};
51          end
52          falpha_prev = falpha;
53          fprintf('%3d %8.3f %8.3f % 8.3f \n',k,x,falpha)
54          end
55          %% Output
56          fprintf('_____\n')
```

Appendix E

Source code for the implementation of the Davidon-Fletcher-Powell method in MATLAB

```matlab
1    % Chapter 6: Numerical Techniques For Unconstrained
         Optimization
2    % Applied Optimization with Matlab Programming Dr. P.
         Venkataraman Second Edition,  John Wiley
3    %%%%Usage DFP(functname,dvar0,niter,tol,lowbound,intvl
         ,ntrials)
4    %  functname   :  where specific example can be found
5    %  dvar0       : initial guess for the design vector
6    %             : length of this vector defines number
         of variables
```

```matlab
7      %   niter        : number of random Walk iterations
8      %   tol          : tolerance for golden section and
           conjugate gradient
9      %   lowbound     : for golden section and upper bound
           calculations
10     %   intv         :  for upper bound calculations
11     %   ntrials      : for upper bound calculations
12     %-------------------------------------------
13     %%%   Example
14     % DFP('superellipse_cost',[0.5 0.5],20, 1e-08, 0,1 ,20)
15     %
16     function ReturnValue = DFP(functname,dvar0,niter,tol,
           lowbound,intvl,ntrials)
17     %%%%%%%%%%    global
18     global xfp ffp afp convgfp lenXfp A
19     %         can be used in calling program to print
20     %         iterative information if necessary
21     %%% management functions
22     clc    % position the cursor at the top of the screen
23     clf    %  closes the figure window
24     format compact   % avoid skipping a line when writing to
           the command window
```

```matlab
25          warning off  % don't report any warnings like divide by
                zero etc.
26          %%%% convergence/stopping criteria
27          e1 = 1.0e-08; e2 = 1.0e-08; e3 = 1.0e-08;
28          nvar = length(dvar0); % length of design vector or
                number of variables
29          % obtained from start vector
30          if (nvar == 2)
31          %************************
32          %  plotting contours -
33          %  only for two variables
34          %  previous generation code is left in place
35          %************************
36          % the plot is centered around initial guess
37          % with (+-) delx1, delx2 on either side
38          % this can be reset by user
39          delx1 = 6;
40          delx2 = 5;
41          x1 = (dvar0(1)-delx1):0.1:(dvar0(1)+delx1);
42          x2 = (dvar0(2)-delx2):0.1:(dvar0(2)+delx2);
43          x1len = length(x1);
44          x2len = length(x2);
45          for i = 1:x1len
```

```matlab
46          for j = 1:x2len
47          x1x2 =[x1(i) x2(j)];
48          fun(j,i) = feval(functname,x1x2);
49          end
50          end
51          % note that contour values are problem dependent
52          % the range is problem dependent
53          %*************************
54          % finished plotting contour
55          %**************************
56          end
57          %********************
58          %   Numerical Procedure
59          %********************
60          % design vector, alpha , and function value is stored
61          xfp(1,:) = dvar0;
62          x = dvar0;
63          gradNaN = 0;
64          Lc = 'r';
65          ffp(1) = feval(functname,x); % value of function at
                start
66          afp(1)=0;
67          grad = (gradfunction(functname,x)); % steepest descent
```

```matlab
68         A = eye ( nvar ) ;   % initial metric

69         % uses MATLAB built in identity matrix function

70         convgfp (1) = grad * grad ';

71         for i = 1: niter -1

72         % determine search direction

73         %    fprintf ( 'iteration number:   ') , disp ( i )

74         s = ( -A* grad ') '; % s is used afp a row vector

75         output = GoldSection_nVar ( functname , tol ,x ,s , lowbound ,
              intvl , ntrials ) ;

76         afp ( i +1) = output (1) ;

77         ffp ( i +1) = output (2) ;

78         for k = 1: nvar

79         xfp ( i +1 , k ) = output (2+ k ) ;

80         x ( k ) = output (2+ k ) ;

81         end

82         grad = ( gradfunction ( functname , x )) ;% steepest descent

83         if isnan ( grad ) == 1

84         fprintf ( '\n***  Gradient does not exist: iteration %2i
              - EXITING  ***\n ',i )

85         gradNaN = 1;

86         break ;

87         end

88         convgfp ( i +1) = grad * grad ';
```

```matlab
89          % print convergence value
90          %      fprintf('gradient length squared:  '),disp(
                convgfp(i+1));
91          %      fprintf('objective function value:  '),disp(ffp(i
                +1));
92          %***********
93          % draw lines
94          %***********
95          if (nvar == 2)
96
97          width = 12; % Width in inches
98          height = 9; % Height in inches
99          alw = 0.75; % AxesLineWidth
100         fsz = 35; % Fontsize
101         lw =1.5; % LineWidth
102         msz = 8; % MarkerSize
103         Font='time new roman';
104         pos = get(gcf, 'Position');
105         set(gcf, 'Position', [pos(1) pos(2) width*100,height
                *100]); %<- Set size
106         figure (1)
107         line([xfp(i,1) xfp(i+1,1)],[xfp(i,2) xfp(i+1,2)],'
                LineWidth',5, ...
```

126

```matlab
108             'Color',Lc);
109         set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
                ;
110         set(gca, 'FontSize', fsz, 'LineWidth',lw,'fontName','
                times'); %<- Set properties
111         xlabel('m - parameter')
112         ylabel('n - parameter')
113         itr = int2str(i);
114         x1loc = 0.5*(xfp(i,1)+xfp(i+1,1));
115         x2loc = 0.5*(xfp(i,2)+xfp(i+1,2));
116         %text(x1loc,x2loc,itr);
117         % writes iteration number on the line
118         if strcmp(Lc,'r')
119         Lc = 'r';
120         else
121         Lc = 'k';
122         end
123         axis([0 14 0 3])
124         pause(1)
125         %*********************
126         % finished drawing lines
127         %*********************
128         end
```

```matlab
129         width = 12; % Width in inches
130         height = 9; % Height in inches
131         alw = 0.75; % AxesLineWidth
132         fsz = 35; % Fontsize
133         lw =1.5; % LineWidth
134         msz = 8; % MarkerSize
135         Font='time new roman';
136         pos = get(gcf, 'Position');
137         set(gcf, 'Position', [pos(1) pos(2) width*100,height
               *100]); %<- Set size
138         %-----I-V curve plot from manufacturer's specification
               -------%
139         figure(2);
140         % clf;
141         subplot(2,1,1)
142         plot(i,xfp(i+1,1),'r-s','LineWidth',3); %% Plot of the
               Super Ellipse Curve
143         set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
               ;
144         % Removing unnecessary white space
145         text(0.26332,0.237026,'0.5','FontSize',20)
146         set(gca, 'FontSize', fsz, 'LineWidth',lw,'fontName','
               times'); %<- Set properties
```

```matlab
147        hold on;
148        ylabel('m - parameter')
149        axis([0 20 0 15])
150        subplot(2,1,2)
151        plot(i,xfp(i+1,2),'k-*','LineWidth',3); %% Plot of the
               Super Ellipse Curve
152        set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
               ;
153        % Removing unnecessary white space
154        text(0.26332,0.237026,'0.5','FontSize',20)
155        set(gca, 'FontSize', fsz, 'LineWidth',lw,'fontName','
               times'); %<- Set properties
156        hold on;
157        ylabel('n - parameter')
158        axis([0 20 0 3])
159
160        %*****************************************
161        %  apply stopping criteria
162        %*****************************************
163        delx = x - xfp(i,:);  % difference in the design vector
164        lenXfp(i+1) = sqrt(sum(delx.^2));  % length of
               difference
165        if(convgfp(i+1)<= tol)
```

```matlab
166         fprintf('Kuhn-Tucker Conditions met\n');
167         break;
168         end % convergence criteria
169         if(lenXfp(i+1)<= tol)
170         fprintf('Exit: Design not changing\n');
171         break;
172         end % convergence criteria
173         if(i == niter-1)
174         fprintf('Exit: Increase number of Iterations\n');
175         break;
176         end % convergence criteria
177         % update the metric here
178         % the semicolon hafp been added for Example 6.4
179         delx = (x - xfp(i,:))';
180         Y = (grad - gradfunction(functname,xfp(i,:)))'; %
                  column vector
181         Z = A*Y;
182         B = (delx*delx')/(delx'*Y);
183         if (norm(Y) <= 1e-08) || (norm(Z) <= 1e-08)
184         C = zeros(nvar);
185         else C = -(Z*Z')/(Y'*Z);
186         end
187         A = A + B + C;
```

```matlab
188         end
189         if gradNaN == 1
190         len = length(afp)-1;
191         else
192         len=length(afp);
193         end
194         designvar=xfp(len,:);
195         ReturnValue = [designvar ffp(len)];
```

Appendix F

Source code for the implementation of the Newton-Raphson method in MATLAB

```matlab
% n_of_var -> number of design variables
% x = [-3 2] -> starting value of x
% epsilon1, epsilon2 -> constant used for terminating
% the algorithm
% delx -> required for gradient computation
% f_prev -> function value at first/previous iteration
% deriv -> gradient vector
% sec_deriv -> hessian matrix
clear all
clc
```

```matlab
11          n_of_var = 2;

12          x = [1 1];

13          epsilon1 = 1e-6;

14          epsilon2 = 1e-6;

15          delx = 1e-3;

16          f_prev = superellipse_cost(x);

17          fprintf('Initial function value = %7.4f\n',f_prev)

18          fprintf('No. x-vector f(x) Deriv \n')

19          fprintf('_____\n')

20          for i = 1:50

21          f_prev = func_multivar(x);

22          deriv = grad_vec(x,delx,n_of_var);

23          sec_deriv = hessian(x,delx,n_of_var);

24          x = (x' - inv(sec_deriv)*deriv')';

25          f = func_multivar(x);

26          if abs(f-f_prev)<epsilon1 || norm(deriv)<epsilon2

27          break;

28          end

29          fprintf('%3d %8.3f %8.3f % 8.3f %8.3f\n',i,x,f,norm(
                deriv))

30          end

31          fprintf('%3d %8.3f %8.3f % 8.3f %8.3f \n',i,x,f,norm(
                deriv))
```

133

```
32          fprintf('_____\n')
```

Appendix G

Manufacturer's Datasheet Curves

To obtain the reference data points required for evaluating the model accuracy of the superellipse model, a specialized imaging processing software is utilized. By tracing the datasheet curves (see Figs. G.1 – G.5), all the reference data points are easily extracted using the WebPlotDigitizer website (https://apps.automeris.io/wpd/).

Nonetheless, the limitation of this approach is that the performance evaluation can therefore be only carried out at specified ambient conditions.
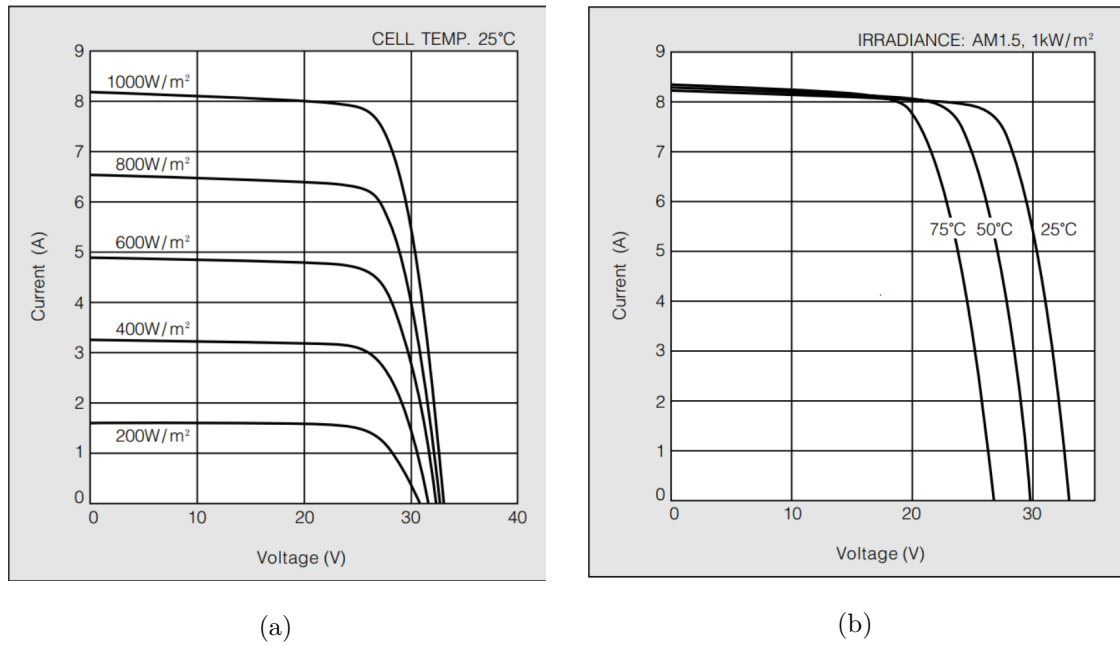
Fig. G.1 A plot of the experimental I–V curve as provided in the KC200GT manufacturer's datasheet (a) varying irradiance condition (b) varying temperature condition.
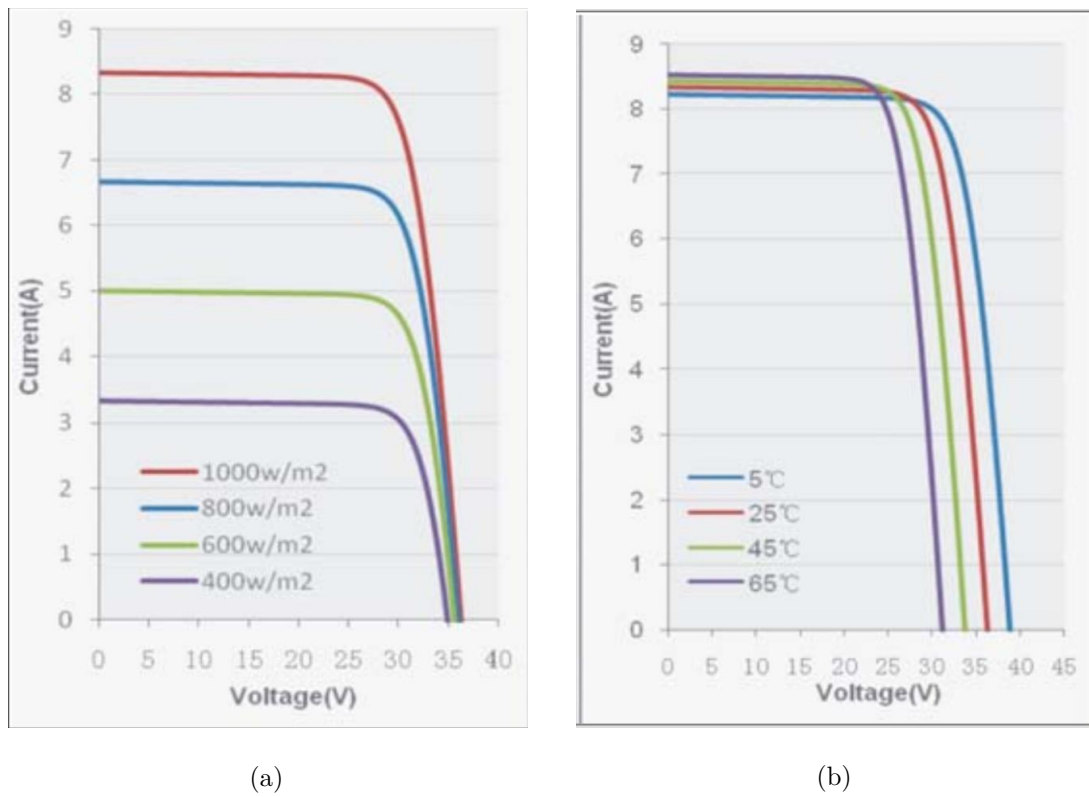
Fig. G.2 A plot of the experimental I–V curve as provided in the CS6P-230P manufacturer's datasheet (a) varying irradiance condition (b) varying temperature condition.
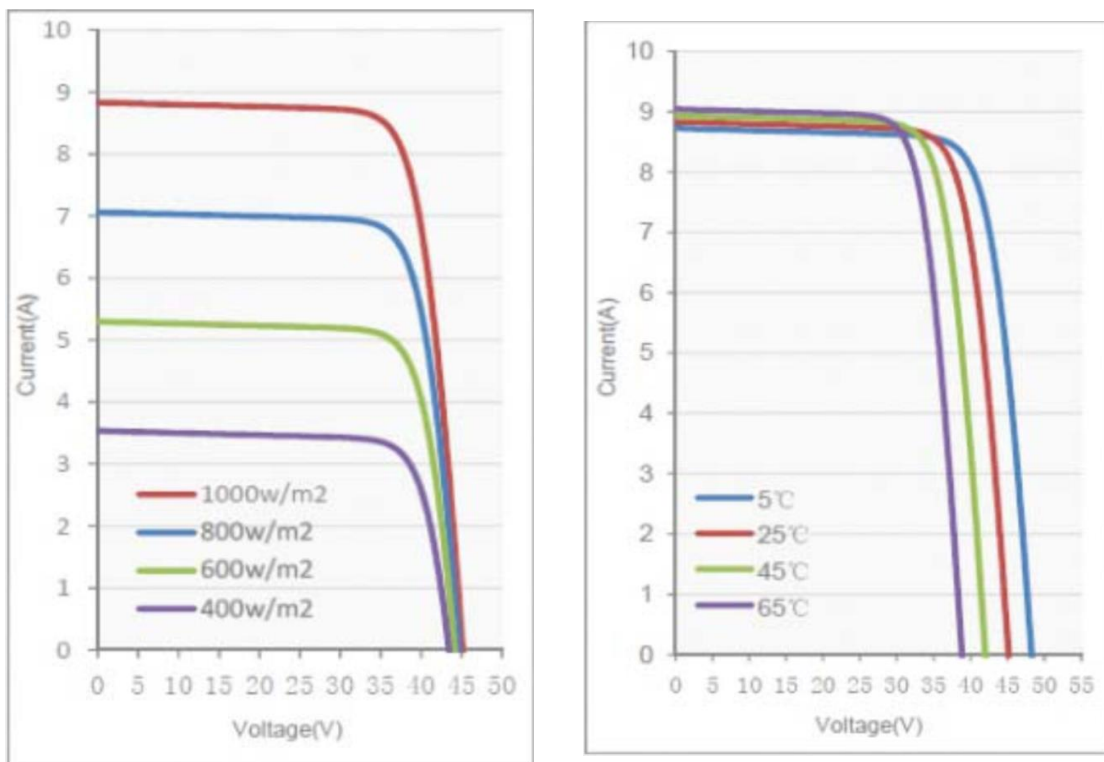
(a)

(b)

Fig. G.3 A plot of the experimental I–V curve as provided in the CS6P-305M manufacturer's datasheet (a) varying irradiance condition (b) varying temperature condition.
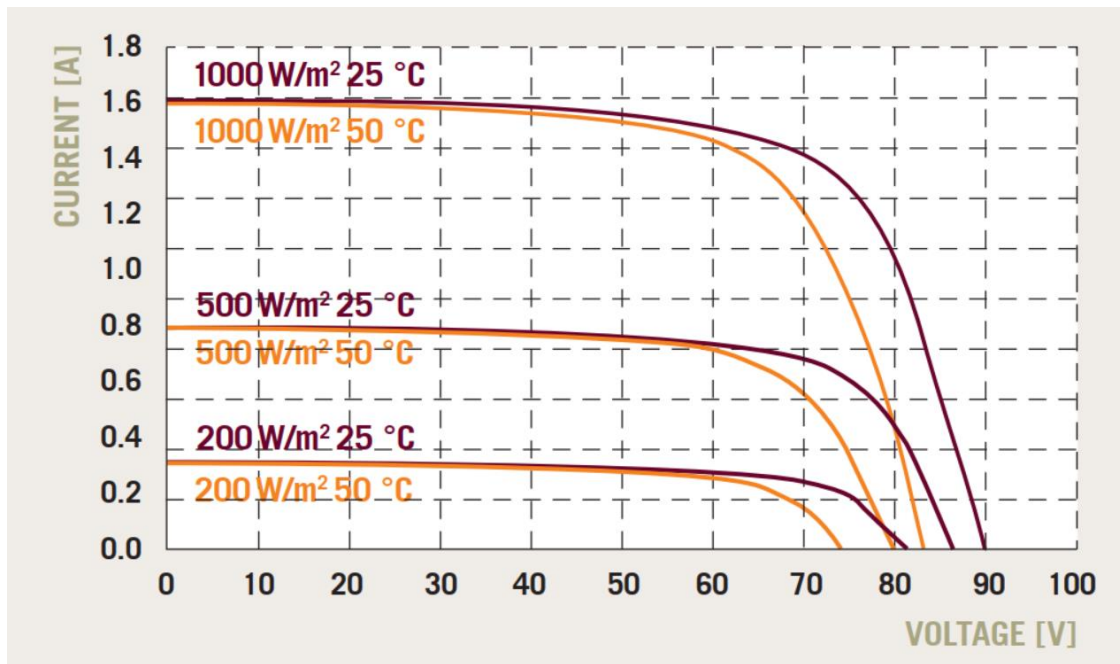
(a)
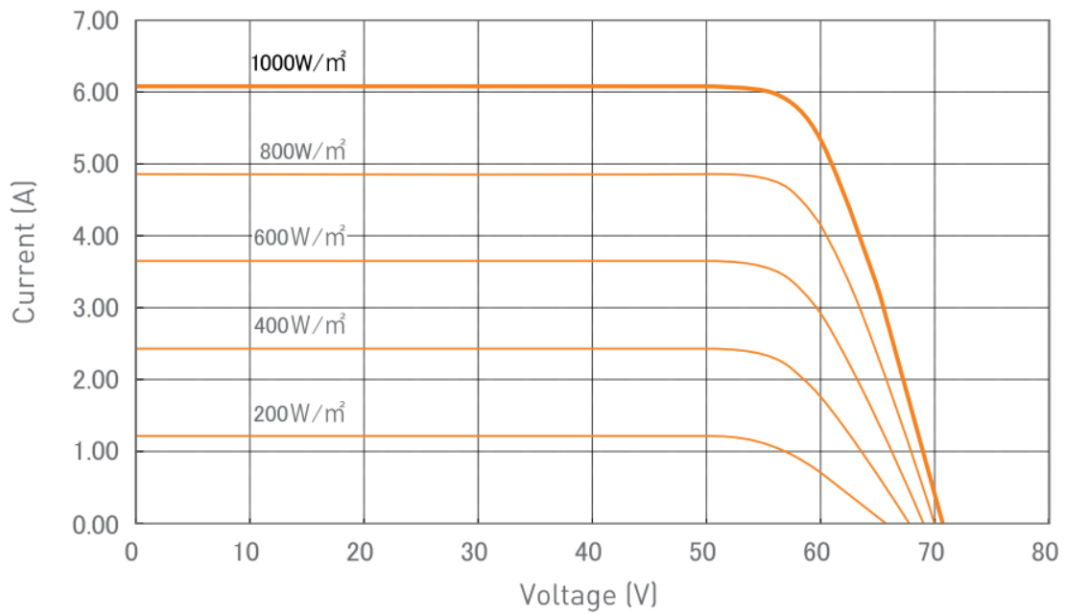
(b)

Fig. G.4 A plot of the experimental I–V curve as provided in the manufacturer's datasheet
(a) Q.SMART UFL100 (b) VBHN330SA16 PV panel.
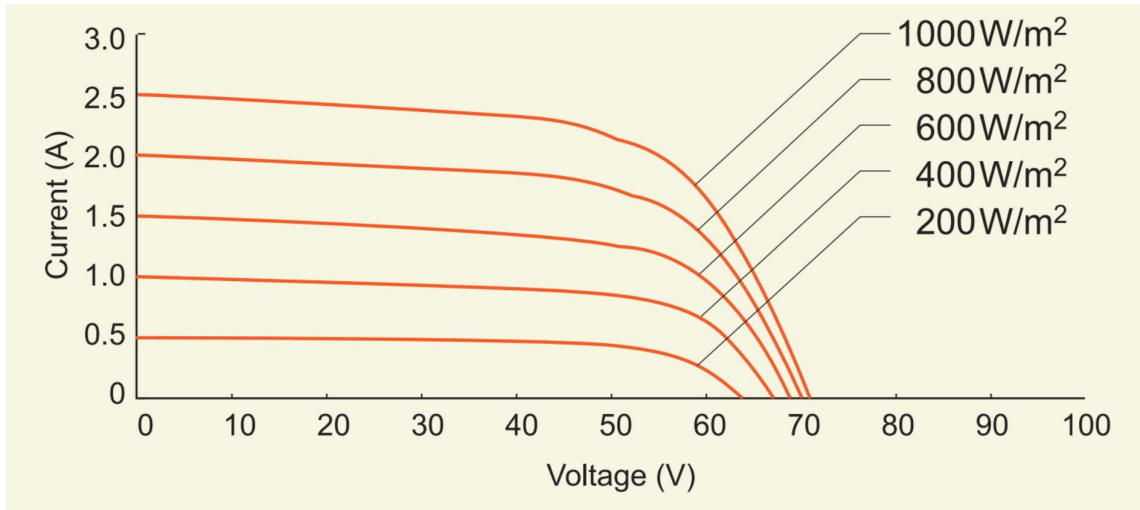
(a)
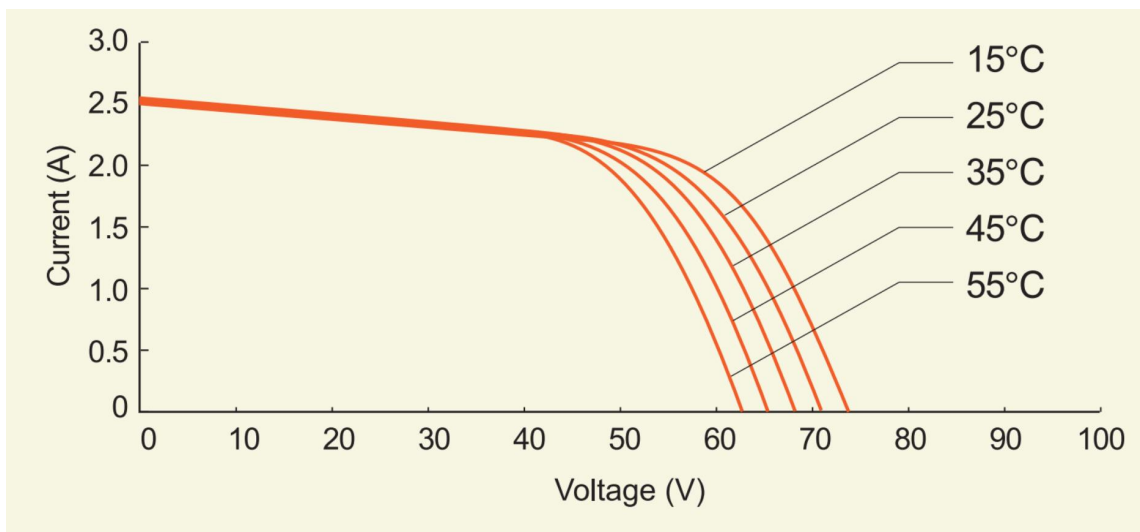
(b)

Fig. G.5 A plot of the experimental I–V curve as provided in the U-EA110 manufacturer's datasheet (a) varying irradiance condition (b) varying temperature condition.
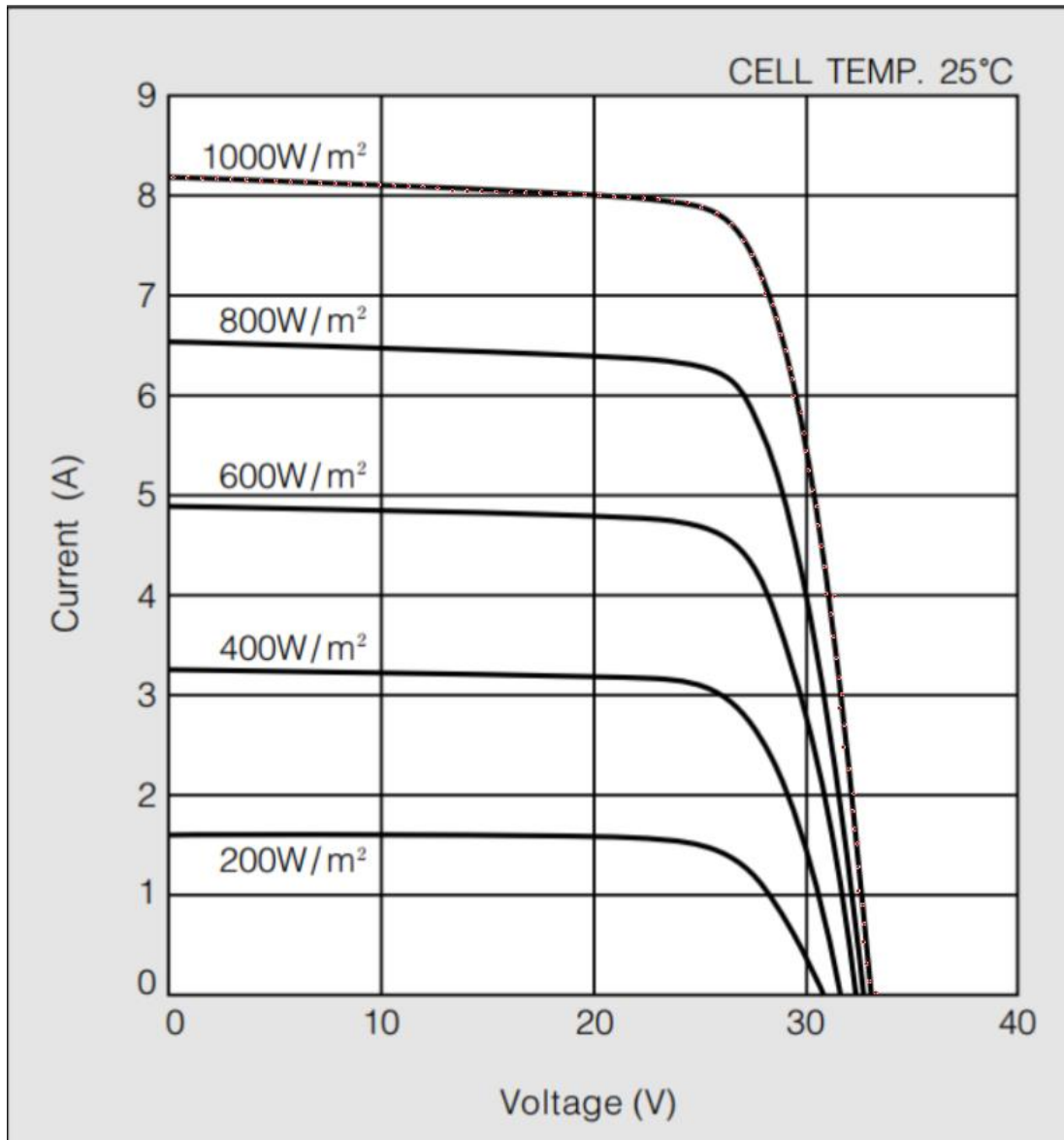
Fig. G.6 Extraction of the experimental data points of the I–V curve as provided by the KC200GT PV panel using WebPlotDigitizer.

Appendix H

IEC EN 50530 standard - overall efficiency of grid connected photovoltaic inverters

To evaluate the performance of PV installations, the accurate and effective emulation of the stationary and dynamic behavior of the PV characteristic curves are essential. The IEC EN 50530 standard provides the set of requirements and conditions that must always be met regarding the dependency of the MPP voltage on the irradiation,the relation of MPP voltage to open circuit voltage and the relation of MPP current to short circuit current.

The actual current/voltage characteristic of the PV simulator must not derive more the 1% in the power within the voltage range of $0,9 \times V_{mp}$ to $1,1 \times V_{mp}$ related to the predetermined characteristic at rated conditions.

Table H.1 General requirements on the simulated I–V characteristic of the PV generator.

	cSi-technology	Thin-film technology	Tolerance
$\dfrac{V_{mp}\big|_{G=200W/m^2}}{V_{mp}\big|_{G=1000W/m^2}}$	0,95	0,98	$\pm 1\%$
$\dfrac{V_{mp,STC}}{V_{mp,STC}}$	0,8	0,72	$\leq 1\%$
$\dfrac{I_{mp,STC}}{I_{sc,STC}}$	0,9	0,8	$\leq 1\%$

Appendix I

MATLAB code for computing the absolute errors at MPP in accordance with the IEC EN 50530 standard

```matlab
1    clear all;
2    clc;
3    %% KC200GT PV panel Datasheet
4    Imp = 7.61; %% Peak Maximum Current
5    Vmp = 26.3; %% Peak Maximum Voltage
6    Isc = 8.21; %% Short Circuit Current
7    Voc = 32.9; %% Opem Circuit Voltage
```

```matlab
8       %% Experimemtal Data
9       filename = 'KC200GT.xlsx'; %% Extracted data points
            from the KC200GT PV panel datasheet
10      V_datasheet = xlsread(filename,'A:A');
11      I_datasheet = xlsread(filename, 'B:B');
12      P_datasheet = V_datasheet.*I_datasheet;
13      % Obtain the maximum power point
14      maxF_datasheet = max(P_datasheet);   % Find max value
            over all elements.
15      indexOfFirstMax_datasheet = find(P_datasheet ==
            maxF_datasheet, 1, 'first');   % Get first element
            that is the max.
16      % Get the x and y values at that index.
17      maxP_datasheet = P_datasheet(indexOfFirstMax_datasheet)
            ;
18      maxV_datasheet = V_datasheet(indexOfFirstMax_datasheet)
            ;
19      maxI_datasheet = I_datasheet(indexOfFirstMax_datasheet)
            ;
20      %% Newton-Raphson Method
21      A = Voc;
22      B = Isc;
23      v=linspace(0,Voc,4096);
```

```matlab
24          %% Parameters of the superellipse
25          m=13.0287;
26          n=0.7416;
27          i=[];
28          i=Isc.*exp((1/n).*log(1-exp(m.*log(v./Voc))));
29          p = v.*i;
30          % Obtain the maximum power point
31          maxF = max(p);  % Find max value over all elements.
32          indexOfFirstMax = find(p == maxF, 1, 'first');  % Get
                first element that is the max.
33          % Get the x and y values at that index.
34          maxP = p(indexOfFirstMax); %% Power at MPP
35          maxV = v(indexOfFirstMax); %% Voltage at MPP
36          maxI = i(indexOfFirstMax); %% Current at MPP
37          %% IEC EN 50530 standard
38          maxP_calculate = maxP;
39          maxV_calculate = maxV;
40          maxI_calculate = maxI;
41          %% Trapizodial Rule Equations
42          %----Parameter definition-------%
43          n_interval_calculate = 50;
44          b_interval_calculate = Vmp+(0.1*Vmp);
45          a_interval_calculate = Vmp-(0.1*Vmp);
```

```matlab
46        h_interval_calculate = (b_interval_calculate -
             a_interval_calculate)./n_interval_calculate;
47        q_0_calculate =(b_interval_calculate -
             a_interval_calculate)./(2*n_interval_calculate);
48        %% Current Error near MPP
49        i_error_superellipse0_calculate = maxI_calculate -
             maxI_datasheet;
50        i_error_superellipse1_calculate =
             i_error_superellipse0_calculate./maxI_datasheet;
51        i_error_superellipse_ABS_calculate = abs(
             i_error_superellipse1_calculate);
52        superellipse_f_aa_i_calculate =
             i_error_superellipse_ABS_calculate.*
             a_interval_calculate;
53        superellipse_f_bb_i_calculate =
             i_error_superellipse_ABS_calculate.*
             b_interval_calculate;
54        superellipse_f_aa_i1_calculate = 2*(
             i_error_superellipse_ABS_calculate.*23.67);
55        superellipse_f_aa_i2_calculate = 2*(
             i_error_superellipse_ABS_calculate.*23.7752);
56        superellipse_f_aa_i3_calculate = 2*(
             i_error_superellipse_ABS_calculate.*23.8804);
```

```
57          superellipse_f_aa_i4_calculate = 2*(

                i_error_superellipse_ABS_calculate.*23.9856);

58          superellipse_f_aa_i5_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.0908);

59          superellipse_f_aa_i6_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.196);

60          superellipse_f_aa_i7_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.3012);

61          superellipse_f_aa_i8_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.4064);

62          superellipse_f_aa_i9_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.5116);

63          superellipse_f_aa_i10_calculate = 2*(

                i_error_superellipse_ABS_calculate.*24.6168);

64          superellipse_f_aa_i1_10_calculate =

                superellipse_f_aa_i1_calculate+

                superellipse_f_aa_i2_calculate+

                superellipse_f_aa_i3_calculate+

                superellipse_f_aa_i4_calculate+

                superellipse_f_aa_i5_calculate ...

65          +superellipse_f_aa_i6_calculate+

                superellipse_f_aa_i7_calculate+

                superellipse_f_aa_i8_calculate+
```

```
                    superellipse_f_aa_i9_calculate+

                    superellipse_f_aa_i10_calculate;
66          superellipse_f_aa_i11_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*24.722);
67          superellipse_f_aa_i12_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*24.8272);
68          superellipse_f_aa_i13_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*24.9324);
69          superellipse_f_aa_i14_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.0376);
70          superellipse_f_aa_i15_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.1428);
71          superellipse_f_aa_i16_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.248);
72          superellipse_f_aa_i17_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.3532);
73          superellipse_f_aa_i18_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.4584);
74          superellipse_f_aa_i19_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.5636);
75          superellipse_f_aa_i20_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*25.6688);
```

```
76          superellipse_f_aa_i11_20_calculate =

                superellipse_f_aa_i11_calculate+

                superellipse_f_aa_i12_calculate+

                superellipse_f_aa_i13_calculate+

                superellipse_f_aa_i14_calculate+

                superellipse_f_aa_i15_calculate ...
77          +superellipse_f_aa_i16_calculate+

                superellipse_f_aa_i17_calculate+

                superellipse_f_aa_i18_calculate+

                superellipse_f_aa_i19_calculate+

                superellipse_f_aa_i20_calculate;
78          superellipse_f_aa_i21_calculate = 2*(

                i_error_superellipse_ABS_calculate.*25.774);
79          superellipse_f_aa_i22_calculate = 2*(

                i_error_superellipse_ABS_calculate.*25.8792);
80          superellipse_f_aa_i23_calculate = 2*(

                i_error_superellipse_ABS_calculate.*25.9844);
81          superellipse_f_aa_i24_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.0896);
82          superellipse_f_aa_i25_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.1948);
83          superellipse_f_aa_i26_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.3);
```

```
84          superellipse_f_aa_i27_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.4052);
85          superellipse_f_aa_i28_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.5104);
86          superellipse_f_aa_i29_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.6156);
87          superellipse_f_aa_i30_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.7208);
88          superellipse_f_aa_i21_30_calculate =
                superellipse_f_aa_i21_calculate+
                superellipse_f_aa_i22_calculate+
                superellipse_f_aa_i23_calculate+
                superellipse_f_aa_i24_calculate+
                superellipse_f_aa_i25_calculate ...
89          +superellipse_f_aa_i26_calculate+
                superellipse_f_aa_i27_calculate+
                superellipse_f_aa_i28_calculate+
                superellipse_f_aa_i29_calculate+
                superellipse_f_aa_i30_calculate;
90          superellipse_f_aa_i31_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.826);
91          superellipse_f_aa_i32_calculate = 2*(
                i_error_superellipse_ABS_calculate.*26.9312);
```

```matlab
92          superellipse_f_aa_i33_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.0364);
93          superellipse_f_aa_i34_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.1416);
94          superellipse_f_aa_i35_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.2468);
95          superellipse_f_aa_i36_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.352);
96          superellipse_f_aa_i37_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.4572);
97          superellipse_f_aa_i38_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.5624);
98          superellipse_f_aa_i39_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.6676);
99          superellipse_f_aa_i40_calculate = 2*(
                i_error_superellipse_ABS_calculate.*27.7728);
100         superellipse_f_aa_i31_40_calculate =
                superellipse_f_aa_i31_calculate+
                superellipse_f_aa_i32_calculate+
                superellipse_f_aa_i33_calculate+
                superellipse_f_aa_i34_calculate+
                superellipse_f_aa_i35_calculate ...
```

```
101         +superellipse_f_aa_i36_calculate+
            superellipse_f_aa_i37_calculate+
            superellipse_f_aa_i38_calculate+
            superellipse_f_aa_i39_calculate+
            superellipse_f_aa_i40_calculate;
102         superellipse_f_aa_i41_calculate = 2*(
            i_error_superellipse_ABS_calculate.*27.878);
103         superellipse_f_aa_i42_calculate = 2*(
            i_error_superellipse_ABS_calculate.*27.9832);
104         superellipse_f_aa_i43_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.0884);
105         superellipse_f_aa_i44_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.1936);
106         superellipse_f_aa_i45_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.2988);
107         superellipse_f_aa_i46_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.404);
108         superellipse_f_aa_i47_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.5092);
109         superellipse_f_aa_i48_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.6144);
110         superellipse_f_aa_i49_calculate = 2*(
            i_error_superellipse_ABS_calculate.*28.7196);
```

```
111        superellipse_f_aa_i50_calculate = 2*(
               i_error_superellipse_ABS_calculate.*28.8248);
112        superellipse_f_aa_i41_50_calculate =
               superellipse_f_aa_i41_calculate+
               superellipse_f_aa_i42_calculate+
               superellipse_f_aa_i43_calculate+
               superellipse_f_aa_i44_calculate+
               superellipse_f_aa_i45_calculate ...
113        +superellipse_f_aa_i46_calculate+
               superellipse_f_aa_i47_calculate+
               superellipse_f_aa_i48_calculate+
               superellipse_f_aa_i49_calculate+
               superellipse_f_aa_i50_calculate;
114        superellipse_f_aa_ii_calculate =
               superellipse_f_aa_i1_10_calculate+
               superellipse_f_aa_i11_20_calculate+
               superellipse_f_aa_i21_30_calculate+
               superellipse_f_aa_i31_40_calculate+
               superellipse_f_aa_i41_50_calculate;
115        superellipse_f_aa_iii_calculate =
               superellipse_f_aa_i_calculate+
               superellipse_f_aa_ii_calculate+
               superellipse_f_bb_i_calculate;
```

```matlab
116         superellipse_q_1_calculate = q_0_calculate.*
                superellipse_f_aa_iii_calculate;
117         superellipse_q_2_calculate = 1/(0.2*Vmp);
118         E_I_superellipse_calculate = superellipse_q_2_calculate
                .*superellipse_q_1_calculate;
119         %% Power error near MPP
120         p_error_superellipse0_calculate = maxP_calculate-
                maxP_datasheet;
121         p_error_superellipse1_calculate =
                p_error_superellipse0_calculate./maxP_datasheet;
122         p_error_superellipse_ABS_calculate = abs(
                p_error_superellipse1_calculate);
123         superellipse_f_cc_i_calculate =
                p_error_superellipse_ABS_calculate.*
                a_interval_calculate;
124         superellipse_f_dd_i_calculate =
                p_error_superellipse_ABS_calculate.*
                b_interval_calculate;
125         superellipse_f_cc_i1_calculate = 2*(
                i_error_superellipse_ABS_calculate.*23.67);
126         superellipse_f_cc_i2_calculate = 2*(
                i_error_superellipse_ABS_calculate.*23.7752);
```

```matlab
127        superellipse_f_cc_i3_calculate = 2*(

               i_error_superellipse_ABS_calculate.*23.8804);

128        superellipse_f_cc_i4_calculate = 2*(

               i_error_superellipse_ABS_calculate.*23.9856);

129        superellipse_f_cc_i5_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.0908);

130        superellipse_f_cc_i6_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.196);

131        superellipse_f_cc_i7_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.3012);

132        superellipse_f_cc_i8_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.4064);

133        superellipse_f_cc_i9_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.5116);

134        superellipse_f_cc_i10_calculate = 2*(

               i_error_superellipse_ABS_calculate.*24.6168);

135        superellipse_f_cc_i1_10_calculate =

               superellipse_f_cc_i1_calculate+

               superellipse_f_cc_i2_calculate+

               superellipse_f_cc_i3_calculate+

               superellipse_f_cc_i4_calculate+

               superellipse_f_cc_i5_calculate ...
```

```
136            +superellipse_f_cc_i6_calculate+
               superellipse_f_cc_i7_calculate+
               superellipse_f_cc_i8_calculate+
               superellipse_f_cc_i9_calculate+
               superellipse_f_cc_i10_calculate;
137        superellipse_f_cc_i11_calculate = 2*(
               i_error_superellipse_ABS_calculate.*24.722);
138        superellipse_f_cc_i12_calculate = 2*(
               i_error_superellipse_ABS_calculate.*24.8272);
139        superellipse_f_cc_i13_calculate = 2*(
               i_error_superellipse_ABS_calculate.*24.9324);
140        superellipse_f_cc_i14_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.0376);
141        superellipse_f_cc_i15_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.1428);
142        superellipse_f_cc_i16_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.248);
143        superellipse_f_cc_i17_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.3532);
144        superellipse_f_cc_i18_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.4584);
145        superellipse_f_cc_i19_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.5636);
```

```
146        superellipse_f_cc_i20_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.6688);
147        superellipse_f_cc_i11_20_calculate =
               superellipse_f_cc_i11_calculate+
               superellipse_f_cc_i12_calculate+
               superellipse_f_cc_i13_calculate+
               superellipse_f_cc_i14_calculate+
               superellipse_f_cc_i15_calculate ...
148        +superellipse_f_cc_i16_calculate+
               superellipse_f_cc_i17_calculate+
               superellipse_f_cc_i18_calculate+
               superellipse_f_cc_i19_calculate+
               superellipse_f_cc_i20_calculate;
149        superellipse_f_cc_i21_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.774);
150        superellipse_f_cc_i22_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.8792);
151        superellipse_f_cc_i23_calculate = 2*(
               i_error_superellipse_ABS_calculate.*25.9844);
152        superellipse_f_cc_i24_calculate = 2*(
               i_error_superellipse_ABS_calculate.*26.0896);
153        superellipse_f_cc_i25_calculate = 2*(
               i_error_superellipse_ABS_calculate.*26.1948);
```

```
154        superellipse_f_cc_i26_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.3);

155        superellipse_f_cc_i27_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.4052);

156        superellipse_f_cc_i28_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.5104);

157        superellipse_f_cc_i29_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.6156);

158        superellipse_f_cc_i30_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.7208);

159        superellipse_f_cc_i21_30_calculate =

                superellipse_f_cc_i21_calculate+

                superellipse_f_cc_i22_calculate+

                superellipse_f_cc_i23_calculate+

                superellipse_f_cc_i24_calculate+

                superellipse_f_cc_i25_calculate ...

160       +superellipse_f_cc_i26_calculate+

                superellipse_f_cc_i27_calculate+

                superellipse_f_cc_i28_calculate+

                superellipse_f_cc_i29_calculate+

                superellipse_f_cc_i30_calculate;

161        superellipse_f_cc_i31_calculate = 2*(

                i_error_superellipse_ABS_calculate.*26.826);
```

```
162        superellipse_f_cc_i32_calculate = 2*(

               i_error_superellipse_ABS_calculate.*26.9312);

163        superellipse_f_cc_i33_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.0364);

164        superellipse_f_cc_i34_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.1416);

165        superellipse_f_cc_i35_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.2468);

166        superellipse_f_cc_i36_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.352);

167        superellipse_f_cc_i37_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.4572);

168        superellipse_f_cc_i38_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.5624);

169        superellipse_f_cc_i39_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.6676);

170        superellipse_f_cc_i40_calculate = 2*(

               i_error_superellipse_ABS_calculate.*27.7728);

171

172        superellipse_f_cc_i31_40_calculate =

               superellipse_f_cc_i31_calculate+

               superellipse_f_cc_i32_calculate+

               superellipse_f_cc_i33_calculate+
```

```
                    superellipse_f_cc_i34_calculate+

                    superellipse_f_cc_i35_calculate ...

173             +superellipse_f_cc_i36_calculate+

                    superellipse_f_cc_i37_calculate+

                    superellipse_f_cc_i38_calculate+

                    superellipse_f_cc_i39_calculate+

                    superellipse_f_cc_i40_calculate;

174         superellipse_f_cc_i41_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*27.878);

175         superellipse_f_cc_i42_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*27.9832);

176         superellipse_f_cc_i43_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.0884);

177         superellipse_f_cc_i44_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.1936);

178         superellipse_f_cc_i45_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.2988);

179         superellipse_f_cc_i46_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.404);

180         superellipse_f_cc_i47_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.5092);

181         superellipse_f_cc_i48_calculate = 2*(

                    i_error_superellipse_ABS_calculate.*28.6144);
```

```
182        superellipse_f_cc_i49_calculate = 2*(
               i_error_superellipse_ABS_calculate.*28.7196);
183        superellipse_f_cc_i50_calculate = 2*(
               i_error_superellipse_ABS_calculate.*28.8248);
184        superellipse_f_cc_i41_50_calculate =
               superellipse_f_cc_i41_calculate+
               superellipse_f_cc_i42_calculate+
               superellipse_f_cc_i43_calculate+
               superellipse_f_cc_i44_calculate+
               superellipse_f_cc_i45_calculate ...
185        +superellipse_f_cc_i46_calculate+
               superellipse_f_cc_i47_calculate+
               superellipse_f_cc_i48_calculate+
               superellipse_f_cc_i49_calculate+
               superellipse_f_cc_i50_calculate;
186        superellipse_f_cc_ii_calculate =
               superellipse_f_cc_i1_10_calculate+
               superellipse_f_cc_i11_20_calculate+
               superellipse_f_cc_i21_30_calculate+
               superellipse_f_cc_i31_40_calculate+
               superellipse_f_cc_i41_50_calculate;
187        superellipse_f_cc_iii_calculate =
               superellipse_f_cc_i_calculate+
```

```matlab
            superellipse_f_cc_ii_calculate+

            superellipse_f_dd_i_calculate ;

188         superellipse_y_1_calculate = q_0_calculate.*

            superellipse_f_cc_iii_calculate ;

189         superellipse_y_2_calculate = 1/(0.2*Vmp);

190         %% Absolute Power Error at MPP

191         E_P_superellipse_calculate = superellipse_y_2_calculate

            .*superellipse_y_1_calculate ;
```

Appendix J

MATLAB code for computing the errors across the full range of the PV characteristic curves

```matlab
1    clear all;
2    clc;
3    %% KC200GT PV panel Datasheet
4    Imp = 7.61; %% Peak Maximum Current
5    Vmp = 26.3; %% Peak Maximum Voltage
6    Isc = 8.21; %% Short Circuit Current
7    Voc = 32.9; %% Opem Circuit Voltage
8    filename = 'KC200GT.xlsx'; %% Extracted data points of
         the I--V curve
9    V_datasheet = xlsread(filename,'A:A');
```

```matlab
10          I_datasheet = xlsread(filename, 'B:B');
11          P_datasheet = V_datasheet.*I_datasheet;
12          maxF_datasheet = max(P_datasheet);   % Find max value
               over all elements.
13          indexOfFirstMax_datasheet = find(P_datasheet ==
               maxF_datasheet, 1, 'first');   % Get first element
               that is the max.
14          % Get the x and y values at that index.
15          maxP_datasheet = P_datasheet(indexOfFirstMax_datasheet)
               ;
16          maxV_datasheet = V_datasheet(indexOfFirstMax_datasheet)
               ;
17          maxI_datasheet = I_datasheet(indexOfFirstMax_datasheet)
               ;
18          %% Superellipse Approximation
19          v=linspace(0,Voc,97);
20          m=13.0287;
21          n=0.7416;
22          i=[];
23          i=Isc.*exp((1/n).*log(1-exp(m.*log(v./Voc))));
24          p = v.*i;
25          maxF_FRAC = max(p);   % Find max value over all elements
               .
```

164

```matlab
26          indexOfFirstMax_FRAC = find(p == maxF_FRAC, 1, 'first')
               ;  % Get first element that is the max.
27          % Get the x and y values at that index.
28          maxP_FRAC = p(indexOfFirstMax_FRAC);
29          maxV_FRAC = v(indexOfFirstMax_FRAC);
30          maxI_FRAC = i(indexOfFirstMax_FRAC);
31          %% I--V curve Error
32          IV_dy = I_datasheet-transpose(i); % error
33          IV_abs_dy = abs(IV_dy);    % absolute error
34          IV_relerr = IV_abs_dy./I_datasheet;  % relative error
35          IV_mean_err = mean(IV_abs_dy);    % mean absolute error
36          IV_MSE = mean(IV_dy.^2);        % Mean square error
37          IV_RMSE = sqrt(IV_MSE); % Root mean square error
38          %% P--V curve Error
39          PV_dy = P_datasheet-transpose(p); % error
40          PV_abs_dy = abs(PV_dy);    % absolute error
41          PV_relerr = PV_abs_dy./P_datasheet;  % relative error
42          PV_mean_err_FRAC = mean(PV_abs_dy);    % mean absolute
               error
43          PV_MSE = mean(PV_dy.^2);        % Mean square error
44          PV_RMSE = sqrt(PV_MSE); % Root mean square error
```

Biography

Tofopefun Nifise Olayiwola is a current Master's student at the Energy Conversion and Circuit Laboratory (ECCL), under Professor Sung-Jin Choi at the University of Ulsan, South Korea. In 2019, Tofopefun received his Bachelor of Engineering degree in Electrical and Electronic Engineering at the University of Ilorin, Nigeria. His interest in photovoltaic systems was started during his undergraduate study in which he researched mitigating the effects of soiling. Currently, Tofopefun's research interests include modeling and approximation of photovoltaic panels, maximum power point tracking, and power converters for PV systems.

For further information, please contact

(A) tofopefungraduate@gmail.com

(B) LinkedIn

(C) ResearchGate

(D) Google Scholar

Publications

Journals

[1] T. N. Olayiwola and S.-J. Choi, "Superellipse model: An accurate and easy-to-fitempirical model for photovoltaic panels," *Solar Energy*, vol. 262, p. 111749, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X23003547.

Conferences

[2] T. N. Olayiwola and S.-J Choi, "Effective Approximation of the Photovoltaic Characteristic Curves using a Double-shaped Superellipse," in *11th International Conference on Power Electronics - ECCE Asia 2023*, May 2023. https://www.researchgate.net/publication/371044807_Effective_Approximation_of_the_Photovoltaic_Characteristic_Curves_using_a_Double-shaped_Superellipse

[3] T. N. Olayiwola and S.-J Choi, "Fast I-V Curve Approximation Technique for Photovoltaic Panels using Superellipse," *In proceedings of the KIPE Conference*, pp. 408-409, 2022. https://www.researchgate.net/publication/361810779_Fast_I-V_Curve_Approximation_Technique_for_Photovoltaic_Panels_using_Superellipse

[4] T. N. Olayiwola and S.-J Choi, "Fast Calculation of I-V Curve Computation Technique for Photovoltaic Panels using Superellipse," *In proceedings of the KIPE Power Electronics Symposium*, 2022. **(Poster only)**

References

[1] M. Afolayan, T. Olayiwola, Q. Nurudeen, O. Ibrahim, and I. Madugu, "Performance evaluation of soiling mitigation technique for solar panels." *Arid Zone Journal of Engineering, Technology and Environment*, vol. 16, no. 4, pp. 685–698, 2020.

[2] T. N. Olayiwola and S.-J. Choi, "Fast iv curve approximation technique for photovoltaic panels using superellipse," 전력전자학술대회 논문집, p. 7, 2022.

[3] J.-Y. Park and S.-J. Choi, "A novel datasheet-based parameter extraction method for a single-diode photovoltaic array model," *Solar Energy*, vol. 122, pp. 1235–1244, 2015.

[4] E. Batzelis, "Non-iterative methods for the extraction of the single-diode model parameters of photovoltaic modules: A review and comparative assessment," *Energies*, vol. 12, no. 3, p. 358, 2019.

[5] A. Laudani, F. R. Fulginei, and A. Salvini, "Identification of the one-diode model for photovoltaic modules from datasheet values," *Solar energy*, vol. 108, pp. 432–446, 2014.

[6] A. Laudani, F. Mancilla-David, F. Riganti-Fulginei, and A. Salvini, "Reduced-form of the photovoltaic five-parameter model for efficient computation of parameters," *Solar Energy*, vol. 97, pp. 122–127, 2013.

[7] A. H. ALQahtani, "A simplified and accurate photovoltaic module parameters extraction approach using matlab," in *2012 IEEE International Symposium on Industrial Electronics.* IEEE, 2012, pp. 1748–1753.

[8] S. Lineykin, M. Averbukh, and A. Kuperman, "An improved approach to extract the single-diode equivalent circuit parameters of a photovoltaic cell/panel," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 282–289, 2014.

[9] M. G. Villalva, J. R. Gazoli, and E. Ruppert Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Transactions on power electronics*, vol. 24, no. 5, pp. 1198–1208, 2009.

[10] E. I. Batzelis, G. Anagnostou, C. Chakraborty, and B. C. Pal, "Computation of the lambert w function in photovoltaic modeling," in *ELECTRIMACS 2019: Selected Papers-Volume 1.* Springer, 2020, pp. 583–595.

[11] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambert w function," *Advances in Computational mathematics*, vol. 5, pp. 329–359, 1996.

[12] S. Winitzki, "Uniform approximations for transcendental functions," in *Computational Science and Its Applications—ICCSA 2003: International Conference Montreal, Canada, May 18–21, 2003 Proceedings, Part I 3.* Springer, 2003, pp. 780–789.

[13] A. Jain and A. Kapoor, "Exact analytical solutions of the parameters of real solar cells using lambert w-function," *Solar Energy Materials and Solar Cells*, vol. 81, no. 2, pp. 269–277, 2004.

[14] E. I. Batzelis, I. A. Routsolias, and S. A. Papathanassiou, "An explicit pv string model based on the lambert w function and simplified mpp expressions for operation under

partial shading," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 1, pp. 301–312, 2013.

[15] E. I. Batzelis and S. A. Papathanassiou, "A method for the analytical extraction of the single-diode pv model parameters," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 504–512, 2015.

[16] D. Barry, J.-Y. Parlange, L. Li, H. Prommer, C. Cunningham, and F. Stagnitti, "Analytical approximations for real values of the lambert w-function," *Mathematics and Computers in Simulation*, vol. 53, no. 1-2, pp. 95–103, 2000.

[17] E. Moshksar and T. Ghanbari, "A model-based algorithm for maximum power point tracking of pv systems using exact analytical solution of single-diode equivalent model," *Solar Energy*, vol. 162, pp. 117–131, 2018.

[18] H. M. Ridha, "Parameters extraction of single and double diodes photovoltaic models using marine predators algorithm and lambert w function," *Solar Energy*, vol. 209, pp. 674–693, 2020.

[19] M. Elyaqouti, E. Arjdal, A. Ibrahim, H. Abdul-Ghaffar, R. Aboelsaud, S. Obukhov, A. A. Z. Diab *et al.*, "Parameters identification and optimization of photovoltaic panels under real conditions using lambert w-function," *Energy Reports*, vol. 7, pp. 9035–9045, 2021.

[20] W. Borsch-Supan, "On the evaluation of the function," *J. of Research of NBS*, vol. 65, 1961.

[21] H. K. Mehta and A. K. Panchal, "Pv panel performance evaluation via accurate v–i polynomial with efficient computation," *IEEE Journal of Photovoltaics*, vol. 11, no. 6, pp. 1519–1527, 2021.

[22] K.-i. Ishibashi, Y. Kimura, and M. Niwano, "An extensively valid and stable method for derivation of all parameters of a solar cell from a single current-voltage characteristic," *Journal of applied physics*, vol. 103, no. 9, p. 094507, 2008.

[23] S.-x. Lun, C.-j. Du, G.-h. Yang, S. Wang, T.-t. Guo, J.-s. Sang, and J.-p. Li, "An explicit approximate i–v characteristic model of a solar cell based on padé approximants," *Solar energy*, vol. 92, pp. 147–159, 2013.

[24] S. Fan, "A new extracting formula and a new distinguishing means on the one variable cubic equation," *Natural science journal of Hainan teachers college*, vol. 2, no. 2, p. 91, 1989.

[25] A. K. Das, "An explicit j–v model of a solar cell using equivalent rational function form for simple estimation of maximum power point voltage," *Solar energy*, vol. 98, pp. 400–403, 2013.

[26] L. E. Mathew and A. K. Panchal, "An exact and explicit pv panel curve computation assisted by two 2-port networks," *Solar Energy*, vol. 240, pp. 280–289, 2022.

[27] W. Xiao, W. G. Dunford, and A. Capel, "A novel modeling method for photovoltaic cells," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No. 04CH37551)*, vol. 3. IEEE, 2004, pp. 1950–1956.

[28] A. Bellini, S. Bifaretti, V. Iacovone, and C. Cornaro, "Simplified model of a photovoltaic module," in *2009 Applied Electronics*. IEEE, 2009, pp. 47–51.

[29] T. N. Olayiwola and S.-J. Choi, "Effective approximation of the photovoltaic character-istic curves using a double-shaped superellipse."

[30] M. H. Mobarak and J. Bauman, "A fast parabolic-assumption algorithm for global mppt of photovoltaic systems under partial shading conditions," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 8066–8079, 2021.

[31] M. Bashahu and P. Nkundabakura, "Review and tests of methods for the determination of the solar cell junction ideality factors," *Solar energy*, vol. 81, no. 7, pp. 856–863, 2007.

[32] E. Meyer, "Extraction of saturation current and ideality factor from measuring voc and isc of photovoltaic modules," *International Journal of Photoenergy*, vol. 2017, 2017.

[33] H. Bayhan and M. Bayhan, "A simple approach to determine the solar cell diode ideality factor under illumination," *Solar Energy*, vol. 85, no. 5, pp. 769–775, 2011.

[34] F. A. Lindholm, J. G. Fossum, and E. L. Burgess, "Application of the superposition principle to solar-cell analysis," *IEEE transactions on electron devices*, vol. 26, no. 3, pp. 165–171, 1979.

[35] P. Venkataraman, *Applied optimization with MATLAB programming*. John Wiley & Sons, 2009.

[36] D. Tabak and B. C. Kuo, *Optimal control by mathematical programming*. SRL Publishing Company, 1971.

[37] C. Lemaréchal, "Cauchy and the gradient method," *Doc Math Extra*, vol. 251, no. 254, p. 10, 2012.

[38] M. R. Hestenes, E. Stiefel *et al.*, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[39] W. C. Davidon, "Variable metric method for minimization," *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 1–17, 1991.

[40] R. Fletcher, *Practical methods of optimization.* John Wiley & Sons, 2013.

[41] T. N. Olayiwola and S.-J. Choi, "Superellipse model: An accurate and easy-to-fit empirical model for photovoltaic panels," *Solar Energy*, vol. 262, p. 111749, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X23003547

[42] S. G. Johnson, "Notes on the convergence of trapezoidal-rule quadrature," 2010.

[43] I. Mysovskikh, "Trapzium formula," *Encyclopedia of mathematics. Springer-Verlag, Berlin, Germany*, 2001.

[44] M. Louazni and S. Al-Dahidi, "Approximation of photovoltaic characteristics curves using bézier curve," *Renewable Energy*, vol. 174, pp. 715–732, 2021.

[45] M. Uoya and H. Koizumi, "A calculation method of photovoltaic array's operating point for mppt evaluation based on one-dimensional newton–raphson method," *IEEE Transactions on Industry Applications*, vol. 51, no. 1, pp. 567–575, 2014.

[46] E. W. Weisstein, "Lambert w-function," *https://mathworld. wolfram. com/*, 2002.

[47] C. Moler, "Cleve's corner: Cleve moler on mathematics and computing," *Splines and Pchips*, 2012.

[48] S.-x. Lun, C.-j. Du, T.-t. Guo, S. Wang, J.-s. Sang, and J.-p. Li, "A new explicit i–v model of a solar cell based on taylor's series expansion," *Solar energy*, vol. 94, pp. 221–232, 2013.

[49] J. Gow and C. Manning, "Development of a photovoltaic array model for use in power-electronics simulation studies," *IEE Proceedings-Electric Power Applications*, vol. 146, no. 2, pp. 193–200, 1999.

[50] X. H. Nguyen and M. P. Nguyen, "Mathematical modeling of photovoltaic cell/module/arrays with tags in matlab/simulink," *Environmental Systems Research*, vol. 4, pp. 1–13, 2015.

[51] E. I. O. Rivera and F. Z. Peng, "Algorithms to estimate the temperature and effective irradiance level over a photovoltaic module using the fixed point theorem," in *2006 37th IEEE Power Electronics Specialists Conference.* IEEE, 2006, pp. 1–4.

[52] S. Jenkal, M. Kourchi, D. Yousfi, A. Benlarabi, M. L. Elhafyani, M. Ajaamoum, and M. Oubella, "Development of a photovoltaic characteristics generator based on mathematical models for four pv panel technologies," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6101–6110, 2020.

초록

태양광 발전을 포함한 신재생에너지 시스템은 각 국의 탄소중립 기조와 함께 더욱 강조되고 있다. 특히 태양전지 패널은 직류전압 또는 전류 출력을 내기 때문에 배터리 에너지저장장치와 함께 직류마이크로 그리드의 핵심 요소기술로 자리매김하고 있다. 특히 태양광패널을 모사하는 하드웨어시뮬레이터나 소프트웨어 시뮬레이터 기술에서는 빠르고 정확한 태양광 패널 모델의 구현이 매우 중요하며, 이는 태양광 발전 패널의 효과적인 성능평가를 가능하게 한다. 태양광 패널모델로는 대부분 다이오드모델기반 등가회로를 사용하는 데, 그 물리적인 현상을 표현하는데에는 유리하지만, 두가지 면에서 단점이 있었다. 첫째, 측정데이터를 통해 새로운 패널의 모델정수를 추출하고자 할때, 그 파라메터 추출상수가 불필요하게 많이 필요하고, 각 모델정수는 물리적인 현상위주로 정의되었기 때문에, 특성곡선의 형태에서 의미하는 바가 명확하지 않았다. 따라서 모델정수 추출이 오래걸리고 불편하였다. 둘째, 태양광 패널모델의 수식에서 전류와 전압관계가 음함수의 형태로 구성되어, 곡선의 동작점 계산이 복잡하고 직관적이지 않아, 시뮬레이션 속도가 빠르지 않았으며, Lambert 함수와 같은 다양한 양함수화 방법을 필요로 하였다.

이와 같은 단점을 해결하기 위해, 본 논문에서는 태양광패널의 I-V특성곡성과 초타원(Superellipse)함수의 기하학적 유사성을 이용한 직관적인 태양광 패널 모델링 방법을 제안하였다. 본 논문에서는 제안된 태양광패널모델의 모델정수추출의 편의성과 추출된 모델의 정확도 측면에서의 장점을 기존 방법들과 비교하였다. 특히, 다양한 재료로 만들어진 패널에 대한 여러가지

일사량과 온도조건을 변화시키면서 측정된 제조사의 데이터시트기반값과 비교 검토하여 모델의

정확도를 검증하였다.