

**People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA – Boumerdes**



**Institute of Electrical and Electronic Engineering
Department of Power and Control**

Final Year Project Report Presented in Partial Fulfilment of
the Requirements for the Degree of

MASTER

In Electrical and Electronic Engineering
Option: Power Engineering
Title:

PV Parameters extraction using Metaheuristics

Presented by:

- **KOUTB-EDDINE BOUTRIK**

Supervised by:

- **Pr. AISSA KHELDOUN**

Dedication

I dedicate this work to my family and friends.

A Great feeling of love goes to my loving parents, who have supported me in my educational journey since the very beginning. Special thanks to my sister and brother who were always of great support.

I also dedicate this dissertation to my dear friends who have, along the years, become my second family.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Pr. AISSA KHELDOUN, for his valuable guidance, orientation, and support.

Much appreciation goes to Pr. KHELDOUN who has been a great teacher who provided me with all the necessary information that helped achieve this project.

Special thanks to everyone who had an impact on this work

Abstract

Interest in photovoltaics has been increasing hugely over the past years. Among the many interests developed, theoretical understanding of PVs has taken much attention in research. To help in design and assess the performance of PV panels, a developed model is used. The model is none other than an equivalent electrical circuit with basic components (a source, resistors, and one diode or more). Single-diode and double-diode models are the most popular in the literature. Equivalent circuit parameters must be obtained, from either a set of experimental data or a manufacturer's datasheet, in order to construct a model. The aim is to obtain values that yield an accurate model. The problem is tackled as an optimization one, where the Root mean square error (RMSE), between the experimental and the calculated data, is the function to be optimized. Optimization is achieved using five different meta-heuristic algorithms: Particle swarm optimization (PSO), Wind driven optimization (WDO), Exchange market algorithm (EMA), Differential evolution (DE), and Marine predators algorithm (MPA). The aforementioned algorithms are adapted to PV parameters extraction using MATLAB. Algorithms are then compared based on the accuracy of the obtained results.

Table of contents:

Dedication.....	I
Acknowledgement.....	II
Abstract.....	III
Table of contents	IV
List of figures.....	VIII
List of tables	XII
Nomenclature.....	XIII
General introduction	1

Chapter I: PV modelling

1.1 introduction.....	3
1.2 PV principle.....	3
1.3 PV technology	5
1.3.1 Monocrystalline silicon	5
1.3.2 Polycrystalline silicon.....	6
1.3.3 Amorphous silicon.....	6
1.3.4 Cadmium telluride	6
1.3.5 Copper Indium Gallium Selenide	7
1.3.6 Polymer and organic PV	7
1.4 PV models	7
1.4.1 Ideal PV model	7
1.4.2 Single diode model	8
1.4.3 Double-diod model	8
1.5 Model parameters	9
1.5.1 Ideality factor.....	9
1.5.2 Photocurrent.....	9
1.5.3 Diode saturation current	10

1.5.4 Series resistance	11
1.5.5 Parallel resistance	12
1.6 Conclusion	14

Chapter II: Optimization algorithms

2.1 Introduction	15
2.2 Problem formulation	15
2.3 Classification of algorithms	15
2.3.1 Local optimization algorithms	16
2.3.2 Global optimization algorithms	16
I. PSO	16
II. WDO	19
III. EMA	21
IV. DE	26
V. MPA	29
2.4 Conclusion	32

Chapter III: PV parameters extraction:

3.1 Introduction	33
3.2 Parameters extraction methods	33
3.2.1 Analytical method	33
3.2.2 Iterative method	34
3.2.3 Metaheuristics	34
3.3 Results	36
3.3.1 PSO results	36
a) Condor CEM150M	36
b) <i>Kyocera KC125GHT</i>	37

Table of contents

c) Siliken SLK60P6L.....	38
d) Sanyo HIP-190B2-BO-01	39
3.3.2 WDO results	41
a) Condor CEM150M	41
b) <i>Kyocera KC125GHT</i>	42
c) Siliken SLK60P6L.....	43
d) Sanyo HIP-190B2-BO-01.....	44
3.3.3 EMA results	45
a) Condor CEM150M	45
b) <i>Kyocera KC125GHT</i>	46
c) Siliken SLK60P6L.....	47
d) Sanyo HIP-190B2-BO-01.....	48
3.3.4 DE results.....	50
a) Condor CEM150M	50
b) <i>Kyocera KC125GHT</i>	51
c) Siliken SLK60P6L.....	52
d) Sanyo HIP-190B2-BO-01.....	53
3.3.5 MPA results	
a) Condor CEM150M	54
b) <i>Kyocera KC125GHT</i>	55
c) Siliken SLK60P6L	56
d) Sanyo HIP-190B2-BO-01	57
3.4 Results discussion	59
3.4.1 Comparative tables	59
a) Condor CEM150M	59
b) <i>Kyocera KC125GHT</i>	59
c) Siliken SLK60P6L.....	60
d) Sanyo HIP-190B2-BO-01.....	60

Table of contents

3.5 Conclusion	61
General conclusion	62
References.....	63

List of figures:

Fig 1.1: Monocrystalline silicon solar cell	3
Fig 1.2: Principle of operation of a solar cell	4
Fig 1.3: I-V curve of a PV cell	5
Fig 1.4: ideal solar cell model	7
Fig 1.5: Single-diode PV model	8
Fig 1.6: Double-diode PV model.....	9
Fig 1.7: The effect of irradiance on cell IV curve	10
Fig 1.8: I-V curve with no series resistance $R_s=0$	11
.....	
Fig 1.9: Effect of a 2Ω series resistance on the previous IV curve	12
Fig 1.10: IV curve of a cell with high parallel resistance (1000Ω)	13
Fig 1.11: Effect of low parallel resistance (10Ω) on cell IV curve	13
Fig 2.1: Schematic representation of the motion of a particle in PSO	18
Fig 2.2: Flowchart of PSO algorithm	19
Fig 2.3: Flowchart for WDO algorithm.....	21
Figure 2.4 : EMA flowchart.....	25
Fig 2.5: DE algorithm flowchart	28
Fig 3.1: RMSE vs Iteration graph using PSO on Condor CEM150M PV module data.	32
Fig 3.2: I-V curve showing both calculated (through PSO) and measured currents of..... Condor CEM150M PV module	33
Fig 3.3: RMSE vs Iteration graph using PSO on Kyocera KC125GHT module data....	33
Fig 3.4: I-V curve showing both calculated (through PSO) and measured currents of..... Kyocera KC125GHT PV module	34

Fig 3.5: RMSE vs Iteration graph using PSO on Siliken SLK60P6L PV module data	
Fig 3.6: I-V curve showing both calculated (through PSO) and measured currents of Siliken SLK60P6L PV module	35
Fig 3.7: RMSE vs Iteration graph using PSO on Sanyo HIP-190B2-BO-01 PV module data.	35
Fig 3.8: I-V curve showing both calculated (through PSO) and measured currents of Sanyo HIP-190B2-BO-01 PV module	36
Fig 3.9: RMSE vs Iteration graph using WDO on Condor CEM150M PV module data.	37
Fig 3.10: I-V curve showing both calculated (through WDO) and measured currents of Condor CEM150M PV module	37
Fig 3.11: RMSE vs Iteration graph using WDO on Kyocera KC125GHT module data.	38
Fig 3.12: I-V curve showing both calculated (through WDO) and measured currents of Kyocera KC125GHT PV module	38
Fig 3.13: RMSE vs Iteration graph using WDO on Siliken SLK60P6L PV module data.	39
Fig 3.14: I-V curve showing both calculated (through WDO) and measured currents of Siliken SLK60P6L PV module	39
Fig 3.15: RMSE vs Iteration graph using WDO on Sanyo HIP-190B2-BO-01 PV module data.....	40
Fig 3.16: I-V curve showing both calculated (through WDO) and measured currents of Sanyo HIP-190B2-BO-01 PV module	40
Fig 3.17: RMSE vs Iteration graph using EMA on Condor CEM150M PV module data.	41
Fig 3.18: I-V curve showing both calculated (through EMA) and measured currents of Condor CEM150M PV module	42

Fig 3.19: RMSE vs Iteration graph using EMA on Kyocera KC125GHT PV module data.	42
Fig 3.20: I-V curve showing both calculated (through EMA) and measured currents of Kyocera KC125GHT PV module	43
Fig 3.21: RMSE vs Iteration graph using EMA on Siliken SLK60P6L PV module data.	43
Fig 3.22: I-V curve showing both calculated (through EMA) and measured currents of Siliken SLK60P6L PV module	44
Fig 3.23: RMSE vs Iteration graph using EMA on Sanyo HIP-190B2-BO-01 PV module data.	44
Fig 3.24: I-V curve showing both calculated (through EMA) and measured currents of Sanyo HIP-190B2-BO-01 PV module	45
Fig 3.25: RMSE vs Iteration graph using DE on Condor CEM150M PV module data.	46
Fig 3.26: I-V curve showing both calculated (through DE) and measured currents of Condor CEM150M PV module	46
Fig 3.27: RMSE vs Iteration graph using DE on Kyocera KC125GHT PV module data.	47
Fig 3.28: I-V curve showing both calculated (through DE) and measured currents of Kyocera KC125GHT PV module	47
Fig 3.29: RMSE vs Iteration graph using DE on Siliken SLK60P6L PV module data.	48
Fig 3.30: I-V curve showing both calculated (through DE) and measured currents of Condor CEM150M PV module	48
Fig 3.31: RMSE vs Iteration graph using DE on Sanyo HIP-190B2-BO-01 PV module data	49
Fig 3.32: I-V curve showing both calculated (through DE) and measured currents of Condor CEM150M PV module	49

Fig 3.33: RMSE vs Iteration graph using MPA on Condor CEM150M PV module data .

.....

Fig 3.34: I-V curve showing both calculated (through MPA) and measured currents of Condor CEM150M PV module

Fig 3.35: RMSE vs Iteration graph using MPA on Kyocera KC125GHT PV module data

Fig 3.36: I-V curve showing both calculated (through MPA) and measured currents of Kyocera KC125GHT PV module

Fig 3.37: RMSE vs Iteration graph using MPA on Siliken SLK60P6LPV module data. ..

Fig 3.38: I-V curve showing both calculated (through MPA) and measured currents of Siliken SLK60P6LPV module data

Fig 3.39: RMSE vs Iteration graph using MPA on Sanyo HIP-190B2-BO-01 PV module data.....

Fig 3.40: I-V curve showing both calculated (through MPA) and measured currents of Sanyo HIP-190B2-BO-01 PV module

List of tables:

Table 1: extracted model parameters and RMSE using PSO.....	36
Table 2: extracted model parameters and RMSE using WDO	41
Table 3: extracted model parameters and RMSE using EMA	45
Table 4: extracted model parameters and RMSE using DE.....	50
Table 5: extracted model parameters and RMSE using MPA	50
Table 6: Comparative table of the parameters extracted and RMSE values of the five algorithms for Condor CEM150M.....	50
Table 7: Comparative table of the parameters extracted and RMSE values of the five algorithms for Kyocera KC125GHT	51
Table 8: Comparative table of the parameters extracted and RMSE values of the five algorithms for Siliken SLK60P6L	51
Table 9: Comparative table of the parameters extracted and RMSE values of the five algorithms for Sanyo HIP-190B2-BO-01	51

Nomenclature:

a	<i>Diode ideality factor</i>
R_s	<i>PV model Series resistance</i>
R_p	<i>PV model Shunt resistance</i>
I_{ph}	Photo-generated current
I_o	diode saturation current
k	Boltzmann's constant $1.38064852 \times 10^{-23}$
q	Elementary charge 1.6×10^{-19} C
I_{sc}	Short-circuit current
V_{oc}	Open-circuit voltage
I_D	Diode current
N_s	Number of series connected cells in a PV module

General introduction

The world is making development in huge leaps; a development that has raised the need for more and more power. Satisfying such a need comes at great cost; pollution has been increasing proportionally with the increase in power demand. Fortunately, many environment-friendly alternatives for electricity generation exist. Potentially, in the future, power demand will be mainly covered by eco-friendly sources; since we receive a staggering 1.74×10^{17} watts of energy from the sun [1], for example.

Photovoltaics deal with the generation of electricity from solar energy; the field has become of great interest over the past years. PV panels cost has been decreasing, while improvements in efficiency are observed; this has led many firms and people investing in them. The theory about photovoltaics is very important; whether it is for educational or technical purposes; it is a key aspect in improving the quality of energy obtained from PVs. An accurate model gives large insights about the behaviour of solar panels; a behaviour usually expressed in mathematical relationships relating current and power to voltage (I-V and P-V), under different thermal and illumination conditions. Modelling gives large insights in terms of design and performance assessment. Along the years, various models have been proposed; Single and double-diode models have been widely used. No matter whatever circuit model is adopted for the PV system, finding the optimum circuit parameters requires solution of transcendental equations relating I-V characteristic of the PV device [2]. For a single-diode model the parameters to be estimated are: the diode ideality factor a , the series resistance R_s , the shunt resistance R_p , the photocurrent I_{ph} , and the saturation current of the diode I_o .

In this project, emphasis is given to the single-diode model; also known as the five-parameter model. The main objective is to obtain the five parameters, with optimal values, to build an accurate model that depicts the behaviour of solar panels. In the literature of PV parameters estimation, many methods have emerged; analytical methods, iterative-based methods and meta-heuristic methods. In this work, interest is devoted to the latter method. There are actually two types of data inputs to perform calculations; it is either done using experimental data obtained from the module, which is the way chosen here, or by the data provided on the manufacturer's datasheet. Data are input into MATLAB programs that are adaptations of optimization algorithms to suit the purpose of parameters extraction.

General introduction

1. **Modelling of PV modules:** In this chapter, the principle of PV, PV technology as well as one-diode and two-diode models will be presented.
2. **Optimization algorithms:** this chapter gives information on how the optimization algorithms used (PSO, WDO, EMA, DE, and MPA) work.
3. **Identification of PV parameters:** this chapter presents methods of estimating PV parameters, and shows results of the work done using the aforementioned algorithms.

CHAPTER I: PV MODELLING

1.1 Introduction

Photovoltaics can be defined simply as the process of using solar cells to convert sunlight into electricity. Modelling of PV panels is very important in order to understand that process, and study it. The more accurate the model is the more accurate will the results be. This is why modelling is a crucial part when it comes to knowing the behaviour of PV panels in different conditions, without even needing to experience those conditions practically.

1.2 PV principle:

PV cells are the unit components of PV panels. They are made by slicing a very thin layer of a silicon rod called an ingot. A single silicon wafer does not act like a solar cell yet; it lacks additional layers to be able to operate as a solar cell. A silicon based solar cell consists of the c-Si absorber layer, a pn-junction to separate the light-excited charge carriers, and a metal front and back contact.



Fig1.1 Monocrystalline silicon solar cell

The light enters the solar cell from the front side (the top side); and is transmitted into the absorber layer where its energy is absorbed. The energy is used to excite charge carriers in the semiconductor material, which are negatively charged electrons, and positively charged holes. These charge carriers diffuse and need to be separated. This separation occurs at the depletion region between the n- and p-type doped silicon and the depletion region at the back

of the solar cell. Doped layers are areas where we intentionally have put impurities that make either the electron or the holes the dominant charge carriers in these regions. In n-type materials, the electrons are the dominant charge carrier and in p-type the holes are. Therefore, the electrons are collected at the n-type layer and holes at the p-type layer. Then the charge carriers have to be collected at the contacts. The electron will move through the load and back to the solar cell. Then, both charge carriers recombine at the metal/p-layer interface.

Hence, we deduce that the photovoltaic process is based on three important principles: First, the excitation of free mobile charge carriers- the hole and electron- due to light Absorption. Second, the separation of the charge carriers (holes and electrons). Third, the collection of the charge carriers at the contacts.

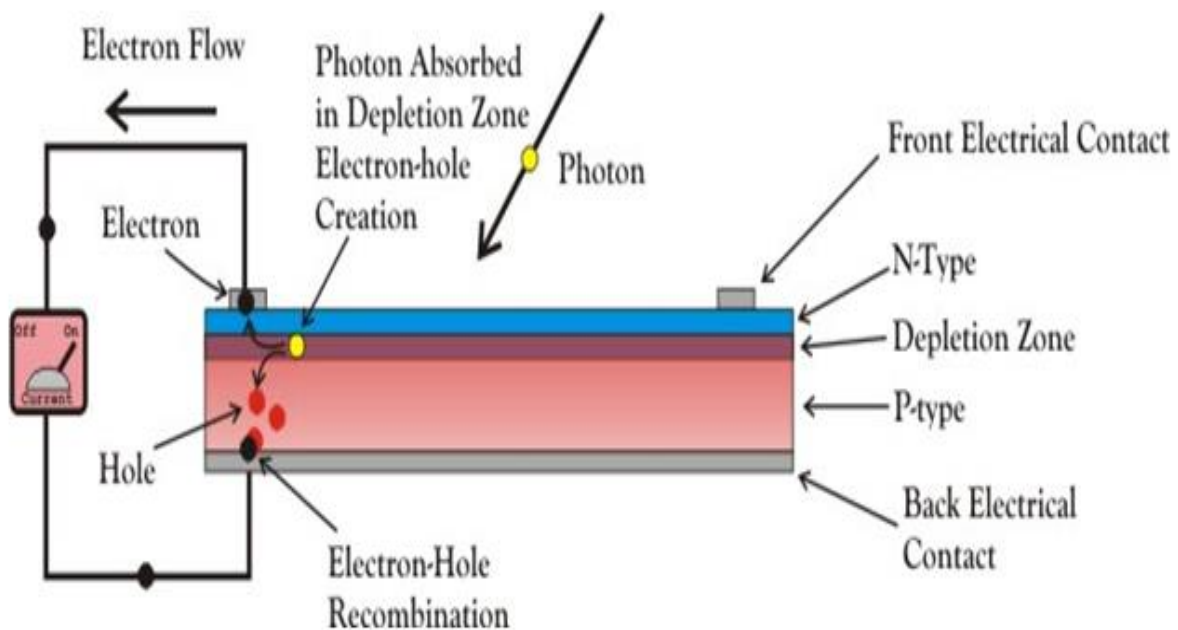


Fig1.2: Principle of operation of a solar cell

The I-V plot (current versus voltage plot) helps understand the characteristics of solar panels. It is an appreciation of how the current changes with respect to voltage.

A solar cell behaves much like a diode under non-illumination. It will block the current under reverse bias conditions and will produce a current under forward bias conditions. However,

when the solar cell is illuminated, it conducts additional current related to the excited charge carriers.

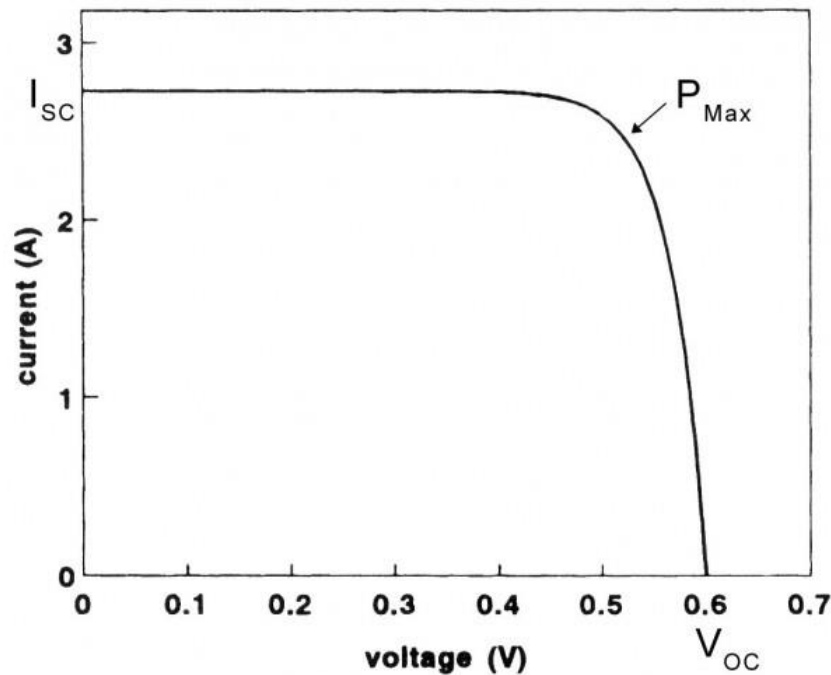


Fig 1.3: I-V curve of a PV cell

1.3PV technology

PV technology has, and will surely continue to make big advancements in the future. Many types of PV have emerged from that progress [3].

1.3.1 Monocrystalline silicon

Monocrystalline silicon solar cells are probably the oldest type of solar cells. They are made from pure silicon crystal, which has continuous lattice and almost no defects. Its properties provide high efficiency of light conversion (up to 22-24%). Manufacturing of the Si crystals is rather complicated, which is responsible for high cost of this type of photovoltaics. Recent developments have decreased the total thickness of Si material used in monocrystalline cells to reduce cost. The monocrystalline silicon cells have a typical black or iridescent blue color. The monocrystalline silicon cells are believed to be very durable and last over 25 years. However, their efficiency will gradually decrease (about 0.5% per year), therefore replacement of operating modules might be needed sooner. The main disadvantages of the monocrystalline silicon panels are high initial cost and mechanical vulnerability (brittle).

1.3.2 Polycrystalline silicon

Polycrystalline cells are made by assembling multiple grains and plates of silicon crystals into thin wafers. Smaller pieces of silicon are easier and cheaper to produce, so the manufacturing cost of this type of PV is less than that of monocrystalline silicon cells. The polycrystalline cells are slightly less efficient (12%). These cells can be recognized by their mosaic-like appearance. Polycrystalline cells are also very durable and may have a service life of more than 25 years. The drawbacks of this type of PV technology are mechanical brittleness and low conversion efficiency.

1.3.3 Amorphous silicon (Thin-film)

Thin film photovoltaic cells are produced by depositing silicon film onto substrate glass. In this process, less silicon is used for manufacturing compared to mono- or polycrystalline cells, but this economy comes at the expense of conversion efficiency. Thin-film PV cells have efficiency of 6% versus 15% for single crystal Si-cells. One way to improve the cell efficiency is to create a layered structure of several cells. The main advantage of the thin-film PV technology is that the amorphous silicon can be deposited on a variety of substrates, which can be made flexible and come in different shapes and therefore can be used in many applications. The amorphous silicon is also less prone to overheating, which usually decreases the solar cell performance. Amorphous silicon is most developed among the thin-film PV.

1.3.4 Cadmium Telluride, CdTe (thin-film)

CdTe PV is another kind thin-film solar technology. It has become quite popular due to the lower cost per kW-hour. The best efficiency obtained with CdTe cells is around 16%. One of the advantages of the CdTe cells is that they capture shorter wavelengths of light than silicon cells can do. There are some environmental concerns related to the limited supply of tellurium and potential toxic impact of cadmium at the stage of CdTe panel disposal. Developing effective closed-loop recycling technologies can be a game-changing factor in favour of this technology.

1.3.5 Copper Indium Gallium Selenide (CIGS)

CIGS PV have become a popular new material for solar cells, as it does not contain toxic Cd, and has higher efficiency (just under 20%). At this moment, the CIGS are the most efficient among the thin-film PV technologies. While lab results confirmed high promise of this kind of photovoltaics, the mass production of CIGS proved to be a problem. The CIGS cells are manufactured by thin film deposition on a substrate, which can also be flexible (unlike the silicon cells). Similar to CdTe cells, the CIGS cells demonstrate good resistance to heating.

1.3.6 Polymer and organic PV

Organic materials are quite attractive since they can be involved in high-output manufacturing and also because they can be made in various thicknesses and shapes. These types of cells are relatively lightweight (compared to silicon cells). Also, they offer flexibility and relatively low fabrication cost. They, however, are much less efficient (about 1/3 of a typical Si cell efficiency) and sometimes prone to quicker degradation (shorter service life).

These are the main well-known types of PV technology; but with innovation and research new types with better characteristics will surely emerge. Breakthrough in PV industry is dependent on advents in other fields like chemistry.

1.4 PV models

1.4.1 Ideal PV model

An ideal solar cell can be represented simply as a light-generated current source in parallel with a diode as shown in **Fig 1.4**

Eq 1.1 gives the output current:

$$I = I_{ph} - I_0 \left[\exp \left(\frac{V}{aV_T} \right) - 1 \right] \quad (1.1)$$

Where V_T is the diode thermal voltage given by:

$$V_T = \frac{N_s k T}{q} \quad (1.2)$$

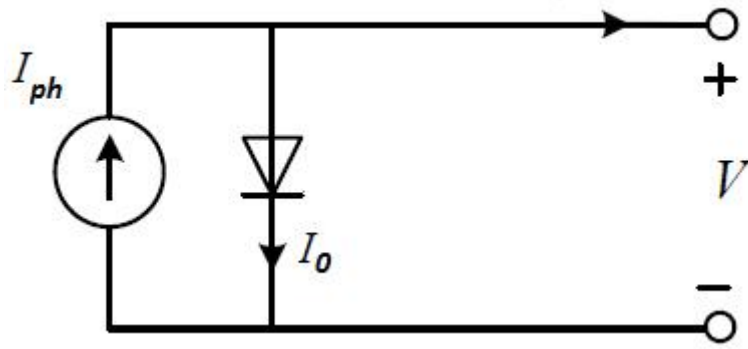


Fig 1.4: ideal solar cell model

1.4.2 Single-diode model

In fact, the solar cell is not exactly ideal. This is why more accurate models have been developed to give more realistic results in analysis. In other words, Eq. (1.1) does not accurately depict the behaviour of a PV cell. A more practical model is the single-diode model that is represented in **Fig 1.5**. Series and parallel resistances have been added to the previous model. Results that are more accurate are obtained from this model, although it consumes more computational time. Eq (1.3) gives the output current:

$$I = I_{ph} - I_0 \left[\exp \left(\frac{V + IR_s}{aV_T} \right) - 1 \right] - \left(\frac{V + IR_s}{R_p} \right) \quad (1.3)$$

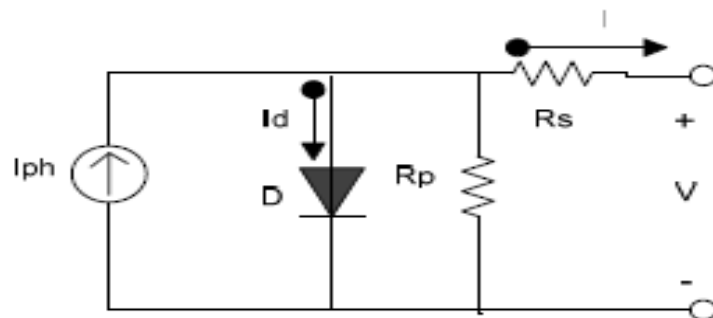


Fig 1.5: Single-diode PV model

1.4.3 Double-diode model

This model is a modified version of the previous one by adding a second diode in parallel with the first one as shown in Fig 1.6. The additional diode stands for the effect caused by the recombination of charge carriers. However, the two-diode model makes computations longer despite its accuracy. Eq (1.4) in this case gives the output current:

$$I = I_{ph} - I_{01} \left[\exp \left(\frac{V + IR_s}{a_1 V_T} \right) - 1 \right] - I_{02} \left[\exp \left(\frac{V + IR_s}{a_2 V_T} \right) - 1 \right] - \left(\frac{V + IR_s}{R_p} \right) \quad (1.4)$$

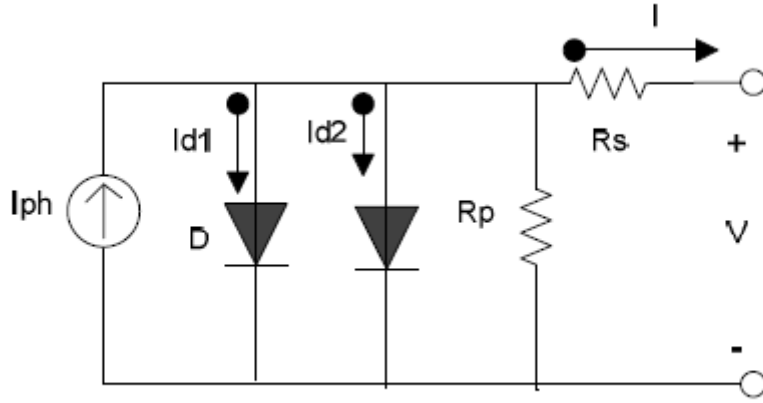


Fig 1.6: Double-diode PV model

1.5 Model parameters:

The solar cell model is constructed using the following parameters [4].

1.5.1 Ideality Factor (a):

The ideality factor (a) is a unitless parameter. It is a measure of how closely the diode follows the ideal diode equation. It accounts for the different mechanisms responsible for moving carriers across the junction. The value of a equal to one means the transport process is purely diffusion, and a value equal to two if it is primarily recombination in the depletion region. The parameter a represents one of parameters to be computed in our work. The ideality factor appears in the diode current component of equation (1.3).

1.5.2 Photo Current I_{ph}

As its appellation suggests, the photocurrent (light-generated current) is the current generated due to the absorption of photons by a solar cell. It depends on reference first and second temperatures, T_1 and T_2 , respectively, and it is given by:

$$I_{ph} = I_{ph}(T_1) + K_0(T - T_1) \quad (1.5)$$

$$I_{ph}(T_1) = I_{scT_1nom} \left(\frac{G}{G_{nom}} \right) \quad (1.6)$$

Where:

$$K_0 = \frac{I_{scT_2} - I_{scT_1}}{T_2 - T_1} \quad (1.7)$$

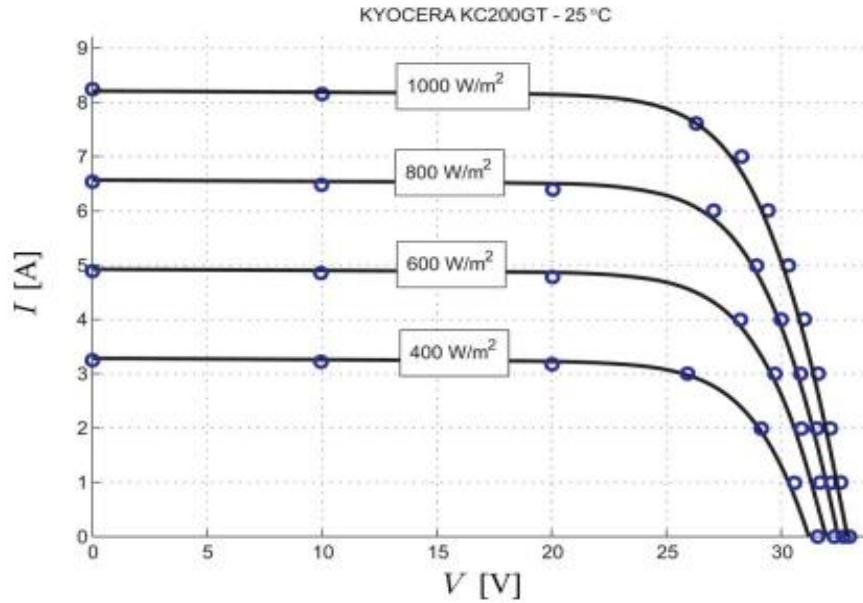


Fig 1.7: The effect of irradiance on cell IV curve

1.5.3 Diode saturation current I_0 :

The saturation current is a combination of the generation current caused by thermal generation of electron-hole pairs within the depletion region of the diode and the diffusion current due to minority carriers in the n and p regions diffusing across the depletion region. Although the saturation current is voltage independent, it does depend on temperature since both the current contributions depend on thermally stimulated carriers.[5]

$$I_o = I_{oT_1} \left(\frac{T_1}{T} \right)^3 \exp \left(\frac{qE_g}{ak} \left(\frac{1}{T} - \frac{1}{T_1} \right) \right) \quad (1.8)$$

Where:

$$I_{oT_1} = \frac{I_{scT_1}}{\exp \left(\frac{qV_{ocT_1}}{akT_1} \right) - 1} \quad (1.9)$$

1.5.4 Series resistance R_s

Series resistance in a solar cell has three causes: firstly, the movement of current through the emitter and base of the solar cell; secondly, the contact resistance between the metal contact and the silicon; and finally the resistance of the top and rear metal contacts. The main impact of series resistance is to reduce the fill factor, although excessively high values may also reduce the short-circuit current.

The value of R_s can be calculated using the following equation:

$$R_s = - \frac{dV}{dI_{V_{oc}}} - \frac{1}{X_v} \quad (1.10)$$

Where:

$$X_v = I_{oT_1} \left(\frac{q}{akT_1} \right) \exp \left(\frac{qV_{ocT_1}}{akT_1} \right) \quad (1.11)$$

The equation of current with a series resistance only (no parallel resistance) reduces to the following:

$$I = I_{ph} - I_o \left[\exp \left(\frac{qV + R_s I}{akT} \right) \right] \quad (1.12)$$

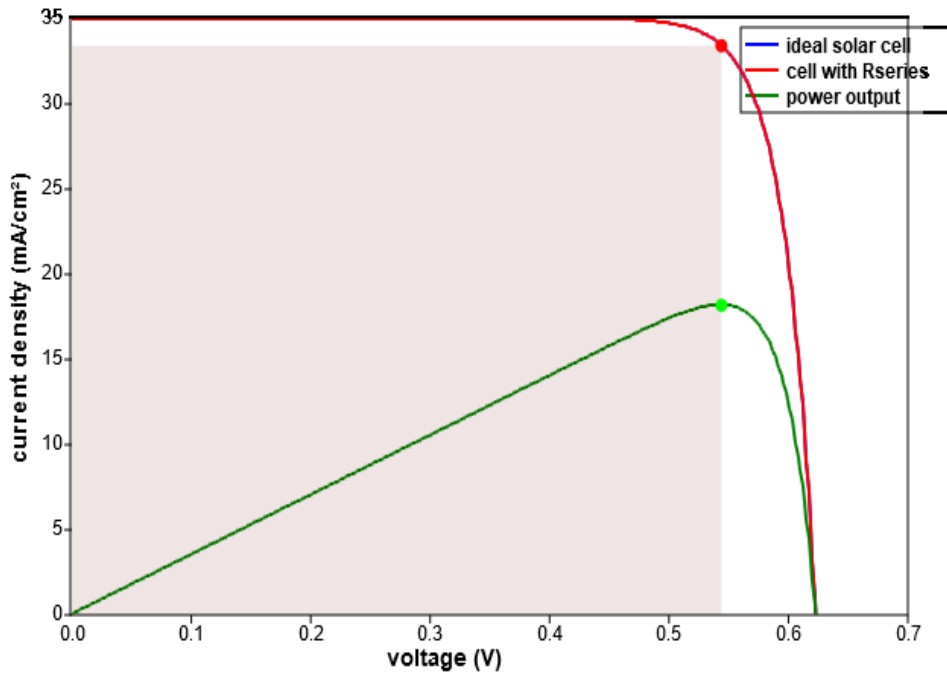


Fig 1.8: I-V curve with no series resistance $R_s = 0$

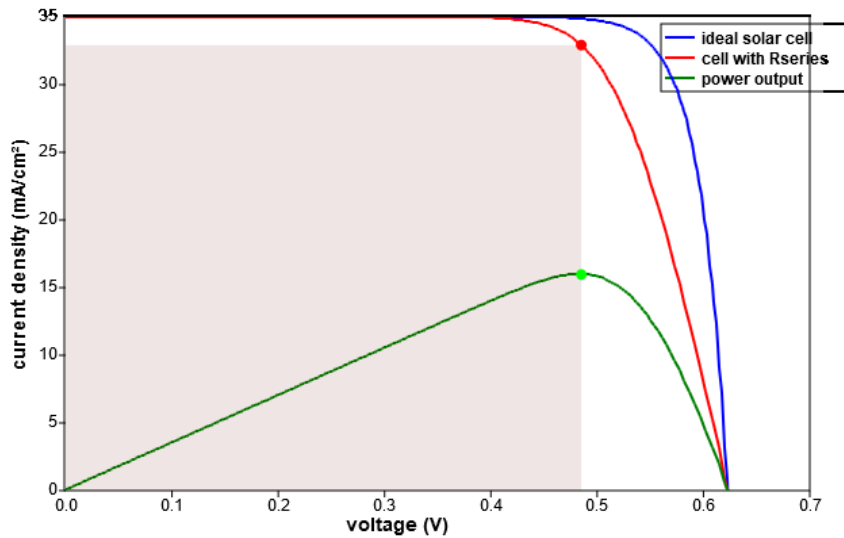


Fig 1.9: I-V curve with a series resistance $R_s = 2\Omega$

We have taken only series resistance (no parallel resistance) in consideration in order to appreciate its effect clearly. This simplifies simulation of dynamic system where PV source is involved such as water pumping [6].

Series resistance does not affect the solar cell at open-circuit voltage since the overall current flow through the solar cell, and therefore through the series resistance is zero. However, near

the open-circuit voltage, the I-V curve is strongly affected by the series resistance. A straightforward method of estimating the series resistance from a solar cell is to find the slope of the IV curve at the open-circuit voltage point.[7]

1.5.5 Parallel resistance R_p

Significant power losses caused by the presence of a shunt resistance, R_p is typically due to manufacturing defects, rather than poor solar cell design. Low shunt resistance causes power losses in solar cells by providing an alternate current path for the light-generated current. Such a diversion reduces the amount of current flowing through the solar cell junction and reduces the voltage from the solar cell. The effect of a shunt resistance is particularly severe at low light levels, since there will be less light-generated current. The loss of this current to the shunt therefore has a larger impact. In addition, at lower voltages where the effective resistance of the solar cell is high, the impact of a resistance in parallel is large. The equation of a solar cell with a shunt resistance only is as follows:

$$I = I_{ph} - I_0 * \exp\left(\frac{qV}{akT}\right) - \frac{V}{R_p} \quad (1.13)$$

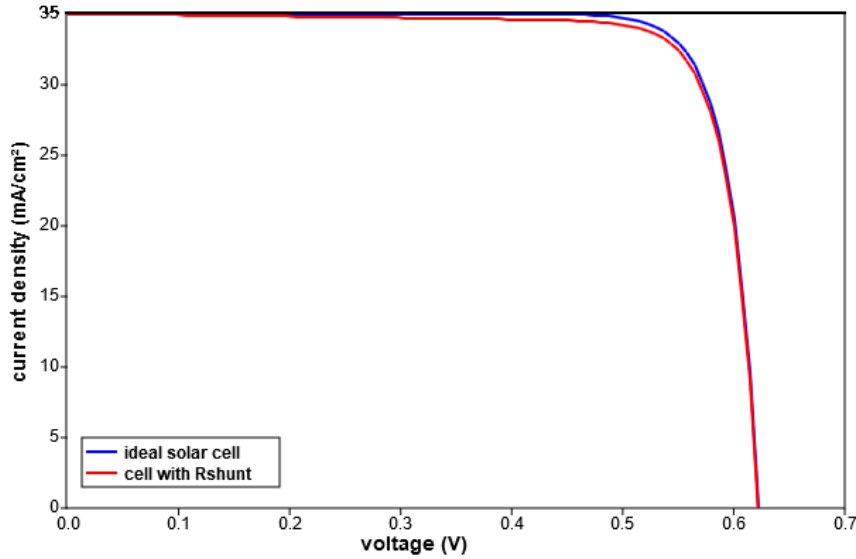


Fig 1.10: IV curve of a cell with high shunt resistance of 1000Ω

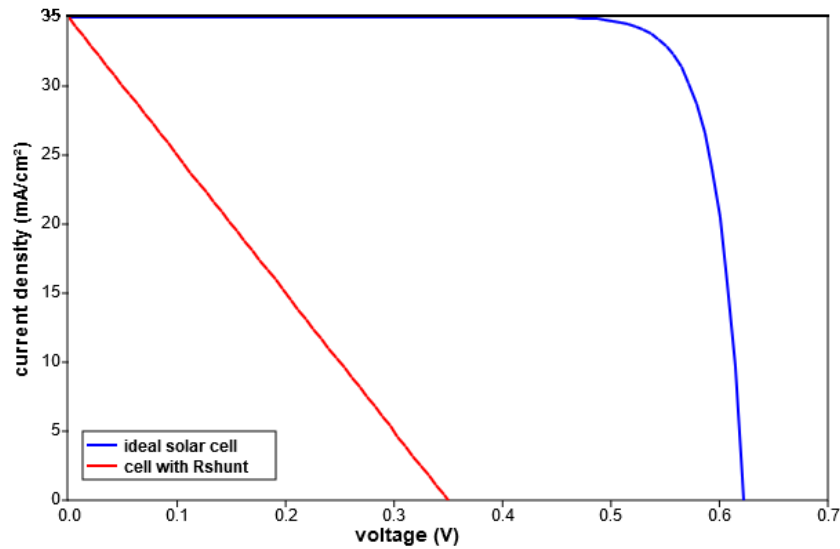


Fig 1.11: Effect of low parallel resistance(10Ω) on cell IV curve

1.6 Conclusion

This chapter has dealt with the different aspects of Photovoltaics; from understanding the principle behind PV to exploring some models used in the literature. It goes without doubt to say that modelling is crucial in designing and studying photovoltaic systems behaviour.

Chapter II: Optimization algorithms

2.1. Introduction:

In mathematics, optimization is defined as finding the best solution for a problem from all feasible solutions [8]. Depending on whether the variables are continuous or discrete, the process of finding values of variables that satisfy an objective function is called optimization.

An optimization algorithm is a procedure that is executed iteratively by comparing various solutions until an optimum or a satisfactory solution is found [9]. With the advent of computers, optimization has become a part of computer-aided design activities.

2.2. Problem formulation

A simple optimization is done by comparing a small set of potential solutions generated by using some background knowledge of the problem (initial guesses). Since problems differ, it is not possible to apply the same formulation to all of them. The purpose of formulation is to create a mathematical model of the optimal design problem, which then can be solved using an optimization algorithm [9].

2.2.1. Design variables: identifying design variables is the first step in formulating an optimization problem. The different combinations of these variables are the potential solutions. Design variables are updated in search for an optimal solution.

2.2.2. Constraints: Design variables must satisfy certain constraints. These constraints can be boundary conditions for example.

2.2.3. Objective function: This is the function that indicates how accurate a solution is. It is expressed in terms of design variables.

2.3. Classification of optimization algorithms:

There are many options for classifying optimization algorithms. Mainly, they can be classified into two categories, which are: Local optimization algorithms and Global optimization algorithms.

CHAPTER II: Optimization algorithms

the difference between the two types can be clarified using a simple analogy; the peak of every mountain is a local optimum since it is the highest level on its neighbourhood, but only the peak of the highest mountain is a global optimum [10]. Following the analogy, local optimization finds local optimum that is not very far from the initial guess. Whereas, global optimization searches a bigger space to find a global optimum.

In our work, all the algorithms used are global optimization ones.

2.3.1. Local optimization algorithms

Local optimization finds the optimal value within the neighbouring set of candidate solutions. The performance of these methods, generally, strongly depends on the initial values supplied. This means that optimization might need to be run several times, with different initial guesses.

Newton Raphson (NR) method is an example in this category. NR uses an initial guess to start with, and the n-iterations are performed until a local optimum is obtained. The drawback is if the initial guess is far from the actual solution this method will be stuck and diverge.

2.3.2. Global optimization algorithms

For this project, metaheuristics are used. Metaheuristics can be adapted to solve a wide range of optimization problems. These methods are designed to find a good solution among a large set of feasible solutions with less computational effort than other optimization techniques.

I. PSO

Particle Swarm Optimization (PSO) is a metaheuristic global optimization algorithm that has gained prominence in the last two decades due to its ease of application in unsupervised, complex multidimensional problems, which cannot be solved using traditional deterministic algorithms [11]. The canonical particle swarm optimizer is based on the flocking behaviour and social co-operation of birds and fish schools and draws heavily from the evolutionary behaviour of these organisms.

CHAPTER II: Optimization algorithms

This algorithm searches the space of an objective function by adjusting the trajectories of individual agents, called particles, as these trajectories form piecewise paths in a quasi-stochastic manner. The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. Each particle is attracted toward the position of the current global best g^* and its own best location x in history, while at the same time it has a tendency to move randomly. When a particle finds a location that is better than any previously found locations, then it updates it as the new current best for particle i . There is a current best for all n particles at any time t during iterations. The aim is to find the global best among all the current best solutions until the objective no longer improves or after a certain number of iterations. The movement of particles is schematically represented in Figure 2.1 where x^* is the current best for particle i , and $g^* = \min\{f(x_i)\}$ for $(i = 1, 2, \dots, n)$ is the current global best.

The position of individual particles is updated as follows [8]

:

$$x_{n+1}^i = x_n^i + v_{n+1}^i \quad (2.1)$$

The velocity is calculated as follows:

$$v_{n+1}^i = v_n^i + c_1 r_1 (p_n^i - x_n^i) + c_2 r_2 (p_n^g - x_n^i) \quad (2.2)$$

Where :

x_n^i - particle position

v_n^i - particle velocity where

p_n^i - best remembered individual particle position

p_n^g - best remembered swarm position

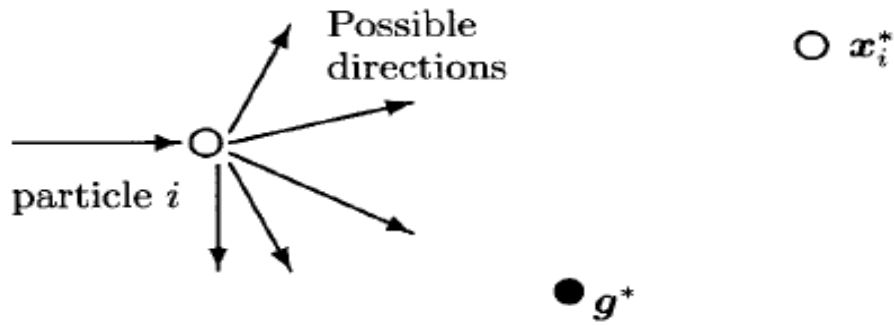


Fig 2.1: Schematic representation of the motion of a particle in PSO

PSO steps:

1-Initialize:

- a) Set constants c_1, c_2, w , number of iterations
- b) Randomly initialise particle positions x_0^i
 $i=1, \dots, n$
- c) Randomly initialize particle velocities $0 \leq v_0^i \leq v_0^{max}$
 $i=1, \dots, n$
- d) Initialize population members

2- optimize:

- a) Evaluate the objective function value f_k^i using design space coordinates x_k^i
- b) If $f_n^i \leq f_{best}^i$ then $f_{best}^i = f_n^i, p_n^i = x_n^i$
 $f_n^i \leq f_{best}^i$ then $f_{best}^g = f_n^i, p_n^g = x_n^i$
- c) If the stopping condition is met then go to (3).
- d) Update all particle velocities v_n^i for $i=1, \dots, p$
- e) Update all particle positions x_n^i for $i=1, \dots, p$
- f) Increment n
- g) Go to 2 (a)

3- terminate

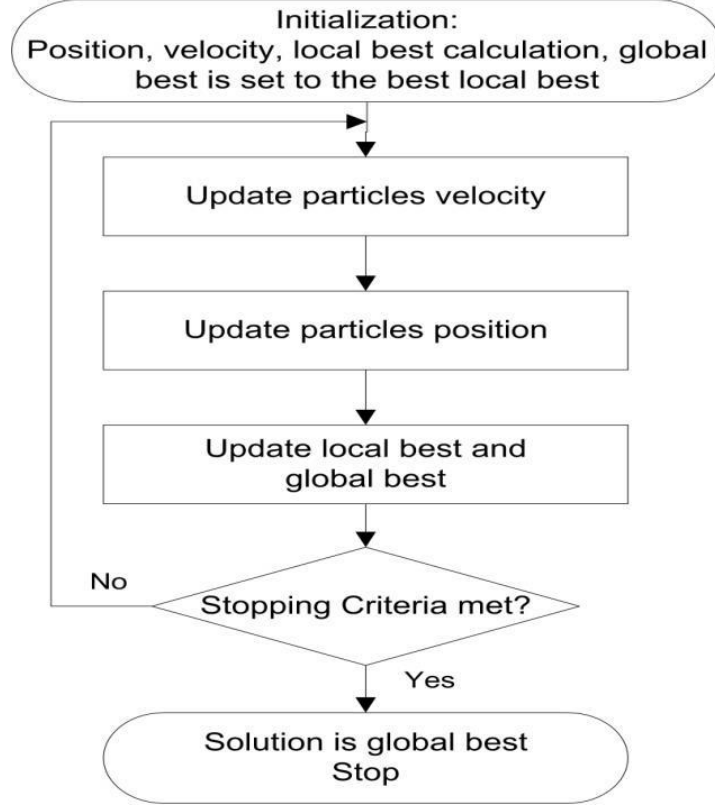


Fig 2.2: Flowchart of PSO algorithm

II. Wind Driven Optimization algorithm (WDO)

The inspiration of the proposed WDO derives from the atmosphere [12]. In the atmosphere, wind blows in an attempt to balance the imbalance of pressure. It flows from high-pressure areas to low pressure areas at a velocity. Depending on the above analysis, some theoretical assumptions are formulated in derivation of the

The Wind Driven Optimization (WDO) algorithm is inspired from the modelling of the climate. In our living environment, wind blows from the high-pressure zone to the low-pressure zone at various speeds to equalize the air pressure imbalance. Based on Newton's second law of motion and some simplifications, the velocity vector, v , and the position vector, x , of the WDO algorithm are updated using

$$v(k+1) = (1 - \alpha)v(k) - gx(k) + \left(RT \left| 1 - \frac{1}{i} \right| (x_{opt} - x(k)) \right) + \left(\frac{cv^{otherdim(k)}}{i} \right) \quad (2.3)$$

Where:

i represents the ranking among all air parcels.

CHAPTER II: Optimization algorithms

In this scheme, the best solution has the lowest pressure with rank 1 and is located at point x_{opt} . Equation (2.3) represents the final form of the velocity update utilized in WDO. The first term in Eq (2.3) states that if there are no other forces acting on the air parcel, then it would continue on its current path with its velocity proportionally reduced by friction. The friction coefficient term could be fixed to a constant value, or one could implement an adaptive friction coefficient, which could vary depending on the velocity of the air parcel. The second term states that gravity constantly pulls the air parcel from its current location towards the centre of the coordinate system at a magnitude proportional to the constant g . This term becomes particularly beneficial if the air parcels are stuck at the boundaries. The third term in Eq (2.3) implies that the higher ranked air parcels will most likely be at a location closer to the x_{opt} and, hence, the effect of the pressure gradient would be smaller. The last term allows the velocity direction to be altered by other dimensions, with a larger influence on higher ranked air parcels. As can be clearly seen in Eq (2.3), there are multiple coefficients that must be chosen prior to starting an optimization, namely: α , g , RT , and c . At each iteration, the velocity and the position of all air parcels need to be updated. Once the new velocity is calculated according to Eq (2.3), the position can be updated by utilizing the following equation,

$$x(k + 1) = x(k) + \Delta t * v(k + 1) \quad (2.4)$$

where, α is a friction coefficient, g is the gravitational constant, R is the universal gas constant, T is the temperature, c is a constant, i is the ranking among all air parcels, x_{opt} is the best parcel so far searched, Δt is the step length.

WDO steps:

- 1- **Initialization:** the parameters given an initial value in this step are : population size, number of iterations, algorithm coefficients alpha RT g .
- 2- **Randomization:** randomize initial population assign random position and velocity for particles.
- 3- **Fitness test:** evaluate the pressure(fitness) of each air parcel (population member)
- 4- **Update:** update velocity and check its limits, update position and check its limits.
- 5- **Check termination criteria:** If the maximum number of iterations is reached end the program, else go to(3)

A flowchart of WDO is represented in Fig

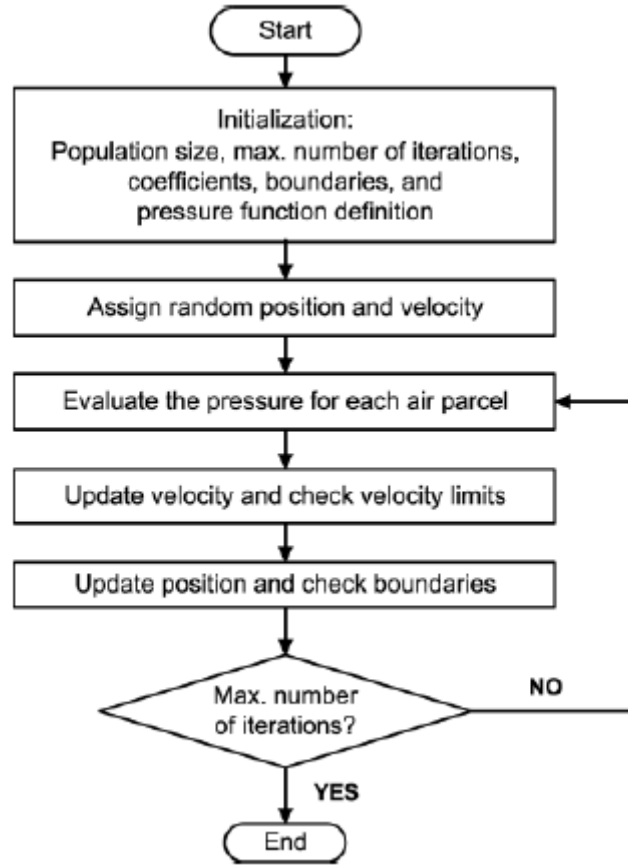


Fig 2.3: Flowchart for WDO algorithm

III. Exchange market algorithm(EMA)

This algorithm is based on stock market behaviour [13]. In a stock market, the shares trading manner is completely sophisticated, different and unique according to mental conditions of several individuals. In the proposed algorithm, it is assumed that the people who are active in the exchange market act similar to the elite stock dealers. In this algorithm, in each market mode the fitness (objective function) of each individual is reviewed, ranked and sorted according to their properties values. In the EMA, all of the shareholders try to introduce themselves as the most successful individuals to market and then the individuals have less fitness tend to do greater risks. Shareholders are arranged into three categories according to their rank after each fitness test. The first group comprises the individual with the best fitness, whereas the third one englobes individuals with the least fitness levels. The individuals in the first group as successful people in the market remain unchanged in all stages of the market. The second and third groups trade with separate equations. In a non-oscillated market, the individuals in second and third groups select stocks which are same or close to the shares of

CHAPTER II: Optimization algorithms

the first group. In other words, the algorithm has the duty to recruit members toward the elite members. For an oscillated market, the individuals in second and third groups trade with separate relationship at high risk, in other words, the algorithm searches for unknown points.

In a stock market, a stable market that is not subject to oscillations can be easily anticipated, and individuals do not take unusual risks to increase their shares. In contrast to that, in an oscillating market the behaviour is not so obvious to be predicted, since individuals take high trading risks to try increasing their shares.

In this algorithm, each individual represents a solution to the problem, and each share stands for a problem variable. There are two states of stock trading: non-oscillating market and oscillating market.

1. Non-oscillating market: In this section, each individual is ranked based on the fitness function and sorted as group 1, group 2, and group 3.

- a) **The first group** is Elite (high rank) shareholders. They represent 10-30% of total population. The members of this group are the best answers for the problems which are necessary to stay intact and unchangeable
- b) **The second group** is shareholders with mean rank. They represent 20-50% of total population. This group updates its population according to the following equation:

$$pop_j^{group(2)} = r * pop_{1,i}^{group(1)} + (1 - r) * pop_{2,i}^{group(1)} \quad (2.4)$$

- c) **The third group** is shareholders with weak rank. This group of shareholders composes 20–50% of the population. The members of this group utilize the differences of share values of the first group as well as their share values' differences compared to the first group individuals and change their shares according to the following equations:

$$pop_k^{group(3),new} = pop_k^{group(3)} + 0.8 * S_k \quad (2.5)$$

$k=1,2,3,n_k$

$$S_k = 2 * r_1 * (pop_{1,i}^{group(1)} - pop_k^{group(3)}) + 2 * r_2 * (pop_{1,2}^{group(1)} - pop_k^{group(3)}) \quad (2.6)$$

CHAPTER II: Optimization algorithms

r_1 and r_2 are random numbers in the interval $[0, 1]$ and n_k the n_{th} member of the third group. $pop_k^{group(3)}$ is the k_{th} member and S_k is the share variations of the k_{th} member of the third group.

2. Oscillating market: In this section, having assessed the shareholders and ranking them based on their fitness, the shareholders would start trading their shares. With regard to their fitness, shareholders will be categorized into 3 separate groups again:

- a) **First Group:** Shareholders with high ranks: This part of the population includes the elite stockbrokers and they do not change their shares and do not undergo the trade risk.
- b) **Second Group:** Shareholders with mean ranks: In this section the sum of the shares held by people tends to be constant and only the number of some of each type of shares increase and some decrease in a way that the sum remains constant. At first, the number of shares held by each person increases based on the following equation:

$$\Delta n_{t1} = n_{t1} - \delta + (2 * r * \mu * \eta_1) \quad (2.7)$$

$$\mu = \left(\frac{t_{pop}}{n_{pop}} \right) \quad (2.8)$$

$$n_{t1} = \sum_{y=1}^n |s_{ty}|_{y=1, 2, 3 \dots n} \quad (2.9)$$

$$\eta_1 = n_{t1} * g_1 \quad (2.10)$$

$$g_1^k = g_{1,max} - \left(\frac{g_{1,max} - g_{1,min}}{iter_{max}} \right) * k \quad (2.11)$$

Δn_{t1} is the amount of shares should be added randomly to some shares, n_{t1} is total shares of t^{th} member before applying the share changes. S_{ty} is the shares of the t^{th} member, δ is the information of exchange market. η_1 is risk level related to each member of the second group, t_{pop} is the number of the t^{th} member in EMA. n_{pop} is the number of the last member, μ is a constant coefficient for each member and g_1 is the common market risk amount which decreases as iteration number increases. $iter_{max}$ is the last iteration number and k is the number of program iteration. $g_{1,max}$ and $g_{1,min}$ indicate the maximum and minimum values of risk in market, respectively. In the second part of this section, it is necessary that each person

CHAPTER II: Optimization algorithms

randomly sells some of his shares equal to the number he has purchased so that the sum of each person's shares remains constant. In this section, it is essential that each person reduces the number of his shares in Δn_{t2} amount. In this status, the Δn_{t2} of each person equals: $\Delta n_{t2} = n_{t2} - \delta$

where Δn_{t2} is the amount of shares that should be decreased randomly from some shares and n_{t2} is the sum share amount of t^{th} member after applying the share variations.

- c) **Third Group:** Shareholders with low ranks: In this section, unlike group 2, the sum of the person's number of shares would change after each trade. In other words, in each section, the person purchases or sells a number of shares. The shareholders of this group change some of their shares based on the following

equation:

$$\Delta n_{t3} = (4 * r_s * \mu * \eta_2) \quad (2.12)$$

$$r_s = 0.5 - rand \quad (2.13)$$

$$\eta_2 = n_{t1} * g_2 \quad (2.14)$$

$$g_2^k = g_{2,max} - \left(\frac{g_{2,max} - g_{2,min}}{iter_{max}} \right) * k \quad (2.15)$$

Where Δn_{t3} is the share amount should be randomly added to the shares of each member, r_s is a random number in $[-0.5, 0.5]$ and η_2 is the risk coefficient related to each member of the third group. g_2 is the variable risk of the market in the third group.

CHAPTER II: Optimization algorithms

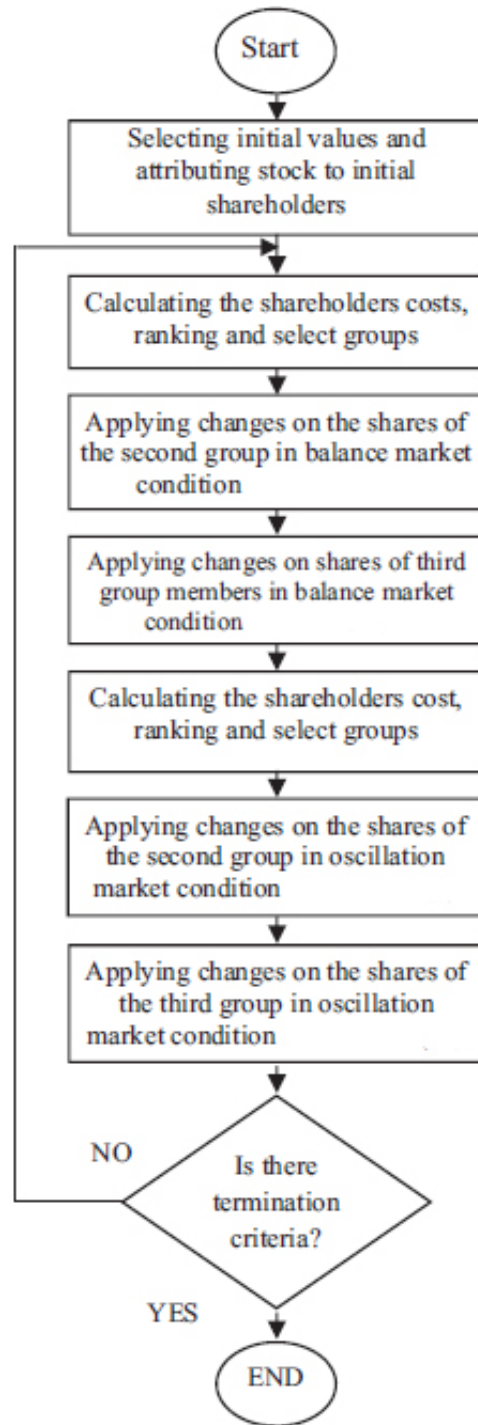


Fig 2.4 : EMA flowchart

IV. Differential evolution (DE) algorithm

Differential evolution is a population-based optimisation evolution algorithm [14], developed to optimise real parameter, real valued functions. DE begins with an initial random population, indicated by P , which comprises NPD -dimension candidate individuals. Thus, the individuals of the G^{th} generation in the initial population can be represented by (x_1^G, x_2^G, x_{NP}^G) , where $G = 0, 1, \dots, G_{max}$ denotes the generation times and G_{max} is the maximal times of the generation. Each candidate individual is actually a D -dimension parameter vector (target vector) within the boundaries denoted by $x_i^G = (x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G)$, where i indicates the i^{th} parameter vector. Then, the individuals enter a loop of the evolutionary process until meeting the termination criterion.

Steps of DE:

1-initialization: upper and lower bounds of each parameter are defined: $x_j^L \leq x_{j,i,1} \leq x_j^U$ initial population is generated using the following equation:

$$x_{i,j} = LB_j + rand * (UB_j - LB_j) \quad (2.16)$$

$j=1,2,\dots,D$

where $rand$ returns a random number uniformly distributed on the interval (0,1), and LB_j, UB_j are the lower and upper bound in the j^{th} dimension, respectively.

2-mutation: After initialization, the mutation strategy is utilized in every individual x_i^G to obtain the mutant vector v_i^G at the G generation. Below are the mutation strategies:

- DE/rand/1

$$v_i^G = x_{r1}^G + F * (x_{r2}^G - x_{r3}^G) \quad (2.17)$$

- DE/rand/2

$$v_i^G = x_{r1}^G + F * (x_{r2}^G - x_{r3}^G) + F * (x_{r4}^G - x_{r5}^G) \quad (2.18)$$

- DE/best/1

$$v_i^G = x_{best}^G + F * (x_{r1}^G - x_{r2}^G) \quad (2.19)$$

- DE/current-to-rand/1

$$v_i^G = x_i^G + F * (x_{r1}^G - x_i^G) + F * (x_{r2}^G - x_{r3}^G) \quad (2.20)$$

- DE/current-to-best/1

$$v_i^G = x_i^G + F * (x_{best}^G - x_{ri}^G) + F * (x_{r1}^G - x_{r2}^G) \quad (2.21)$$

CHAPTER II: Optimization algorithms

The subscripts r_1, r_2, r_3, r_4, r_5 are mutually distinct integers randomly produced from $\{1, 2, \dots, NP\}$ and they are unequal to the index i . The real number F is the scaling factor of difference vector. X_{best}^G denotes the best individual, that is the parameter vector which obtains the best fitness value in the current population.

3-crossover: For the sake of increasing the diversity of the population, the crossover operation is performed by mixing the target vector x_i^G with the mutated vector v_i^G to generate the crossover vector (trial vector) u_i^G as follows:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } rand * u_{i,j} \leq CR \text{ or } j = j_{rand} \\ x_{i,j}^G, & \text{otherwise} \end{cases} \quad (2.22)$$

where $i \in [1, NP], j \in [1, D]$

CR represents the crossover rate determined by the user, which controls the number of dimensions inherited from a mutant vector.

4- Selection

In the end, DE implements the greedy selection to decide whether the target vector or the trial vector is reserved to the next generation. For a minimization problem, the selection operator is as follows:

$$x_i^{G+1} = \begin{cases} u_i^G, & \text{if } f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{otherwise} \end{cases} \quad (2.23)$$

where $f(\mathbf{x})$ is the objective function (fitness function) to be minimized.

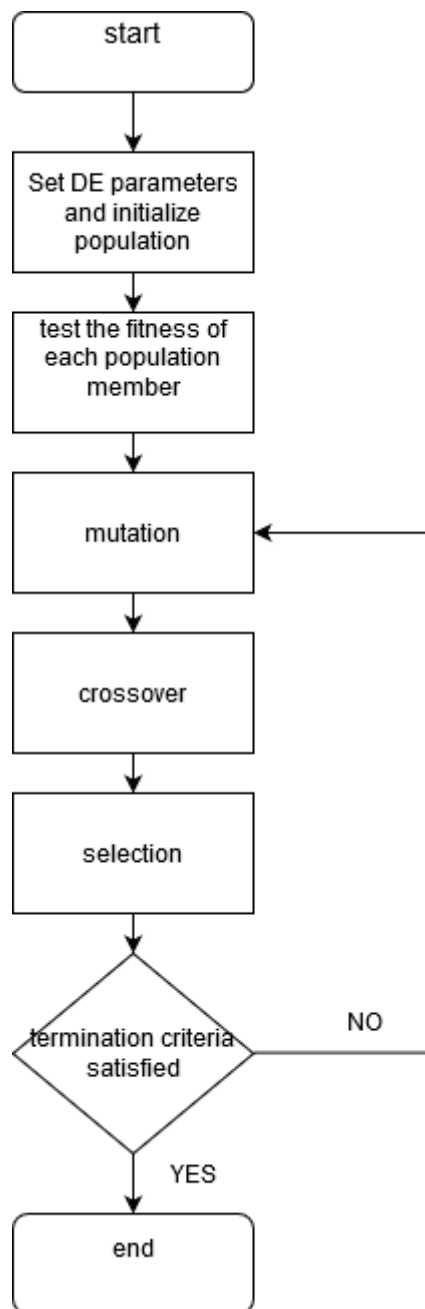


Fig 2.5: DE algorithm flowchart

V. Marine Predators Algorithm (MPA):

Marine Predators algorithm is a new metaheuristic algorithm developed in 2019 and published in 2020, even used for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea [15]

As its name indicates, it is based on the behaviour of marine predators looking for prey. Based on “surviving of the fittest” predators try to select the best strategy that maximizes their chances of encountering a prey.

MPA steps:

1-initialization: the initial solution is uniformly distributed over the search space as the first trial according to Eq (2.24)

$$X_0 = X_{min} + rand(X_{max} - X_{min}) \quad (2.24)$$

Where X_{min} and X_{max} are the lower and upper limits for variables and rand is a random number in the range of 0 to 1.

Based on the *survival of the fittest theory*, it is said that top predators in nature are more talented in foraging. Thus, the fittest solution is nominated as a top predator to construct a matrix which is called Elite. Arrays of this matrix oversee searching and finding the prey based on the information on prey’s positions

$$Elite = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \dots & X_{1,d}^I \\ X_{2,1}^I & X_{2,2}^I & \dots & X_{2,d}^I \\ \dots & \dots & \dots & \dots \\ X_{n,1}^I & X_{n,2}^I & \dots & X_{n,d}^I \end{bmatrix}_{n \times d} \quad (2.25)$$

Where $\vec{X}^I$ represents the top predator vector, which is replicated n times to construct the *Elite* matrix. n is the number of search agents while d is the number of dimensions. It is noted that both predator and prey are considered as search agents. Because by the time that a predator is looking for its prey, the prey is looking for its own food. At the end of each iteration, the *Elite* will be updated if the top predator is substituted by the better predator.

Another matrix with the same dimension as *Elite* is called *Prey* which the predators update their positions based on it. In a simple word, the initialization creates the initial *Prey* of which the fittest one (predator) constructs the *Elite*. The *Prey* is shown as follows:

$$\text{Prey} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \\ \dots & \dots & \dots & \dots \\ X_{n,1} & X_{n,2} & \dots & X_{n,d} \end{bmatrix}_{n \times d} \quad (2.26)$$

It should be noted that the whole process of the optimization is mainly and directly related to these two matrices.

2- Optimization phases: MPA optimization process is divided into three main phases of optimization considering different velocity ratio and at the same time mimicking the entire life of a predator and prey:

- a) Phase 1:** In high-velocity ratio or when predator is moving faster than prey. This scenario happens in the initial iterations of optimization, where the exploration matters. The following mathematical model is applied:

While $Iter < \frac{1}{3} Max_{iter}$

$$\overrightarrow{stepsize_i} = \overrightarrow{R_B} \otimes (\overrightarrow{Elite_i} - \overrightarrow{R_B} \otimes \overrightarrow{Prey_i}) \quad i = 1, \dots, n \quad (2.27)$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P \cdot \overrightarrow{R} \otimes \overrightarrow{stepsize_i} \quad (2.28)$$

Where R_B is a vector containing random numbers based on Normal distribution representing the Brownian motion. The notation $\otimes$ shows entry-wise multiplications. The multiplication of by prey simulates the movement of prey. $P=0.5$ is a constant number, and R is a vector of uniform random numbers in $[0,1]$. This scenario happens in the first third of iterations when the step size or the velocity of movement is high for high exploration ability. $Iter$ is the current iteration while Max_iter is the maximum one.

- b) Phase 2:** In this stage, the prey and predator are moving in the same area, and this movement simulates the process of searching for the prey/food. Furthermore, this refers to the process of changing the status of the MPA from exploration to exploitation.

The following mathematical model is applied:

While $\frac{1}{3} Max_{iter} < Iter < \frac{2}{3} Max_{iter}$

For the first half of population

$$\overrightarrow{stepsize_i} = \overrightarrow{R_L} \otimes (\overrightarrow{Elite_i} - \overrightarrow{R_L} \otimes \overrightarrow{Prey_i}) \quad i = 1, \dots, \frac{n}{2} \quad (2.29)$$

Then Eq 2.28 is re-applied.

where $\overrightarrow{R_L}$ is a vector of random numbers based on Lévy distribution representing Lévy movement. The multiplication of $\overrightarrow{R_L}$ and $Prey$ simulates the movement of prey in Lévy manner while adding the step size to prey position simulates the movement of prey.

For the second half of population:

$$\overrightarrow{stepsize_i} = \overrightarrow{R_B} \otimes (\overrightarrow{R_B} \otimes Elite_i - \overrightarrow{Prey_i}) \quad i = \frac{n}{2}, \dots, n \quad (2.30)$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P.CF \otimes \overrightarrow{stepsize_i} \quad (2.31)$$

where CF is the parameter that controls the step size of movement for the predator.

c) phase 3: In low-velocity ratio or when predator is moving faster than prey. This scenario happens in the last phase of the optimization process which is mostly associated with high exploitation capability.

The following mathematical model is applied:

$$While \text{Iter} > \frac{2}{3} Max_{Iter}$$

$$\overrightarrow{stepsize_i} = \overrightarrow{R_L} \otimes (\overrightarrow{R_L} \otimes Elite_i - \overrightarrow{Prey_i}) \quad i = 1, \dots, n \quad (2.32)$$

Eq (2.31) is re-applied

3- Eddy formation and FADs' effect:

Another point which causes a behavioural change in marine predators is environmental issues such as the eddy formation or Fish Aggregating Devices (FADs) effects. the FADs effect is mathematically presented as:

$$\overrightarrow{Prey_i} = \begin{cases} \overrightarrow{Prey_i} + CF[\vec{X}_{min} + \vec{R}(\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} & \text{if } r \leq FADs \\ \overrightarrow{Prey_i} + [FADs(1 - r) + r](\overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}}) & \text{if } r > FADs \end{cases} \quad (2.33)$$

Where $FADs = 0.2$ is the probability of FADs effect on the optimization process. $\vec{U}$ is the binary vector with arrays including zero and one. This is constructed by generating a random vector in $[0,1]$ and changing its array to zero if the array is less than 0.2 and one if it is greater than 0.2. r is the uniform random number in $[0,1]$. $\vec{X}_{min}$ and $\vec{X}_{max}$ are the vectors containing the lower and upper bounds of the dimensions. Subscripts r_1 and r_2 denote random indexes of $prey$ matrix.

4- Marine memory: Based on the highlighted points, marine predators have a good memory in reminding the place where they have been successful in foraging. This capability is simulated by memory saving in MPA. After updating the $Prey$ and implementing FADs

CHAPTER II: Optimization algorithms

effect, this matrix is evaluated for fitness to update the *Elite*. The fitness of each solution of the current iteration is compared to its equivalent in prior iteration, and the current one replaces the solution if it is more fitted.

2.4.Conclusion: the algorithms mentioned may differ in their working process. They may differ in the number of steps, and the number of parameters required to be tuned. However, they can all be adapted for our optimization purposes, which will be shown in the next chapter.

CHAPTER III: PV parameters extraction

3.1.Introduction:

As we have seen in chapter two, to model a PV module several parameters must be obtained. Modelling is important whether it is for educational purposes or for actually assessing the performance of a PV panel. Parameter extraction of photovoltaic (PV) models, which remains a multi-variable, nonlinear, and multi-modal problem, has recently gained considerable attention in the simulation and calculation of solar PV systems. Extracting these parameters can be done by using either the manufacturer's data sheet or actual experimental data. In this project, the algorithms mentioned in chapter three (PSO, WDO, EMA, DE) have been adapted to suite the purpose of PV parameters extraction of single-diode model using experimental data. Methods used in extraction can be classified into three categories: analytical methods, iterative-based methods, and meta-heuristic methods.

3.2.Parameters extraction methods:

3.2.1 Analytical method:

$$I = I_{ph} - I_s * \exp\left(\frac{V+R_s I}{aV_t} - 1\right) - \frac{V+IR_s}{R_p} \quad (3.1)$$

Analytical expressions for the extraction of the model parameters:

$$a = \frac{V_m + I_m R_{s0} - V_{oc}}{V_T \left[\ln\left(I_{sc} - \frac{V_m}{R_{p0}} - I_m\right) - \ln\left(I_{sc} - \frac{V_{oc}}{R_p}\right) + \frac{I_m}{I_{sc} - \left(\frac{V_{oc}}{R_{p0}}\right)} \right]} \quad (3.2)$$

$$I_0 = \left(I_{sc} - \frac{V_{oc}}{R_p}\right) * \exp\left(-\frac{V_{oc}}{nV_T}\right) \quad (3.3)$$

$$R_s = R_{s0} - \frac{nV_T}{I_0} \exp\left(-\frac{V_{oc}}{nV_T}\right) \quad (3.4)$$

$$R_{s0} = \left(-\left(\frac{dV}{dI}\right)_{V=V_{oc}}\right) \quad (3.5)$$

$$R_p = R_{p0} = -\left(\frac{dV}{dI}\right)_{I=I_{sc}} \quad (3.5)$$

$$I_{ph} = I_{sc}\left(1 + \frac{R_s}{R_p}\right) \quad (3.6)$$

[16]

3.2.2 Iterative method: this method involves minimizing the objective square error where the error function is defined as the difference between estimated and experimental currents. It is expressed as follows:

$$S(\theta) = \sum_{i=1}^N (I_{i,measured} - I_{i,estimated})^2 \quad (3.7)$$

where:

$S(\theta)$: is the objective function to minimize

N: is the number of points measured (V_i, I_i)

$I_{i,measured}$: is the measured current

$I_{i,estimated}$: is the estimated current

$\theta=[a, I_0, I_{ph}, R_p, R_s]$: parameters to estimate.

Eq (3.1) is implicit in I, so for computation purposes $I_{i,measured}$ and $V_{i,measured}$ are substituted in Eq (3.1 1)

$$I(\theta) = I_{ph} - I_s * \exp\left(\frac{V_{i,measured} + R_s I_{i,measured}}{a V_t} - 1\right) - \frac{V_{i,measured} + R_s I_{i,measured}}{R_p} \quad (3.8)$$

Eq. (3.8) is nonlinear in its parameters and hence the resulting set of normal equations $F(\theta)$, derived from multivariate calculus for a minimum to occur, will also be nonlinear and no exact solution can be generally found. Newton's method can be used to obtain an approximation to the exact solution for the nonlinear set of equations $F(\theta) = 0$. The Newton functional iteration procedure evolves from:

$$[\theta_k] = [\theta_{k-1}] - [J(\theta_{k-1})]^{-1} [F(\theta_{k-1})] \quad (3.9)$$

where $[J(\theta)]$ is the Jacobian matrix. The NR method converges rapidly, but it is only a local optimization technique that also requires a sufficiently accurate starting vector θ . [17]

3.3.3 Meta-heuristics: this is the method chosen in this work. It is rather a new method compared to the previous ones. In this method, no initial guess is needed, only the allowed range of the parameters up for optimization is provided. This allows more flexibility to look on many potential candidate solutions and improve their fitness accordingly.

The objective function to be minimized, in this project, is the Root mean square error (RMSE) between experimental and estimated currents. RMSE is given by equation

$$F(\theta) = \sqrt{\frac{1}{N} \sum_{i=1}^N (I_{i,measured} - I_{i,estimated})^2} \quad (3.10)$$

$$I = I_{ph} - I_s * \exp\left(\frac{V + R_s I}{N_s a V_T} - 1\right) - \frac{V + I R_s}{R_p} \quad (3.11)$$

CHAPTER III: PV Parameters extraction

$$I(\theta) = I_{ph} - I_s * \exp\left(\frac{V_{i,measured} + R_s I_{i,measured}}{aV_t} - 1\right) - \frac{V_{i,measured} + R_s I_{i,measured}}{R_p} \quad (3.12)$$

$\theta=[a, I_0, I_{ph}, R_p, R_s]$: parameters to be computed.

Below, is the way this method has been used in PV parameters extraction

- All the algorithms used in this section are population-based ones.
- Experimental data (current and voltage) of PV modules are read from excel files.
The four PV modules used are: *CondorCEM150M*, *KyoceraKC125GHT*, *SanyoHIP-190B2-BO-01*, and *SilikenSLK60P6L*
- The number of iterations and number of population are chosen.
As a rule of thumb, since the number of parameters is five, the number of population is chosen to be ten times that number (number of population=50).
The maximum number of iterations is chosen to be 1000.
- Algorithm coefficients are initialized; these coefficients can be tuned along the program depending on the algorithm.
- A random initial population is initialized within the parameters boundaries.
Each population member pop_i is a vector of the five parameters of interest:
$$pop_i = [a \ R_s \ R_p \ I_{pv} \ I_o]$$
- The fitness (RMSE) of each population member (candidate solution) is evaluated.
- Population members are updated; with the boundary conditions checked and respected, in order to get a better RMSE.
- The algorithms run for a defined number of iterations and give the smallest obtained value of RMSE
- In the end of the MATLAB programs running, the following results are obtained:
The single diode model 5 parameters, I-V curves with both experimental and calculated currents on them, and a graph of RMSE vs the number of iterations.

3.3. Results

In this section, the results of PSO, WDO, EMA, DE, and MPA based PV single-diode model parameters extraction are shown. The results consist of I-V curves depicting both estimated and measured current data, and graphs showing RMSE vs iteration number.

The results are organised as follows:

- Each algorithm has four sections of results (for the four PV modules)
- Each section comprises two graphs: RMSE vs iteration and I-V curve.

3.3.1. PSO results:

a) Condor CEM150M

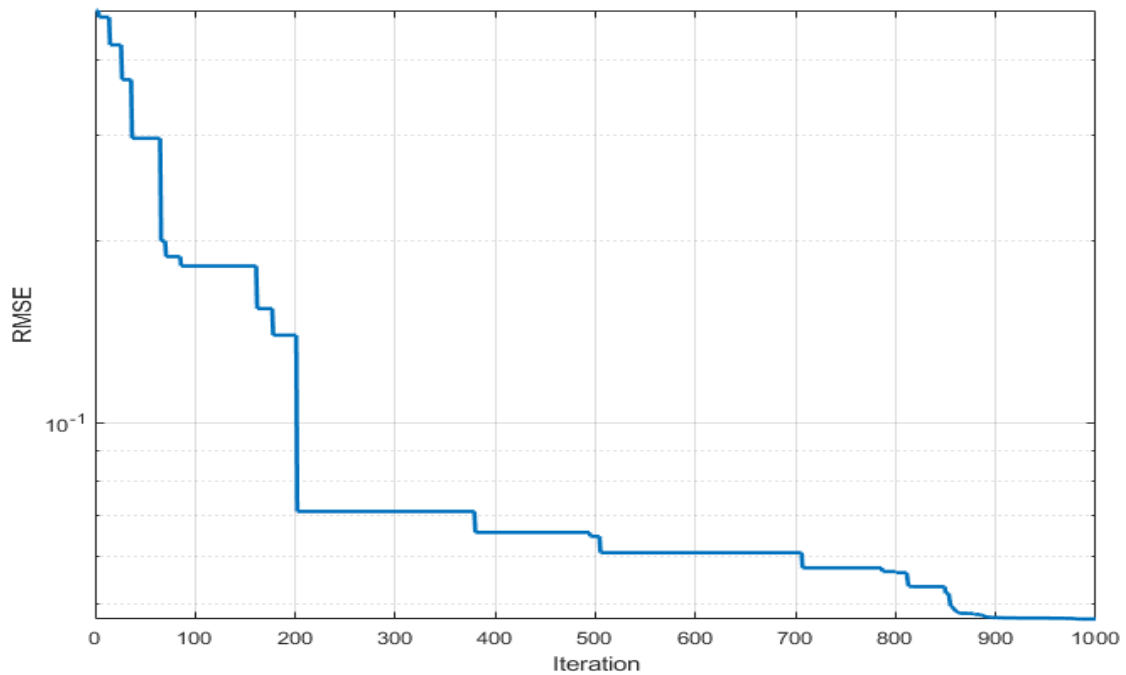


Fig 3.1: RMSE vs Iteration graph using PSO on Condor CEM150M PV module data.

CHAPTER III: PV Parameters extraction

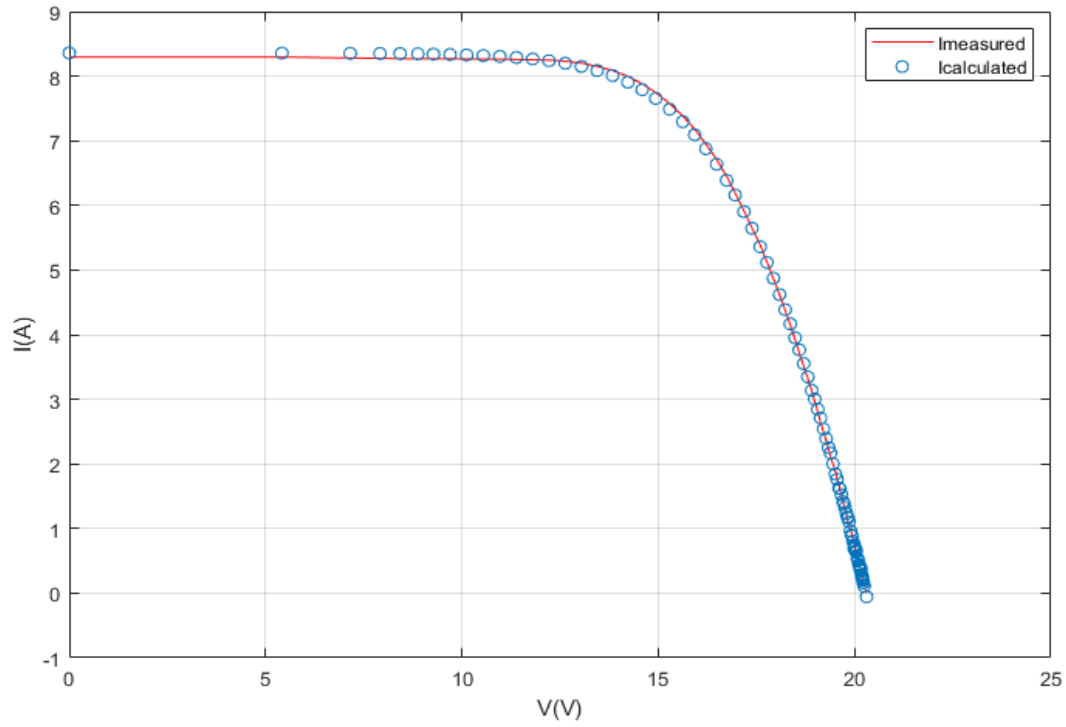


Fig 3.2: I-V curve showing both calculated (through PSO) and measured current of Condor CEM150M PV module

b) KyoceraKC125GHT:

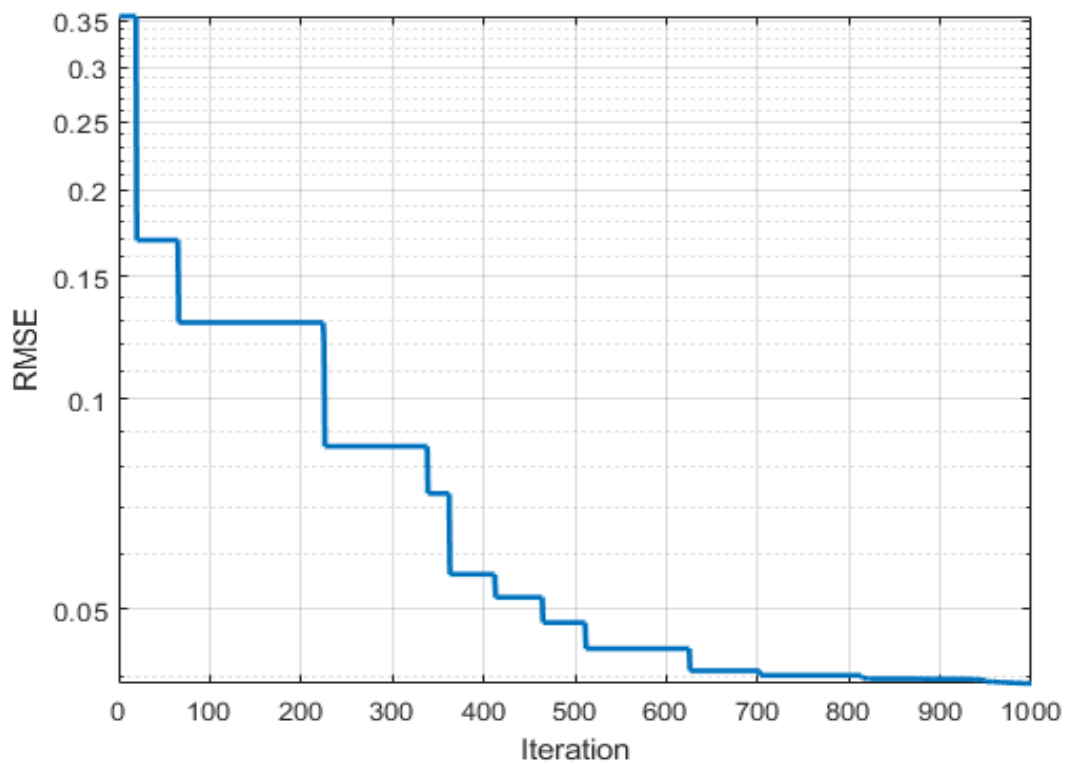


Fig 3.3: RMSE vs Iteration graph using PSO on KyoceraKC125GHT module data.

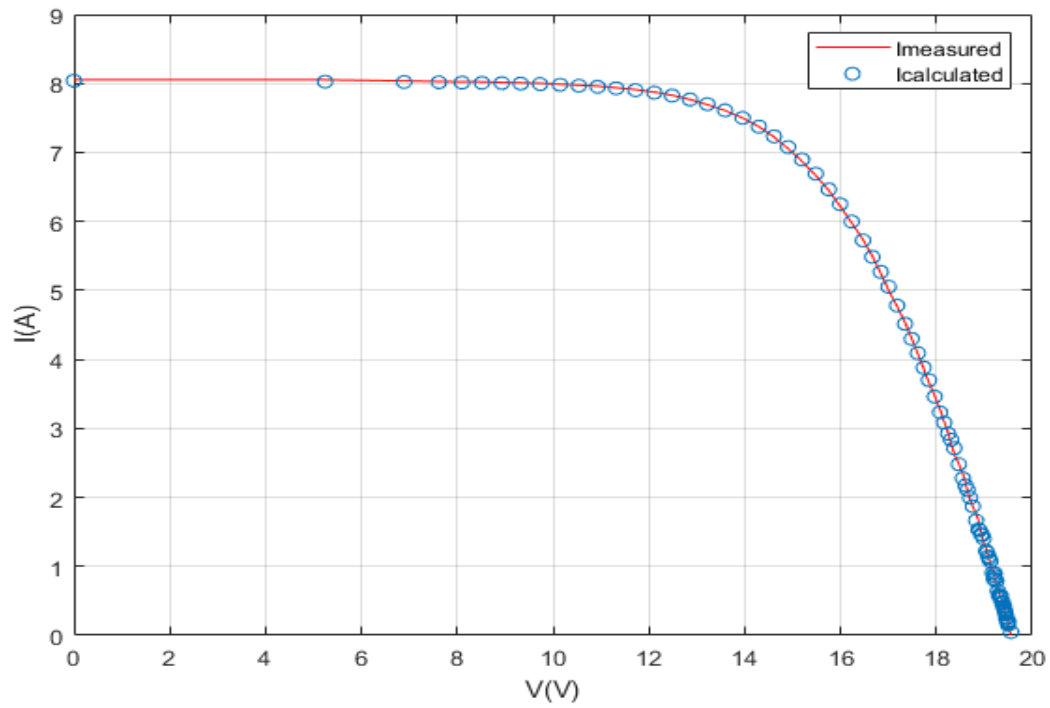


Fig 3.4: *I-V curve showing both calculated (through PSO) and measured currents of Kyocera KC125GHT PV module*

c) SilikenSLK60P6L:

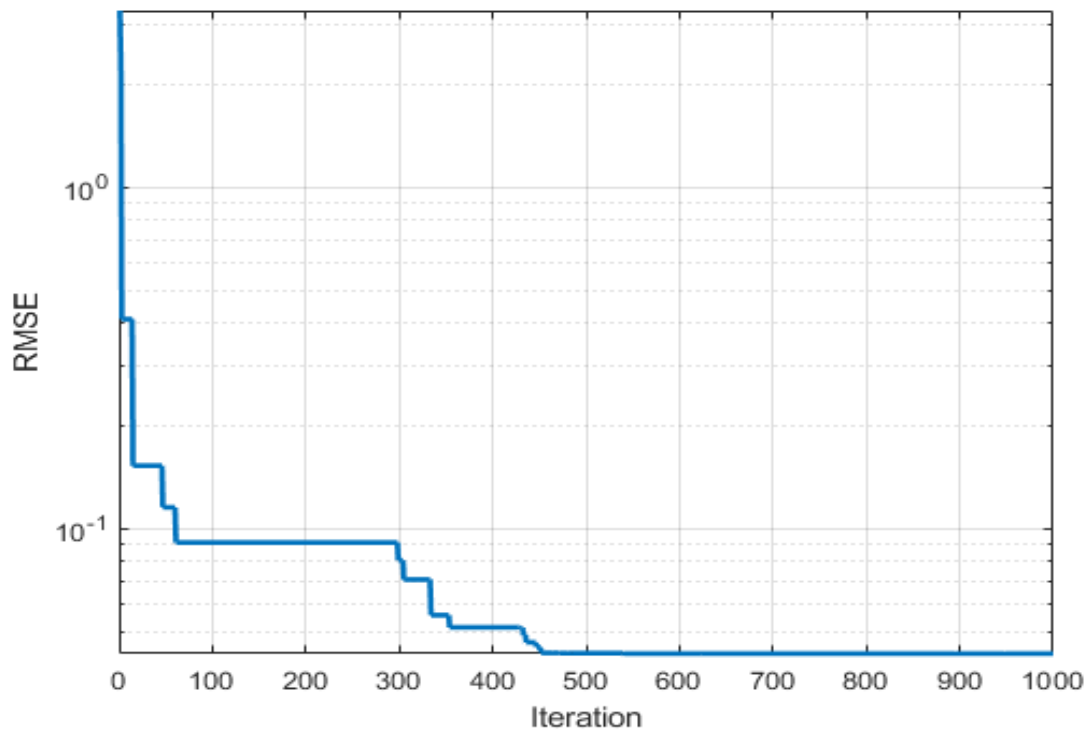


Fig 3.5: *RMSE vs Iteration graph using PSO on Siliken SLK60P6L PV module data.*

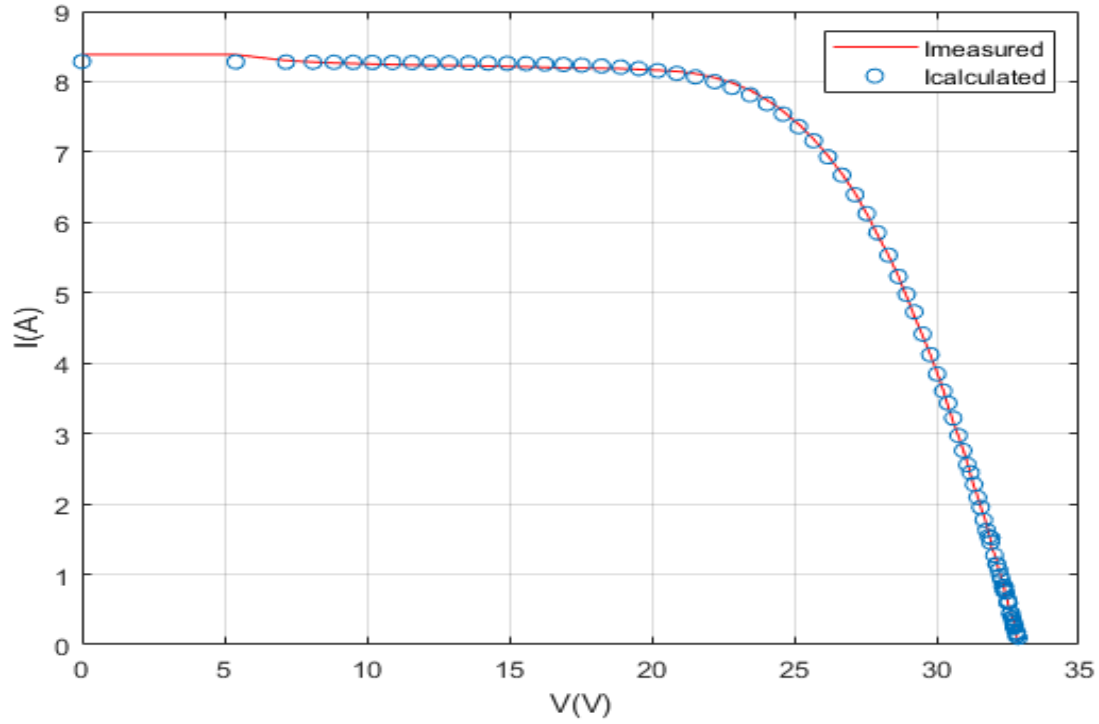


Fig 3.6: I-V curve showing both calculated (through PSO) and measured currents of SilikenSLK60P6L PV module

d) Sanyo HIP-190B2-BO-01:

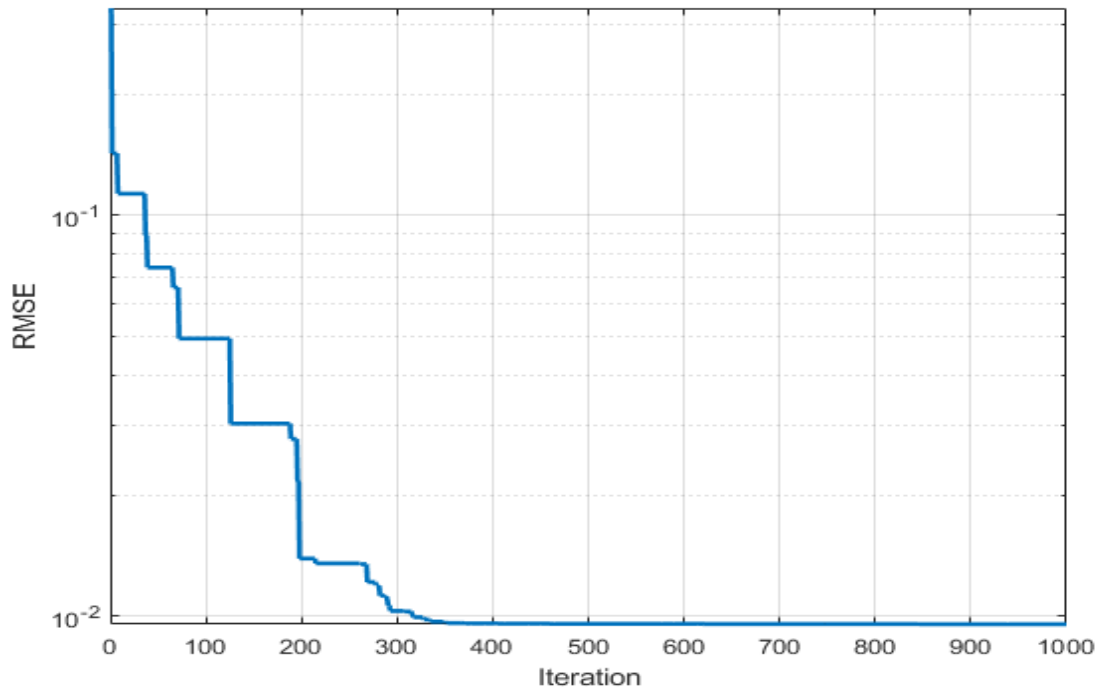


Fig 3.7: RMSE vs Iteration graph using PSO on Sanyo HIP-190B2-BO-01 PV module data.

CHAPTER III: PV Parameters extraction

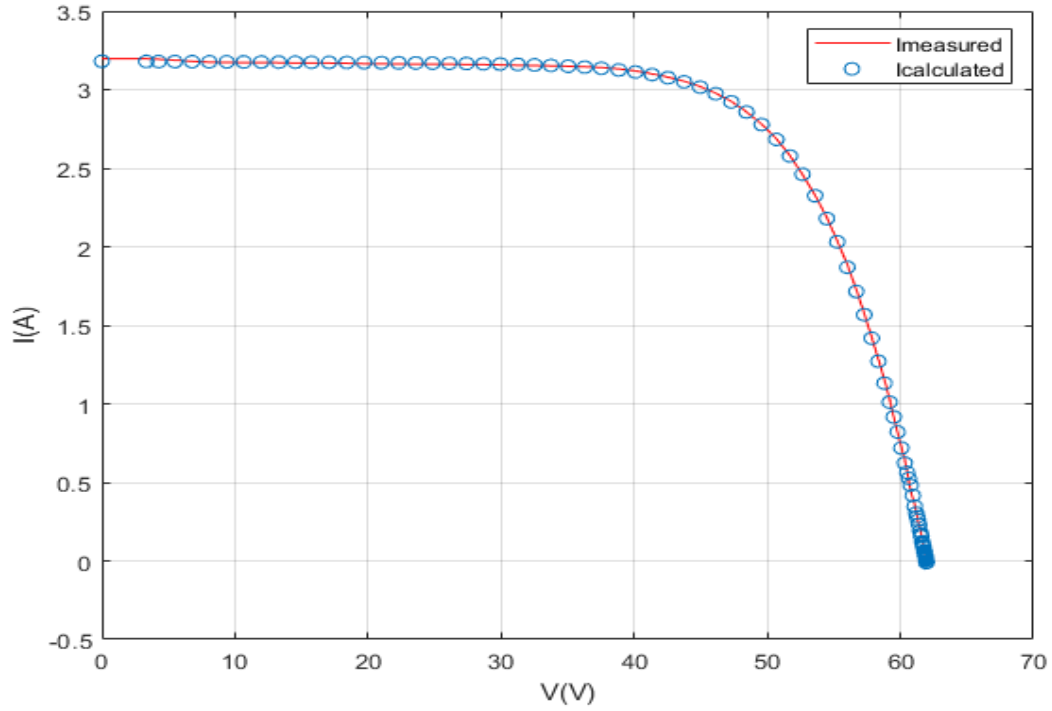


Fig 3.8: I-V curve showing both calculated (through PSO) and measured currents of Sanyo HIP-190B2-BO-01 PV module

PV module	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
CondorCEM150M	1.4483	0.2216	$4.223 \cdot 10^3$	8.3594	$8.2937 \cdot 10^{-6}$	0.0472
Kyocera KC125GHT	1.3578	0.2335	$4.471 \cdot 10^2$	8.0433	$6.8308 \cdot 10^{-6}$	0.0390
Siliken-SLK60P6L 210Wp	1.2818	0.3912	$9.356 \cdot 10^2$	8.2890	$2.4152 \cdot 10^{-6}$	0.0429
Sanyo-HIP-190B2- BO-01	1.6765	0.9788	$2.397 \cdot 10^3$	3.1822	$4.1867 \cdot 10^{-6}$	0.0095

Table 1:extracted model parameters and RMSE using PSO

3.3.2 WDO results:

a) Condor CEM150M:

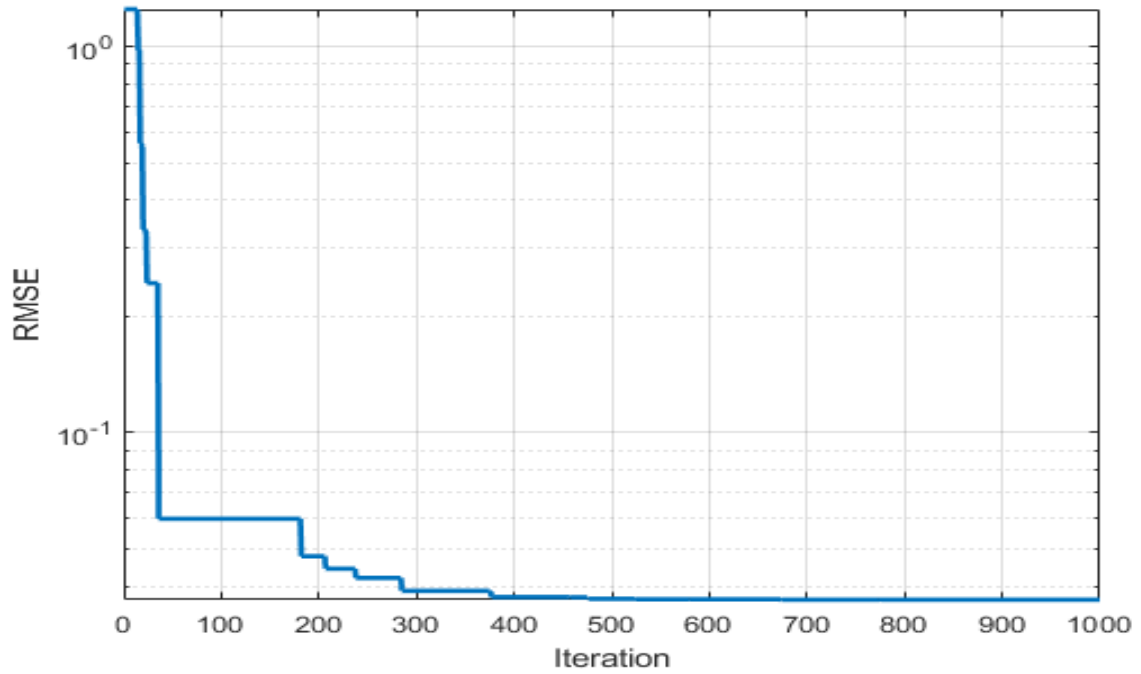


Fig 3.9: RMSE vs Iteration graph using WDO on Condor CEM150M PV module data.

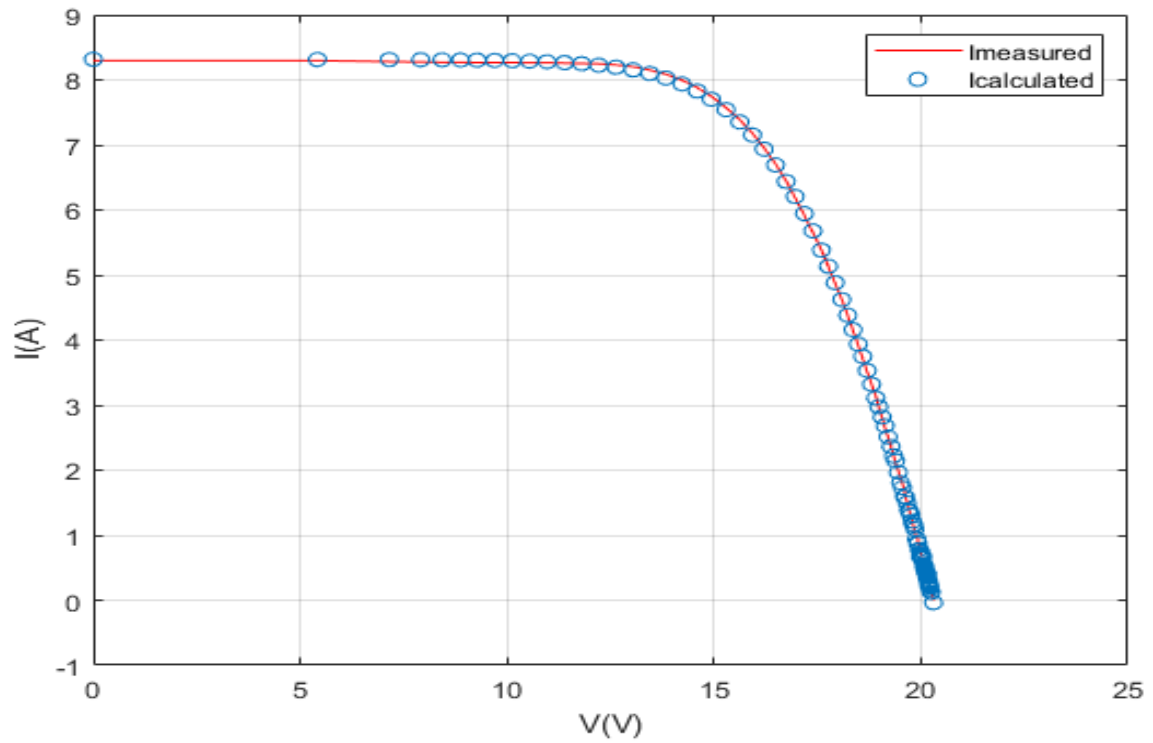


Fig 3.10: I-V curve showing both calculated (through WDO) and measured currents of Condor CEM150M PV module

b) Kyocera KC125GHT:

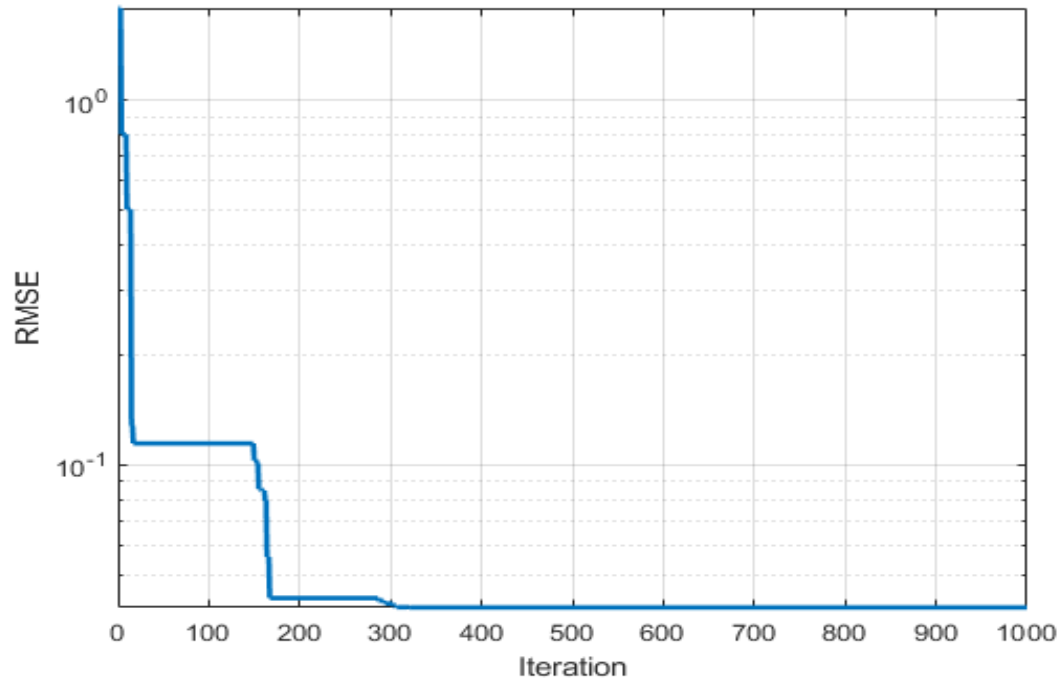


Fig 3.11: RMSE vs Iteration graph using WDO onKyocera KC125GHT module data.

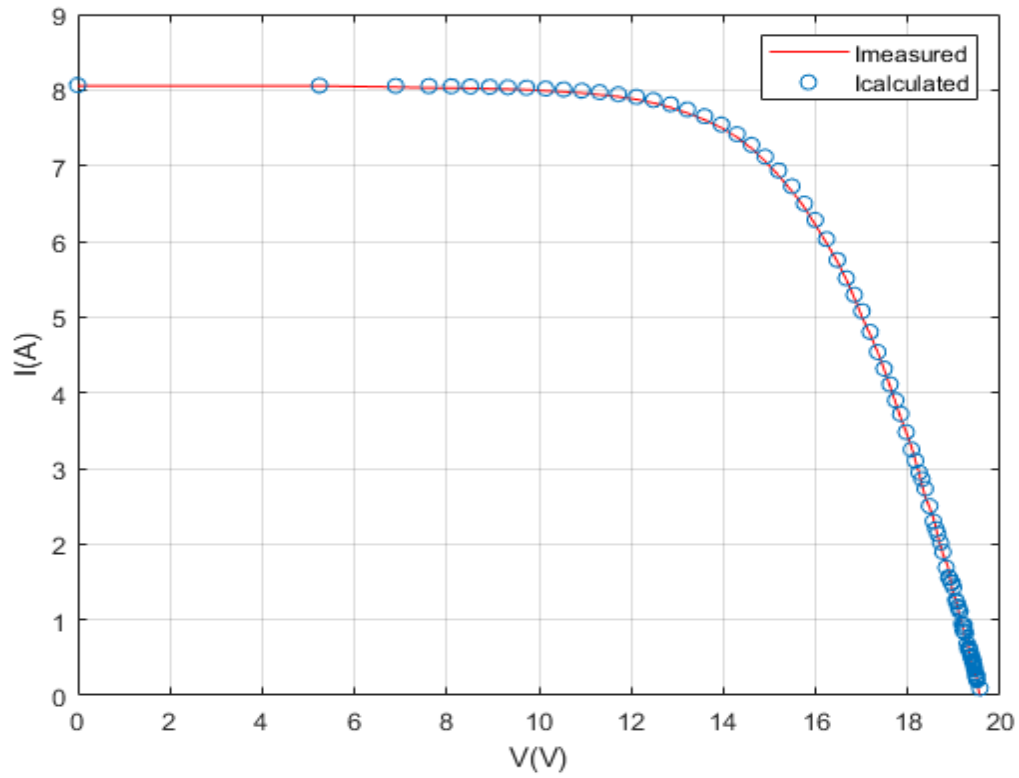


Fig 3.12: I-V curve showing both calculated (through WDO) and measured currents of Kyocera KC125GHT PV module

c) Siliken SLK60P6L:

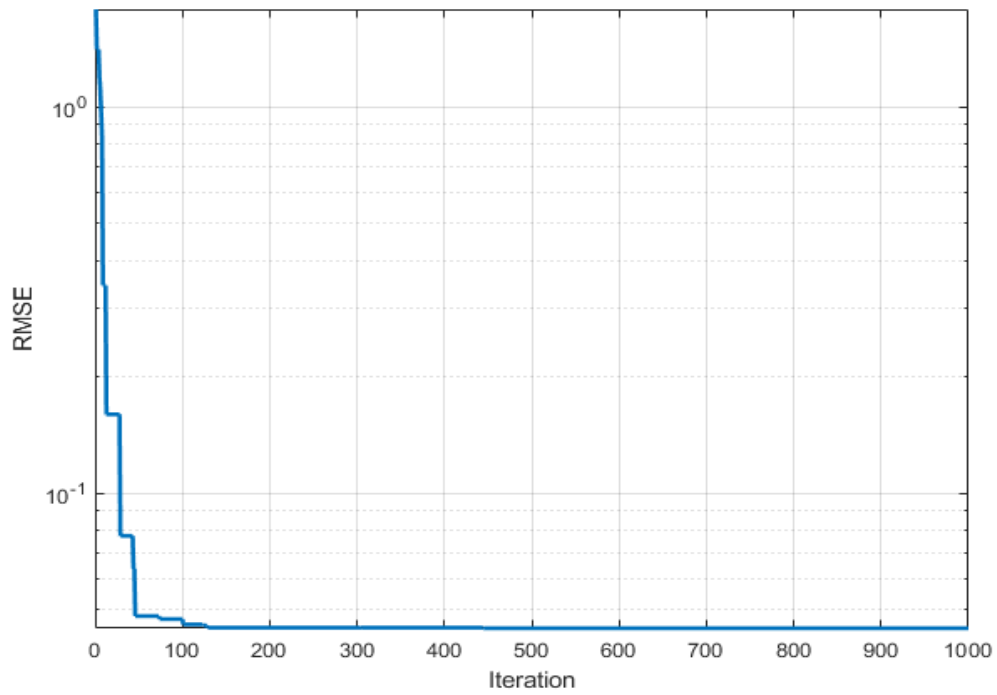


Fig 3.13: RMSE vs Iteration graph using WDO on Siliken SLK60P6LPV module data.

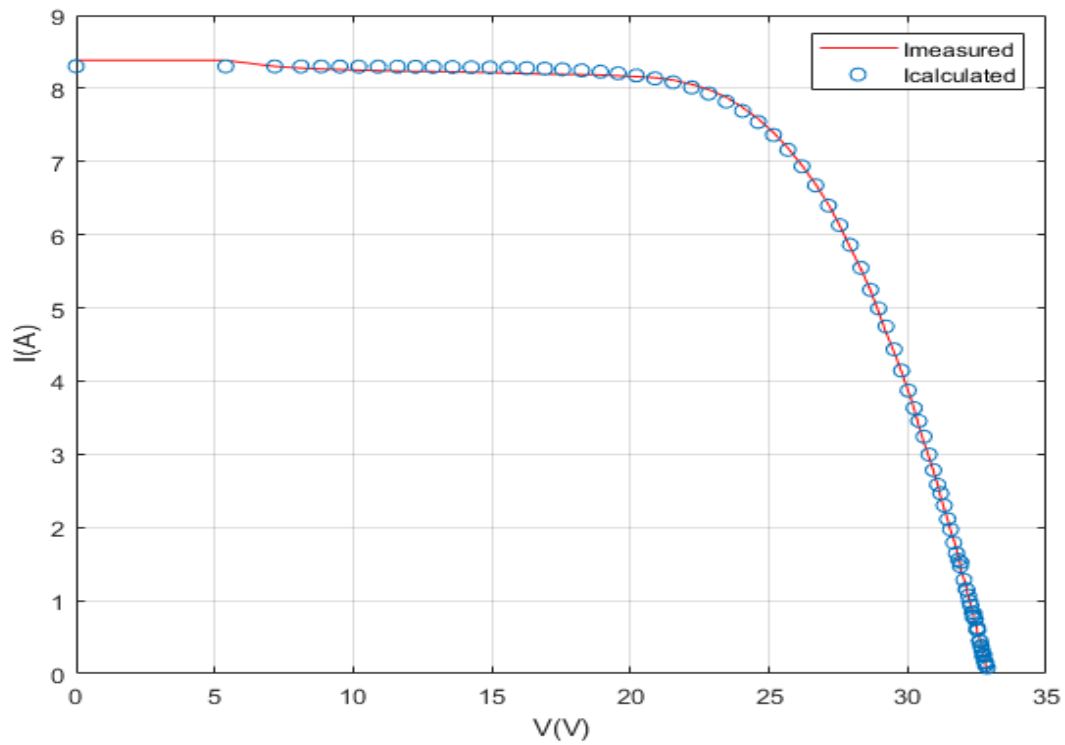


Fig 3.14: I-V curve showing both calculated (through WDO) and measured currents of Siliken SLK60P6L PV module

d) Sanyo HIP-190B2-BO-01:

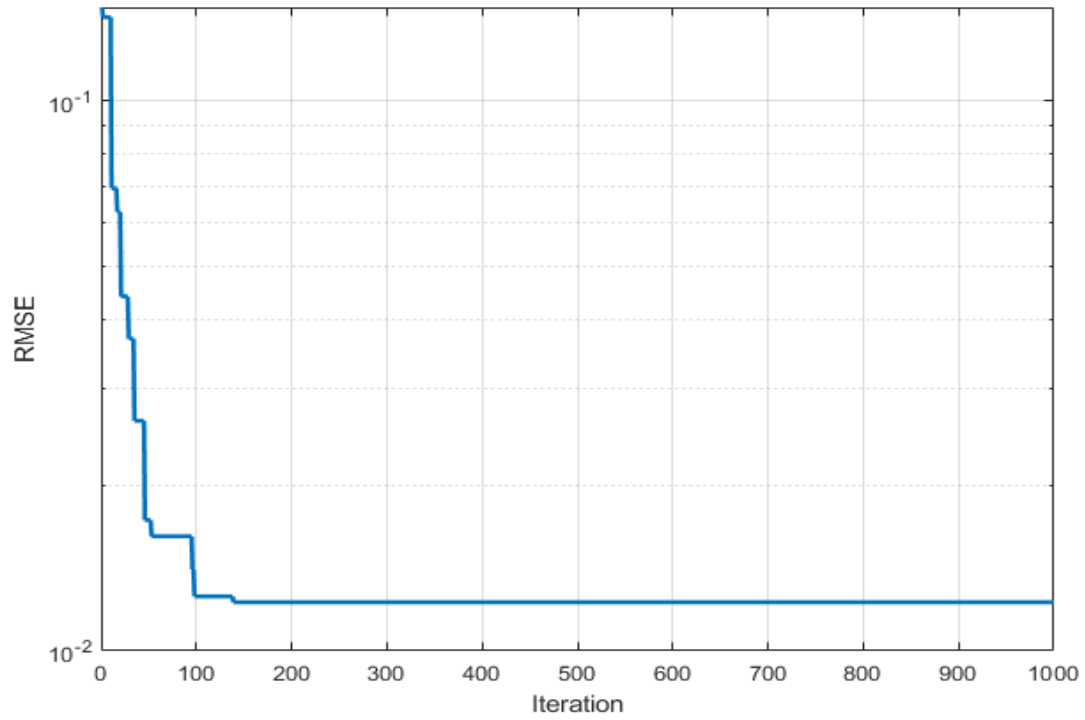


Fig 3.15: RMSE vs Iteration graph using WDO on Sanyo HIP-190B2-BO-01 PV module data.

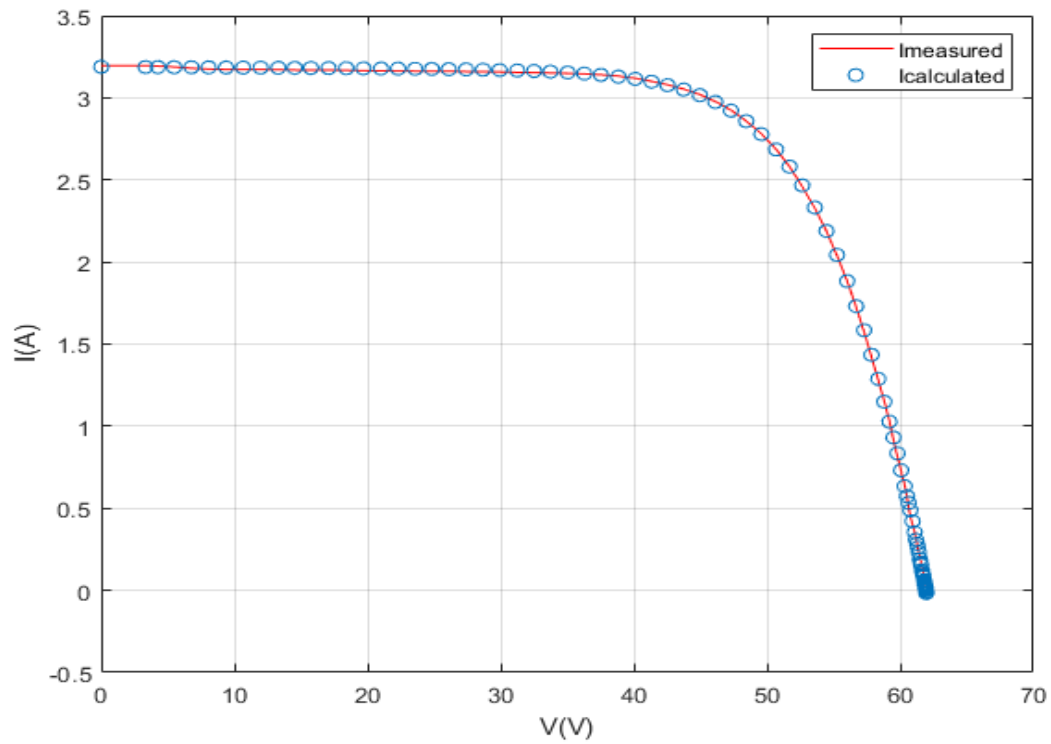


Fig 3.16: I-V curve showing both calculated (through WDO) and measured currents of Sanyo HIP-190B2-BO-01 PV module

CHAPTER III: PV Parameters extraction

PV module	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
Condor CEM150M	1.3033	0.2460	$2.070 \cdot 10^3$	8.3183	$1.7658 \cdot 10^{-6}$	0.0369
Kyocera KC125GHT	1.3479	0.2376	$1.065 \cdot 10^3$	8.0657	$6.1658 \cdot 10^{-6}$	0.0406
Siliken SLK60P6L 210Wp	1.3301	0.3759	$3.397 \cdot 10^3$	8.3035	$4.1984 \cdot 10^{-6}$	0.0446
Sanyo HIP-190B2-BO- 01	1.7343	0.8780	$2.252 \cdot 10^3$	3.1901	$6.5997 \cdot 10^{-6}$	0.0122

Table 2: extracted model parameters and RMSE using WDO

3.3.3 EMA results:

a) Condor CEM150M:

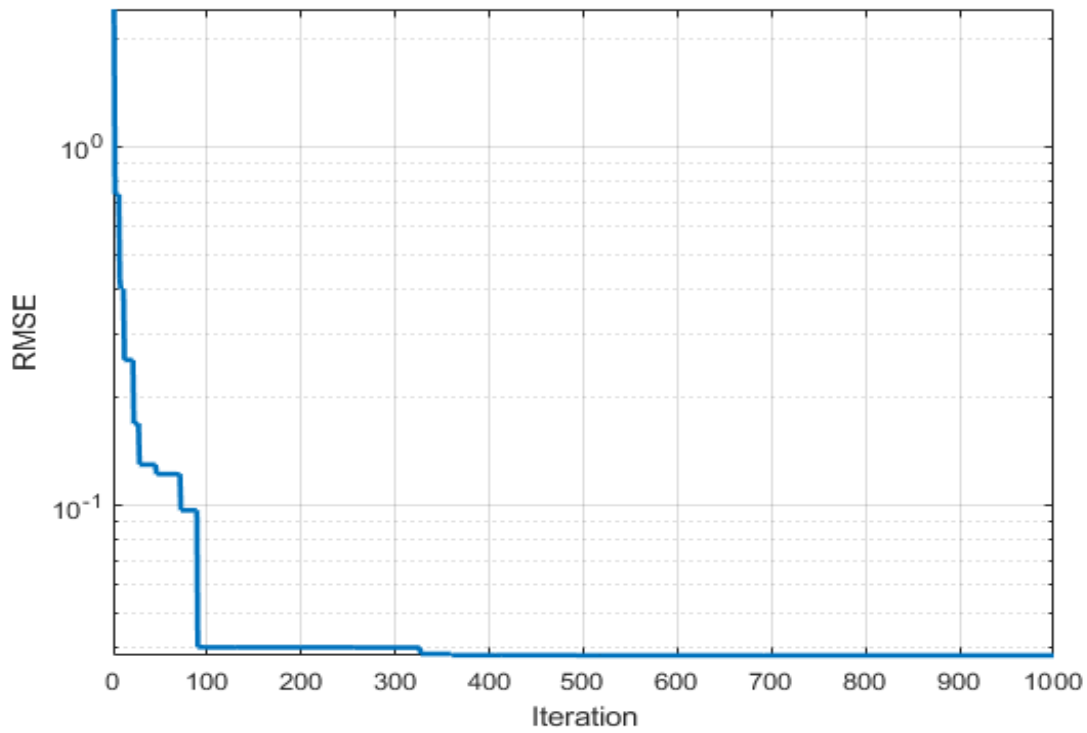


Fig 3.17: RMSE vs Iteration graph using EMA on Condor CEM150M PV module data.

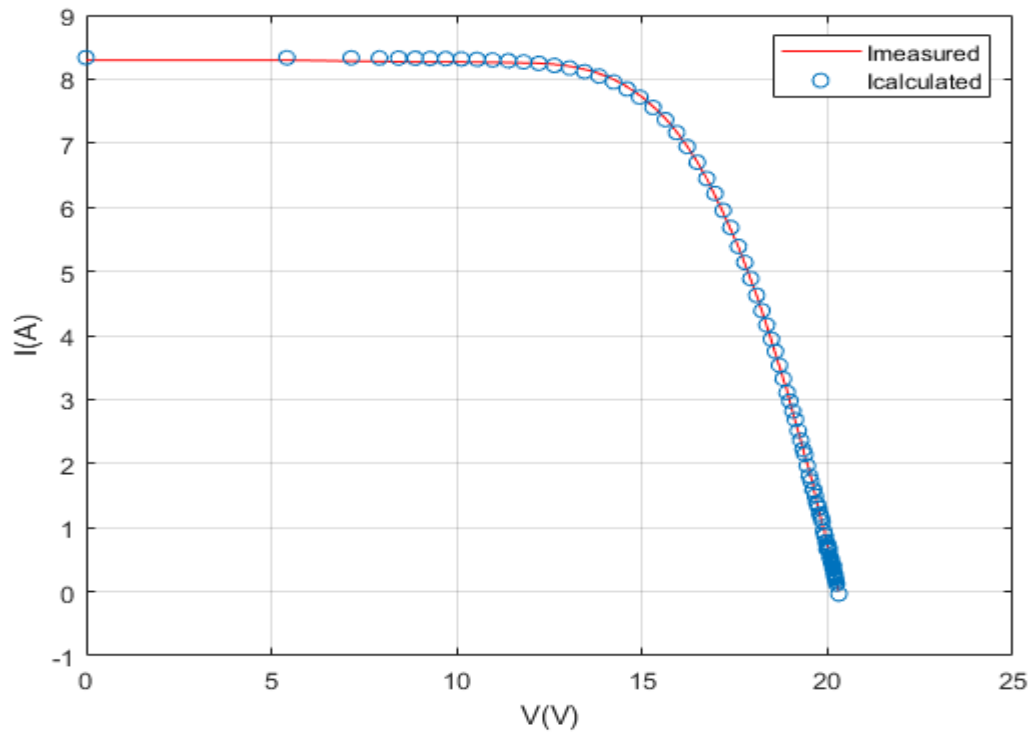


Fig 3.18: *I-V curve showing both calculated (through EMA) and measured currents of Condor CEM150M PV module*

b) Kyocera KC125GHT:

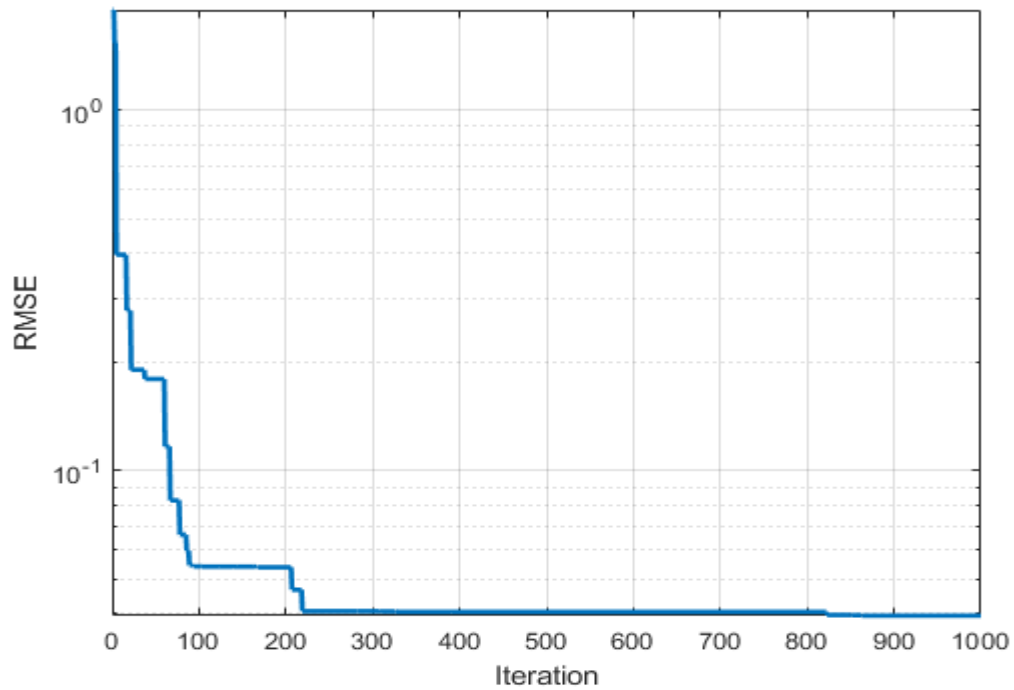


Fig 3.19: *RMSE vs Iteration graph using EMA on Kyocera KC125GHT PV module data.*

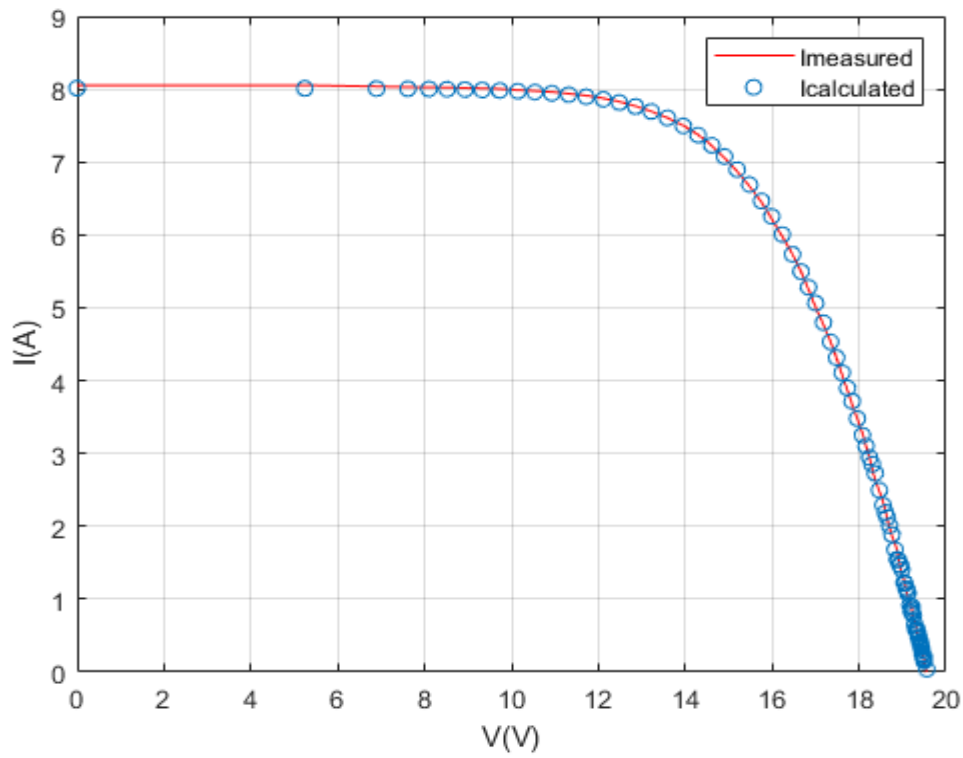


Fig 3.20: *I-V curve showing both calculated (through EMA) and measured currents of Kyocera KC125GHTPV module*

c) Siliken SLK60P6L:

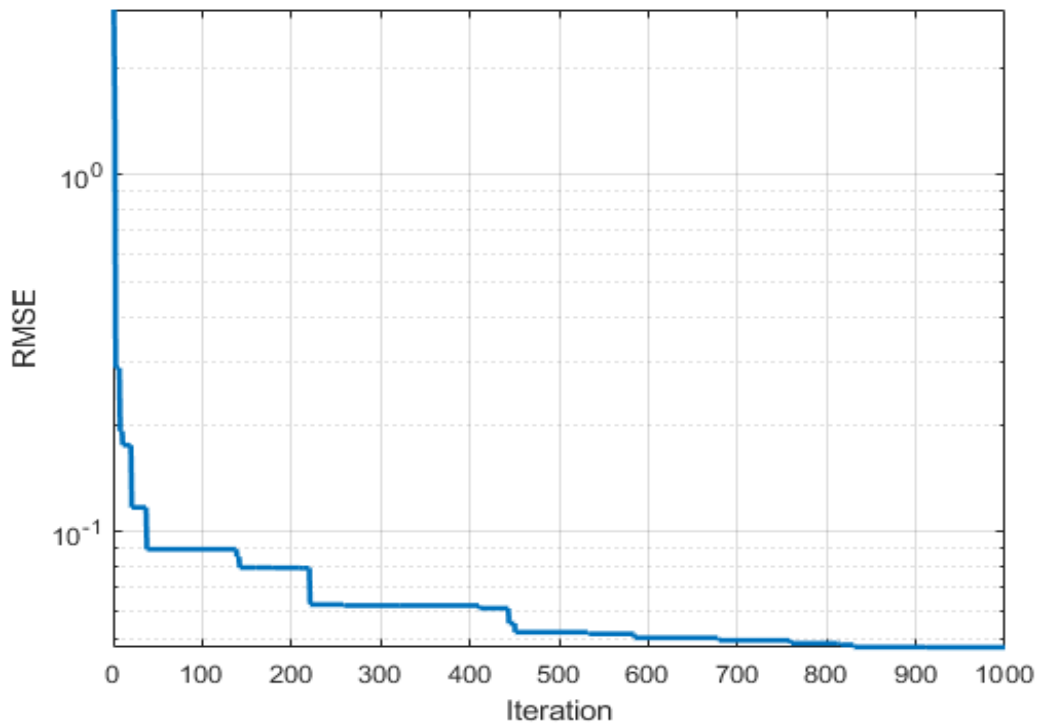


Fig 3.21: *RMSE vs Iteration graph using EMA on Siliken SLK60P6LPV module data.*

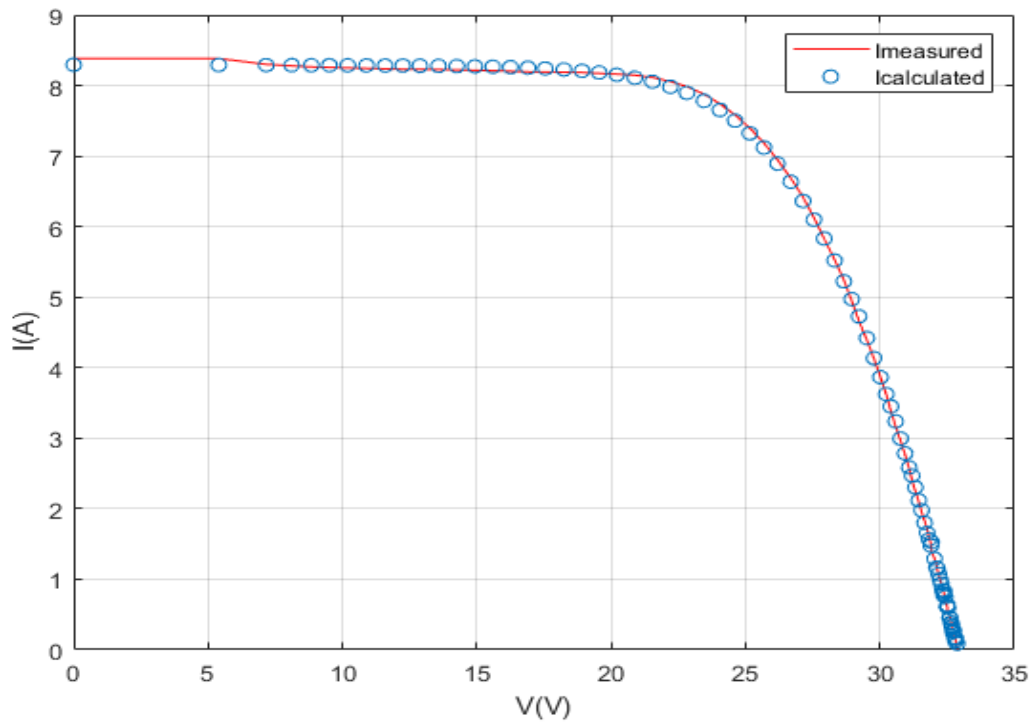


Fig 3.22: I-V curve showing both calculated (through EMA) and measured currents of Siliken SLK60P6L PV module

d) Sanyo HIP-190B2-BO-01:

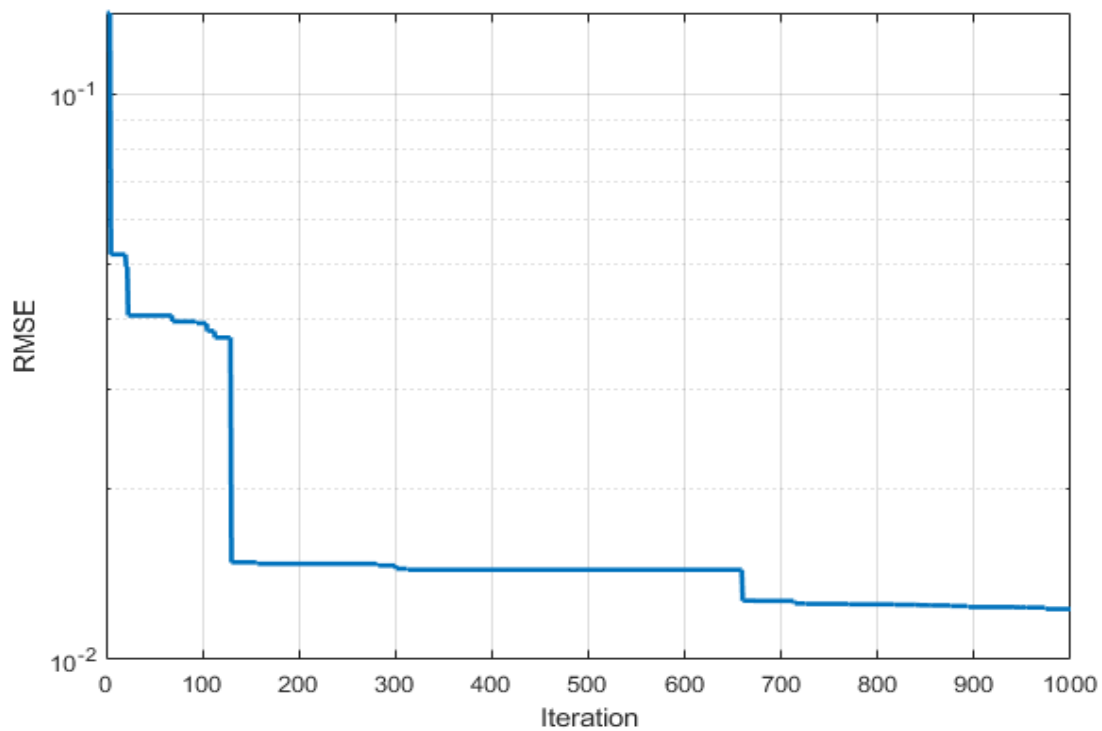


Fig 3.23: RMSE vs Iteration graph using EMA on Sanyo HIP-190B2-BO-01 PV module data.

CHAPTER III: PV Parameters extraction

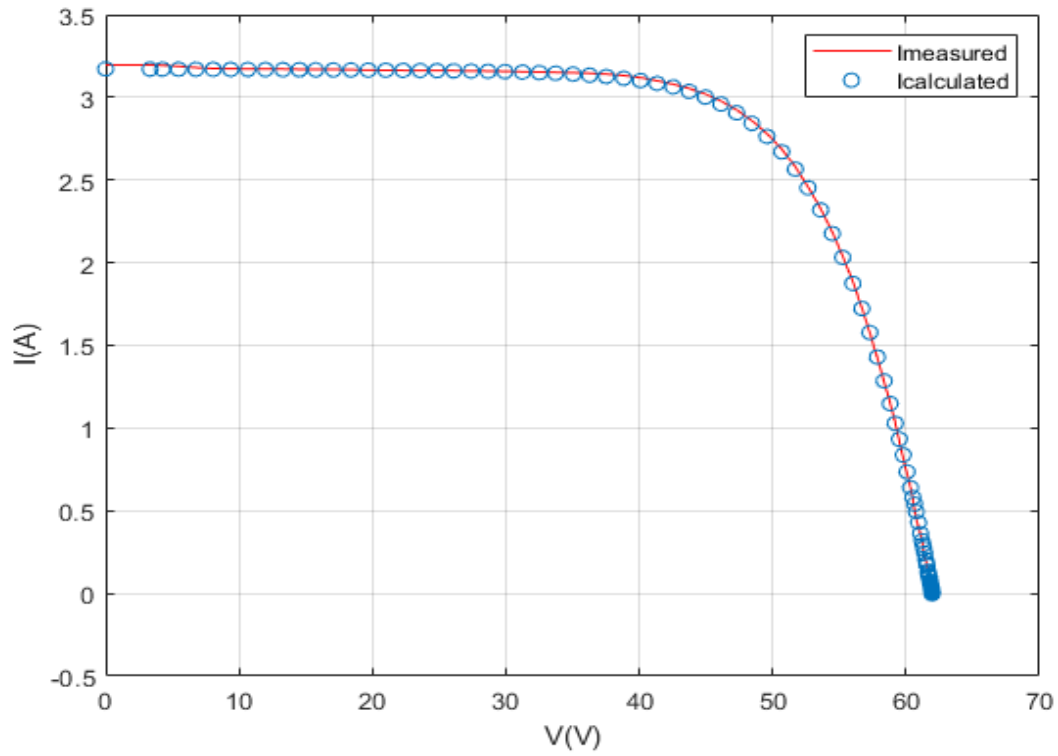


Fig 3.24: I-V curve showing both calculated (through EMA) and measured currents of Sanyo HIP-190B2-BO-01 PV module

PV module	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
Condor CEM150M	1.3058	0.2462	$3.089 \cdot 10^3$	8.3370	$1.8240 \cdot 10^{-6}$	0.0381
Kyocera KC125GHT	1.3741	0.2289	$2.290 \cdot 10^3$	8.0143	$8.0754 \cdot 10^{-6}$	0.0395
Siliken SLK60P6L 210Wp	1.3688	0.3655	$1.723 \cdot 10^3$	8.2957	$6.3057 \cdot 10^{-6}$	0.0474
Sanyo HIP-190B2-BO- 01	1.7512	0.8738	$3.095 \cdot 10^3$	3.1729	$7.4343 \cdot 10^{-6}$	0.0122

Table 3:extracted model parameters and RMSE using EMA

3.3.4. DE results:

a) Condor CEM150M

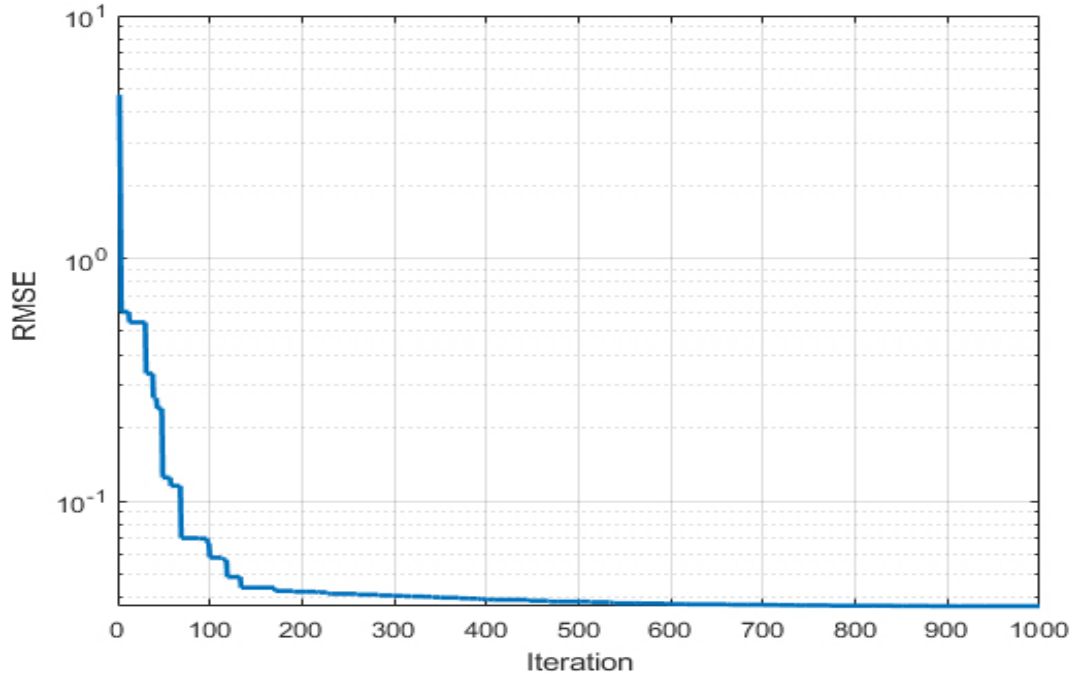


Fig 3.25: RMSE vs Iteration graph using DE on Condor CEM150M PV module data.

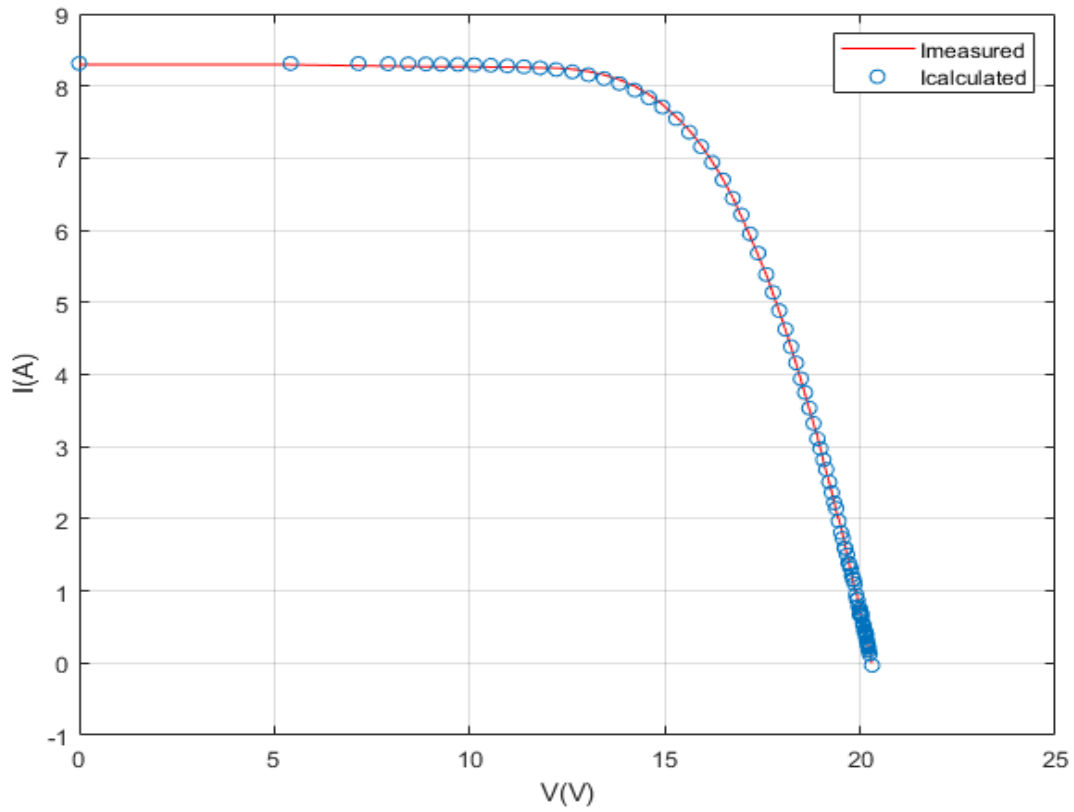


Fig 3.26: I-V curve showing both calculated (through DE) and measured currents of Condor CEM150M PV module

b) Kyocera KC125GHT

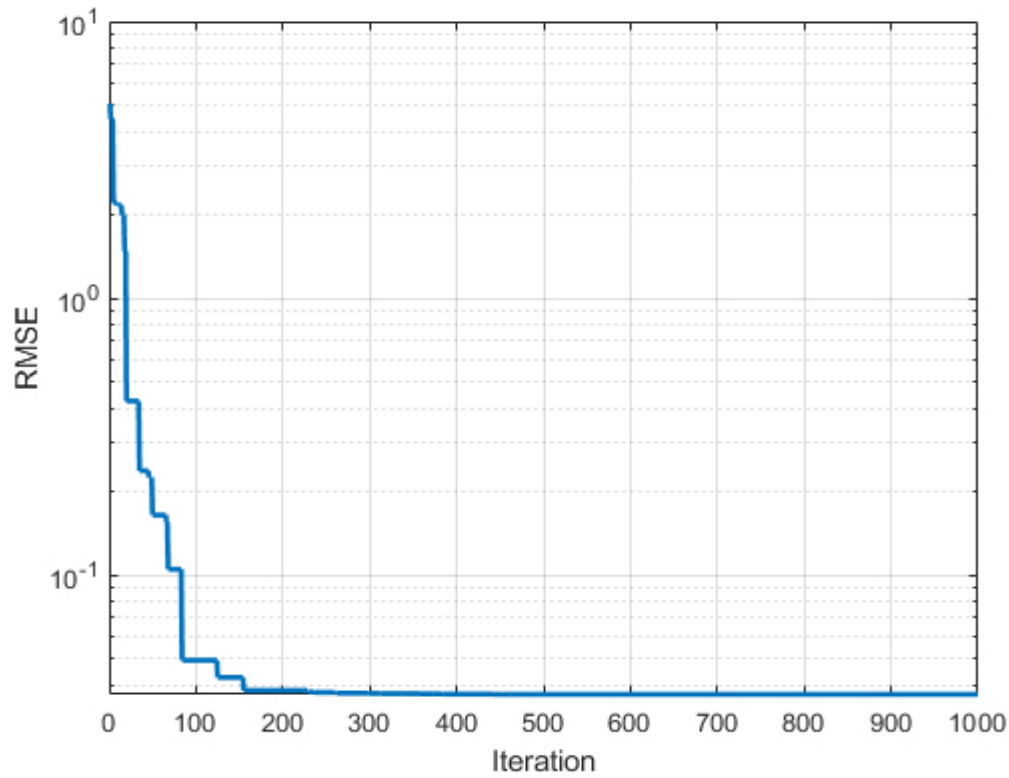


Fig 3.27: RMSE vs Iteration graph using DE on Kyocera KC125GHT PV module data.

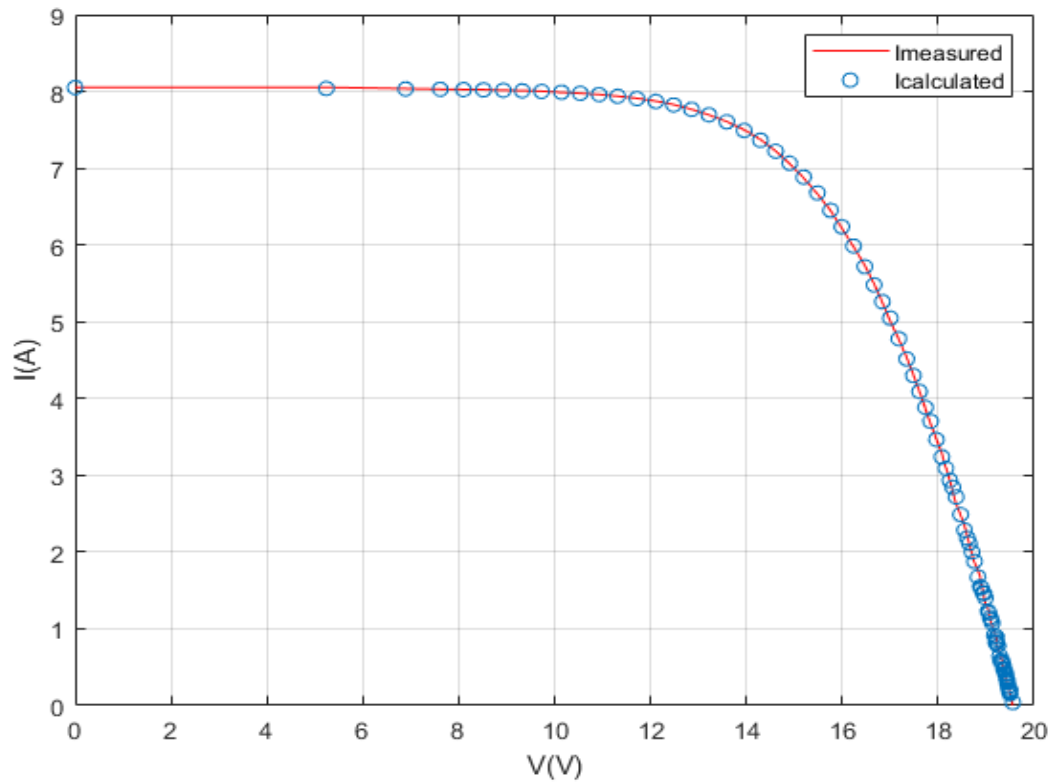


Fig 3.28: I-V curve showing both calculated (through DE) and measured currents of Kyocera KC125GHT PV module

c) Siliken SLK60P6L:

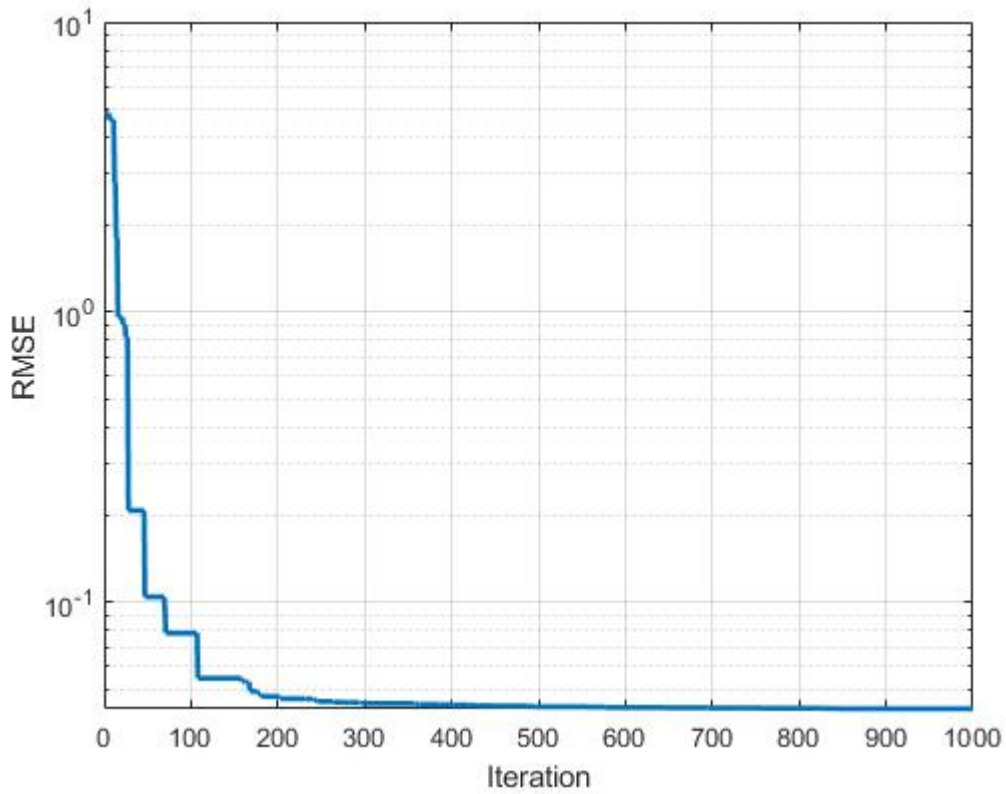


Fig 3.29: RMSE vs Iteration graph using DE on Siliken SLK60P6LPV module data.

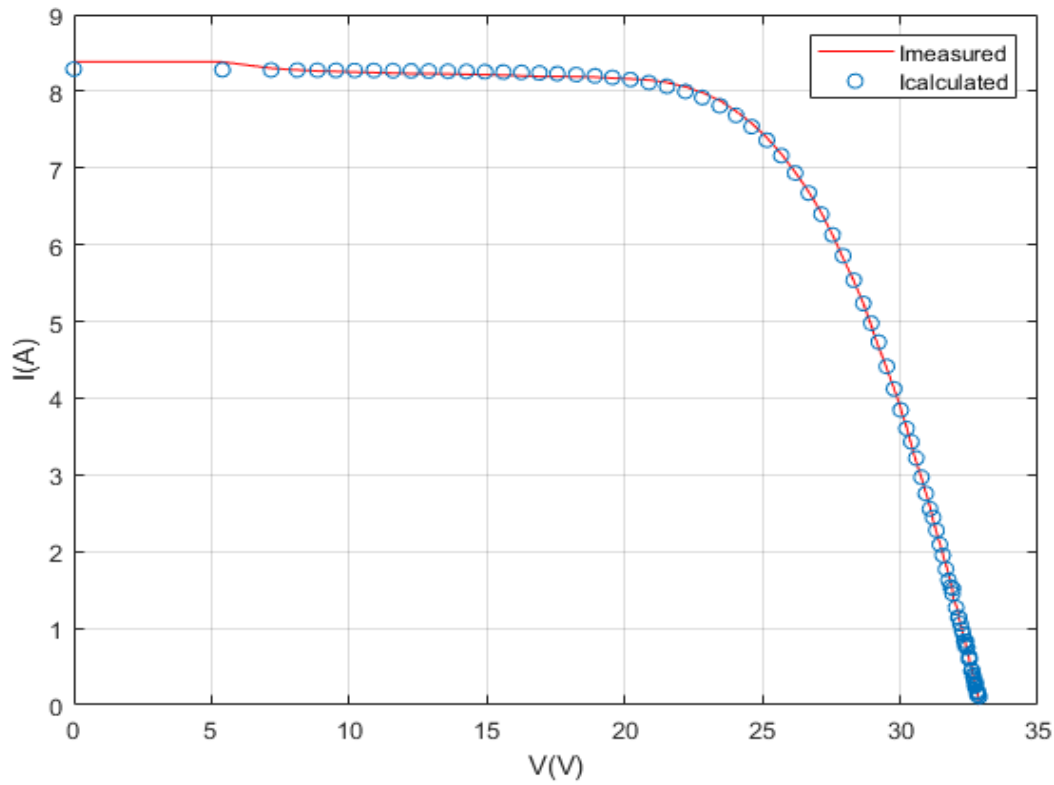


Fig 3.30: I-V curve showing both calculated (through DE) and measured currents of Siliken SLK60P6L PV module

d) Sanyo HIP-190B2-BO-01:

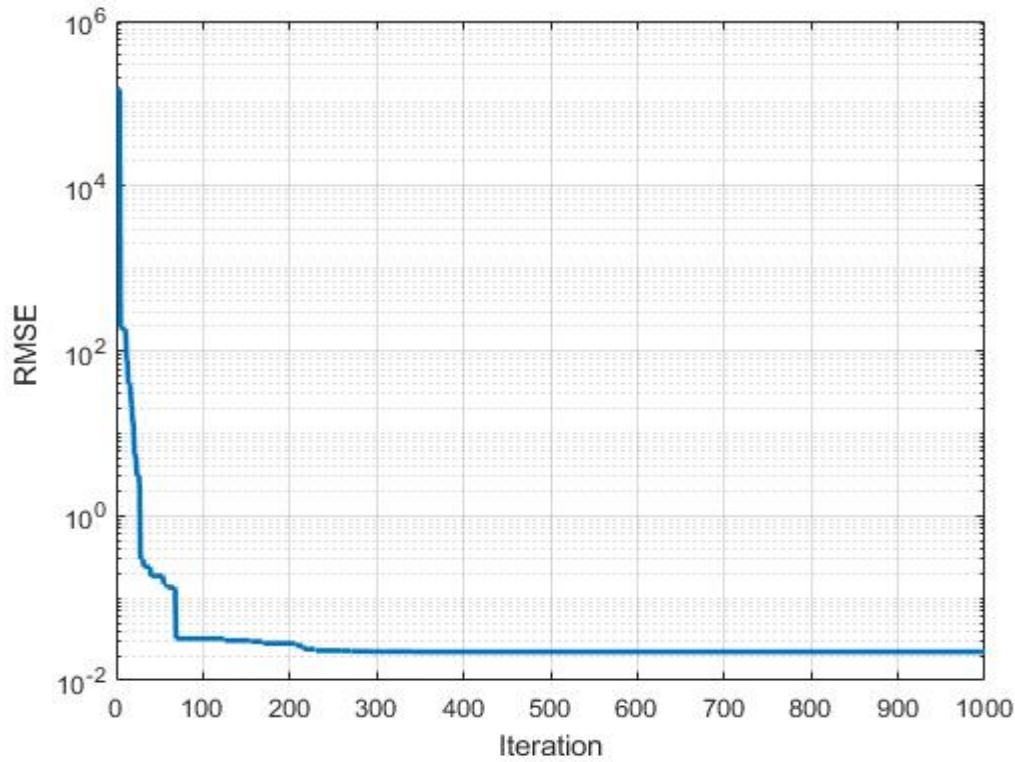


Fig 3.31: RMSE vs Iteration graph using DE on Sanyo HIP-190B2-BO-01 PV module data.

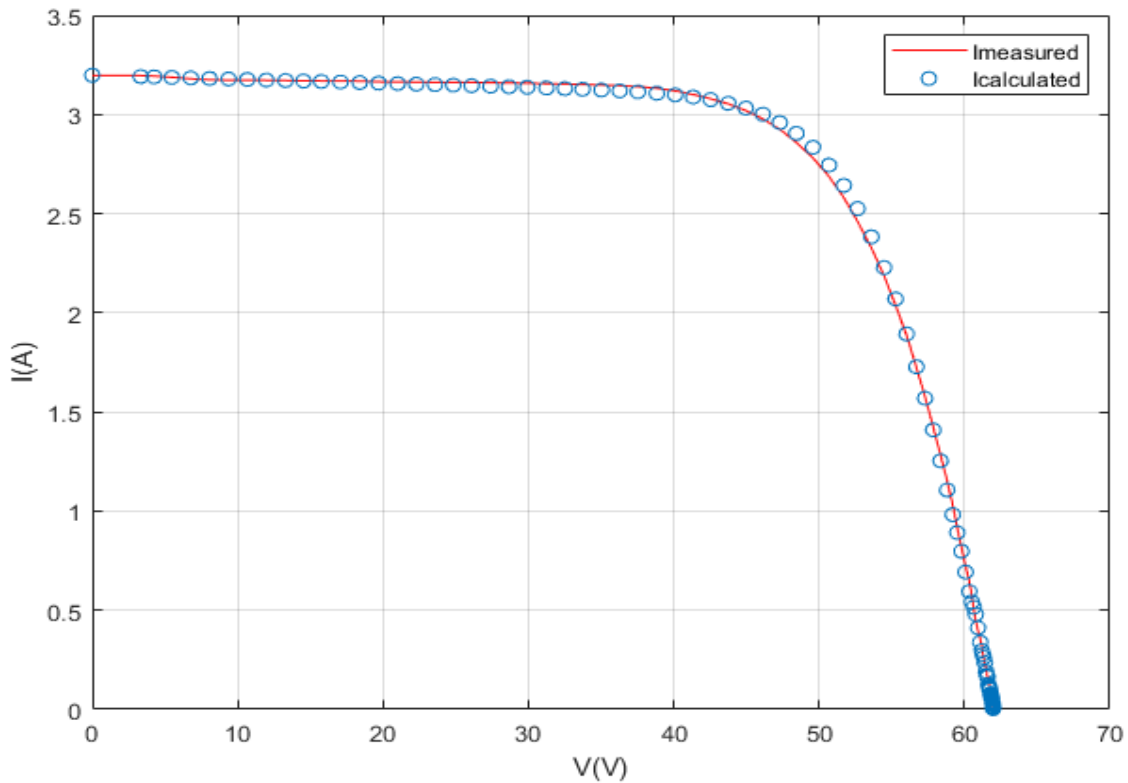


Fig 3.32: I-V curve showing both calculated (through DE) and measured currents of Sanyo HIP-190B2-BO-01 PV module

PV module	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
Condor CEM150M	1.2942	0.2477	$4.902 \cdot 10^3$	8.3140	$1.5837 \cdot 10^{-6}$	0.0205
Kyocera KC125GHT	1.3956	0.2269	$5.00 \cdot 10^2$	8.0534	$1.0000 \cdot 10^{-5}$	0.0070
Siliken SLK60P6L 210Wp	1.2672	0.3950	$5.924 \cdot 10^2$	8.2955	$2.0269 \cdot 10^{-6}$	0.0274
Sanyo HIP-190B2-BO- 01	2.0000	1.4566	$5.00 \cdot 10^2$	3.2069	$4.0000 \cdot 10^{-8}$	0.0151

Table 4: extracted model parameters and RMSE using DE

3.3.5. MPA Results:

a) Condor CEM150M:

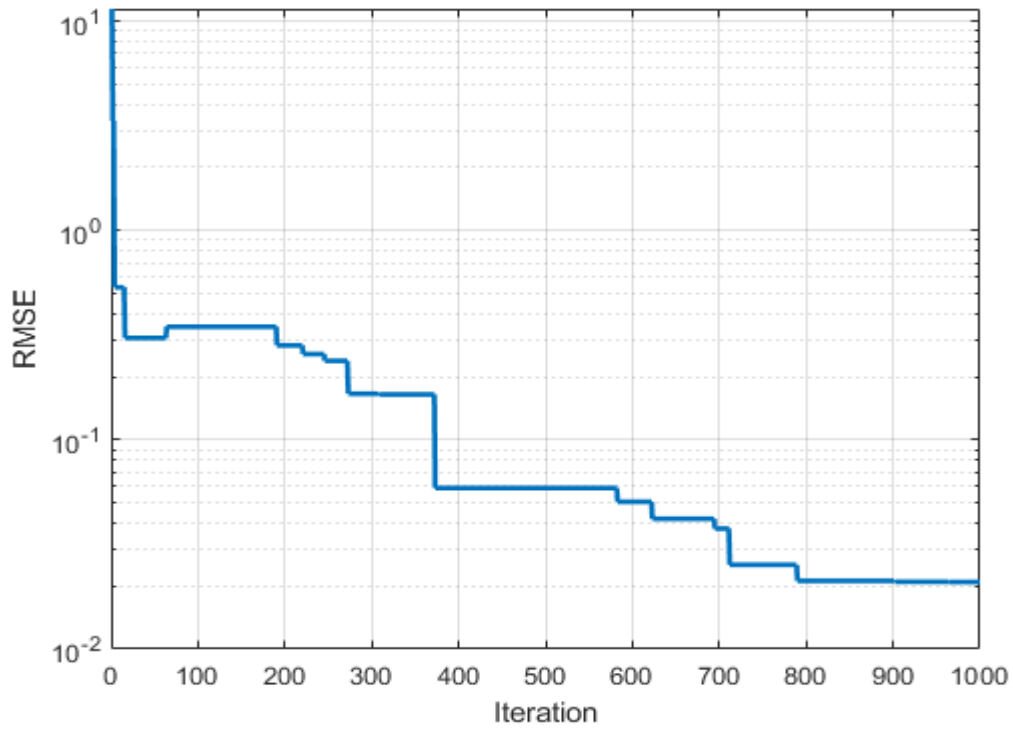


Fig 3.33: RMSE vs Iteration graph using MPA on Condor CEM150M PV module data.

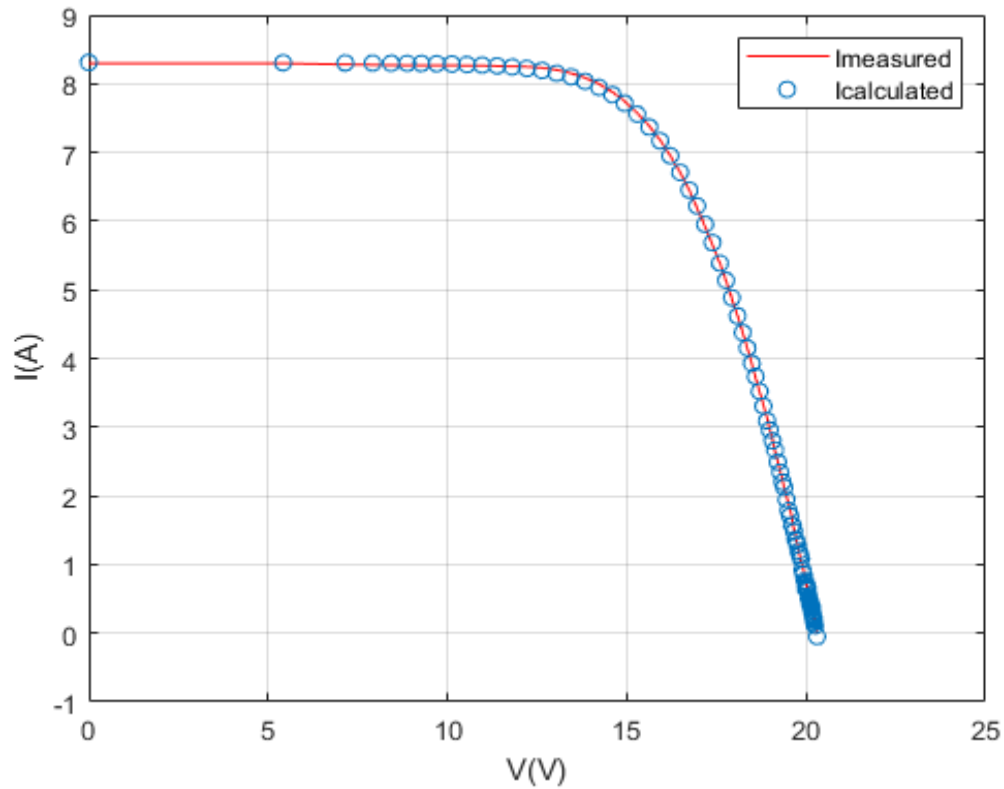


Fig 3.34: I-V curve showing both calculated (through MPA) and measured currents of Condor CEM150M PV module

b) Kyocera KC125GHT:

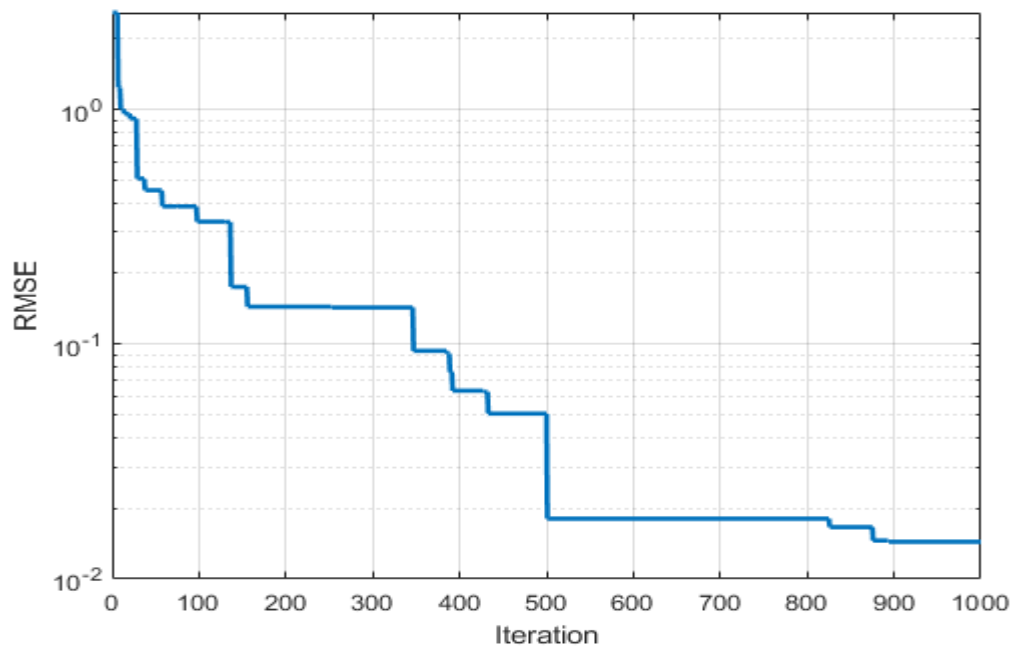


Fig 3.35: RMSE vs Iteration graph using MPA on Kyocera KC125GHT PV module data.

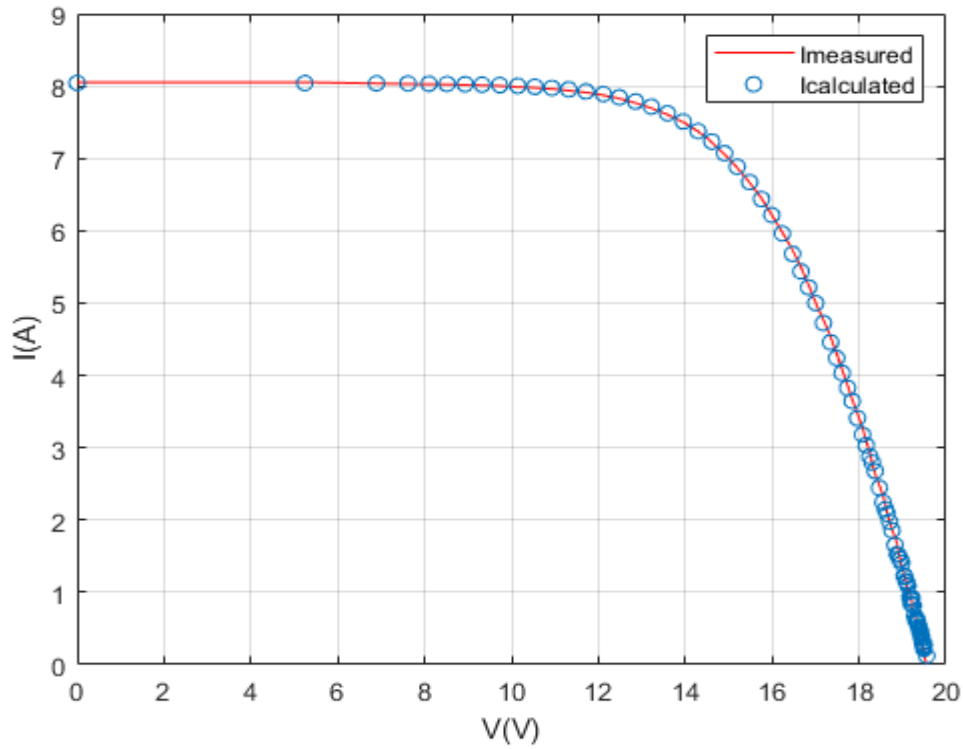


Fig 3.36: I-V curve showing both calculated (through MPA) and measured currents of Kyocera KC125GHT PV module

c) Siliken SLK60P6L:

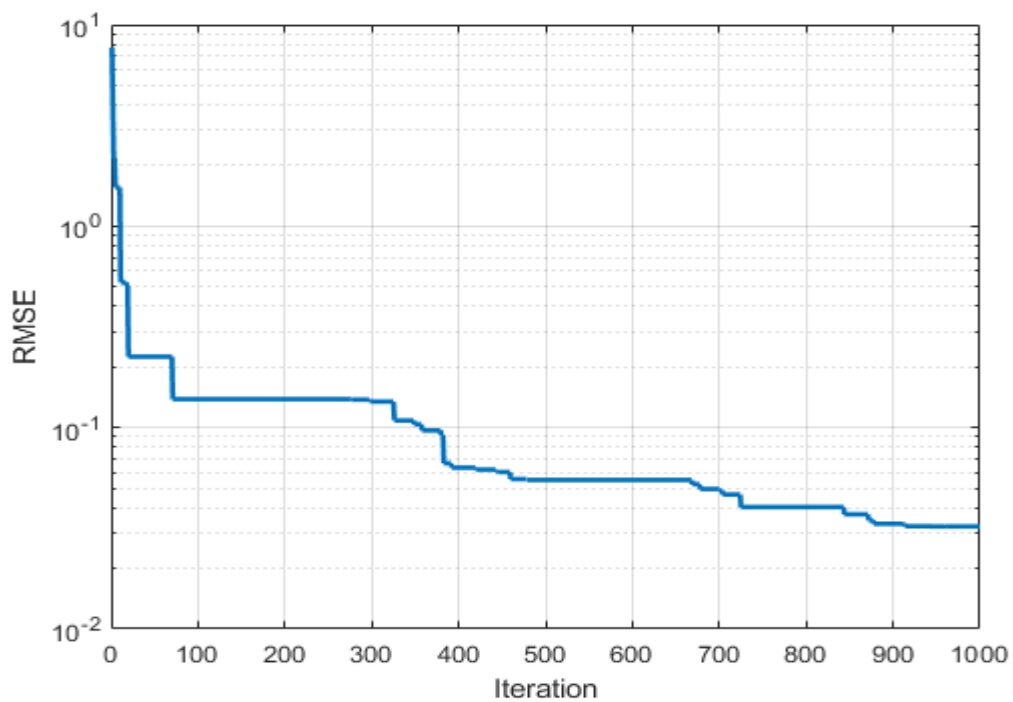


Fig 3.37: RMSE vs Iteration graph using MPA on Siliken SLK60P6LPV module data.

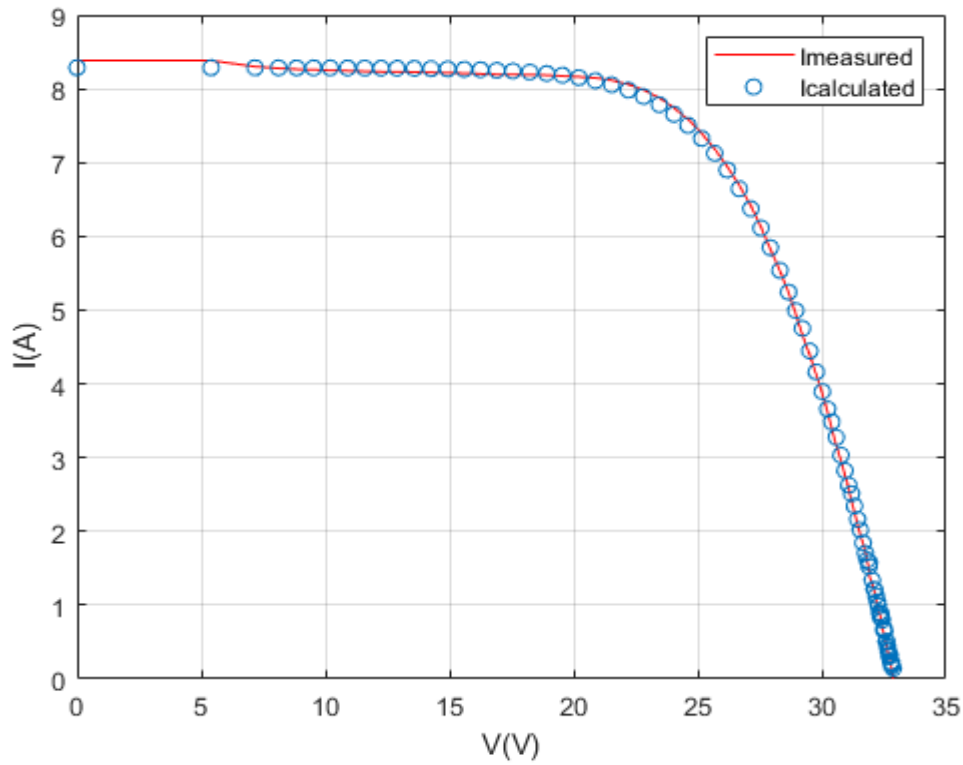


Fig 3.38: I-V curve showing both calculated (through MPA) and measured currents of Siliken SLK60P6LPV module.

d) Sanyo HIP-190B2-BO-01:

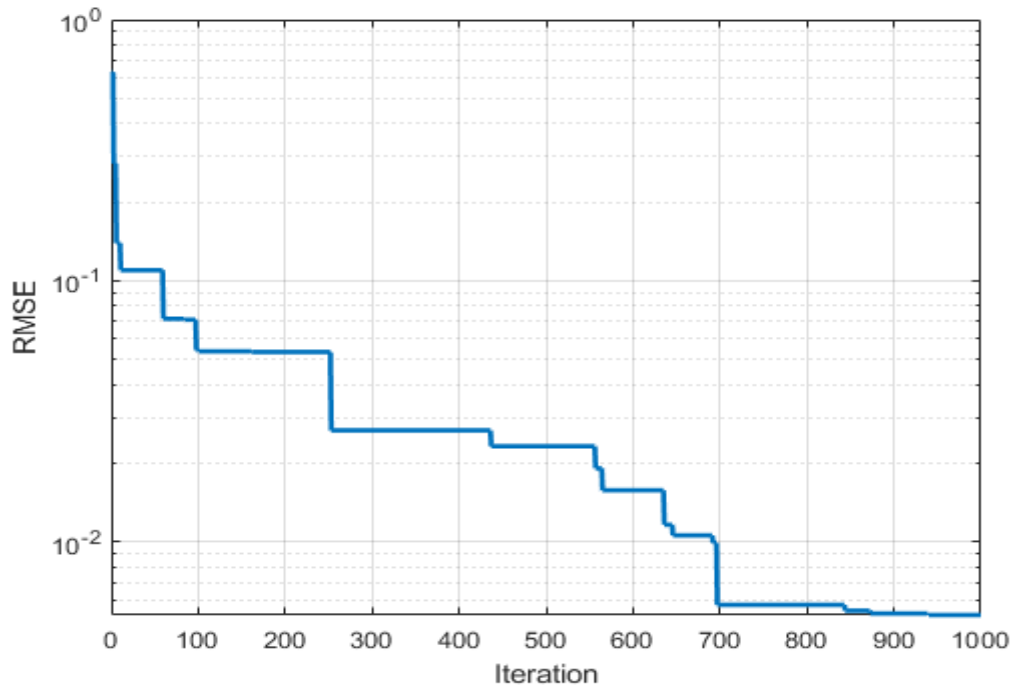


Fig 3.39: RMSE vs Iteration graph using MPA on Sanyo HIP-190B2-BO-01 PV module data.

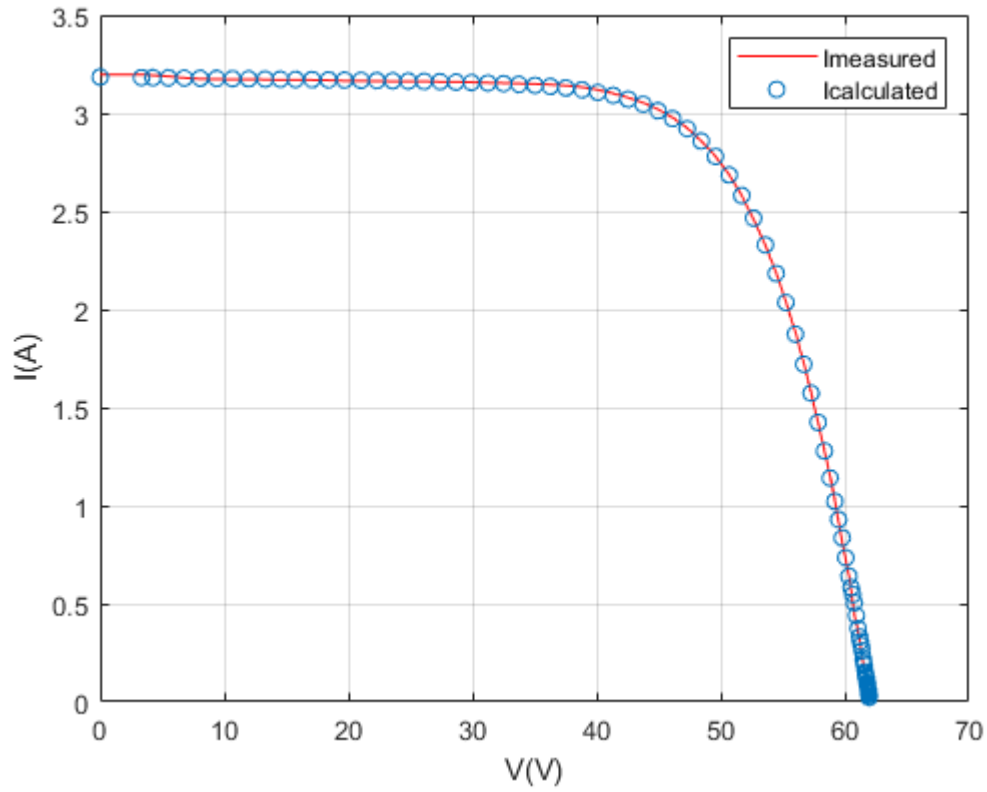


Fig 3.40: I-V curve showing both calculated (through MPA) and measured currents of Sanyo HIP-190B2-BO-01 PV module

PV module	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
Condor CEM150M	1.2644	0.2523	$1.189 \cdot 10^3$	8.3120	$1.1010 \cdot 10^{-6}$	0.0208
Kyocera KC125GHT	1.3535	0.2440	$4.108 \cdot 10^3$	8.0439	$6.5088 \cdot 10^{-6}$	0.0144
Siliken SLK60P6L 210Wp	1.3751	0.3637	$4.290 \cdot 10^2$	8.2873	$6.6865 \cdot 10^{-6}$	0.0322
Sanyo HIP-190B2-BO- 01	1.6237	1.0496	$1.290 \cdot 10^3$	3.1868	$2.6520 \cdot 10^{-6}$	0.0052

Table 5:extracted model parameters and RMSE using MPA

3.4. Results discussion:

The algorithms used have yielded very good RMSE values, which shows that the difference between the experimental and calculated data is very small; this means that the parameters obtained are accurate. Hence, the models are reliable and can be used to represent the modules accurately; by inspecting the obtained I-V curves, it can be seen that the data points calculated using the extracted parameters align almost perfectly with the experimental I-V curves, which confirms the accuracy of the models obtained.

The curves representing the RMSE value versus the number of iterations give an overview about the behaviour of the five algorithms in the process of finding optimal parameter values to achieve the best RMSE value possible. It can be said that the algorithms keep the same pattern while optimizing; for example PSO has a stair looking graph behaviour, it is like the RMSE value is driven downstairs (to lower values) but in relatively large treads. This causes PSO, generally, to take a greater number of iterations to find its best possible RMSE, compared to the other algorithms. Another observation is that almost all graphs show that the best possible value of RMSE is reached way before the determined number of maximum iterations; so there is no need to increment that number looking for better results.

3.4.1 Comparative tables:

a) Condor CEM150M:

Algorithm	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
PSO	1.4483	0.2216	$4.223 \cdot 10^3$	8.3594	$8.2937 \cdot 10^{-6}$	0.0472
WDO	1.3033	0.2460	$2.070 \cdot 10^3$	8.3183	$1.7658 \cdot 10^{-6}$	0.0369
EMA	1.3058	0.2462	$3.089 \cdot 10^3$	8.3370	$1.8240 \cdot 10^{-6}$	0.0381
DE	1.2942	0.2477	$4.902 \cdot 10^3$	8.3140	$1.5837 \cdot 10^{-6}$	0.0205*
MPA	1.2644	0.2523	$1.189 \cdot 10^3$	8.3120	$1.1010 \cdot 10^{-6}$	0.0208

Table 6: Comparative table of the parameters extracted and RMSE values of the five algorithm for Condor CEM150M

a) Kyocera KC125GHT:

Algorithm	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
PSO	1.3578	0.2335	$4.471 \cdot 10^2$	8.0433	$6.8308 \cdot 10^{-6}$	0.0390
WDO	1.3479	0.2376	$1.065 \cdot 10^3$	8.0657	$6.1658 \cdot 10^{-6}$	0.0406

CHAPTER III: PV Parameters extraction

EMA	1.3741	0.2289	2.290×10^3	8.0143	8.0754×10^{-6}	0.0395
DE	1.3956	0.2269	5.00×10^2	8.0534	1.0000×10^{-5}	0.0070*
MPA	1.3535	0.2440	4.108×10^3	8.0439	6.5088×10^{-6}	0.0144

Table 7: Comparative table of the parameters extracted and RMSE values of the five algorithm for Kyocera KC125GHT

b) Siliken SLK60P6L:

Algorithm	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
PSO	1.2818	0.3912	9.356×10^2	8.2890	2.4152×10^{-6}	0.0429
WDO	1.3301	0.3759	3.397×10^3	8.3035	4.1984×10^{-6}	0.0446
EMA	1.3688	0.3655	1.723×10^3	8.2957	6.3057×10^{-6}	0.0474
DE	1.2672	0.3950	5.924×10^2	8.2955	2.0269×10^{-6}	0.0274*
MPA	1.3751	0.3637	4.290×10^2	8.2873	6.6865×10^{-6}	0.0322

Table 8: Comparative table of the parameters extracted and RMSE values of the five algorithm for Siliken SLK60P6L

c) Sanyo HIP-190B2-BO-01:

Algorithm	a	$R_s(\Omega)$	$R_p(\Omega)$	$I_{ph}(A)$	$I_0(A)$	RMSE
PSO	1.6765	0.9788	2.397×10^3	3.1822	4.1867×10^{-6}	0.0095
WDO	1.7343	0.8780	2.252×10^3	3.1901	6.5997×10^{-6}	0.0122
EMA	1.7512	0.8738	3.095×10^3	3.1729	7.4343×10^{-6}	0.0122
DE	2.0000	1.4566	5.00×10^2	3.2069	4.0000×10^{-8}	0.0151
MPA	1.6237	1.0496	1.290×10^3	3.1868	2.6520×10^{-6}	0.0052*

Table 9: Comparative table of the parameters extracted and RMSE values of five the algorithm for Sanyo HIP-190B2-BO-01

*: best obtained RMSE value for the module.

From the comparative tables above, DE is clearly on top in terms of RMSE values; it has three best RMSE values out of four. It has not even reached a value above 0.274; this shows the effectiveness of this optimization method for the purpose of single-diode model PV parameters extraction. MPA has also given three second-best results and one best, which shows that this newly developed algorithm is very effective. However, the statement does not discredit the other optimization techniques used.

CHAPTER III: PV Parameters extraction

Another observation is the narrow gap in RMSE values overall, but especially between WDO and EMA; the gap is very narrow between the two values on all tables.

The widely varying parameter, between all algorithms results, is the parallel resistance R_p ; this is mainly due to its wider range.

The less varying parameter between the five algorithms results is the photo-generated current I_{ph} ; this is logical because I_{ph} must be very close to the short-circuit current of the module.

3.5. Conclusion:

This chapter has covered our approach on single-diode parameters extraction; the approach is based on using metaheuristic techniques on the problem formulated as an optimization one. Different algorithms have been used, their behaviours were observed some yielding more accurate results, but overall all the results obtained have good fitness and are satisfactory.

General conclusion:

Solar energy is labelled as the main energy for the future; this labelling has led the photovoltaic literature to grow fast, over the years. In this project, interest was oriented towards one particular aspect; that is the modelling of photovoltaics. That has lead us to gather a great understanding of PVs, in our way to realise this project. Going from broad areas of research to narrow ones kept the work very interesting; for seeing the bigger picture always helps in understanding more details put together.

Our work started by explaining the very basic principle of PVs, which is essential before going into more in-depth aspects. After that, different equivalent model were presented; the ideal PV cell representation, single-diode model, and double-diode model. Following that, the parameters that represent these models (ideality factor, series resistance, parallel resistance, photo-generated current, and diode saturation current were introduced. Then, light was shed on optimization techniques, but mainly the techniques of interest for us; that are known as metaheuristics. Those optimization algorithms used were defined in depth; since they are the main tool for solving our problems later on. Coming to the main section, which was adapting those algorithms to suite the main purpose of the project, which is single-diode PV model parameters extraction, the problem was formulated as an optimization one with the RMSE as an objective function. Five algorithms were used (PSO, WDO, EMA, DE, MPA), and the results obtained were accurate as it was graphically shown. In the end, some comparisons were carried out, finding out that DE method has given best overall objective function values; without discrediting the accuracy of the other techniques.

The topic of this project is very important and interesting, because obtaining models with high accuracy results in better design and assessment of photovoltaics, which will lead to big advents in the field.

References:

- [1]- David Thorpe, Solar Technology: The Earth scan Expert Guide to Using Solar Energy for Heating, Cooling and Electricity, Routledge, 1 edition, October 12, 2011

- [2]- Biswas PP, Suganthan PN, Wu G, Amaratunga GAJ, Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm, Renewable Energy (2018), doi: 10.1016/j.renene.2018.07.152 an adaptive differential evolution algorithm

- [3]- www.e-education.psu.edu/eme812/node/608, Types of PV technology and recent innovations, accessed on June 7, 2020

- [4] S. Madi, A. Kheldoun, Bond graph based modeling for parameter identification of photovoltaic module, Energy, 141, pp.1456-1465, 2017.

- [5]- Experiment 15 temperature dependence of the saturation current of a junction diode, Solid state physics laboratory, California Polytechnic state university.

- [6]- S Djeriou, A Kheldoun, A Mellit, Efficiency improvement in induction motor-driven solar water pumping system using golden section search algorithm, Arabian Journal for Science and Engineering, 43, pp. 3199–3211, 2018.

- [7]- www.pveducation.org/pvcdrom/solar-cell-operation, Series resistance. Accessed on June 7, 2020

- [8] [.wikipedia.org/wiki/Optimization_problem](https://wikipedia.org/wiki/Optimization_problem), accessed on June 10, 2020

- [9] - Deb Kalyanmoy, Optimization for Engineering Design: Algorithms and Examples, edition 2, PHI Learning Pvt. Ltd., 2012

- [10]- Daniel Ashlock, Evolutionary Computation for Modeling and Optimization, Springer-Verlag New York, 2006, DOI 10.1007/0-387-31909-3

- [11]- Xin-She Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley & Sons, 2010, DOI:10.1002/9780470640425

References

- [12]- S. L. Ho, and Shiyong Yang, and Yanan Bai, .A Wind Driven Optimization Algorithm for Global Optimization of Electromagnetic Devices, IEEE, IEEE Transactions on Magnetics , Volume: 54 , Issue: 3 , March 2018, DOI: 10.1109/TMAG.2017.2750543
- [13]- Exchange market algorithm Naser Ghorbani, Ebrahim Babaei, Applied Soft Computing 19 (2014) 177–187.
- [14]- Comparative study on parameter extraction of photovoltaic models via differential evolution, Xi Yanga, Wenying Gong, Ling Wang, Energy Conversion and Management 201 (2019) 112113
- [15]Al-qaness, M.A.A.; Ewees, A.A.; Fan, H.; Abualigah, L.; Abd Elaziz, M. Marine Predators Algorithm for Forecasting Confirmed Cases of COVID-19 in Italy, USA, Iran and Korea. *Int. J. Environ. Res. Public Health* 2020, 17, 3520.
- [16]-D.S.H. Chan ; J.C.H. Phang, Analytical methods for the extraction of solar-cell single- and double-diode model parameters from I-V characteristics ,IEEE, IEEE Transactions on Electron Devices, Volume: 34, Issue: 2 , Feb 1987 .
- [17]- T. EASWARAKHANTHAN , J. BOTTIN , I. BOUHOUC & C. BOUTRIT (1986),Nonlinear Minimization Algorithm for Determining the Solar Cell Parameters with Microcomputers, International Journal of Solar Energy, 4:1, 1-12, DOI: 10.1080/01425918608909835