DEVELOPMENT AND ANALYSIS OF AN IMPROVED PV-ARRAY MODEL WITH SHADING EFFECTS

BY

NDUBISI NWAJIAKU, B.Eng (UNIBEN) 2006

M.Sc /ENG/36240/12-13

ndubisinwajiaku@gmail.com

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES AHMADU BELLO UNIVERSITY, ZARIA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc) DEGREE IN CONTROL ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY,
ZARIA, NIGERIA

JANUARY, 2017

DECLARATION

I, NWAJIAKU Ndubisi, hereby declare that this dissertation titled "**DEVELOPMENT AND ANALYSIS OF AN IMPROVED PV-ARRAY MODEL WITH SHADING EFFECTS**" was carried out by me under the supervision of Professor. M. B. Mu'azu and Dr. T. H. Sikiru as part of the requirements for the award of degree of Master of Science in Control Engineering. To the best of my knowledge, this research work has never been submitted anywhere for the award of any certificate. All literatures used and cited in this report, are duly acknowledged in the reference pages.

_____	_____	_____
NWAJIAKU Ndubisi	**Signature**	**Date**

CERTIFICATION

This dissertation titled **"DEVELOPMENT AND ANALYSIS OF AN IMPROVED PV-ARRAY MODEL WITH SHADING EFFECTS"** meets the requirements for the award of degree of Master of Science (M.Sc) in Control Engineering by Ahmadu Bello University, Zaria and it is approved for its contribution to knowledge and literature.

(Chairman, Supervisory Committee) Prof. M. B. Mu'azu	_____ Signature	_____ Date
(Member, Supervisory Committee) Dr. T. H Sikiru	_____ Signature	_____ Date
(Head of Department) Dr. Y. Jibril	_____ Signature	_____ Date
(Dean School of Postgraduate Studies) Prof. K. Bala	_____ Signature	_____ Date

DEDICATION

This dissertation is dedicated to God for bestowing me with health, knowledge and patience to complete this work.

ACKNOWLEDGEMENT

My M.Sc has been a multifaceted and enriching journey. I am happy to see this journey finished. This is my chance to recognize those who contributed to make it possible.

My heartfelt-gratitude goes to my supervisors Prof. M.B. Mu'azu and Dr. T.H Sikiru, for giving me generously of their time, directing and encouraging me. Their kind supervision throughout the research period, useful counsel and constructive criticism helped me improve my dissertation report; an endless source of help. Working with them was one of the most rewarding experiences of these few years.

Special thanks to my colleagues and especially my fellow M.Sc students in Electrical and Computer Engineering Department, A.B.U Zaria for the lively environment and camaraderie.

I would like to express my appreciation to Prof. S.A Jonah, Mr. Edmund Harold and Engr. Rocco Bechila for providing me an opportunity to set important benchmark for my professional career.

Thereafter, am thankful to Abdulrashid Muhammad, Valentine Ikpo, Salawudeen Tijani, Abdulmumini Yesufu, Busayo Adebiyi, Umar Abubakar, Austine Ejegwa, John Linus, Abdul Umoru who always helped me whenever I ask anything regarding my research.

The contribution of all my teachers in enhancing my knowledge is well appreciated, for without you my life would have been meaningless.

And finally, to my family, I would like to articulate my deepest gratitude for the unceasing confidence and encouragement which I have received from them throughout my life.

The errors and inconsistencies contained in this work remain my own.

NWAJIAKU Ndubisi
January, 2017

ABSTRACT

The performance of photovoltaic (PV) module is dependent on shading conditions of the cells. This research work is aimed at developing an improved PV-model considering the various degree of shading on PV-cells connected in a PV-module. This is based on modelling the different degrees of shading of cells in a module into the general model by introducing degree of shading parameter. A MATLAB-based graphical user interface (GUI) is developed using Artificial Fish Swarm Algorithm (AFSA). This is used to accurately predict current-voltage (i-v) curves, power-voltage (p-v) curves, maximum power point values, short-circuit current and open-circuit voltage. A set of simulation scenarios involving uniform and non-uniform shading across PV-modules with/without bypass diodes were carried out in order to demonstrate the effectiveness of the developed models and their accuracy. The developed AFSA based model was compared with the conventional dynamic programming (DP) based model, using the output parameters of the PV-system presented in the work of El-Saady *et al.* (2013) to evaluate the performance of the proposed approach using three performance metrics (open circuit voltage, short-circuit current and maximum power point). Results from AFSA based model showed 59.7303W, 3.8A and 21.0897V for maximum power point, short-circuit current and open circuit voltage respectively. This showed that AFSA based model produced increase in accuracy improvement of 5.1%, 0.03% and 0.22% over DP based model. Even though the developed AFSA based approach possessed some degree of truncation error, it has been shown to outperform the conventional DP based approach.

TABLE OF CONTENTS

DECLARATION **II**

CERTIFICATION **III**

DEDICATION **IV**

ACKNOWLEDGEMENT **V**

ABSTRACT **VI**

LIST OF FIGURES **XI**

LIST OF TABLES **XIII**

LIST OF ABBREVIATION **XIV**

APPENDIX **X**

CHAPTER ONE : INTRODUCTION **1**

1.1 BACKGROUND 1

1.2 MOTIVATION 5

1.3 STAMENT OF PROBLEM 5

1.4 AIMS AND OBJECTIVES 6

1.5 DISSERTATION ORGANIZATION 6

CHAPTER TWO : LITERATURE REVIEW **8**

2.1 INTRODUCTION 8

2.2 REVIEW OF FUNDAMENTAL CONCEPTS 8

2.2.1 PV Cell Operation 8

2.2.2 PV Cell Model 11

2.2.3 Diodes and their Functions 14

2.2.4 Effects of Bypass Diodes on Photovoltaic Cells 14

2.2.5 Maximum Repetitive Reverse Voltage (V_{RRM}) 17

2.2.6 Maximum Power Point Current (I_{mpp}) and Voltage (V_{mpp}) 17

2.2.7 Ideality and Fill Factor 17

2.2.8 Solar Cell Efficiency 18

2.2.9 Resistance 18

2.2.10 Maximum PowerPoint (MPP) 19

2.3 REVIEW OF SIMILAR WORKS 26

CHAPTER THREE : MATERIALS AND METHODS 34

3.1 INTRODUCTION 34

3.2 METHODOLOGY 34

3.3 PROPOSED MATHEMATICAL MODEL OF PV-CELLS WITH SHADING EFFECTS 35

3.4 PROPOSED AFSA BASED APPROACH FOR I-V CURVES 39

3.5 PROPOSED AFSA BASED APPROACH FOR P-V CURVES 46

3.6 PROPOSED AFSA APPROACH FOR MAXIMUM POWER POINT 47

3.7 PROPOSED DP BASED APPROACH FOR PV –ARRAY 50

3.8 THE PROPOSED MODEL FOR BYPASS DIODE IN PV-MODULES 52

3.9 1NFLECTION IN I-V CURVE 52

3.10 SHADING EFFECT ON PV-SYSTEM WITH BYPASS DIODES 53

3.11 PROPOSED APPROACH FOR P-V CURVES WITH BYPASS DIODE EFFECT 56

3.12 THE DEVELOPED PV- ARRAY ANALYSIS SIMULATOR 59

3.13 PROPOSED PV- ARRAY ANALYSIS ALGORITHM 60

3.14 SIMULATION 63

CHAPTER FOUR : RESULT AND DISCUSSION 64

4.1 INTRODUCTION 64

4.2 PROPOSED MODEL TESTING RESULTS 64

4.2.1 I-V Curves with Varying Degree of Shading 64

4.2.2 P-V Curves with Varying Degree of Shading 65

4.2.3 Output Characteristics of PV-Array in Different Configuration 66

4.2.4 Effect of Bypass-Diodes on the Output Characteristics of PV-Array 71

4.2.5 Comparison Case A 76

4.2.6 Comparison Case B 77

CHAPTER FIVE : CONCLUSION AND RECOMMENDATION 79

5.1 CONCLUSION 79

5.2 SIGNIFICANT CONTRIBUTIONS 79

5.3 RECOMMENDATION 80

5.4 LIMITATION 81

REFERENCES 82

APPENDIX

A.1 MAIN FUNCTION M-FILE: AFSA 86

A.2 SUB-FUNCTION M-FILE: CHASE 89

A.3 SUB-FUNCTION M-FILE: DISTANCE 90

A.4 SUB-FUNCTION M-FILE: MOVE_FISH 90

A.5 SUB-FUNCTION M-FILE: NEW_FISH 90

A.6 SUB-FUNCTION M-FILE: OBJF 90

A.7 SUB-FUNCTION M-FILE: PARAMETERS 91

A.8 SUB-FUNCTION M-FILE: PREY 91

A.9 SUB-FUNCTION M-FILE: SWARM 92

B.1 SUB-FUNCTION M-FILE: VOC 92

B.2 SUB-FUNCTION M-FILE: P 93

B.3 SUB-FUNCTION M-FILE: ISC 93

C.1 SUB-FUNCTION M-FILE: PV_PLOTTER 93

C.2 SUB-FUNCTION M-FILE: IV_PLOTTER 94

D.1 SUB-FUNCTION M-FILE: OBJF_V 95

D.2 SUB-FUNCTION M-FILE: OBJF_P 96

D.3 SUB-FUNCTION M-FILE: OBJF_I 97

E.1 SUB-FUNCTION M-FILE: PV_OUTPUT_XTICS 98

E.2 SUB-FUNCTION M-FILE: MAXPOWERPOINT 98

LIST OF FIGURES

Figure 1.1 A Simple PV System (Thongam & Jaiswal, 2015) 1

Figure 1.2 PV-Cells in a PV-Module Under Different Degrees of Shading 4

Figure 2.1 Basic PV-Cell Structure (Cubas *et al*., 2014) 9

Figure 2.2 Group of Cells Packed into Modules, Panels and Arrays to Provide a Specific Power Output (Huld et al., 2010) 10

Figure 2.3 Equivalent Circuit Model 11

Figure 2.4 Typical Current –Voltage Curves (Shading of 10 series and 4 Parallel Modules) (Platzer, 2012) 15

Figure 2.5 Typical Power –Voltage Curves (Shading of 10 series and 4 Parallel Modules) (Platzer, 2012) 16

Figure 2.6 Typical Graph of Current vs. Voltage and Power vs. Voltage (Salam et al., 2013) 19

Figure 2.7 Flowchart of Adaptive Control Algorithm (Arikatla &Abu-Qahouq, 2010) 20

Figure 2.8 Relationship between the Duty cycle and Power (Thongam & Jaiswal, 2010) 21

Figure 2.9 Position of MPP in terms of $\frac{dI_{PV}}{dV_{pv}}$ (Isaloo & Amiri, 2016) 22

Figure 2.10 MPP as function of ΔP and ΔV (Christopher & Ramesh, 2013) 23

Figure 2.11 Steps of Fuzzy Control (Hawkes & Leach, 2009) 24

Figure 3.1 Typical PV-Module with 18 PV-Cells (with 6 Parallel Paths) 37

Figure 3.2 Variation of Output Current with Changes in Output Voltage 41

Figure 3.3 I-V Curve Showing the Proposed MPP Search Approach 48

Figure 3.4 I-V Curve for Identical and Non-Identical PV-Array Configuration 52

Figure 3.5 Determination of MPPs on a PV-System with Bypass Diodes 58

Figure 3.6 Screenshot of the Developed GUI Window 59

Figure 3.7 Flowchart for the Proposed AFSA Based PV-Array Analysis Algorithm 62

Figure 4.1 I-V Curves of a Single PV-Module with Varying Degrees of Shading 65

Figure 4.2 Curves of a Single PV-Module with Varying Degrees of Shading 66

Figure 4.3 I-V Curve for Configuration Case 1 68

Figure 4.4 P-V Curve for Configuration Case 1 68

Figure 4.5 I-V Curve for Configuration Case 2 68

Figure 4.6 P-V Curve for Configuration Case 2 68

Figure 4.7 I-V Curve for Configuration Case 3 69

Figure 4.8 P-V Curve for Configuration Case 3 69

Figure 4.9 I-V Curve for Configuration Case 4 69

Figure 4.10 P-V Curve for Configuration Case 4 69

Figure 4.11 AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 1) 69

Figure 4.12 AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 2) 70

Figure 4.13 AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 3) 70

Figure 4.14 AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 4) 70

Figure 4.15 I-V Curve for Case 1 72

Figure 4.16 P-V Curve for Case 1 72

Figure 4.17 I-V Curve for Case 2 73

Figure 4.18 P-V Curve for Case 2 74

Figure 4.19 I-V Curve for Case 3 75

Figure 4.20 P-V Curve for Case 3 75

LIST OF TABLES

Table 3.1: The Choice of Module and AFSA Parameters 63

Table 4.1: Results for the Selected PV-Array Configuration (Cases 1 to 8) 67

Table 4.2: Output Characteristics of the Three Cases Scenario of PV-Array with Bypass Diodes 76

Table 4.3: Comparison of the KC200GT Solar Array Data Sheet Information and the Results of the Developed AFSA based Model 77

Table 4.4: Comparison of the BP SolarMSX-60 Solar Array Data Sheet Information and the Results of the Developed AFSA and DP based Models 78

LIST OF ABBREVATION

Acronyms	Definition
PV	Photovoltaic
AFSA	Artificial Fish Swarm Algorithm
GUI	Graphics User Interphase
MATLAB	Matrix Laboratory
Mppt	Maximum Power Point
Impp	Maximum Power Point current
Vmpp	Maximum Power Point Voltage
Voc	Open Circuit Voltage
Isc	Short circuit Current
DP	Dynamic Programming
Pmax	Maximum Power output

CHAPTERONE

INTRODUCTION

1.1 Background

The evolution in technology and the awareness of global warming have drawn attention towards the use of solar photovoltaic cells as an alternative source of electric power (Diaf *et al.*, 2011). A Photovoltaic (PV) system generates electricity by the direct conversion of solar energy (Theocharis *et al.*, 2012). Photovoltaic systems provide electricity for home appliances, villages, water pumping, desalination and many other applications (Said *et al.*, 2012). Photovoltaic output power depends on many factors, such as sun position, weather conditions, module temperature, thermal characteristics, module material composition, mounting structure, and shading (Platzer, 2012)

A simple solar PV system is shown in Figure 1.1

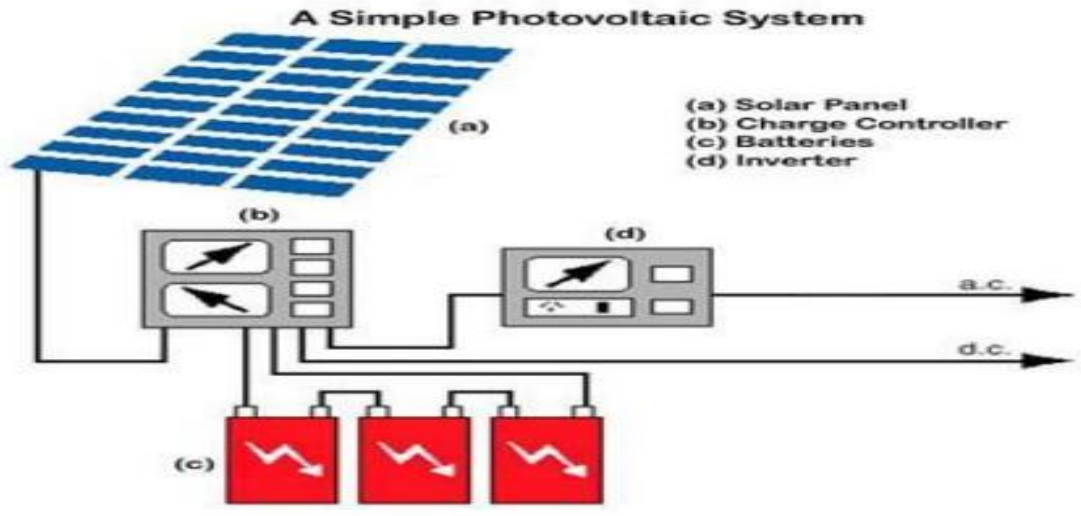


Figure 1.1: A Simple PV System (Thongam & Jaiswal, 2015)

Some of the advantages of solar photovoltaic are (Jayakumar, 2009):

1. Solar PV provides clean-green energy and available almost everywhere there is sunlight.
2. It is appropriate for smart energy networks with Distributed Power Generation-DPG, as next generation power network structure.
3. Solar PV cost is currently on a fast reducing track. Consequently Solar PV has indeed a highly promising future both for economic variability and environmental sustainability
4. Solar PV is totally silent, producing no noise at all, which makes it a perfect solution for urban areas and for residential applications.
5. Solar PV has no mechanically moving parts, except in cases of sun-tracking mechanical bases. Consequently it has far less breakages or requires less maintenance than other renewable energy systems.
6. Solar PV, through photoelectric phenomenon, produces direct electricity.

Despite the advantages of Solar PV, it also has some limitations which include (Jayakumar, 2009);

1. Fragility, which means can be damaged easily.
2. Intermittency and unpredictability of solar energy due to weather, makes solar panels to shaded.
3. Multiple components.

Shading is an effect that is produced when sun light meant to directly hit the PV-cell is obstructed by an opaque object (Parida *et al.*, 2013). Shading obstructions can be classified as soft or hard sources. Soft sources such as a tree leave, a roof vent, a chimney or other item which shades from a distance, are those in which the shadowing is diffused or dispersed (Kelly & Gibson, 2009).These soft sources significantly reduce the amount of light reaching the cells of a module. Hard sources are defined as those that stop light from

reaching the cells, such as a blanket, tree branch, etc sitting directly on top of the glass (Briggs, 2012). Some shading types can cover an entire single cell of a module, some can cover an entire module of the array, and some create various shading pattern on either strings, module, or cells (Morais *et al.*, 2010). Measuring the extent of the shade on a solar array may be challenging due to the fact that shadows move as the sun position moves throughout the day and year. Shading of photovoltaic systems can cause high loss in performance (Lagorse *et al.*, 2010). There are different types of shading, which include:

1. **Direct Shading:** In this case, a shadowing object is placed close to modules and shades the PV array constantly. The closer the shading object is to the array, the darker the shade is and the less diffuse light reaches the module surface, the more problematic the situation gets (Jiang *et al.*, 2011).
2. **Temporal Shading:** This is caused by natural conditions, like snow, leaves, soiling, etc. This effect is especially important in ground mounted systems in rural areas, where the PV arrays come much easier in contact with dust. In snowy regions, horizontal arrangement of the module is recommendable in order to minimize losses (Haeberlin, 2007).
3. **Self-Shading:** A bad system design might cause shading on modules due to other modules that have been placed ahead. As a rule of thumb, the modules located in a row must be separated from each other approximately 4-6 times the height of the tilted module and under this condition, mutual shading is avoided (Firth *et al.*, 2010).
4. **Whole and Partial Shading:** Whole shading and partial shading refer to cell shading ratio $\left(\frac{shadedarea}{ce.llarea}\right)$.Whole shading and partial shading reference to the threshold point below which there is insufficient irradiance to generate cell voltage (Whole Shade) and

above which there is sufficient irradiance to generate cell current (Partial Shade).Whole or partial shading of a cell will lead to different effects on module voltage or current or both (Farret & Simoes, 2006).

Figure 1.2 shows typical PV cells in a PV module under different shading conditions

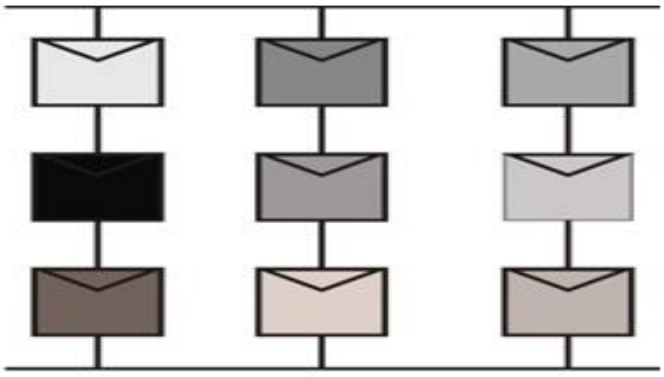


Figure 1.2: PV–Cells in a PV-Module under Different Degrees of Shading

Effect of shading as a significant research issue in optimizing PV output power has been established (Alsayid *et al.*, 2013a; Kelly & Gibson, 2009; Seyedmahmoudian *et al.*, 2013).Some researchers have considered shading of certain group of cells or modules in a PV-array with uniform shading pattern (Khalaf *et al.*, 2014; Ramana & Jena, 2015; Sikiru, 2007). This research proposes the development and analysis of improved PV-array with shading effects which include introduction of degree of shading parameter and use of Artificial Fish Swarm Algorithm (AFSA) for determining the output characteristics, because it has presented capability to avoid local optimum in order to achieve global optimization. The focus of developing the improved PV-model is to help in analyzing the effect of varying degrees of shading across in the individual cells in a PV-module.

1.2 Motivation

Shading conditions affect the performance of Photovoltaic cells. PV-cells must be operated at their maximum power point, for optimal utilization. However, maximum power point is not steady rather it varies with shading degrees, and other degrading effects. Existing models' current and voltage compromises that all the PV-cells exhibit homogeneity for same insolation conditions. In practice, these assumptions are not absolutely correct because of non-uniformity caused by environmental conditions. Therefore, it is almost unachievable to have uniform insolation across the PV-cells. Having these variations of the characteristics of the PV-arrays due to degree of shading, there is demand to effectively determine the Maximum Power Point using high efficient convergent intelligent technique.

1.3 Statement of Problem

Photo-voltaic cell is an indispensable part in the conversion of sun's power into electricity. It provides useful applications in enabling humanity to make use of sunlight in a clean and highly versatile way. Photovoltaic modeling is a challenge due to shading. Most of the PV-models currently available in the open literature are well adequate for uniform shading pattern of PV-modules. However, the accuracy of the output characteristics of the existing models are often been affected by the effect of the degree of shading across individual cell. In order to overcome this shortcoming, an improved PV-array model that incorporates a parameter that can handle the varying degree of shading is proposed in this research as a means of addressing the limitations that are inherent in the existing models.

1.4 Aim and Objectives

The aim of the research is to develop and analyze an improved PV-array model, while considering the effects of varying degrees of shading across the individual cells of a PV-module. In achieving the aim, the following are the set of objectives:

1. To incorporate a shading parameter into the conventional PV-array model that will account for varying degree of shading across individual cells of a given PV-module;
2. To develop Artificial Fish Swarm Algorithm (AFSA) based model for determining the output characteristics (I–V curve; P–V curve; and Maximum Power Point) of PV-cells connected in various configurations with and without bypass diode;
3. To develop Matlab-based AFSA and DP graphical user interface (GUI) models in order to carry out a comparative analysis using the following performance metrics for the validation of the developed model: open-circuit voltage, short-circuit current and maximum power point.

1.5 Dissertation Organization

The general introduction has been presented in Chapter One. The rest of the chapters are structured as follows: First, detail review of related literature and relevant fundamental concepts about PV-Cell Operation, PV-Array, PV-Cell Model, Maximum Power Point, Maximum Power Point Algorithms, Artificial Fish Swarm Algorithm (AFSA) are carried out in Chapter Two. Second, mathematical model for the proposed PV-array output characteristics coupled with that of the improved AFSA algorithm, the steps for the development of the proposed AFSA based I – V curve, P – V curve, and MPP evaluation Algorithm are presented in chapter Three. Third, the analysis and discussion of the result

are shown in Chapter Four. Finally, conclusion and recommendations of further work makes up the Chapter Five. The list of cited references and MATLAB codes in the appendices are provided at the end of this dissertation.

CHAPTER TWO

LITERATUREREVIEW

2.1 Introduction

This chapter will introduce the fundamental concepts related to PV-model and finally discuss the related works done by researchers.

2.2 Review of Fundamental Concepts

In this chapter, concepts fundamental to the research work in addition to similar other works are reviewed.

2.2.1 PV Cell Operation

The photoelectric effect is the ability to generate a power across a material under illumination, and is the core principle on which all PV cells operate (Bayrak & Cebec, 2013). Although this effect was fully explained by Einstein in 1921 (Rui *et al.*, 2010), it was not until 1954 that the first PV cell for the visible spectrum was constructed by Chapin, Fuller, and Pearson at Bell Labs (Gabriele *et al.*, 2009). PV cell is very similar to that of the classical diode with a PN junction (Salah & Ouali, 2011). When the junction absorbs light, the energy of absorbed photons is transferred to the electron–proton system of the material, creating charge carriers that are separated at the junction. The charge carriers are electron–ion pairs for liquid electrolyte or electron–hole pairs for solid semiconducting materials (Nabulsi & Dhaouadi, 2012). The charge carriers in the junction region create a potential gradient, get accelerated under the electric field and circulate as current through an external circuit (Rodriguez *et al.*, 2012). The square of the current multiplied by the resistance of the circuit is the power converted into electricity. The remaining power of the photon elevates

the temperature of the cell and dissipates into the surroundings (Vongkoon & Liutanakul, 2012).

2.2.1.1 PV cells

PV cell is basically a semiconductor *p-n* junction-based photodiode (Brito *et al.*, 2011). This semiconductor photodiode generates electrical power when exposed to light (wang *et al.*, 2012). PV-cell is a core building block of larger PV-arrays ranging from a few watts to hundreds of MW. PV cells can be made up of various semiconductor materials, such as amorphous and crystalline materials. But mono-crystalline silicon and poly-crystalline silicon are the most common types known commercially as can be seen in Figure 2.1 (Cubas *et al.*, 2014).

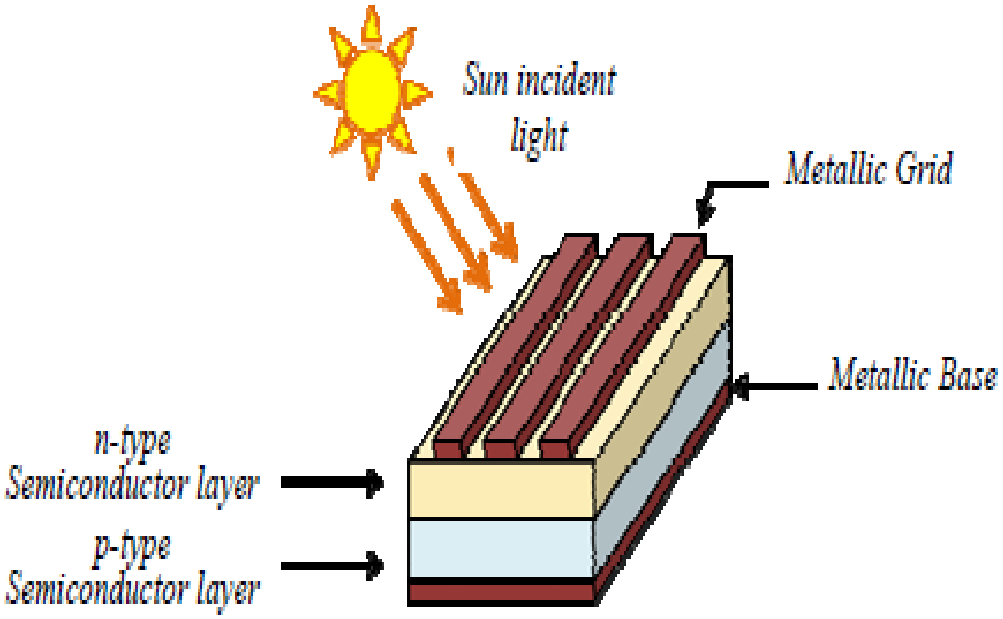


Figure 2.1: Basic PV Cell Structure (Cubas *et al.*, 2014)

2.2.1.2 PV module

Generally, a combined series and parallel connection of PV cells is known as a module (Gabriele *et al.*, 2009). Mostly, commercial modules consist of 36 or 72 cells (Basri *et al.*, 2010). The modules consist of transparent front side, encapsulated PV cells, and back side. The front side material is usually made up of low-iron and tempered glass. The efficiency of a PV module is less than a PV cell due to the fact that some solar irradiation is reflected by the glass cover and frame shadowing (Keyrouz *et al.*, 2013).

2.2.1.3 PV array

The PV cell is the basic unit of a solar panel, but it is intended to produce a small power of a few watts. The need for large power causes the need for collection of identical cells to form a module or a solar panel, in order to increase the power generated (Dio *et al.*, 2009). As shown in Figure 2.2, a PV array (system) is an interconnection of solar panels which in turn is made up of several modules that it is consisted of many PV cells connected in series and parallel (Khalaf *et al.*, 2014). The power produced by a single module is seldom enough for commercial use, so modules are connected to form a solar panel to supply the load.

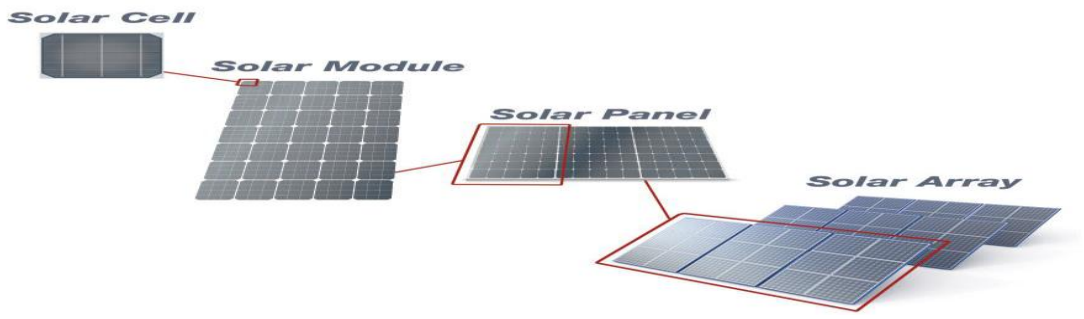


Figure 2.2: Group of Cell Packed into Modules, Panels and Arrays to Provide a Specific Power Output (Huld *et al.*, 2010)

2.2.2 PV Cell Model

The purpose of modelling the PV cell is to predict the behavior and power output of the PV system under various operating conditions, and to evaluate the effects of partial shadowing, module mismatch, and cell or module failure, on a PV system's output p-v and I - V characteristics (Macabebe & Dyk, 2008). The modelling of a PV cell involves the estimation of the I - V and P - V characteristics curves to emulate the real cell under various operating conditions. The most popular approach is to utilize the electrical equivalent circuit, which is primarily based on the diode (lshaque *et al.*, 2011).

The electrical performance of a photovoltaic cell can be approximated by the equivalent circuit shown in Figure 2.3

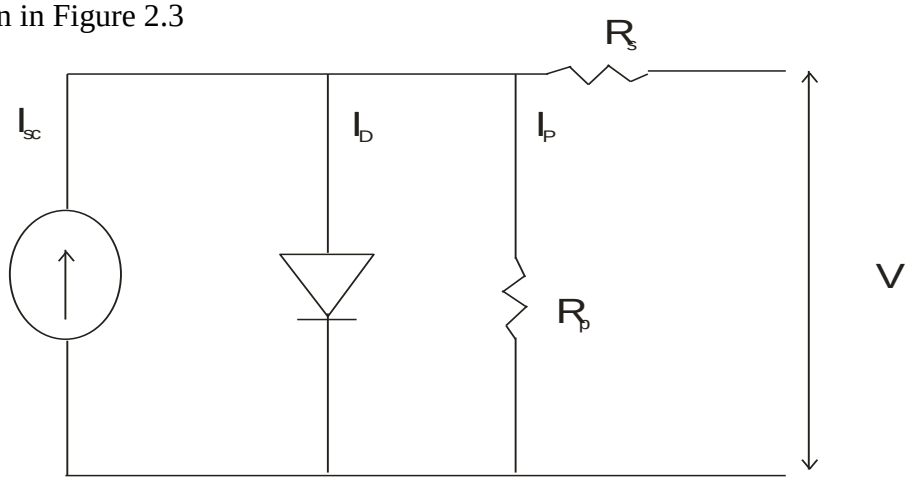


Figure 2.3: Equivalent Circuit Model

I_{SC} short circuit current (photonic current) is the PV-cell current source. It is dependent on solar irradiance. There is a diode shunt connected across the current source representing the diffusion current through the p-n junction. Internal series and parallel resistances are represented by R_S and R_P respectively. The former is due to transmission of electric current produced by the solar cell involves ohmic losses (Durischa *et al.*, 2007). It is seen that series resistance affects the cell operation mainly by reducing the fill-factor (Diaf *et al.*,

2011). The current I_D through the diode can be expressed by Shockley's diode equation (Singh & Tripathi, 2012).

According to the equivalent circuit model shown in Figure 2.3, the equations for the circuit operation can be shown as below (Adelmo *et al.*, 2014):

$$I_D = I_S \left[\exp q \left(\frac{V + IR_S}{NKT} \right) - 1 \right] \tag{2.1}$$

where I_s is the reverse saturation current, V is the output voltage, q is the charge of one electron, T is the solar cell temperature in Kelvin, and *K* is the Boltzmann constant, and N is the junction perfection factor or ideality factor, which determines the diode deviation from the ideal p-n junction . I_{SC} denotes the photocurrent which is dependent upon the light spectrum and the spectral response of the solar cell. The dependence of the photocurrent on the irradiance and cell temperature can be described by the following empirical equation (Faranda *et al.*, 2008).

$$I_{SC} = \left[I_{SCR} + Ki \;\; T - Ti \;\right] \frac{G}{100} \tag{2.2}$$

Where I_{SCR} is the short-circuit current generated at Ti which is the reference temperature in Kelvin, the factor Ki is the temperature coefficient of the short-circuit current and *G* is the irradiance in mW/cm^2

Reverse saturation current $I_{S,}$ is related to temperature. Higher temperature increases the concentration of the intrinsic charge carriers and consequently results in higher carrier

recombination. Therefore rising temperature increases the reverse saturation current (Faranda *et al.*, 2008):

$$I_S = I_O \left(\frac{T}{T_i} \right)^3 \exp\left[\frac{qE_g}{NK} \left\{ \frac{1}{T_i} - \frac{1}{T} \right\} \right] \tag{2.3}$$

Where E_g is bandgap, nominal cell temperature is given as T is defined as (Faranda *et al.*, 2008).

$$T = T_a + 0.2XG \tag{2.4}$$

Taking into account the internal resistances R_P and R_S ,the cell current is then expressed as (Mäki *et al.*, 2011).

$$I = I_{SC} - I_S \left(\exp\left[q \frac{V + IR_S}{NKT} \right] - 1 \right) - \frac{V + IR_S}{R_P} \tag{2.5}$$

2.2.2.1 Series-connected cells

Solar cells are connected in series so that voltage across each cell can be accumulated at the output. The cells in series carry the same current irrespective of the fact whether one or more PV cells under shade produce less photonic current (Chicco *et al.*, 2009). As the shading increases in a cell, its output voltage starts to fall.

2.2.2.2 Parallel-connected cells

In order to increase the currents, solar cells are connected in parallel. In parallel connection of cells, the currents across each cell is summed up while the voltage remains constant across them (Joseph & Kamala, 2013).

2.2.3 Diodes and their Functions

A diode is a specialized electronic component with two electrodes called the anode and the cathode. The fundamental property of a diode is its tendency to conduct electric current in only one direction (King *et al.*, 2007), three types of diodes common with PV systems are as follows :

Blocking diode: It is placed in series with a module or string of modules to prevent reverse current flow and protect PV-modules. It conducts current during normal operation. It can be used to prevent the discharge of batteries at night in standalone systems (Lei *et al.*, 2011).

Isolation diode: The isolation diode is another device most commonly found in PV systems. This diode is installed between two series strings to isolate a good series string from one that has failed or become shaded. Because bypass diodes are required in today's modules, the use of external isolation diodes is redundant and seldom practiced (Ngan & Tan, 2011).

Bypass diode: This is a diode connected across one or more solar cells in a photovoltaic module to protect these cells from thermal destruction in case of whole or partial shading of individual cells while other cells are exposed to full light.

2.2.4 Effects of Bypass Diodes on Photovoltaic Cells

Bypass diodes are a standard addition to any crystalline PV module. Modern solar modules come with small components inside them called "bypass diodes" (Ishaque *et al.*, 2011). These diodes allow shaded cells to be bypassed, allowing the current from other modules in the string to continue flowing. The bypass diodes' function is to eliminate the hot-spot phenomena which can damage PV cells and even cause fire if the light hitting the surface of the PV cells in a module is not uniform. The bypass diodes are usually placed on sub-

strings of the PV module. The effect of bypass diodes reduces the output power reduction, and also introduces non-linearity in the I–V and P-V characteristic (Platzer, 2012).

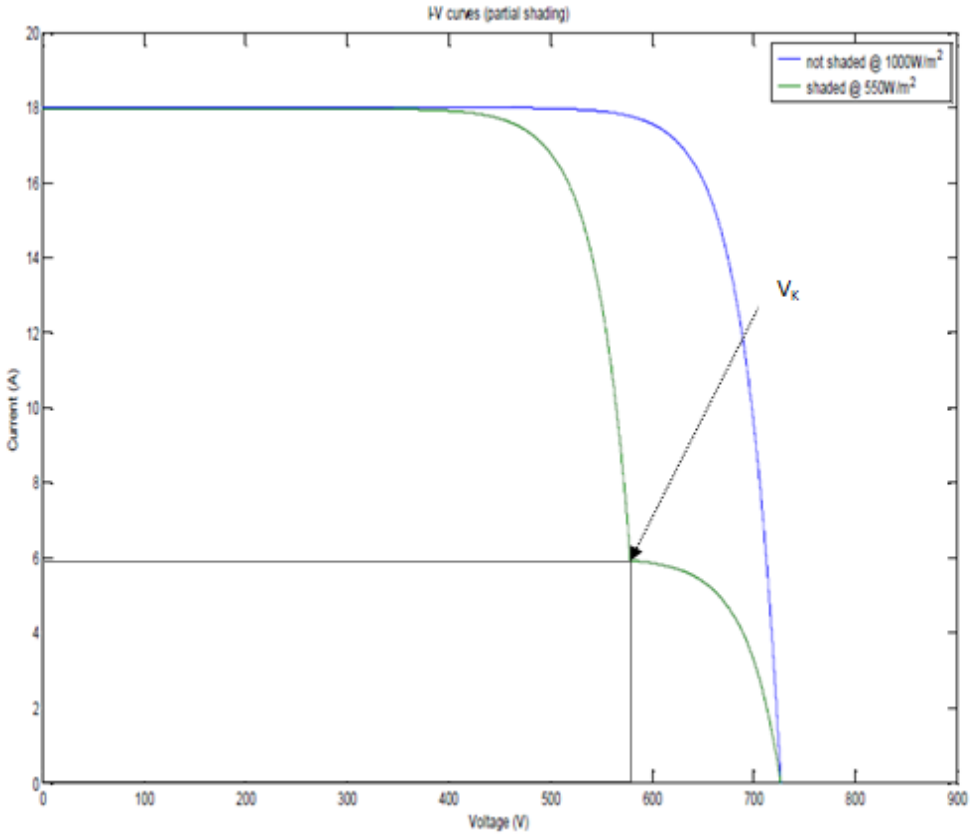


Fig 2.4: Typical Current –Voltage Curves (Shading of 10 Series and 2 Parallel Modules) (Platzer, 2012)

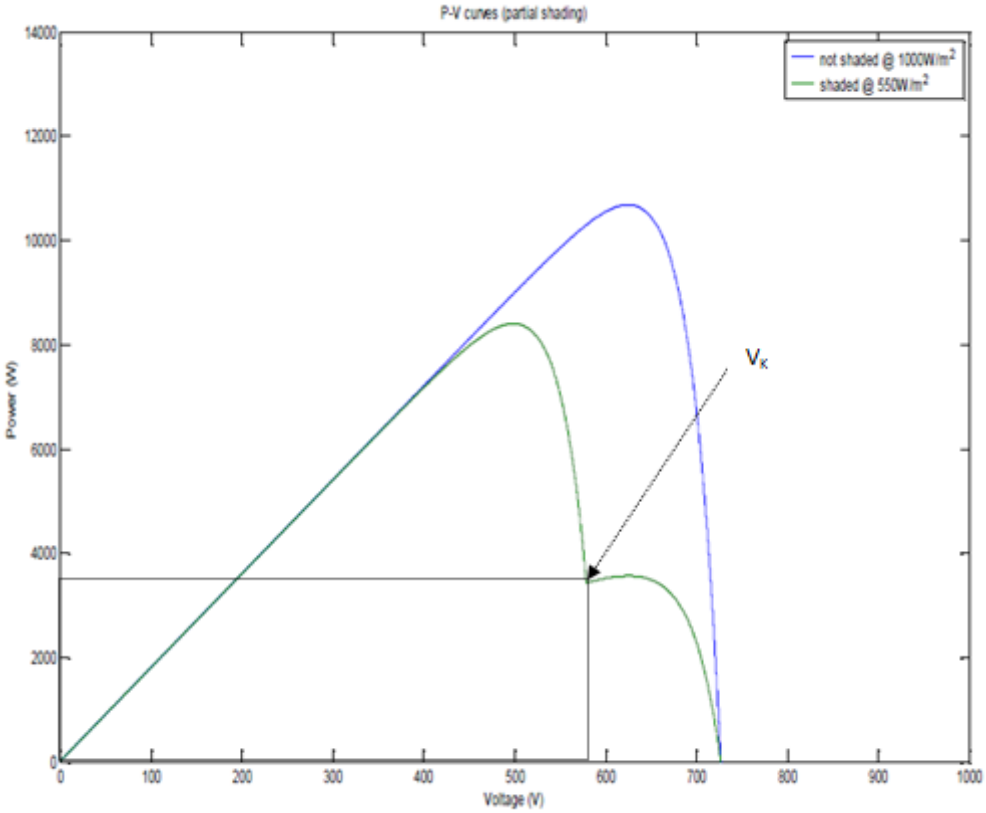


Fig 2.5:Typical Power – Voltage Curves (Shading of 10 Series and 2 Parallel Modules) (Platzer, 2012)

From the graphs presented, it is observed that the I-V curves have multiple steps, while the P-V curves have multiple peaks. This deviation from the ideal (unshaded) I-V and P-V curves accounts for the presence of by-pass diodes in the modules. Furthermore it can be observed that the P-V curve of Figure 2.5 has two maxima power points in contrast to the usual single maximum power point that exists in P-V characteristic curves of unshaded modules. Point V_k in both curves is a point where the string current is higher than the short circuit current of a particular cell and it is called the **Inflection Point**. It defines the current and voltage conditions in which the bypass diode becomes active; therefore the voltage distance from V_{OC} to the inflexed is related to number of bypass modules.

2.2.5 Maximum Repetitive Reverse Voltage (V_{RRM})

The maximum repetitive reverse voltage (V_{RRM}) of the bypass diode is directly linked with the number of cells bridged by the bypass diode. Knowing the number of bypass diodes (nd) and the open circuit voltage (Voc) of the panel, the maximum repetitive reverse voltage (V_{RRM}) of the bypass diode can be calculated as shown (Quesada *et al.*, 2009):

$$V_{RRM} \geq \frac{V_{OC}Max}{nd} \tag{2.6}$$

2.2.6 Maximum Power Point Current (I_{mpp}) and Voltage (V_{mpp})

I_{mpp} is the corresponding current for the maximum power point on an I-V curve. Also V_{mpp} is the corresponding voltage for the maximum power point on a P-V curve.

2.2.7 Ideality and Fill Factor

The diode ideality factor,N, is used to model second order effects in the diode equation (2.1), although it does not have a large impact on the shape of a solar cell's I-V curve. It is typically chosen as a value between 1 and 2 for crystalline and multi crystalline silicon, but up to 3 times that for triple-junction cells due to their structure of three cells in series (Carrero *et al.*, 2010).

The fill factor of a cell is the ratio of the maximum power produced by an ideal PV cell to a real cell. Generally, fill factor can be used to quantify any reduction in performance, such as dust accumulating on the surface of a module (Mekhilef *et al.*, 2012). Graphically, the fill factor is a measure of how square the I-V curve shape is. A fill factor of 1.0 represents a perfectly square I-V curve, a physical impossibility but a useful reference shape. The calculation of fill factor is given as (Orioli & Di Gangi, 2013):

$$\text{Fill factor (FF)} = \frac{I_{MPP} \cdot V_{MPP}}{I_{SC} . V_{OC}} \quad (2.7)$$

where:

I_{MP} is the maximum power point current

V_{MP} is the maximum power point voltage

I_{SC} is the short-circuit current

V_{OC} is the open-circuit voltage

2.2.8 Solar Cell Efficiency

This is the amount of electricity a PV device produces in relation to the amount of light shining on the device, expressed as a percentage. It is defined as maximum power output of the module divided by the irradiance and array surface. (Andre Mermoud *et al.*, 2012) as:

$$\eta = \frac{P_{MAX}}{P_{Light}} = \frac{I_{SC} V_{OC} . FF}{P_{Light}} \quad (2.8)$$

Maximum output power is P_{MAX} and the power of the irradiance light is Plight,

2.2.9 Resistance

It is an electrical device that has the fundamental tendency to oppose the flow of current in electrical circuits. A typical PV- cell is made up of shunt and series resistance.

Shunt resistance mostly arises due to (Seok-Ju *et al.*, 2007):

1. Manufacturing defects, rather than poor solar cell design.
2. Leakage of current through the cell, around the edges of the device and between contacts of different polarity.

Series resistance occurs due to (Seok-Ju *et al.*, 2007):

1. Movement of current through the emitter and base of the solar cell.

2. Contact resistance between the metal contact and silicon.

2.2.10 Maximum PowerPoint (MPP)

PV cells have a single operating point where the values of the current (I) and Voltage (V) of the cell result in a maximum power output. These values correspond to a particular resistance, which is equal to V/I as specified by Ohm's Law. A PV cell has an exponential relationship between current and voltage and the maximum power point (MPP) occurs at the knee of the curve, where the resistance is equal to the negative of the differential resistance (V/I = -dV/dI) (Singh & Tripathi, 2012). Maximum power point trackers utilize some type of control circuit or logic to search for this point and thus to allow the converter circuit to extract the maximum power available from a cell. A typical graph of current versus voltage and power versus voltage curves depict the maximum power point as shown in the Figure 2.6

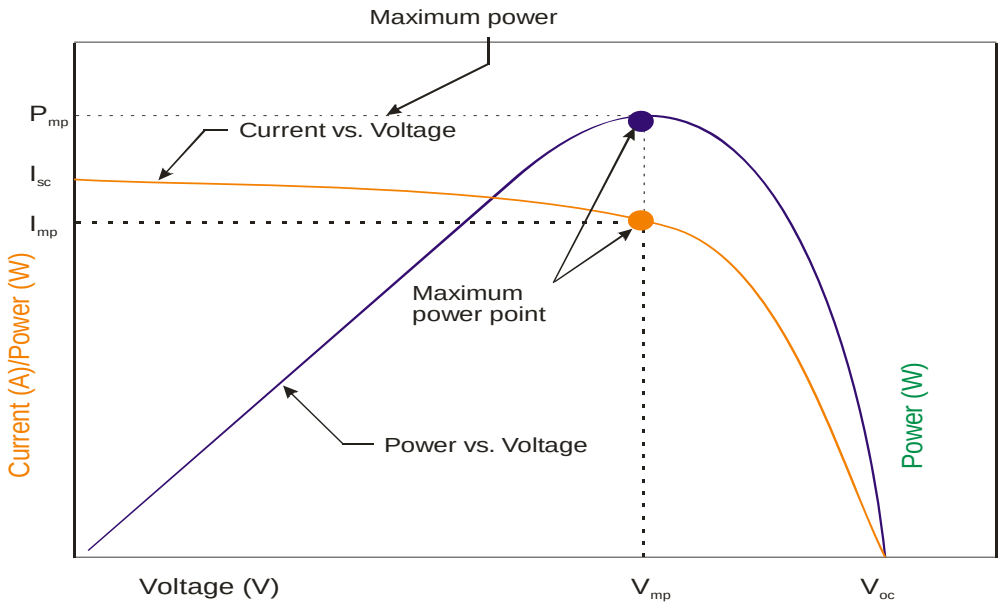


Figure 2.6: Typical Graph of Current vs. Voltage and Power vs. Voltage (Salam *et al.*, 2013)

From Figure 2.6, P_{mp} is the maximum power of the PV-system, V_{mp} and I_{mp} the corresponding voltage and current at the instant of maximum power, I_{sc} the short-circuit current and V_{oc} the open circuit voltage.

2.2.10.1 Maximum power point algorithm

Several algorithms have been exposed to research for MPP such as the algorithm of the adaptive control, Hill Climbing, Increment of Conductance and Perturb and Observe (P & O) algorithm, amongst others. These are discussed as follows:

1. **Adaptive Control Algorithm:** It keeps the system at MPP. Figure 2.7 shows the detailed operation of this algorithm. The aim of this algorithm is to calculate the power at the time t_i from measurements of I_{PV} and V_{PV}, and compare it to the one stored in memory (Salah & Ouali, 2011).

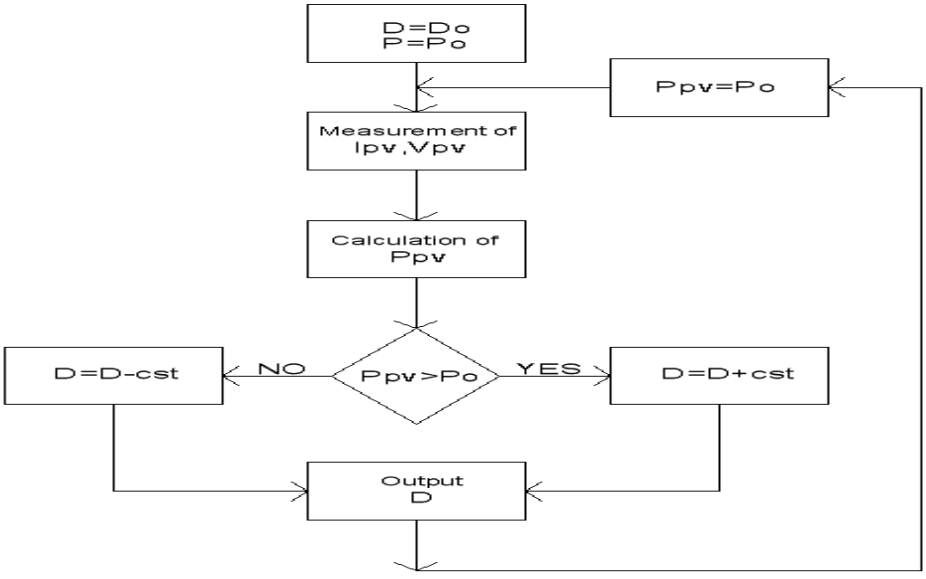


Figure 2.7: Flow Chart of Adaptive Control Algorithm (Arikatla & Abu-Qahouq, 2010).

Hill Climbing Algorithm: This method is based on the relationship between the power and duty cycle illustrated by Figure 2.8. The variable 'P' in Figure 2.8 corresponds to a value "-1" or "1" according to the sign of the slope $\frac{dP}{dD}$. D represents the variation of the desired duty cycle.it is the most commonly used method due to its ease of implementation, but can fail under rapidly changing atmospheric conditions (Cubas *et al.*, 2014). The MPP is reached when $\frac{dP}{dD}$ reaches zero as shown in Figure 2.8

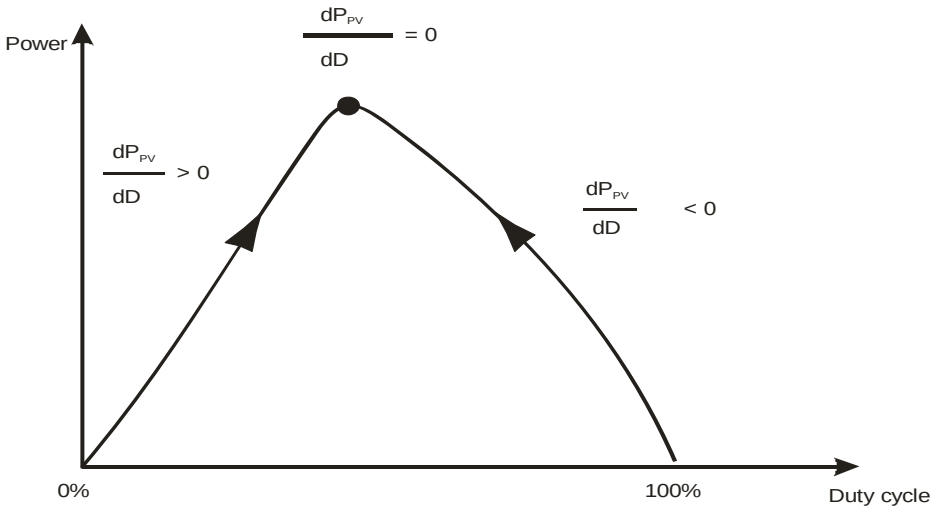


Figure 2.8**:** Relationship between the Duty Cycle and Power (Thongam & Jaiswal, 2015)

2. **Increment of Conductance Algorithm:** This algorithm is based on the observation that the following equation holds for the MPP (Peng *et al.*, 2013):

$$\frac{dI_{PV}}{dV_{PV}} = \frac{I_{PV}}{V_{PV}} \qquad (2.9)$$

When the optimum operating point in the PV plane is at the right of MPP then $\frac{dI_{PV}}{dV_{PV}} < \frac{I_{PV}}{V_{PV}}$, while when the optimum operating point is at the left of MPP then $\frac{dI_{PV}}{dV_{PV}} > \frac{I_{PV}}{V_{PV}}$.The MPP can also be found by comparing the instantaneous conductance $\frac{I_{PV}}{V_{PV}}$ with the incremental conductance $\frac{dI_{PV}}{dV_{PV}}$. Therefore, the sign of $\frac{dI_{PV}}{dV_{PV}} - \frac{I_{PV}}{V_{PV}}$ indicates the correct sense of the perturbation that leads to the MPP (Wang *et al.*, 2013).These are as illustrated in Figure 2.9

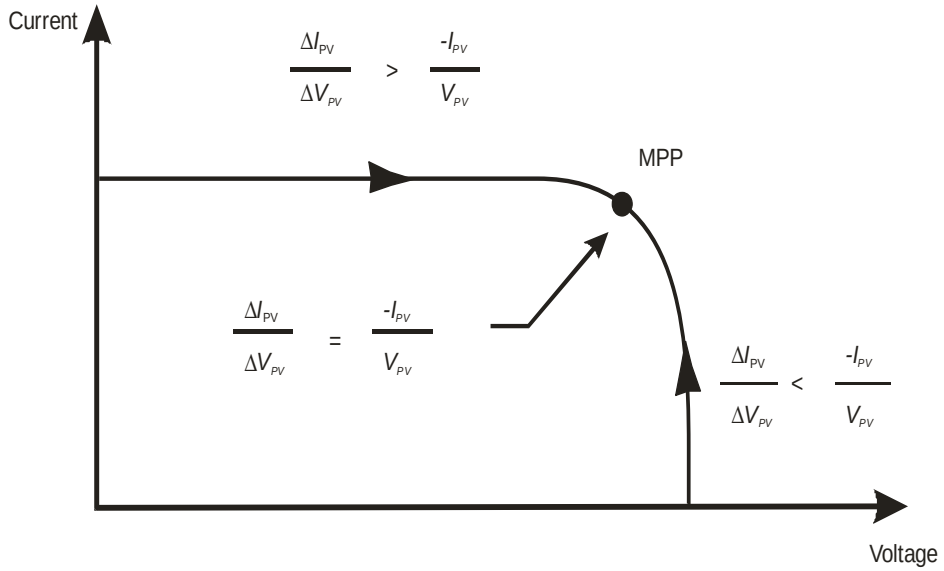


Figure 2.9: Position of MPP in terms of $\frac{dI_{PV}}{dV_{PV}}$ (Isaloo & Amiri, 2016)

3. **Perturb and Observe Algorithm:** The principle of this type of control, as its name indicates, is based on disturbing the voltage value of the photovoltaic generator (PVG) and observing the behaviour of the resulting power. The example in Figure 2.10, illustrates this principle in a simple way. Moreover, from the P-V curve, if the voltage

increases, the power increases. There can be an issue at low irradiance that result in oscillation and also at fast changing irradiance which can result in initially choosing of wrong direction of search (Zainudin & Mekhilef, 2010).

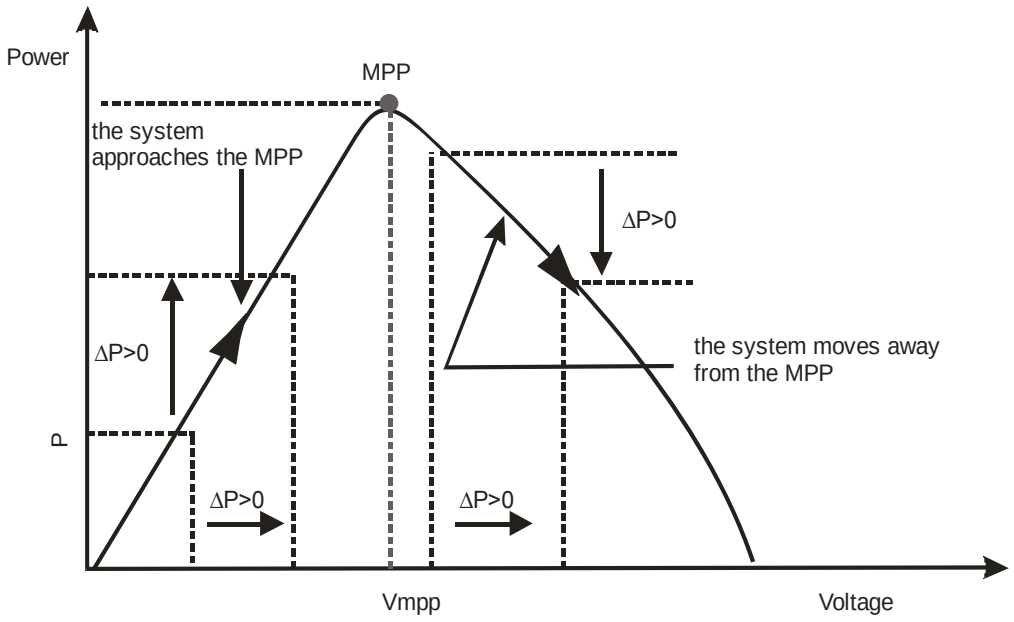


Figure 2.10: MPP as Function of ΔP and ΔV (Christopher & Ramesh, 2013)

When the power increases due to disturbance of the voltage, the direction of the disturbance is kept, otherwise it is reversed to resume the convergence to the new MPP. On the other hand, it has some disadvantages, such as in the characteristic curve P-V of the PVG, $\Delta P = 0$ is never reached. Whenever V increases or decreases, power is changed which makes the implementation of step $PPV_n = PPV_{n-1}$ in the algorithm without importance (Paraskevadaki & Papathanassiou, 2011) .This instability in the value of P will lead to instability around the optimum value of the power. However, this instability can be reduced by minimizing the value of the increment in the search

algorithm. In addition, the increment value is decreased, this algorithm will slow down. As a result a compromise between speed and precision is needed.

4. **Fuzzy Logic Method:** Fuzzy logic asserts itself as an operational technique. Used alongside other techniques like genetic algorithm for advance control, it makes a discrete entrance but popular in industrial control automation. Fuzzy controller is divided into three steps illustrated in Figure 2.11

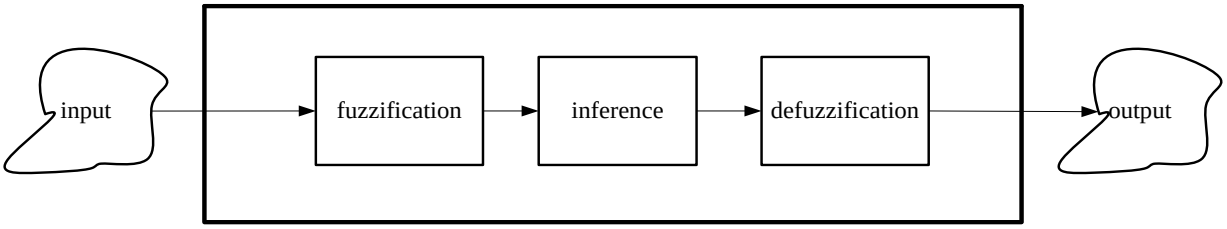


Figure 2.11: Steps of Fuzzy Control (Hawkes & Leach, 2009)

This system implements the fuzzy logic control in three stages: fuzzification, decision-making, and defuzzification. During fuzzification, crisp input variables are converted into linguistic variables based on a membership function. In the decision-making stage, the rules which are specified by a set of IF–THEN statements define the controller behavior (Hawkes & Leach, 2009). The rules describing this stage of operation are expressed as linguistic variables represented by fuzzy sets. In the defuzzification stage, the fuzzy logic controller output is converted from a linguistic variable to a numerical variable still using a membership function. This provides an analog signal that will control the power converter and drive the operating point to the MPP. Fuzzy logic controllers offer the following advantages: capability of working with imprecise inputs and lack of requirement of an accurate mathematical model. However, the learning ability and accuracy achieved by the approximation depend on the number of the fuzzy levels and the form of the membership functions (Dumitras & Moschytz, 2007) .

5. **Artificial Fish Swarm Algorithm:** The optimization algorithms which are inspired by intelligent behaviour of school of fishes, (Artificial Fish Swarm - AFS) are among the most recently introduced techniques (Mehdi *et al.,* 2012).Several approaches have been proposed to model the specific intelligent behaviours of fishes since its invention, AFSA algorithm has been successfully applied to many kinds of problems such as complex optimization domains, data mining, image processing, scheduling in (Yu & Li, 2014; Zeng & Xu, 2012; Zhu & Lin, 2013). According to the various applications mentioned, AFS algorithm confirmed its good performance.

The basic behavior of fish in water is to discover region with more food, by vision or sense; the place with more fish is generally the place with most food. The environment where an Artificial Fish (AF) lives is mainly the solution space and is the states of other AFs (Tijani & Mua'zu, 2015).Its next behaviour depends on its current state and its local environmental state (including the quality of the question solutions at present and the states of nearby companions). The individual behaviour can be represented as follows (Xu *et al.*, 2014):

1. **Synergetic move rule:** Basic communication is kept between individuals. If an individual in the swarm receive a call sent by the other individuals, it moves forward to the call individual's position by a step with a certain probability.
2. **Reconnaissance move rule**: If the individual does not receive a call, it implements reconnaissance according to its own swarm historical experiment. It finds a better goal and then moves forward to the position by a step.
3. **Stochastic move rule:** if the searching individual does not receive a call or does not find a better goal, it moves a step randomly. If it however, finds a better move, it sends a call to the searching swarm.

An artificial fish would influence the environment via its own activities and its companions' activities. The system is started first in a set of randomly generated potential solutions, and then performs the search for the optimum interactively.

The idea of Artificial Fish Algorithm (AFA) is to imitate the fish behaviours, such as preying, swarming and chasing (Tijani & Mua'zu, 2015). Suppose the problem under consideration has *D*-dimensions. Initialize the state vector of the swarm with *N* artificial fish such that the state of one artificial fish can be formulated as: $Xi= (x_n, x_{i2},\ldots x_{iD})$ for i=1,2…N, where Xi is the status of the fish, which represent the target variable for the problem under consideration (Wu *et al.*, 2011). The current food concentration in the position of fish is expressed as $y= f(X_i)$, which is the objective function. The visual distance between the artificial fish is di,j = (Xi-Xj), where *i* and *j* is a randomly generated fish. Step is the maximum size of the movement of artificial fish. δ is the degree of congestion (crowd factor).

Some of the advantages of using Artificial Fish Swarm Algorithm (AFSA) are as follows (Mehdi *et al.*, 2012).

1. Artificial Fish Swarm Algorithm (AFSA) tends to converge quickly and efficiently.
2. It has high flexibility and precision accuracy.
3. Artificial Fish Swarm Algorithm (AFSA) has high fault tolerance.

2.3 REVIEW OF SIMILAR WORKS

Shading in photovoltaic modeling has been an active research topic over the years (Huld *et al.*, 2010).Some of the relevant research works are reviewed as follows:

Sikiru (2007) proposed PV-array model and studied the performance of power conditioning system, the method utilized the development of novel current-voltage

characteristics models for parallel, series and series-parallel connected photovoltaic (PV) cells. The model accounted for ambient temperature, parametric variations of insolation and shading. Mean value theorem was used to track the maximum power point in a single step. Equally, harmonic balance technique and conventional switching function were used in analyzing the power conditioner. On the other hand, the work did not consider varying degree of shading on individual cell with respect to P-V characteristics.

Meanwhile, **Nguyen *et al.* (2009)** proposed a neural network-based approach for modeling solar PV-array under partial shading condition. In their work, a neural network based shading function that characterized the relationship between the maximum output power of the shaded solar PV-array and input such as sun's position angles (solar azimuth and solar altitude) and ambient temperature was used in lieu of shading factor. The maximum output power of the solar PV- array was the output of the neural networks, while the measurement of the input and output signal of the neural network were done several times a day for several days when the solar PV- array was partly shaded by a nearby object. Comparing the neural network output to the measured data, satisfactory result was obtained, however, the work only took into account some modules shaded with uniform pattern.

Subsequently, **Saloux *et al.* (2011)** presented a simple photovoltaic formulation for characterizing and dimensioning cells and arrays. The method permitted the short-circuit current, the open-circuit voltage, the maximum cell power and the optimum cell-operation conditions to be determined. In addition, the model also allowed the quantifying effects of panel temperature and solar irradiance on key parameters, which allowed for easy determination of the performance of any photovoltaic array without performing tedious numerical calculations. The model was based on an ideal cell, where effects of series and

parallel resistances were neglected. The performance of the model was evaluated as a function of standard and weighted mean errors observed between reference and estimated values, and the model provided accurate results. Effective as this work was, effect of degree of shading across the individual cells in a PV-module was not considered.

However, **Mäki *et al.* (2011)** proposed operation of string-connected photovoltaic module under partial shading. They developed I-V characteristics of the bypass diodes by assuming series resistance of the diode to be infinite and the photonic current which its proportional energy gap was equal to zero. The rest of the parameters, like diode ideality factor, series resistance and diode saturation current were obtained by means of curve fitting to a measured I-V characteristics of a Schottky diode. Both bypass diode and PV-cell were independently modelled since they are spectral components of the photovoltaic module. Validity of the simulation model was verified experimentally by using two string-connected Raloss SR30-36 PVM. The overall agreement was good, which justified the use of the simulation model. The proposed model captured the occurrence and properties of multiple maximum power points using one uniform shading strength value for the shading at a time. However, the work did not consider the non-uniform shading across the cells in the PV-module.

In their work, **Pachpande and Zope (2012)** proposed the effect of shading on solar panel. They studied various maximum power point trackers (MPPT) for PV array systems, using MATLAB to simulate the model characteristics, with respect to effect of change in irradiation, temperature and uniform shading across some modules. Series resistance was determined by the contacts and bulk semi-conductor material of the solar cell, while the shunt resistance was estimated by the non-ideal nature of p-n junction that provided a short-

circuit path around the junction. They equally worked on the PV modeling and its characteristics at uniform insolation and showed the effect of different insolation condition on I-V and P-V curves at constant temperature. Their work was a remarkable improvement on the work proposed by **Saloux *et al.* (2011)**, but the effect of the degree of shading across the cells in a PV-module was not evident in this work.

In a similar vein, **Rodriguez *et al.* (2012)** presented a PV-model for photovoltaic (PV) and a procedure for calculating the parameters of the modules of a string (a group of PV modules connected in series). The model took into account the series and parallel resistors of each PV module, considered the Shockley representation of the blocking diode, and approximated the bypass diode as an ideal switch. They described an algorithm to calculate the current and power of PV fields in non-uniform (mismatched) conditions. That procedure was repeated for each string and the current was evaluated by adding the currents of all strings. The parameter calculation procedure evaluated the voltage and current of each active module from the current and voltage of the string and fits the current equation of a module to the experimental data. The accuracy of the proposed model was tested experimentally with a string of two modules obtaining satisfactory accuracy. Although, the algorithm captured module mismatch in the PV-array system, however, they did not evaluate the effect of the degree of shading across the individual cells in the PV-modules.

Subsequently**, Liu *et al.* (2012)** used particle swarm optimization based-MPPT algorithm for PGS operating under partial shading condition (PSC). The standard version of PSO was modified to meet the practical consideration of PGS operating under PSC. Design procedure which took hardware limitation into account was tested, according to simulation of 1000 test cases the proposed technique could reach the GMPP in less than 27 iterations

and average tracking efficiency higher than 99%. Four different shading patterns were used to test the correctness of the proposed system. Although, the proposed method boasted the merits such as easy to implement and system-independent MPPT algorithm for centralized-type PGS operating under PSC, the technique easily get stucked in the local optimal solution and the validation only considered uniform shading pattern.

Similarly**, Mahor *et al.* (2013)** developed a photovoltaic simulation system using MATLAB/Simulink software in order to simulate and evaluate the behaviour of the real photovoltaic systems. A developed model of the solar module was followed by that of photovoltaic system with DC-DC buck converter. The model took solar radiation intensity and cell temperature as inputs and outputs of the I-V and P-V characteristics. Simulation results showed that the model tracked effectively the maximum power point of the system under temperature and irradiance conditions. The effectiveness and superiority of the model over conventional model was verified, as many researchers adopted it in their various models to reduce power loss. But the research work did not estimate the effect of the degree of shading on the PV cells.

For their contribution, **Alsayid *et al.* (2013b)** analyzed the effects of partial shading on a PV array, using two different PV systems, one group was under STC without shading and another partially shaded PV system. The first aspect was done by simulation using MATLAB/Simulink, and the impact of shading was illustrated experimentally by measurements on a two commercial140 W PV series connected panels. The effect of the bypass diodes on PV characteristics under partially shaded conditions was simulated. The instrument HT I-V 400 Tracer was used to verify the validity of the simulation. Different

irradiance (G = 1000 W/$m^2$and G = 500 W/m^2) were equally simulated. From the simulation results, as the two modules were not equally illuminated, the power contributed by each panel was different and the maximum power was less than 280 W. Their work only considered few module mismatches with the same shading pattern.

Sai *et al.* (2013) modelled and simulated solar module under Non-Uniform Conditions and considered the effect of TiO_2 coated cell. In their study, the parameters and temperature of the solar cells were assumed as unchanged and a numerical model was developed to simulate the effect of passing cloud on a solar module. Further, a single cell was coated with TiO_2nano solution to form an effective anti- reflection coat on the cell surface, thereby increasing the effective absorbance and the current output generated for the given insolation. Simulations based on the developed algorithm clearly described the effect of shadow on the PV cell. The efficiency of plain mc-Si solar cell without TiO_2coating was 11.2%. The mc-Si solar cells with double and single-layer TiO_2 coating showed an increase of efficiency of about 14.5%.and 14.1%, respectively, It was observed that the I–V characteristic with a double-layer anti-reflection coating was better enhanced than single-layer anti-reflection coatings. The work centred on the performance of TiO_2 coated PV-cell and did not address the effect of the degree of shading of the individual cells.

On the same problem, **Khalaf *et al.* (2014)** considered the effect of shading on solar Photovoltaic (PV) modules, by using a simulation model, which was able to simulate both the I-V and P-V characteristics curves for PV panels with three different sizes. Three percentages of shading states (25%, 50%and 75%) and without shading were used as efficiency limitations. One diode equivalent circuit was used to investigate electrical

characteristics of a typical Kyocera 54W and Solara 130W solar modules. The results were extracted and simulated using the MATLAB software. The results showed that the performance of both models was widely decreased and the models cannot charge the batteries if the shading was 75% or more for single panel. The systems with the smallest size percentage had better performance than the rest. The drop voltage due to shading was increased with the biggest size and it was clear that Kyocera PV panel was less affected by shading when the same percentage of shading was applied. Their work demonstrated that output power yield was affected based on the intensity of shading, their model was limited to PV-panel evaluation and steady percentage shading pattern across string of modules utilized in the work.

Ramana and Jena (2015), on the other hand, proposed a modified approach of modelling photovoltaic array for partial shading condition using a single diode model. A method based on adaptively varying the value of series resistance was used to find the equivalent circuit parameters. Initially, for uniform irradiance condition, single diode model was used as equivalent circuit. The proposed model was simulated using MATLAB and results were validated using other models in the literature. The proposed methodology had advantage over other conventional methods in terms of accuracy and lower simulation time. The proposed model and series resistance model exhibited similar results at standard test conditions because both the algorithms were based on maximum power matching, But when irradiance level decreased, It was observed that the proposed model was giving better results than series resistance model, especially at maximum power point conditions. The proposed model showed better results than the existing single diode model in all the cases. It showed better results than existing double diode model in some of the cases, but at

low temperature conditions, the double diode model gave better results or nearly similar results. However, the authors did not consider the effect of the degree of shading of individual cell with respect to I-V and P-V characteristics in their work.

Furthermore, **Pareek and Dahiya (2015)** proposed a means of simulation and performance analysis of individual modules to address partial shading and parameter variations in large photovoltaic fields. In their work, they presented a Simulink simulation model of photovoltaic module which was easy, simple, and also capable to show the effect of insolation temperature variations, and effects of shadow on the I-V and P-V characteristics of individual modules within the array. The effectiveness of the simulated model was verified by model in the literature. The simulation model was capable of predicting the behaviour of the PV-module under uniform and non-uniform insolation. The P-V module result generated for maximum power, open circuit voltage, short circuit current, maximum power voltage and current were in close agreement with the datasheet.The model accounted for module mismatch and did not consider the effect of the degree of shading of individual cell.

Shading in the modelling of photovoltaic has received significant research attention. Developing an improved model that will consider the effect of the degree of shading across the individual cells through introduction of degree of shading parameter is still an open question in the literature. MATLAB-based graphic user interface (GUI) using Artificial Fish Swarm Algorithm (AFSA) is proposed in this research to address the open question of degree of shading in a PV cell modeling.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

In this chapter, mathematical model for the proposed PV-array output characteristics coupled with that of the improved AFSA algorithm are presented and discussed. The steps involved in the development of the proposed AFSA based I – V curve, P – V curve, and MPP evaluation Algorithm and the GUI simulator are also presented. Other materials that were useful in achieving the set objectives are also discussed.

3.2 Methodology

The following methodology is adopted in this work:

1. To incorporate a shading parameter into the conventional PV-array model that accounts for varying degree of shading across individual cells of a given PV- module;
   a) Initializing a PV-module as matrix in terms of cells and parallel path.
   b) Assigning shading coefficient to the i^{th} cell in the P^{th} string / column.
   c) Formulating a subject based on item (b) as shading parameter.
   d) Multiply photonic current by shading parameter in the general model.
2. To develop Artificial Fish Swarm Algorithm (AFSA) based approach, for determining the output characteristics (I–V curve; P–V curve; and Maximum Power Point) of PV-cells connected in various configurations with and without bypass diode;
   a) Generate AFSA based Isc, Voc Estimation and point of inflection Algorithms.
   b) Using AFSA Based Algorithms to trace I-V and P-V characteristics curves.
3. To develop a MATLAB based Graphical User Interphase (GUI) having two chapters (AFSA and Dynamic Programming (DP) based approaches) in order to carry out a

comparative analysis using the following performance metrics: open-circuit voltage, short-circuit current and maximum power point.

a) Incorporate both AFSA and DP approaches into a MATALB based GUI software to aid variety of PV-array analysis simulation.
b) Compare the performance of AFSA and DP based approaches using open-circuit voltage, short-circuit current and maximum power point.

3.3 Proposed Mathematical Model of PV-Cells with Shading Effects

Several PV-cell output characteristics equations have been earlier developed by researchers which are further tested and verified for accuracy. Some of these models / equations are presented in the literature review chapter of this work. In this work, a generalized model of PV-cells is developed to aid the development of a generalized solution strategy.

A PV-Cell can be easily represented as an entity or a current source whose output behaviour is governed by the amount of solar irradiation across its surface. This means that, its output characteristics is a function of diode parameters from which the cell is made up of; and voltage at which it is operating. Consider a typical PV-cell model as in equation (2.5). It can be observed that the output current 'I' is expressed in terms of three components which include:

1. The short circuit current I_{SC}
2. The diode saturation current component as in equation (2.1)
3. The component of current flowing into the parallel resistor $\left[\frac{V + IR_S}{R}\right]$

The third component of current is usually very small as compared to the first due to the high value of parallel resistance R_P. Therefore in most conventional models presented in literature, the third component is usually neglected in order to reduce the complexity.

In general, equation (2.5) can be redefined as follows:

$$I = f\ I_{SC}\ + f\ I_S, q, V, I, R_S, R_P, N, K, T \tag{3.1}$$

But the set of parameters (Is, q, Rs, Rp, N, K,T) are fixed for a given cell operated at a given temperature T and pressure, therefore equation (3.1) can be redefined in terms of fewer independent variables as given by:

$$I = f\ I_{SC}\ + f\ I, V \tag{3.2}$$

However, the property of solar cells are such that when connected in series or parallel configuration, the aggregate output can be expressed in terms of the output of individual cells using the laws of series and parallel combinations. This implies that when the output of a PV-array is to be determined, individual cells forming the array could be considered and solved using equation (3.2) and finally aggregated to obtain the resulting output of the PV-array. Moreover, I_{SC} directly depends on the intensity of sunlight, therefore the effect of shading can be formulated as a factor of I_{SC} Such that equation (3.2) becomes:

$$I = \beta f\ I_{SC}\ + f\ I, V \tag{3.3}$$

Where: β represents the degree of shading of the PV-cells. The parameter β is a matrix with as much elements as the number of cells forming the given PV-module. In order to simplify the description of β, consider PV-module with 18 PV-cells having 6 parallel paths as shown in Figure 3.1

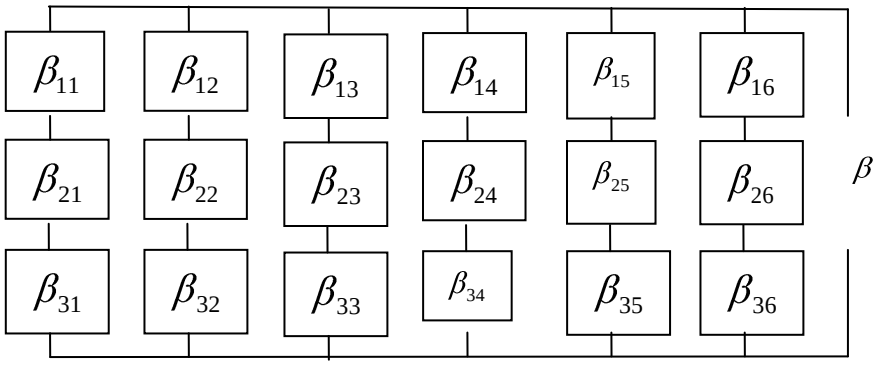


Figure 3.1: Typical PV-Module with 18 PV-Cells (with 6 Parallel Paths)

In Figure 3.1, a PV-module is formed by 6 strings of 3 PV-cells each. It is established in literature that PV-cells connected in series have a resulting output current equal to the current generated by the cell with least output. This corresponds to the cell with the highest degree of shading (i.e. least value of β). β is formulated such that at total shading, it will have a value of zero. In general, β can be expressed using:

$$\beta = \frac{\sum_{P=1}^{n_P} \min[\beta_P]}{n_P} \tag{3.4}$$

And $$\beta_P = [\beta_{1P}, \beta_{2P}, \beta_{3P}, \beta_{4P} \ldots\ldots \beta_{n_s P}] \tag{3.5}$$

Where:

n_p represents the number of parallel paths

β_P represents the P[th] column in the $\beta matrix$

β_{iP} represents the shading coefficient of the i[th] cell in the P[th] string / column.

n_s Represents number of cells per path.

Therefore it is worthy of noting that the value of β must lie between zero and one as described by the following inequality $0 \leq \beta \leq 1$

As an example, let the module shown in Figure 3.1 have shading defined by the $\beta matrix$ given by:

$$\beta = \begin{pmatrix} 0.4 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 \\ 0.3 & 0.4 & 0.5 & 0.7 & 0.8 & 0.8 \\ 0.2 & 0.5 & 0.6 & 0.8 & 0.8 & 0.9 \end{pmatrix} \tag{3.6}$$

min (β_P) is estimated by taking the minimum values across each column. The value of the degree of shading is then calculated using:

$$\beta = \frac{\sum 0.2, 0.4, 0.5, 0.6, 0.7, 0.8}{6} \tag{3.7}$$

$$\beta = 0.5333$$

The pseudo-code for computing the degree of shading is shown in Algorithm 1.

Algorithm 1: β Estimation Algorithm

1. Determine the number of parallel paths n_P ,and the number of cells per parallel path n_S ;
2. Evaluate the degree of shading for each of the cells forming the module. Using equation (3.8) and define an empty matrix β
3. For p = 1 : n_P
4. For I = 1 : n_S
5. If $\beta_{P(-1)} \neq$ empty
6. If $\beta_{Pi} \square \beta_{P(-1)}$
7. $\beta_{\min} = \beta_{Pi}$
8. else
9. $\beta_{\min} = \beta_{P(-1)}$
10. end
11. Else
12. $\beta_{\min} = \beta_{Pi}$
13. End
14. End
15. $\beta = [\beta, \beta_{\min}]$

16. end
17. $\beta = \frac{\sum \beta}{n_P}$
18. Stop

For a given cell (i), the degree of shading β_i can be calculated using:

$$\beta_i = \frac{Intensity\ of\ solar\ Radiation}{Maximum\ Intensity\ of\ Solar\ Radiation} \tag{3.8}$$

β Can also be expressed in terms of the output current of a solar cell as in:

$$\beta_i = \frac{I_i}{I_{i\max}} \tag{3.9}$$

Where:

I_i represents the actual output current of the cell (i)

$I_{i\max}$ represents the output current at the condition when the intensity is highest.

Finally, every I – V characteristics equation can be expressed using equation (3.2) which implies that the shading parameter β can be assigned as in equation (3.3).

3.4 Proposed AFSA based approach for I-V Curves

AFSA as an optimization technique requires an objective function. In order to develop an approach for tracking the I-V curve of any given PV-module, it is important that a suitable objective function is designed. Consider the improved PV-module equation (3.3), an objective function may be defined as:

$$f_{obj} = I - \beta f\ I_{SC}\ - f\ I,V \tag{3.10}$$

Equation (3.10) can be rewritten as:

$$f_{obj} = I - f\ \beta, I_{SC}, I, V \tag{3.11}$$

Therefore, the objective function of the I-V Curve tracking is to minimize equation (3.11) which is expressed using:

$$f_{obj} = \min\ I - f(\beta, I_{SC}, I, V) \tag{3.12}$$

The conventional AFSA is characterized by a randomly generated population of fish (row of N_P real number ranging between zero and one where N_P is the number of parameters usually defined at the beginning of the simulation) which are made to undergo a set of manipulation processes such as; swarming, chasing and preying in order to determine the fish with the best fitness based on the objective function.

Consider an artificial fish with N_P parameters as in:

$$f_i = P_1, P_2, P_3, P_4, ...P_{NP-1}, P_{NP} \tag{3.13}$$

The target is to represent the output current as a fish through a decoding function. This means that each of the population of the artificial fish must be converted to current, so that the output current of the PV-cells could be determined by solving the non-linear I-V characteristics equation shown in equation (3.3).

From equation (3.13),

$$0 \leq P_i \leq 1 \qquad \text{for } i = 1 \text{ to } N_P \tag{3.14}$$

It may be observed that the sum of all the parameter in the ith artificial fish f_i will have a maximum value of N_P and a minimum value of zero (0). Therefore, equation (3.4) can be redefined:

$$f_j = \frac{\sum_{i=1}^{N_P} f_j \ \ i}{N_P} \tag{3.15}$$

Where

N_P represents the number of parameter

f_j represents the decoded version of the j[th] artificial fish f_j

f_j will have a maximum value of 1 and a minimum value of zero. But the maximum value of the PV-cell output current is short circuit current I_{SC} and its minimum is zero. Therefore, equation (3.15) can be modified to represent output current of PV-cell as shown in:

$$I = \left[\frac{\sum_{i=1}^{N_P} f_j \ i}{N_P} \right] . I_{SC} \qquad (3.16)$$

Equation (3.16) represents the AFSA based decoding function (DF) of the I-V curve solution approach. The behaviour of such curve can be used to improve the rate of convergence of the AFSA optimization approach.

Consider the diagram shown in Figure 3.2:

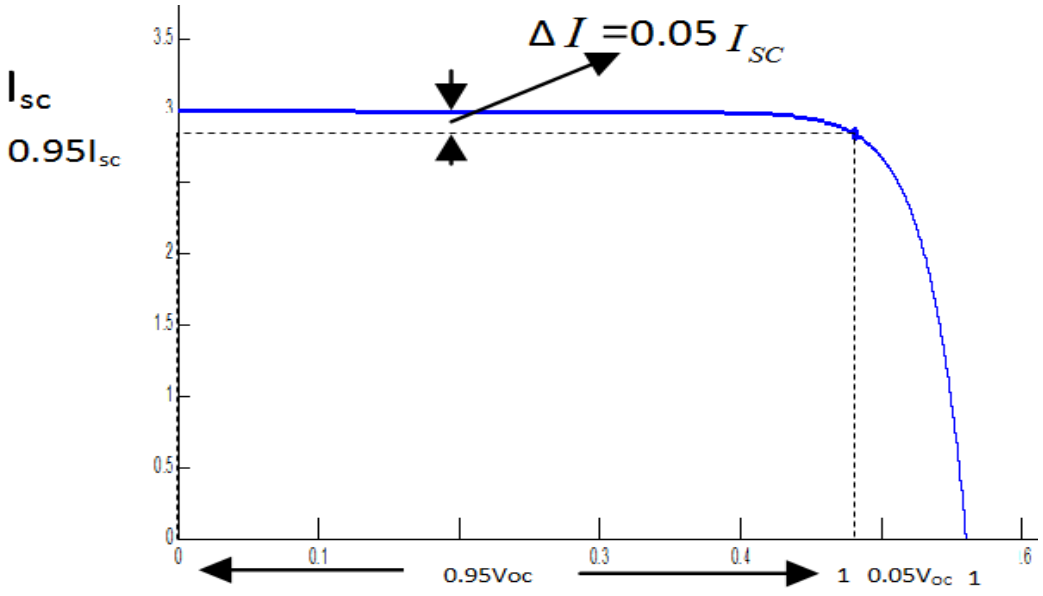


Figure 3.2: Variation of Output Current with Changes in Output Voltage

In Figure 3.2, the output current of PV-cell is shown to vary only slightly for a large percentage variation in output voltage. This shows that about 95% drop in output current usually occur only within a 5% variation in output voltage. This means that artificial fish must be such that: their decoded values vary over a 5% range of I_{SC} with a probability of 0.95.

However, MATLAB software uses the normal distribution to generate random real numbers ranging between zero and one. This of course has to be modified by improving equation (3.16).

The proposed modified decoding function is further described using:

$$I = \begin{cases} I_1 if\ 0 \le V \le 0.95V_{oc} \\ I_2 if\ 0.95V_{oc} \le V \le V_{oc} \end{cases} \quad (3.17)$$

$$I_1 = I_{sc}\left[1 - 0.05\left[\frac{\sum_{i=1}^{N_P} f_j\ \ i}{N_P}\right]\right] \quad (3.18)$$

$$I_2 = 0.95 I_{SC}\left[\frac{\sum_{i=1}^{N_P} f_j\ \ i}{N_P}\right] \quad (3.19)$$

Where:

I_{SC} represents the short circuit current

I_1 and I_2 represents the respective decoding functions at the two intervals defined in equation (3.18) & (3.19)

f_j represents the j[th] artificial fish.

Algorithm 2, described the pseudo code for decoding an artificial fish as follows:

Algorithm 2: AFSA Decoder Algorithm

1. Enter the value of the output voltage V
2. Enter the AFSA parameters
3. If $V \leq 0.95V_{OC}$
4. Decode fish using equation (3.18)
5. Check if the allowable minimum value of equation (3.12) is attained
6. if yes
7. stop executing the algorithm
8. else
9. generate new fish and go to stop 2
10. end
11. else if $V \leq V_{OC}$
12. Decode fish using equation 3.19
13. Repeat stops 5 to 10
14. end

The decoding algorithm presented earlier is used continuously to decode randomly generated artificial fish to their respective PV-cell output current equivalent. The AFSA on the other hand is used to determine the optimal/minimum value of equation (3.12). This makes it possible for the I-V curve to be traced.

The pseudo code for tracing the I-V Curve is provided in Algorithm 3 as follows.

Algorithm 3: AFSA Based I-V Curve Tracing Algorithm

1. Define the parameters of AFSA and the values of I_{SC} and V_{OC}
2. Generate population of artificial fish
3. Decode each fish using Algorithm 2
4. If the decoded fish is optimal according to equation 3.12 and output voltage V is less than V_{OC}

5. Increment V and repeat steps 2 to 4
6. End if $V = V_{OC}$

In some PV-cell model equations, the short circuit current I_{SC} is not usually defined but replaced with the photo current I_{λ}, likewise the open circuit voltage V_{OC} . In such situations, both I_{SC} and V_{OC} have to be computed. In most conventional PV-cell model, V_{OC} can be computed directly by substituting I = 0 and making V the subject. While I_{SC} is always computed through a solution of non-linear equation. In any case, V_{OC} and I_{SC} can be computed using Algorithm 4 and 5 respectively.

Algorithm 4: AFSA based Voc Estimation Algorithm

1. Define the parameters of a PV-module
2. Initialize AFSA and optimize equation 3.12 with $I = 0$ and using the decoding function shown in equation (3.20)
3. Print the values of V as V_{OC} if equation 3.12 is minimized.

$$V_j = \left[\frac{\sum_{i=1}^{N_P} f_j \ i}{q.N_P} \right] In \left[\frac{I_\lambda + I_S}{I_S} \right] .n.k.T \qquad (3.20)$$

Where:

I_{λ} = Solar photo current

I_S = Diode saturation current

n =Diode proportionality factor

k = Boltzmann constant

T = Operating temperature

f_j = The j[th] artificial fish

i = 1to N_P, number of parameter

q = Electron charge

V_j = The equivalent output voltage represented by j[th] artificial fish

Let a new parameter V_O be defined as in equation 3.21

$$V_O = \ln\left[\frac{l_\lambda + l_s}{l_s}\right].\left[\frac{n.k.T}{q}\right] \tag{3.21}$$

Thus: equation 3.20 can be expressed in terms of V_O as in equation (3.22)

$$V_j = \frac{V_O}{N_P}\left[\sum_{i=1}^{N_P} f_j \;\; i\right] \tag{3.22}$$

The parameter V_O is obtained from equation (2.5) by substituting $I = 0$ and $V = V_O$, while neglecting the third right hand term. I_{SC} is also, replaced with I_λ in the same equation. Equation (3.22) represents the AFSA based decoding function for determining the open circuit voltage V_{OC} of any given PV-module.

Algorithm 5: AFSA Based I_{SC} Estimation Algorithm

---

1. Define parameter of PV-module
2. Initialize AFSA using the specified parameter
3. Optimize equation (3.12) with V and using equation (3.18) as AFSA decoding function for current
4. Print the value of $I_1 =$ as I_{SC} with the objective function is minimized
5. Stop.

3.5 Proposed AFSA Based Approach for P-V Curves

In the proposed curve tracing approach, once the open circuit voltage V_{OC} is computed, an increment can be defined over which the output current is calculated using the approach

discussed earlier. The smaller the value of ΔV, the smoother the P-V Curve. In the other hand, smaller value of ΔV leads to more computation owing to the fact that equation (3.12) has to be solved for every increase in output voltage.

The points on the P-V Curve are obtained by multiplying the output current I with the output voltage. The number of such point is calculated using:

$$n_{P-v} = \frac{V_{OC}}{\Delta V} \tag{3.23}$$

Thus:

$$0 \leq \Delta V \leq V_{OC} \tag{3.24}$$

In order to avoid approximation in the steps, the value of n_{P-v} is defined and used in evaluating the increment ΔV. Therefore any point (P_t) on the P-V curve can be defined using:

$$P_t\ P,V\ = I_{vt} * V_t \tag{3.25}$$

Where:

I_{vt} = Output current at voltage V_t

V_t = Output voltage at the t^{th} point

$P_t\ P,V\ =$ Magnitude of power at the t^{th} point

$P =$ Output power of PV-module

$V =$ Output Voltage

The pseudo code for generating the P-V characteristics Curve of a given PV-module is given in Algorithm 6 as follows:

Algorithm 6: AFSA Based P-V Curve Tracing Algorithm

1. Define AFSA and PV- module parameters
2. Run Algorithm 4 and 5 to determine V_{OC} and I_{SC} respectively
3. Run Algorithm3 to determine the respective points P_{iv} on the I-V curve
4. Multiply the current on the I-V curve with the output voltage corresponding to that point to obtain power at each point $P_t \quad p,v$
5. Plot $P_t \quad p,v$ against the output v
6. Stop if the maximum voltage V_{OC} is reached

3.6 Proposed AFSA Approach for Maximum Power Point

In the proposed research, unlike the conventional methods for determining MPP, once the P-V is traced, all the individual power points (PP) are available for MPP computation. The MATLAB command 'max (Pt) can be used to obtain the maximum power from a set of PP obtained from the I-V curve. MPP is characterized by three set of values namely:

1. Maximum Power (MP)
2. Current at MP (I_{MPP})
3. Voltage at MP (V_{MPP})

The first and the last items are obtained from the P-V Curve, while the second item is obtained from the I-V curve by tracing the point at which V_{MPP} meets with the curve, to the current I axis. In most conventional PV-modules, I_{MPP} and V_{MPP} are closer to I_{SC} and V_{OC} respectively depending on module parameters. The proposed AFSA based method for finding the MPP uses I_{SC} and V_{OC} as guide for solution search. This may be further be described using figure 3.3

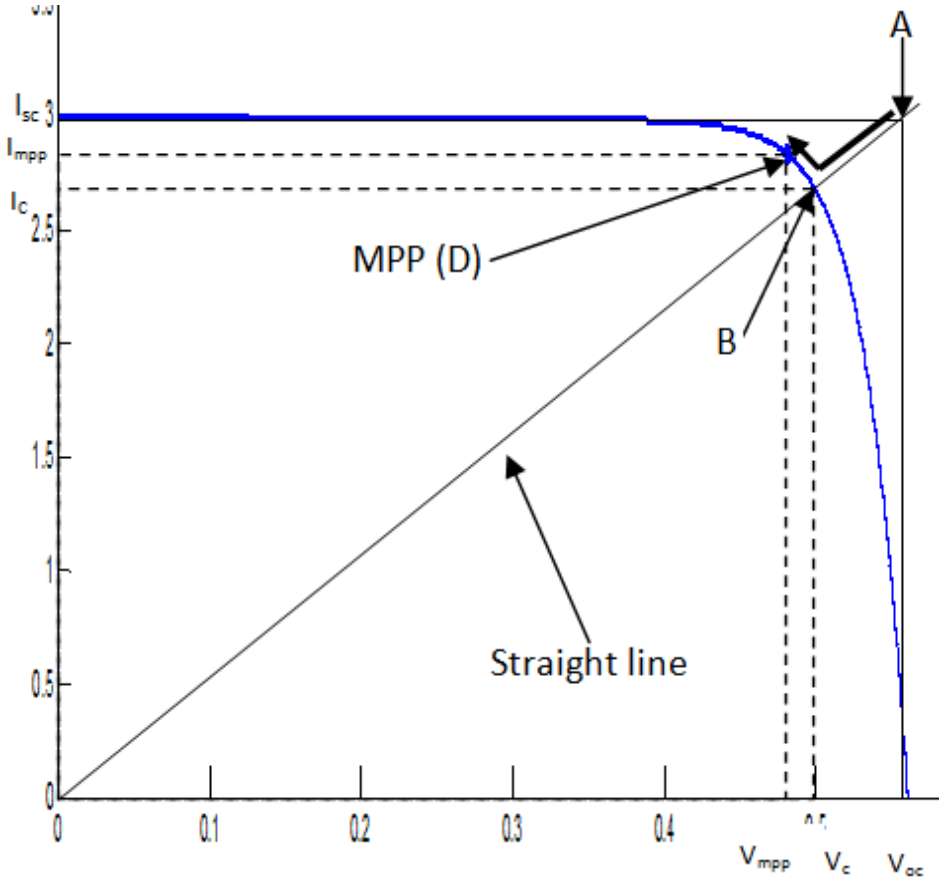


Figure 3.3: I-V Curve showing the Proposed MPP Search Approach

As may be observed from Figure 3.3, the I-V curve intersect the straight line at point B which correspond to (V_C, I_C), this point can be located using the equation of the straight line given in:

$$I_C = MV_C \tag{3.26}$$

Where:

$M = \frac{I_{SC}}{V_{OC}}$ is the slope of the straight line.

Since the output current in a PV-Cell is non-linear relationship of its voltage, I_C could be obtained by optimizing equation (3.12) with $V = \frac{I_C V_{OC}}{I_{SC}}$ after which the voltage V_C is determined accordingly. After I_C and V_C are found, a hill climbing approach is used based on AFSA to obtain the MPP as indicated by point 'D'. Therefore the proposed approach can be summarized as follows:

1. Locate point A V_{OC}, I_{SC} using AFSA;
2. Locate point B V_C, I_C using AFSA based on point A;
3. Make use of hill climbing approach to gradually move from point B to the described destination (point D) based on AFSA.

This approach has the advantage of short travelling distance (few points) as compared to the conventional hill climbing approach which results in fewer computation and increased rate of convergence.

3.7 Proposed DP based approach for PV –array

This approach was designed to serve as a comparison between the proposed AFSA based model. In this approach open circuit voltage V_{OC} is not solved dynamically, but the output current is solved dynamically. The value of V_{OC} is computed directly by making V the subject of the conventional PV-module characteristic formula while substituting $I = 0$ as the output current. The disadvantage of this method is that, if a model equation is designed to have a non-linear relationship of voltage (as in equation 2.5), finding a solution for output voltage becomes very complex. As such, the accuracy of this approach largely depends on the linearity of the I-V characteristics equation. In the other hand, the output

current of PV-module is computed dynamically. Once the value of V_{OC} is known, an incremental changes ΔV in the value of the output voltage V is defined and used to compute the number of steps n_S in the I-V curve. Normally, the higher the value of n_S the smoother the curve. The relationship between $V_{OC}, \Delta V,$ and n_S is given as follows.

$$n_S = \frac{V_{OC}}{\Delta V} \tag{3.27}$$

The characteristics equation of PV-module is solved for output current I over the increase in output voltage V using Algorithm 7 as shown in the following.

Algorithm 7: DP Based Output Current I Estimation Algorithm

1. Define PV-module Parameters
2. Define the error limit (e)
3. Define the Operating Voltage V
4. Define the change in current ΔI □ □Output current I and $n = 0$.
5. While $I_{n+1} - I_n$ □ e
6. If $_n$ □1
7. $I_n = \beta f\ I_\lambda\ + f\ I_{n-1}, \mathrm{V}$
8. Else
9. $I_n = \Delta I$
10. End
11. $I_{n+1} = \beta f\ I_\lambda\ + f\ \ I_n + \Delta I\ , V$
12. $n = n+1$
13. End if condition is fulfilled

In Algorithm 7, the function $\left[I = \beta f\ I_\lambda\ + f\ I_n, V \right]$ and subsequent once represents any output characteristics equation of PV-Cells. I_n and I_{n+1} represent the consecutive during the iterative process.

Furthermore, both the P-V and I-V Curves are generated by incrementing the value of the output voltage over the required n_S steps using the change ΔV and running Algorithm 7 at each point. In the case of P-V Curve, the output current generated by Algorithm 7 (at each stage) is converted to its respective power by multiplying it with the voltage at that stage. Finally the power output obtained is then plotted against the range of voltage. While in I-V Curve case, the output current I generated by Algorithm 7 is directly plotted against the voltages at each step.

In the case of MPP, the proposed DP approach is not too different from the AFSA approach. The heuristic search method in AFSA is replaced with an incremental approach in which set of values that are much smaller than ΔV are defined as increments. The hill climbing approach described using figure 3.3 is still adhered to in this case. This means that points A, B and then D also serves as the manner in which the MPP is determined.

3.8 The Proposed Model for Bypass Diode in PV-Modules

Bypass diodes are inherent in PV-module and has the aim of providing an alternative path for current in order to prevent the effect of one part of the Cell-arrays from propagating itself over the entire PV-module. However, bypass diodes introduces inflection into I-V and P-V characteristics curve of PV-modules. This results in an alteration in maximum power point.

3.9 1nflection in I-V Curve

For simplicity of description, consider a typical I-V curve having an inflection at point 'A' as shown in figure 3.4; point 'A' is traced to the vertical and horizontal axis to indicate I_A and V_A respectively.

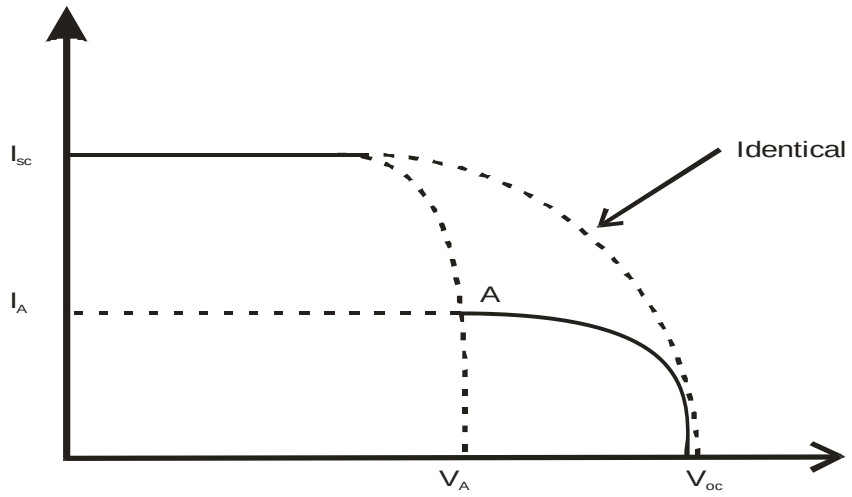


Figure 3.4: I-V Curve for Identical and Non-identical PV-Array Configuration.

Figure 3.4, is obtained from a PV-array with two non-identical parallel paths (one path has higher output). If the paths were identical, an I-V curve such as, the one represented by dotted line in figure 3.4 would have been generated. Point 'A' is called the inflection point and it represents the point at which the highest output current superimposes the other. The Cells forming each of the parallel paths must obey KCL regardless of their output and the resulting output from the parallel paths must obey KVL. Thus V_{OC} represents the open circuit voltage of the path with the highest output voltage, while I_{SC} represents the combine output short circuit current of the entire parallel paths. In order to locate point (A), I_A and V_A must be determined as follows:.

$$I_A = I_{SC} - I_{\min} \tag{3.28}$$

Where:

$I_A =$ Inflection current

$I_{SC} =$ Combined output short circuit current

$I_{\min} =$ Output current of the path with minimum short circuit current $I_{SC\min}$

Also,

$$V_A = V_{OC} - V_{\min} \tag{3.29}$$

Where:

V_A represents inflection voltage

V_{OC} represents the resulting open circuit voltage of the system

$V_{\min}$ represents the output Voltage of the path having the least voltage.

3.10 Shading Effect on PV-System with Bypass Diodes

Consider a PV-System with n individual PV-module connected in series-parallel configuration, such that the system has n_p parallel paths with each path having n_s modules. This implies that the following holds:

$$n = n_s X n_p \tag{3.30}$$

Assuming each module is protected using a bypass diode (meaning that, each module has a bypass diode connected in parallel with it). Let shading cause the output of each module to be degraded by a factor β_{ij}, where i and j represents the position and parallel path on which a module belongs. The output characteristics of the system can be defined at β_{ij} for $i = 1 : n_S$, and $j = 1 : n_P$ as follows:

$$I_{SC_{Sys}} = I_{SC_m} * n_P \tag{3.31}$$

And

$$V_{OC_{Sys}} = V_{OC_m} * n_S \tag{3.32}$$

In the case of partial shading, algorithm 1 may be employed to compute the resulting degree of shading β for the entire system. In this case, since bypass diodes are presents, n_P inflections are introduced into the characteristics curves of the system. The point of inflection on the I-V curve could be determined using Algorithm 8, as follows.

Algorithm 8: Points of Inflection Determination Algorithm

1. Define the input parameter of PV-system
2. Estimate n_S and n_P also $V_{OC_{Sys}}$ and $I_{SC_{Sys}}$
3. For $n=1$ to n_P
4. For $m = 1$ to n_S
5. Determine $I_{SC_{nm}}$ and $V_{OC_{nm}}$
6. If $n_S \neq 1$
7. If $I_{SC_{nm}} \neq I_{SC_{n\,m-1}}$
8. $I_{\min} = I_{SC_{nm}}$
9. else
10. $I_{\min} = I_{SC_{n\,m-1}}$
11. end
12. If $V_{OC_{nm}} \neq V_{OC_{n\,m-1}}$
13. $V_{\min} = V_{OC_{nm}}$
14. else
15. $V_{\min} = V_{OC_{n\,m-1}}$
16. end
17. else
18. $I_{\min} = I_{SC_{nm}}$
19. $V_{\min} = V_{OC_{nm}}$
20. end
21. end
22. If $n=1$
23. $V_{\mathrm{infL}_n} = V_{\min}$
24. $I_{SC_{Sys}} = I_{SC_{Sys}} - I_{\min}$
25. $I_{\inf L_n} = I_{SC_{Sys}}$
26. else
27. $V_{\inf L_n} = V_{\inf L_{n-1}} + V_{\min}$

28. $I_{\inf L_n} = I_{\inf L_{n-1}} - I_{\min}$
29. end
30. End
31. Print a row matrix of $I \inf L$ with columns
32. Print a row matrix of $V \inf L$ with n_P columns

Once the inflection points on the I-V Curve are determined, Algorithm 9 can then be used to trace the I-V Curve of the PV System. With the i[th] inflection point located at $V \inf L_i, I \inf L_i$.

In this approach, the more the number of parallel paths, the higher the complexity of the approach and the higher the number of computations. For smoother curves, high precision is required, since AFSA based method is proposed, the rate of convergence depends on the curve. In this work a few thousands of points are set as limit in order to generate convergence. Algorithm 9 is presented as follows.

Algorithm 9: AFSA Based I-V Curve Tracing Algorithm with Bypass Diode

1. Define the PV-Input parameters
2. Run Algorithm 8 to generate the inflections M_{Inf} matrix
3. Run Algorithm 2 to compute I_{SC} for the system
4. Run Algorithm 3 to compute V_{OC} for the system
5. For $i = 1$ to length M_{Inf}
6. If $i = 1$
7. Trace I-V Curve from $0, I_{SC}$ to V_{Inf}, I_{Inf} using Algorithm 5
8. Else if 1< i < n
9. Trace I-V Curve from $V_{Inf_{i-1}}, I_{Inf_{i-1}}$ to V_{Inf_i}, I_{Inf_i} using Algorithm 5
10. Else
11. Trace the I-V Curve from $V_{Inf_{i-1}}, I_{Inf_{i-1}}$ to V_{Inf_n}, I_{Inf_n} using Algorithm 5
12. End
13. End for

3.11 Proposed Approach for P-V Curves with Bypass Diode Effect

As discussed earlier, bypass diodes introduce inflection into PV- Output characteristics curves such as P-V Curve. Once an I-V Curve is traced, the individual points on the I-V Curve can be used to generate the P-V Curve. This may be achieved by multiplying the current (I) and the Voltage (V) at each point to obtain the power (P) at that point .This power can then be plotted against the voltage to obtain the P-V Curve for a given system.

In the previous model of PV-system with bypass diode, the inflection in the I-V Curves also introduces inflection in the P-V Curve, i.e the point of inflection V_{Inf}, I_{Inf} on the I-V Curve results into a corresponding point of inflection V_{Inf}, I_{Inf} on the P-V Curve. Therefore, it may be further stated that the X-coordinate of any inflection point on P-V Curve is obtained directly

from the X-coordinate of an I-V Curve while the Y-coordinate is obtained using equation 3.32 as follows

$$P_{Inf} = I_{Inf} * V_{Inf} \tag{3.33}$$

The P-V Curve with inflection may be obtained using a similar procedure as that of I-V Curve having inflection as discussed earlier. However, the distinction may be noticed in the procedure for tracing a P-V Curve with the inflection as discussed in Algorithm 10.

Algorithm 10: Algorithm for Tracing P-V Curve with Inflection

1. Define PV-Input parameters
2. Run Algorithm 8 to generate the inflection matrix $M_{Inf} = \begin{bmatrix} V_{Inf} \\ I_{Inf} \end{bmatrix}$
3. Replace I_{Inf} by $P_{Inf} = I_{Inf} \times V_{Inf}$ to form a new matrix $M_{Inf} = \begin{bmatrix} V_{Inf} \\ P_{Inf} \end{bmatrix}$
4. Run Algorithm 3 to compute the V_{OC} for the new system
5. For $i = 1$ to length M_{Inf}

6. If $i = 1$
7. Trace P-V Curve from (0,0) to V_{Inf_i}, P_{Inf_i} using Algorithm 6
8. Else if 1< i < n
9. Trace P-V Curve from $V_{Inf_{i-1}}, P_{Inf_{i-1}}$ to V_{Inf_i}, P_{Inf_i} using Algorithm 6
10. Else
11. Trace P-V Curve from $V_{Inf_{i-1}}, P_{Inf_{i-1}}$ to V_{Inf_i}, P_{Inf_i} using Algorithm 6
12. End if
13. End for
14. Stop.

In the case of PV –modules with bypass diodes the output characteristics of the system such as the V_{OC} and I_{SC} still maintain their original value as when no bypass diodes are not connected. However, inflection points are introduced into the P-V and I-V Curves while the Maximum Power Point MPP shifts. This results into n_P MPP. Where n_P represent the number of Parallel paths in the P-V System. Since there must be one (1) true MPP, this implies that there is one (1) local MPP. In order to determine the true MPP, all the MPPs are determined and the true MPP is the maximum of all the MPPs. The proposed approach for determining the MPP of a system with bypass diodes as further described using figure 3.5 as follows.

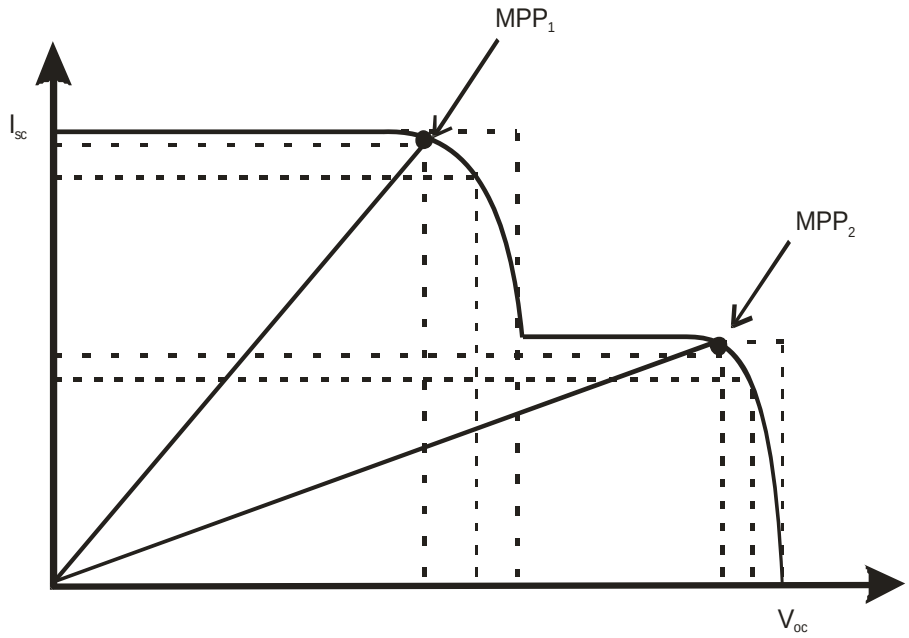


Figure 3.5: Determination of MPPs on a PV-System with Bypass Diodes

The true MPP is determined as follows:

$$MPP = \max \ MPP_1, MPP_2, \tag{3.34}$$

Equation (3.34) is used in the case of two MPPs, for a system with n MPPs, equation (3.34) can be improved to take care of any number of MPPs using:.

$$MPP = \max \ MPP_1, MPP_2 ..... MPP_n \tag{3.35}$$

3.12 The Developed PV- Array Analysis Simulator

Fortunately, MATLAB based GUI has been developed to take care of all the PV-array analysis that were discussed earlier in this write-up. This GUI has been programmed to be able to accommodate changes in user parameter choice and desired scenario of simulation. In the GUI, two set of approaches are incorporated namely:

1. AFSA Based PV-Analysis
2. DP Based PV-Analysis

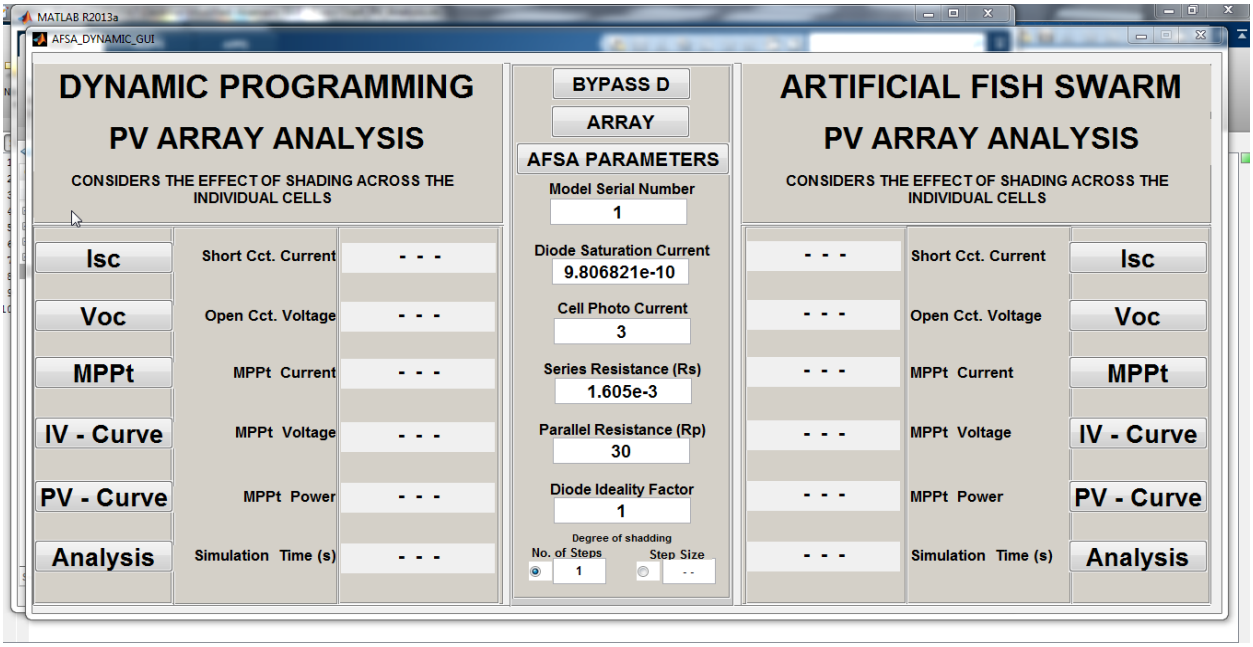


Figure 3.6: Screenshot of the Developed GUI Window

Figure 3.6 shows the face of the developed GUI window. In the GUI, a set of text boxes are provided to enable a user input the PV-array parameters. These inputs are defined as follows.

1. Parallel Resistance
2. Series Resistance
3. Diode saturation Current
4. PV-Cell Photo Current
5. Diode Ideality Factor

However, the GUI can output values for the following output parameters:

1. Short Circuit Current
2. Open Circuit Voltage
3. Maximum Power Point
4. Maximum Power Point Current
5. Maximum Power Point Voltage

6. Simulation Time(s)
7. I-V Curve
8. P-V Curve
9. Degree of Shading (no of steps and step sizes)

Note that: the input and output are utilized and generated respectively by both the approaches (AFSA and DP). The graphics user interphase also output curves as results. Such curves will be generated and discussed in the result chapter of this work. The simulator could also be used to carry out a simulation involving the effect of bypass diode on PV-arrays. This is achieved by clicking the bypass diode button (BYPASS D).

3.13 Proposed PV- Array Analysis Algorithm

The previous discussions have covered the various steps involved in the proposed PV-array Analysis Algorithm (PAAA). The steps involves : Determining the short circuit current (Isc) and open circuit voltage (Voc) ; Using the values of Isc and Voc to trace the I-V Curve ; Using the Voc and Isc to trace the P-V Curve ; Determining the maximum power point (MPP) using the I-V and/or P-V Curve ; Repeating this analysis while incorporating the effect of Bypass

Diode; The following sequential steps form the proposed AFSA based PV- Array Analysis Algorithm.

1. Define the PV- Array and AFSA parameter;
2. Run the Algorithm 1 to estimate β
3. Run Algorithm 2 to determine Isc
4. Run Algorithm 3 to determine Voc
5. Run Algorithm 4 to generate the I-V Curve

6. Run Algorithm 5 to generate the P-V Curve
7. Run Algorithm 6 to determine the MPP
8. Run Algorithm 7 to generate the I-V Curve with inflection
9. Run Algorithm 8 to generate the P-V Curve with inflection
10. Run Algorithm 9 to determine the MPP under the effect of Bypass Diodes;

Figure 3.7 shows the flowchart for the Proposed AFSA based PV-Array Analysis Algorithm.

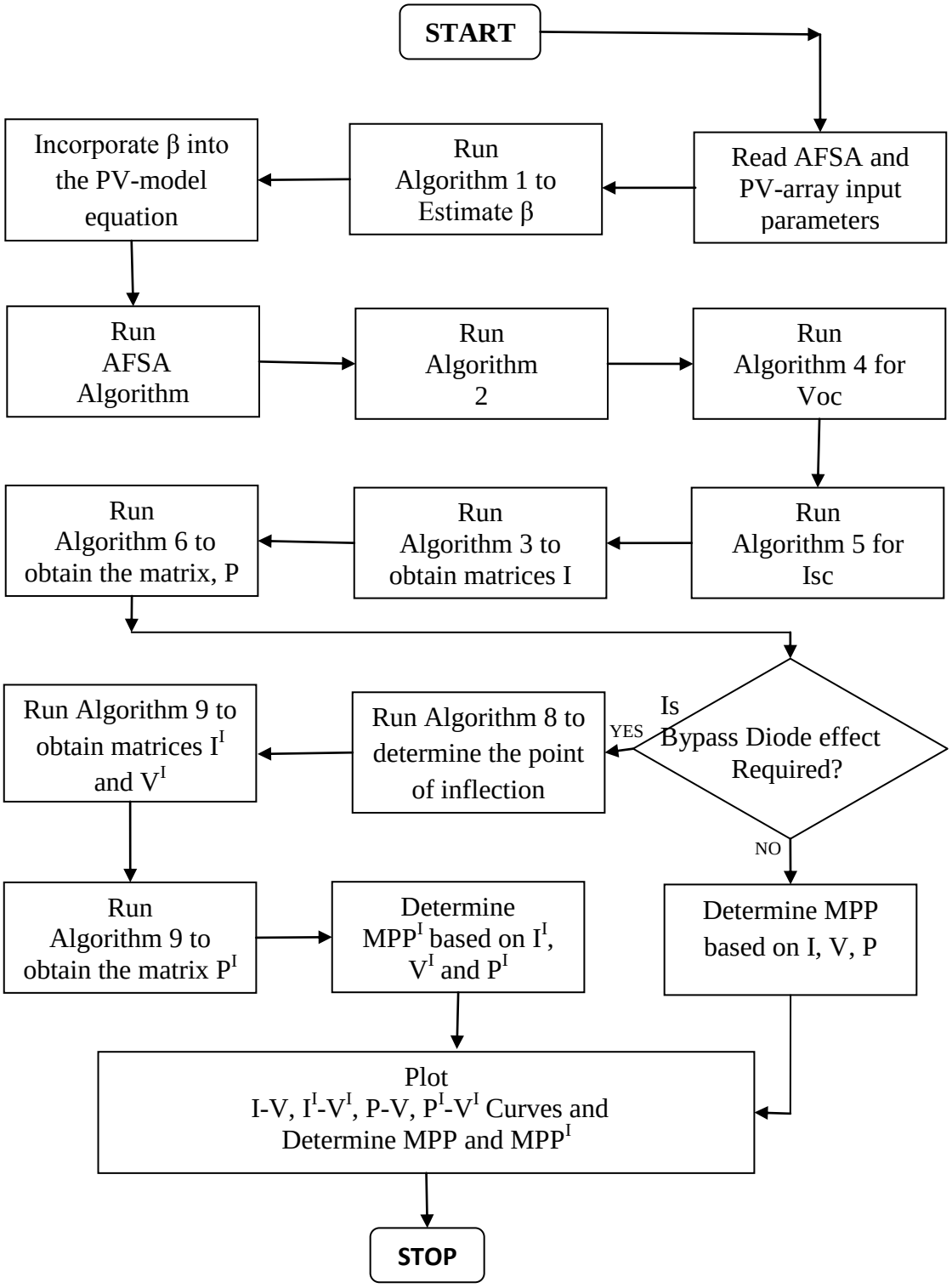


Figure 3.7: Flow Chart for the Proposed AFSA Based PV-Array Analysis Algorithm

3.14 Simulation

In order to demonstrate the effectiveness of the developed models and simulator, a PV-module having the parameters shown in Table 3.1 was chosen to be simulated using the AFSA based parameters shown on the same table. The entire simulation set-up comprises of various configurations of PV-module. It may be observed from Table 3.1 that the Visual Distance and the Step Size of the artificial fish are relatively low. These are two parameters that govern the accuracy of the developed AFSA based model. These values are chosen based on the output characteristics of the individual modules. The higher the values of V_{oc} and I_{sc} of the individual modules, the higher the values of VD and SS. In general higher values of SS result in low accuracy.

Table 3.1: The Choice of Module and AFSA Parameters

MODULE PARAMETER	**Value**
Diode Sat. Current	9.806821e-10
Cell Photo Current	3
Series Resistance (R_S)	1.605e-3
Parallel Resistance (R_P)	30
Diode Ideality Factor	1
AFSA PARAMETERS	**Value**
Number of Fish	10
Number of parameters	5
Visual Distance	0.1
Crowdness Factor	0.5
Step Size	0.01
Maximum Iteration	100
Number of Trials	5
Optimization Type	Minimization (1)

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Introduction

In this chapter, the developed Graphics User Interface (GUI) was used to simulate a number of scenarios of PV-module output characteristics. The results are presented and discussed. The effect of Bypass Diode (BD) on the output characteristics of the PV-modules is also simulated. The accuracy of the developed models was also demonstrated by estimating the output parameters of the PV-system presented in the work of El-Saady *et al.* (2013) and comparing the results. The effectiveness of the proposed AFSA based method was also compared with the conventional Dynamic Programming (DP) based approach.

4.2 Proposed Model Testing Results

In this stage, different degree of shading was analyzed. A typical PV-array data sheet information was also re-verified using the developed AFSA based simulator.

4.2.1 I-V Curves with Varying Degree of Shading

In Figure 4.1, the degree of shading was varied between 1 and 0 to obtain the output characteristics of a single module with the parameters shown in Table 3.1. It may be observed that as the degree of shading decreases, the Short Circuit Current (I_{SC}) decreases widely while the Open Circuit Voltage (V_{OC}) slightly decreases. The MPPt can be seen to vary slightly with decrease in shading parameter. In general, V_{OC} varies between 0.56 and 0.50 when β is varied between 1 and 0.1 while I_{SC} gradually decreases from 3 to 0.3, in figure 4.1.

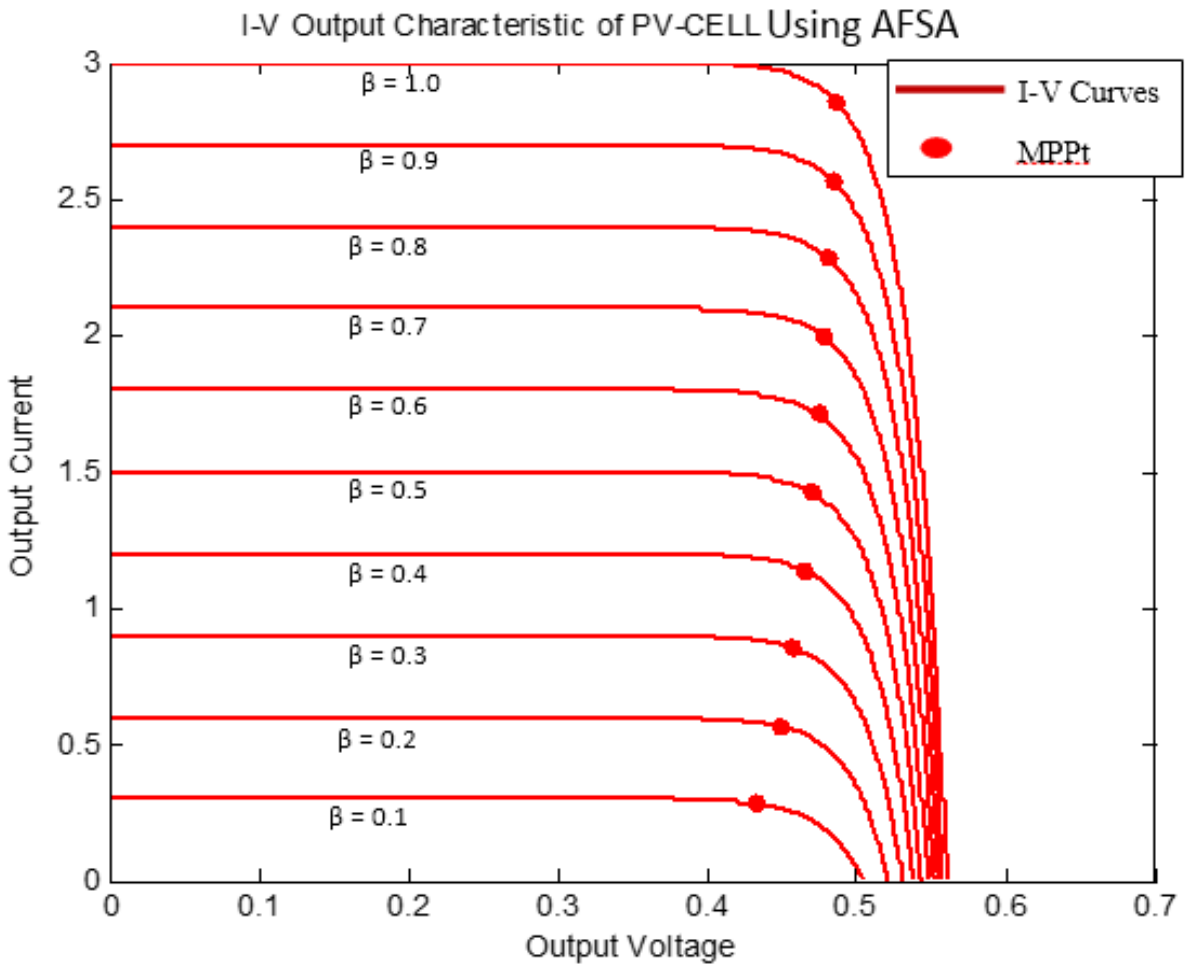


Figure 4.1**:** I-V Curve of a Single PV-Module with Varying Degrees of Shading

4.2.2 P-V Curves with Varying Degree of Shading

Figure 4.2 is a set of P-V curves for a single module with varying degree of shading. The degree of shading was varied between 1 and 0 to obtain the output characteristics of a single module with the parameters shown in Table 3.1. It may be observed that as the degree of shading decreases, the point corresponding to the peak of the curves falls. This results in decrease in the slope of the P-V curves. In general, the maximum power decreases with decrease in degree of shading. It may be observed that the P-V curves vary in the same pattern as the I-V curves. This may be associated with the fact that, the output

power of a PV-module is the product of the current (I) and voltage (V) at any given instance. This justifies the fact that the behaviour of a PV-module can be completely defined using its I-V curves.

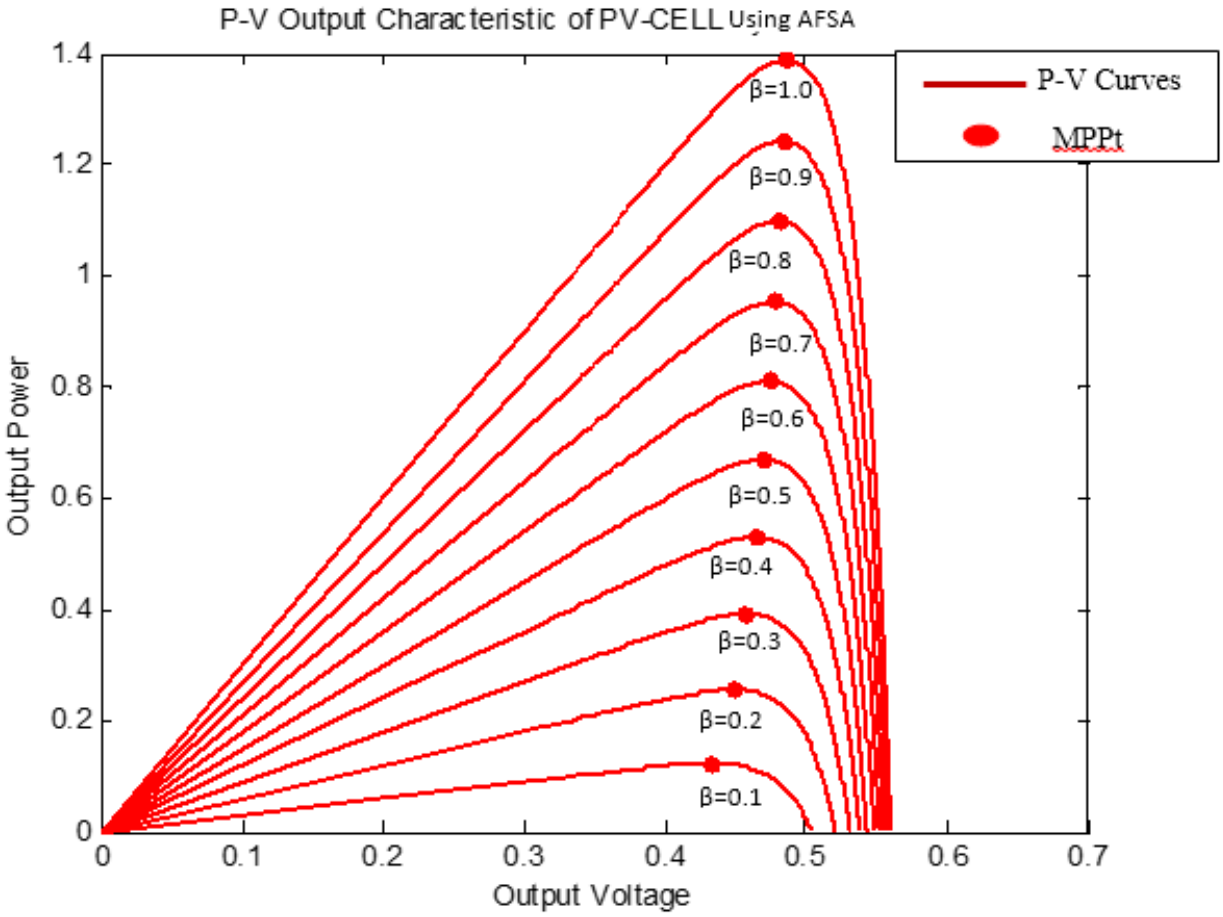


Figure 4.2**:** P-V Curves of a Single PV-Module with Varying Degree of Shading

4.2.3 Output Characteristics of PV-Array in Different Configuration

In order to demonstrate the effectiveness of the developed AFSA based PV-Array analysis simulator, an array of 54 individual modules each having the parameters shown in Table 3.1 was subjected to a number of configurations as shown in Table 4.1. N_S and N_P represent the number of series and parallel paths respectively. While the values of the output

parameters (I_{SC}, V_{OC}, P_{MAX}, I_{MPP} and V_{MPP}) for the resulting configuration and the computational time (CPU seconds) are also presented in table 4.1.

Table 4.1: Results for the Selected PV-Array Configuration (Cases 1 to 8)

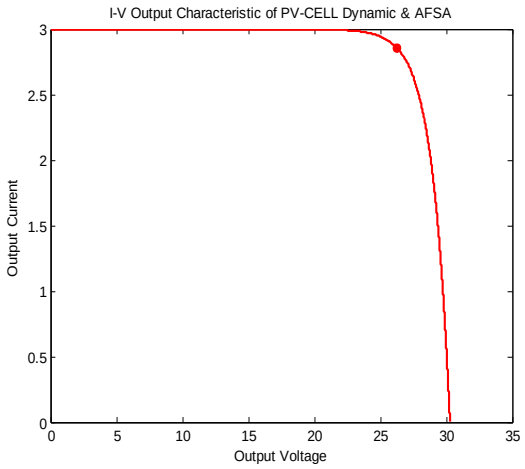
Case S/No.	Configuration		Output Characteristics					
	N_s	N_p	I_{sc}	V_{oc}	P_{max}	I_{mpp}	V_{mpp}	CPU (s)
1	54	1	3	30.2778	73.689	2.8329	26.0118	0.3751
2	27	2	6	15.1366	73.689	5.6658	13.0059	0.3685
3	18	3	9	10.0909	73.689	8.4987	8.6706	0.3676
4	9	6	18	5.0452	73.689	16.9974	4.3353	0.3701
5	6	9	27	3.3636	73.689	25.4962	2.8902	0.3791
6	3	18	54	1.6818	73.689	50.9923	1.4451	0.3882
7	2	27	81	1.1212	73.689	76.4885	0.9634	0.3899
8	1	54	162	0.5605	73.689	152.977	0.4817	0.3954

The data presented in Table 4.1 shows that the voltage (V_{OC} and V_{MPP}) are directly proportional to N_S while the currents (I_{SC} and I_{MPP}) depends on N_P. While the maximum output power (P_{MAX}) is independent of array configuration. This shows that the array configuration parameters (N_S and N_P) control the resulting output of the PV-array for specified maximum output power (P_{MAX}).

The developed simulator generates a number of curves in accordance with the output parameter values while analysing any given PV-array configuration. These curves include the following:

1. I – V Curves;
2. P – V Curves;
3. Optimization Curves for Current;
4. Optimization Curves for Voltage and
5. Optimization Curves for Power.

The curves for the configurations case 1 to 4 as shown in Table 4.1 are represented by Figures 4.3 to 4.14. The I-V and P-V characteristics curves for the configurations in cases 1 to 4 are shown in Figures (4.3, 4.5, 4.7 and 4.9) and (4.4, 4.6, 4.8 and 4.10) respectively; while their AFSA optimization curve for current, voltage and power are represented by (a), (b) and (c) of Figures 4.11 to 4.14 respectively.



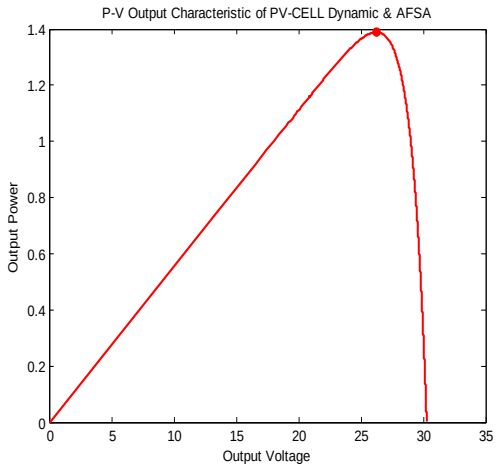
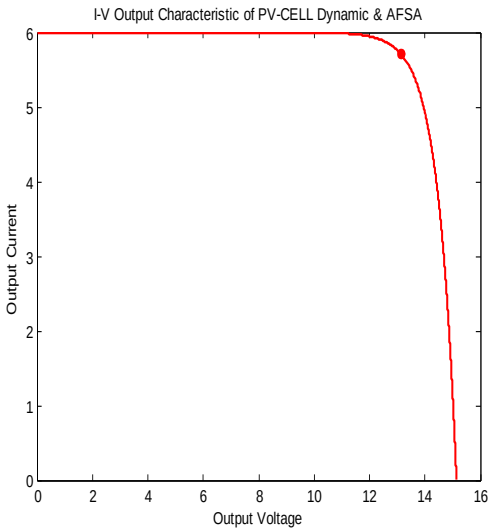


Figure4.3:I-V Curve for ConfigurationCase1 Figure4.4:P-V Curve for Configuration Case1



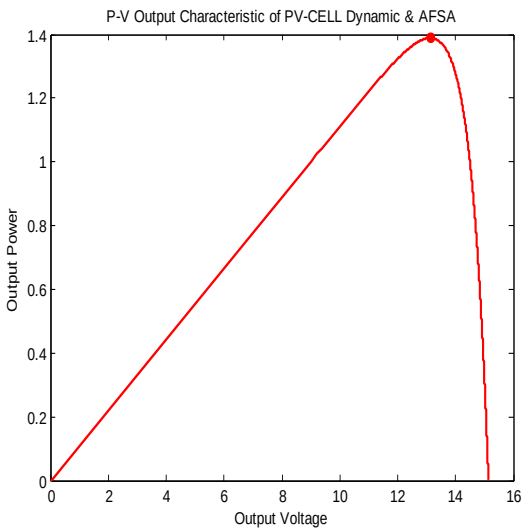
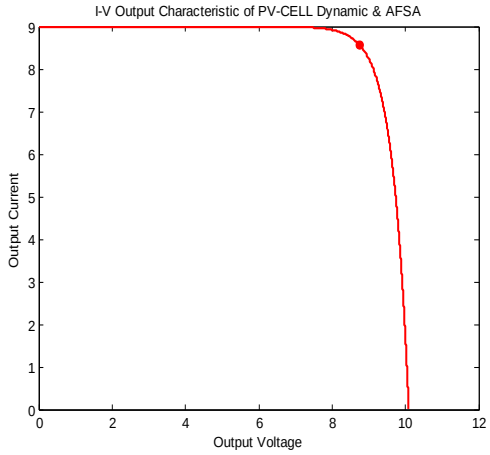


Figure4.5:I-V Curve for ConfigurationCase2 Figure4.6:P-V Curve for Configuration Case2



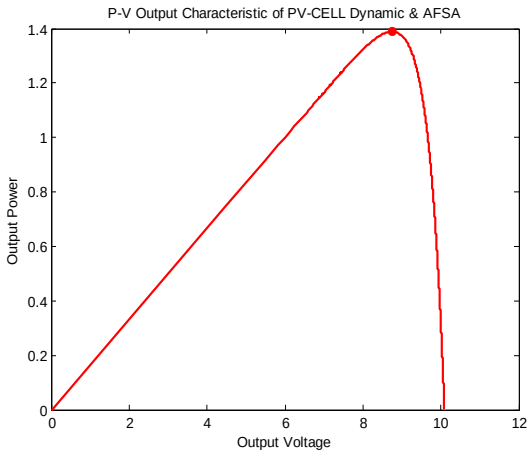
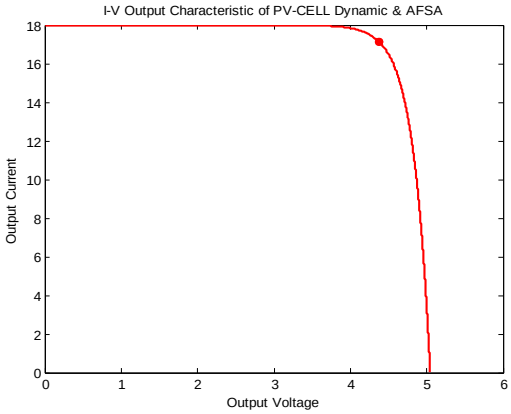


Figure4.7:I-VCurve for Configuration Case3 Figure4.8:P-V Curve for Configuration Case3



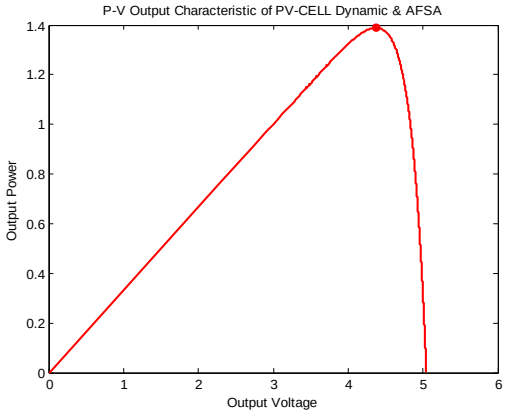
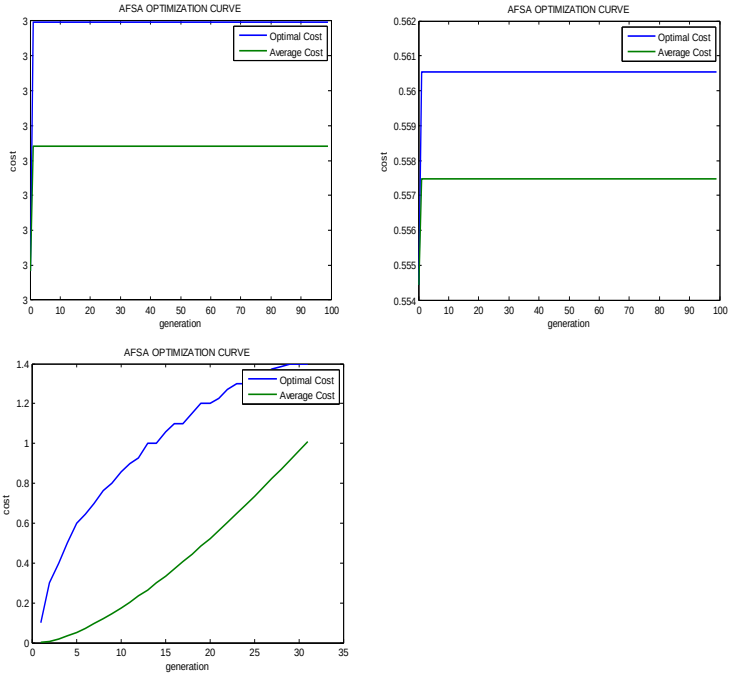
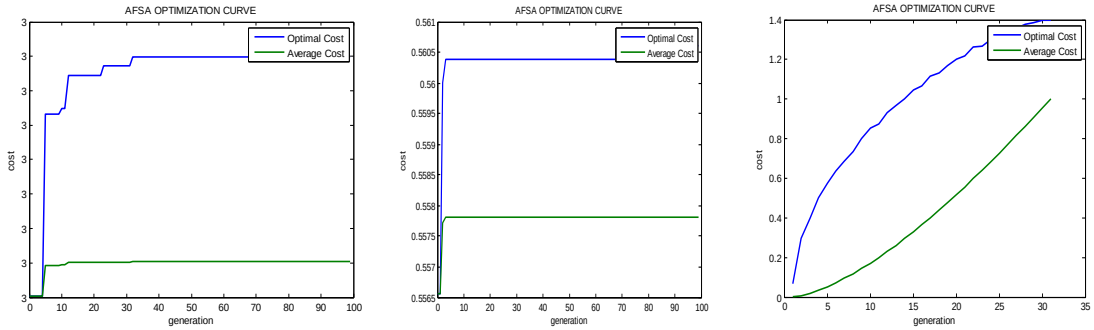


Figure4.9:I-VCurve for Configuration Case4Figure4.10:P-VCurve for Configuration Case4



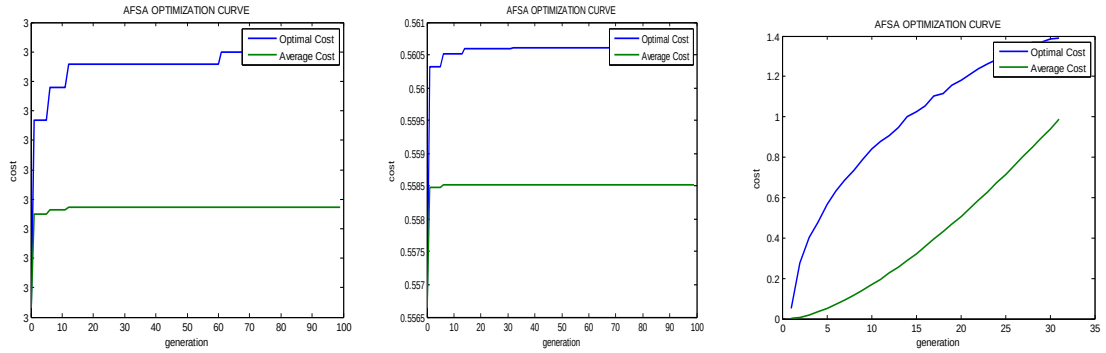
(a) (b) (c)

Figure4.11:AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 1)



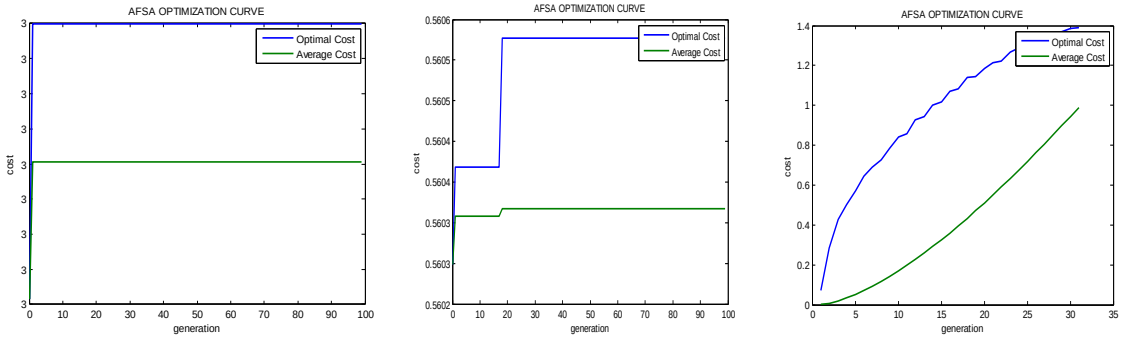
(a) (b) (c)

Figure4.12:AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 2)



(a) (b) (c)

Figure4.13:AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 3)



(a) (b) (c)

Figure4.14:AFSA Optimization Curves for Current, Voltage and Power (Configuration Case 4)

4.2.4 Effect of Bypass-Diodes on the Output Characteristics of PV-Array

Despite the numerous advantages of Bypass-Diodes (BD) to a PV-Array, it has disadvantages. Bypass Diodes affect the output characteristics of PV-Array (in the presence of non-uniform shading) by introducing distortions/inflections in the I-V and P-V curves. These inflections (Inflection Points, IPts.) results in multiple MPPts, with all except one MPPt been a true MPPt (all other MPPts being local MPPt). In order to demonstrate the effect of Bypass Diodes on PV-arrays, three case scenarios of PV-array configurations were

considered and the I-V and P-V curves of each case was generated. Table 4.2 shows the output characteristics of these 3 cases.

A. Case I: 2 Strings of 27 PV-modules

In this case, 54 PV-modules were assumed to be connected such that two parallel paths are formed, with each parallel path comprising of 27 individual PV-modules connected in series. A BD is placed between them. The configuration is then subjected to a non-uniform degree of shading of $\beta = 1$ (in the first path) and $\beta = 0.5$ (in the second path). This degree of shading is simply written as [1 0.5] for a two paths configuration. Note, each module has the same parameters as shown Table 3.1. Figure 4.15 shows the I-V curve of Case I while Figure 4.16 shows its P-V curve. The corresponding values for the Currents, Voltages and Powers at the Maximum Power Points (MPPt) and the Inflection Points (IPt) can also be found in Table 4.2.

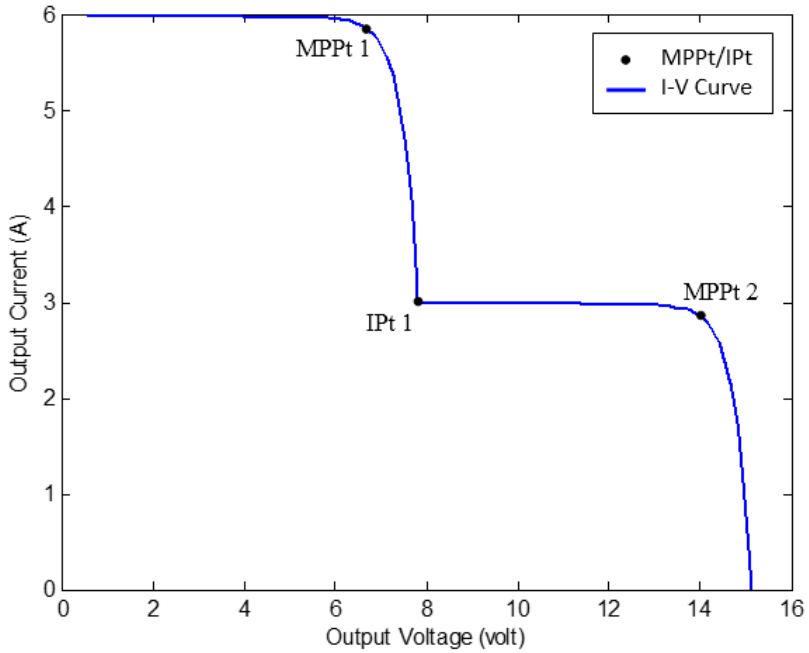


Figure 4.15: I – V Curve for Case 1

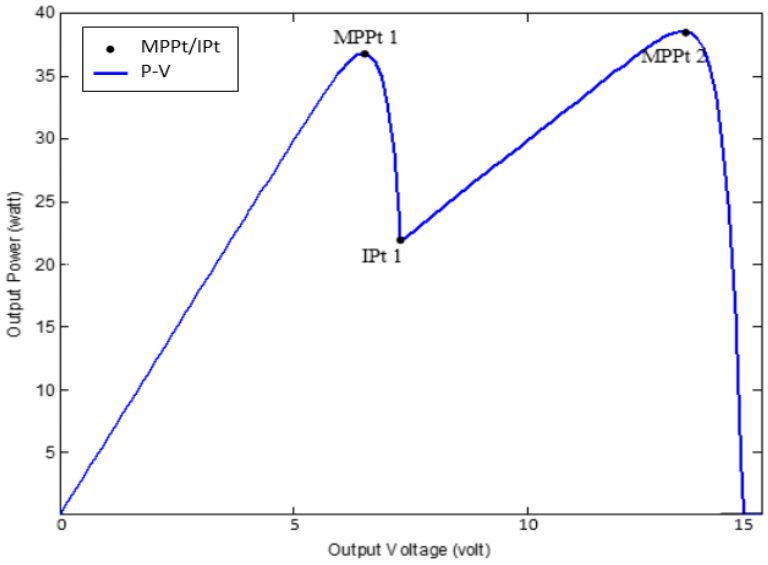


Figure 4.16: P – V Curve for Case 1

B. Case 2: 3 Strings of 18 PV-modules

In this case, 54 PV-modules were assumed to be connected such that three parallel paths are formed, with each parallel path comprising of 18 individual PV-modules connected in series. A BD is placed between them. The configuration is then subjected to a non-uniform degree of shading of $\beta = 1$ (in the first path); $\beta = 0.67$ (in the second path) and $\beta = 0.19$ (In the third path). This degree of shading is simply written as [1 0.67 0.19] for a three paths configuration. Note, each module has the same parameters as shown in Table 3.1. Figure 4.17 shows the I-V curve of Case 2 while Figure 4.18 shows its P-V curve. The corresponding values for the Currents, Voltages and Powers at the Maximum Power Points (MPPt) and the Inflection Points (IPt) can also be found in Table 4.2.

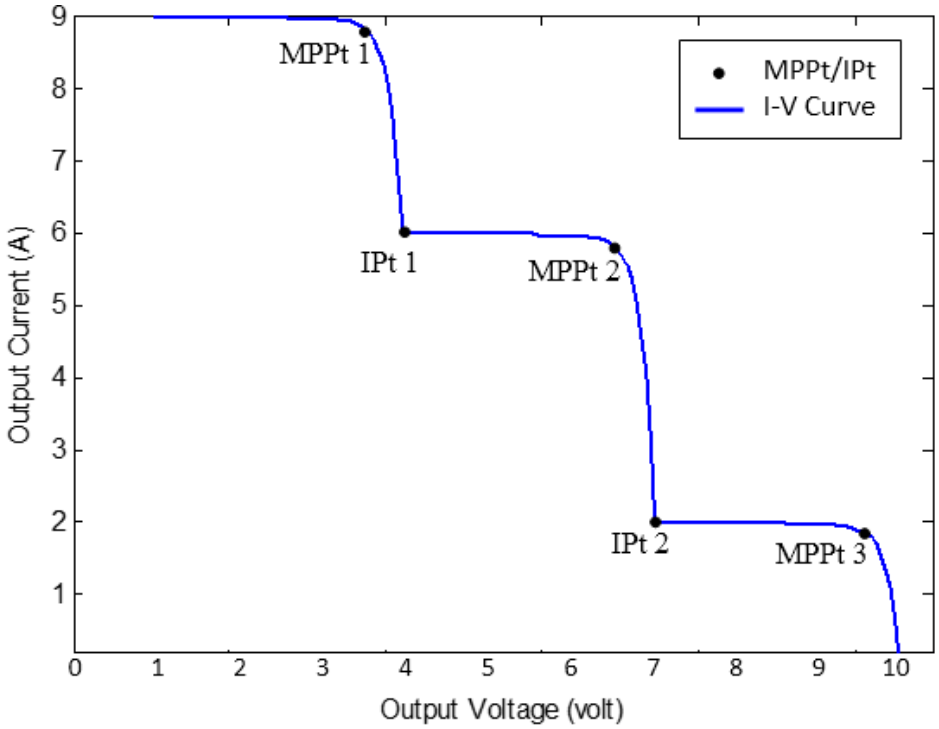


Figure 4.17: I – V Curve for Case 2

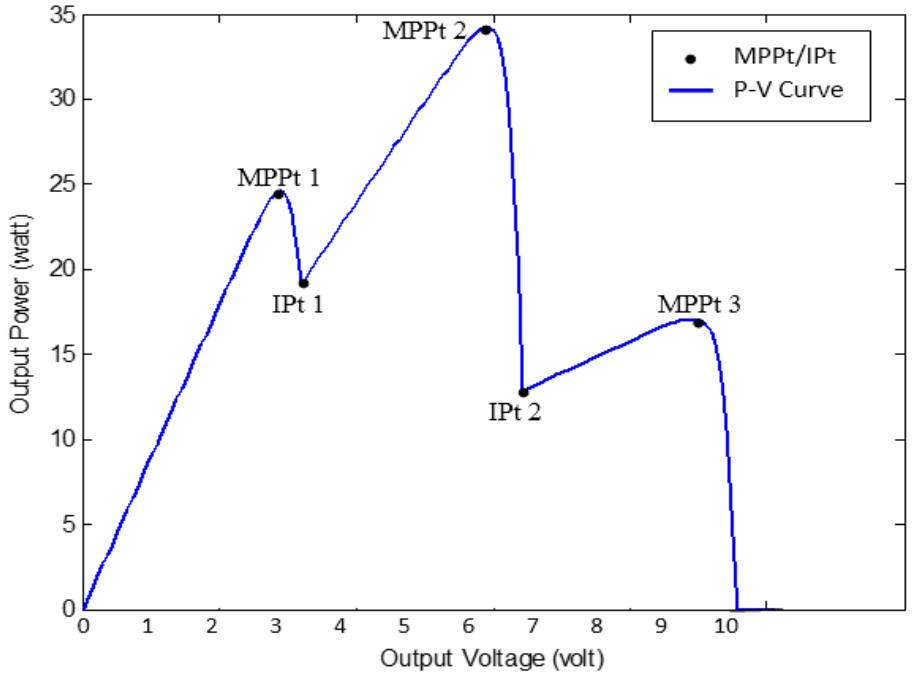


Figure 4.18: P – V Curve for Case 2

C. Case 3: 6 Strings of 9 PV-modules

In this case, 54 PV-modules were assumed to be connected such that six parallel paths are formed, with each parallel path comprising of 9 individual PV-modules connected in series. A BD is placed between them. The configuration is then subjected to a non-uniform degree of shading of $\beta = 1$ (in the first path); $\beta = 0.85$ (in the second path); $\beta = 0.56$ (In the third path); $\beta = 0.28$ (In the fourth path); $\beta = 0.09$ (In the fifth path) and $\beta = 0.04$ (In the six path). This degree of shading is simply written as [1 0.85 0.56 0.28 0.09 0.04] for a six paths configuration. Note, each module has the same parameters as shown in Table 3.1. Figure 4.19 shows the I-V curve of Case 3 while Figure 4.20 shows its P-V curve. The corresponding values for the Currents, Voltages and Powers at the Maximum Power Points (MPPt) and the Inflection Points (IPt) can also be found in Table 4.2.

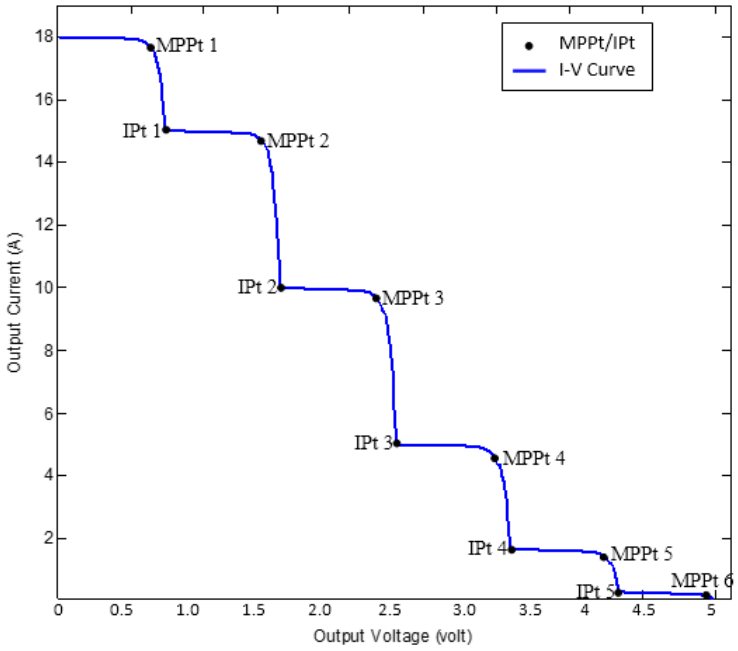


Figure 4.19: I – V Curve for Case 3

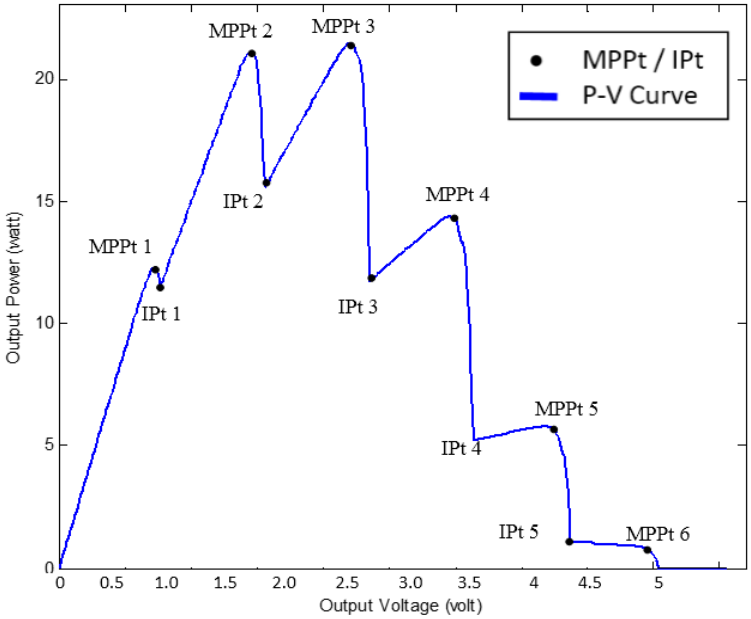


Figure 4.20: P – V Curve for Case 3

Generally, for a PV system with no bypass diodes subjected to uniform degree of shading, the power output of the system of PV-Array is constant regardless of the array configuration. While current and voltage are inversely proportional to each other. This may be noted in Table 4.1. Whereas, for a PV system with bypass diodes subjected to non-uniform degree of shading, the power output of the system of PV-Array varies with array configuration, and I-V and P-V curves of such system has multiple MPPts and IPts. Details of this statement may be confirmed from Table 4.2.

Table 4.2: Output Characteristics of the three cases Scenario of PV-Array with Bypass Diodes.

S/N.	**Scenario Case 1**	**V_{OC} 15.1366**	**I_{SC} 6**	**MPPt (true) 38.573**		
	I_{MPP}	**V_{MPP}**	**P_{MPP}**	**I_{IP}**	**V_{IP}**	**P_{IP}**
1	5.812	6.322	36.743	3	7.568	22.705
2	2.886	13.365	38.573			
S/N.	**Scenario**	**V_{OC}**	**I_{SC}**	**MPPt (true)**		

	Case 2	10.0909	9	34.264		
	I_{MPP}	V_{MPP}	P_{MPP}	I_{IP}	V_{IP}	P_{IP}
1	8.723	3.247	28.324	6	3.712	22.272
2	5.827	5.88	34.264	1.705	6.893	11.753
3	1.586	9.621	15.259			
S/N.	**Scenario Case 3**	**V_{OC} 5.0452**	**I_{SC} 18**	**MPPt (true) `21.478**		
	I_{MPP}	V_{MPP}	P_{MPP}	I_{IP}	V_{IP}	P_{IP}
1	17.498	0.712	12.459	15.226	0.812	12.364
2	14.892	1.424	21.206	10.075	1.524	15.354
3	9.741	2.205	21.478	4.960	2.305	11.433
4	4.626	3.267	15.113	1.663	3.367	5.600
5	1.329	4.198	5.579	0.761	4.298	3.272
6	0.427	4.923	2.102			

4.2.5 Comparison Case A

The data sheet of a KC200GT solar array parameters at $25^{O}C$, generating 1000W/m2 as presented in the work of El-Saady *et al.* (2013) was considered in order to demonstrate the accuracy of the developed AFSA based models. The result from the AFSA based model is compared with the KC200GT solar array data sheet information as shown in Table 4.3.

Table 4.3: Comparison of the KC200GT Solar Array Data Sheet Information and the Results of the Developed AFSA based Model.

S/N.	KC200GT solar array data sheet information		KC200GT solar array AFSA based Model	
1	I_{MPP}	7.61A	I_{MPP}	7.634A
2	V_{MPP}	26.3V	V_{MPP}	26.276V
3	P_{MAX}	200.143W	P_{MAX}	200.362W
4	I_{SC}	8.21A	I_{SC}	8.21A
5	V_{OC}	32.9V	V_{OC}	33.007V
6	K_V	-1230mV/^{O}C	I_{SAT}	9.825E-8A
7	K_I	3.2mA/^{O}C	I_{PHOTO}	8.21A
8	N	54	R_P	591.1123

9	A	1	R_S	0.2277E-3

It may be observed that a little error exists between the proposed AFSA based model and the original data sheet information of a KC200GT solar array. This error may have resulted from truncation during computation.

4.2.6 Comparison Case B

The data sheet of a BP SolarMSX-60solar array Parameters at $25^{O}C$, generating 1000W/m2 as presented in the work of El-Saady *et al.* (2013) was considered in order to demonstrate the accuracy of the developed AFSA based model as compared to the DP based approach. The result from the AFSA and DP based model are compared with the BP Solar MSX-60 solar array data sheet information as shown in Table 4.4

Table 4.4: Comparison of the BP SolarMSX-60 Solar Array Data Sheet Information and the Results of the Developed AFSA and DP based Models.

S/N.	**BP SolarMSX-60 solar array data sheet information**		**BP SolarMSX-60 solar array AFSA based Model**		**BP SolarMSX-60 solar array DP based Model**	
1	I_{MPP}	3.5A	I_{MPP}	3.489A	I_{MPP}	3.5723A
2	V_{MPP}	17.1V	V_{MPP}	17.1196V	V_{MPP}	17.621V
3	P_{MAX}	59.83W	P_{MAX}	59.7303W	P_{MAX}	62.948W
4	I_{SC}	3.8A	I_{SC}	3.8A	I_{SC}	3.7999A
5	V_{OC}	21.1V	V_{OC}	21.0897V	V_{OC}	21.0423V
6	K_V	80m $V/^{O}C$	I_{SAT}	9.825E-8A	I_{SAT}	9.825E-8A
7	K_I	3 $A/^{O}C$	I_{PHOTO}	8.21A	I_{PHOTO}	8.21A
8	N	36	R_P	591.1123	R_P	591.1123
9	a	1	R_S	0.2277E-3	R_S	0.2277E-3

Finally, the developed AFSA based model can be said to possess little computation error that accumulates with increase in number of tracing points due to truncation, but had shown a considerable superiority over the conventional Dynamic Programming (DP) based approach. This can also be noticed from Table 4.4.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this work, an improved PV-array model has been developed by incorporating a shading parameter into the conventional PV-array model. A set of algorithms have been developed for the various steps involved in PV-array analysis. AFSA based approach has been developed and incorporated into a MATLAB based GUI softwareto aid future researchers in replicating the proposed research work and verify its validity. A DP based approach has also been incorporated into the same GUI to aid comparative analysis between the AFSA and the DP based approach. Several scenarios of PV-array analysis have been simulated and presented in Chapter 4 for demonstration. The developed AFSA based approach has

shown to be efficient in tracing the output characteristic curves of PV systems and in generating the design parameters for solar modules used in solar energy conversion applications. Two typical data sheets information (KC200GT and BP solarMSX-60) have been used to compare the accuracy of the developed AFSA based model and the conventional DP approach. Finally, the developed AFSA based approach has been proven to be effective in curve tracing application and non-linear equation solving.

5.2 Significant Contributions

This research work offers the following significant contributions to the existing body of knowledge:

1. Development of a Matlab-based AFSA and DP graphical user interface (GUI) improved PV-array model to aid an accurate shading effect analysis across individual cell/modules of any given PV-system.
2. The AFSA based approach was validated on three performance metrics. All the metrics (maximum power point, short-circuit current and open-circuit voltage) outperformed the DP based approach by 5.1%, 0.03% and 0.22% respectively on the BP Solar MSX-60 array datasheet information.

5.3 Recommendation

The following are some useful recommendations that may serve as guide to the advancement of this research work:

1. The proposed AFSA-based model could be extended to cover the effect of temperature, module type, and other physical parameters;

2. The AFSA based model can be improved for use in on-line practical maximum power point tracking by embedding it into a micro-chip to boost inverter output performance in balancing the maximum power generated with load requirement;
3. The DP and AFSA based model can be hybridized to form a more robust approach that could be integrated into solar/wind farm control and use in solar power plant output capability enhancement.

5.4 Limitation

The following limitations were encountered in the course of this work:

1. Keeping track with some of the partial solutions in dynamic programming in designing the graphical user interphase (GUI) was a challenge.
2. The accuracy of the AFSA based model results were limited due to some degree of truncation error in the convergence process.

REFERENCES

Adelmo, O. C., Francisco, J. G. S., Juan, M., & Andrea, S. G. (2014). A review of diode an solar cell equivalent circuit model lumped parameter extraction procedures. *IEEE* FACTA UNIVERSITY Series Electronic & Energies 27(1) 57-102

Alsayid, B., Alsadi, S. Y., Jallad, J. f. S., & Dradi, M. H. (2013a). Partial Shading of PV System Simulation with Experimental Results. *scientific research*,4(6) 429-435.

Alsayid, B., Alsadi, S. Y., Jallad, J. f. S., & Dradi, M. H. (2013b). Partial Shading of PV System Simulation with Experimental Results. *scientific research, 64*(27), 429-435.

Andre Mermoud, Bruno Wittmer, & Michel Villoz. (2012). *User's Guide PVsyst Contextual Help*

Arikatla, V., & Abu-Qahouq, J. A. (2010). An Adaptive Digital PID controller scheme for power converters. *Energy Conversion Congress and Exposition (ECCE), 21*(76), 223-227.

Basri, Y. E., Petibon, S., Estibals, B., & Alonso, C. (2010). New P&O MPPT Algorithm for FPGA Implementation *IEEE Transction on energy conversion*(43), 2868-2873.

Bayrak, G., & Cebec, M. (2013). Monitoring A Grid Connected PV Power Generation System with Labview. *IEEE Transcation on industrial electronics*(45), 1234-1340.

Briggs, D. (2012). Shade Impact:How Solar Systems Handle Sub-optimal Conditions.EAphcsc Energy.

Brito, M. A. G., Junior, L. G., Sampaio, L. P., & Canesin, C. A. (2011). Evaluation of MPPT techniques for photovoltaic applications. *IEEE International symposium on industrial electronics*(ISIE), 1039-1044.

Carrero, C., Rodriguez, C., Ramirez, D., & Platero, C. (2010). Simple Estimation of PV modules loss resistances for low error modelling. *Renewable Energy*(35), 1103-1108.

Chicco, G., Schlabbach, J., & Spertino, F. (2009). Experimental assessment of the waveform distortion in grid-connected photovoltaic installations. *Solar Energy, 83*(2009), 1026-1039. doi: 10.1016/j.solener.2009.01.005

Christopher, I. W., & Ramesh, R. (2013). Comparative Study of P&O and InC MPPT Algorithms. *American Journal of Engineering Research (AJER), 2*(12), 402-408

.

Cubas, J., Pindado, S., & Victoria, M. (2014). On theAnalytical approach for modelling photovoltaic systems behaviour.Journal of Power Sources, 467-474.

Diaf, S., Notton, G., & Stoyanov, L. (2011). Hybrid photovoltaic/wind energy systems for remote location. *Energy Procedia, 12*(26), 666-677.

Dio, V. D., Cascia, D. L., Miceli, R., & Rando, C. (2009). A Mathematical Model to Determine the Electrical Energy Production in Photovoltaic Fields Under Mismatch Effect.*clean Electrical power,International Conference. (46-51).*

Dumitras, A., & Moschytz, G. (2007). Understanding Fuzzy Logic-An interview with Lofti Zadeh. *IEEE Signal Processing Magazine, 48*(37), 865-913.

Durischa, W., Bitnara, B., Mayora, J. C., Kiessa, H., Lamb, K.-h., & Closea, J. (2007). Efficiency model for photovoltaic modules and demonstration of its application to energy yield estimation. *solar energy materialas and solar cells, 91*(2007), 79-84. doi: 10.1016/j.solmat.2006.05.011

El-Saady, G., El-Nobi Ibrahim, A., & El-Hendawi, M. (2013). Simulated Annealing Modelling and Analog MPPT Simulation for Standalone Photovoltaic Arrays. *International Journal on Power Engineering and Energy(IJPEE), 4*(41), 353-360

Faranda, R., Leva, S., & Maugeri, V. (2008). *MPPT techniques for PV Systems: Energetic and cost comparison*. Paper presented at the 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century.

Farret, F. A., & Simoes, M. G. (2006). integration of alternative sources of Energy,(book). *IEEE Press and John Wiley and Sons,Inc,*.

Firth, S. K., Lomas, K. J., & Rees, S. J. (2010). A simple model of PV system performance and its use in fault detection. *Solar Energy, 84*, 624-635. doi: 10.1016/j.solener.2009.08.004

Gabriele, G., Claudio, R., Darko, O., & Domenico, C. (2009). A new multilevel conversion structure for grid-connected PV applications. *IEEE Transactions on Industrial Electronics* 56(11), 4416-4426.

Haeberlin, H. (2007). *Damages at Bypass Diodes by Induced Voltages and Currents in PV modules Caused by Nearby Lightning Currents*. Paper presented at the 22nd European Photovoltaic Solar Energy Conference, Milano, Italy.

Hawkes, A. D., & Leach, M. A. (2009). Modelling high level system design and unit commitment for a microgrid. *Applied energy, 86*(2009), 1253–1265. doi: 10.1016/j.apenergy.2008.09.006

Huld, T., Gottschalg, R., Beyer, H. G., & Topic, M. (2010). Mapping the performance of PV modules, effects of module type and data averaging. *Solar Energy, 84*(2010), 324-338.

Isaloo, B. A., & Amiri, P. (2016). Improved Variable Step Size Incremental Conductance Mppt Method with High Convergence Speed for PV Systems. *Journal of Engineering Science and Technology, 11*(4), 516-528.

Ishaque, K., Salam, Z., & Taheri, H. (2011). Acurate MATLAB Simulink PV system Simulator based on a two-diode model. *Power Electronics, 11*(4), 1723-1734.

Jayakumar, P. (2009). *Solar Energy Resource Assement Book*.

Jiang, Y., Qahouq, J. A. A., & Orabi, M. (2011). *MATLAB/Pspice Hybrid Simulation modelling of Solar PV Cell/Module*. Paper presented at the IEEE Transactions on Energy Conversion, Applied Power Electronics Conference and Exposition (APEC), 2011 Twenty-Sixth Annual IEEE.

Joseph, M. E. A., & Kamala, J. (2013). PV Array Characteristics Analysis Under Partial Shading & Modelling of P&O MPPT Applied Boost Convertor Using MATLAB/Simulink. *Solar Energy Materials & Solar Cells*(41), 596-601.

Kelly, N. A., & Gibson, T. L. (2009). Improved photovoltaic energy output for cloudy conditions with a solar tracking system. *Solar Energy, 83*, 2092-2102. doi: 10.1016/j.solener.2009.08.009

Keyrouz, F., Hamad, M., & Georges, S. a. (2013). A novel unified maximum power point tracker for controlling a hybrid wind-solar and fuel-cell system. *Renewable Energies*, *78*(19), 405-412.

Khalaf, Y., Ibraheem, O., Adil, M., Mohammed, S., Qasim, M., & Waleed, K. (2014). Maximum Power Point Evaluation of Photovoltaic Modules Under Shading Effect. *European Scientific Journal*, *10*, 451-473.

King, D., Hansen, B., Kratochvil, V., & Quintana, M. (2007). Dark current-voltage measurements on photovoltaic modules as a diagnostic or manufacturing tool, Paper presented at the Photovoltaic Specialists Conference. *11*(23), 795-811.

Lagorse, J., Paire, D., & Miraoui, A. (2010). A multi-agent system for energy management of distributed power sources. *Renewable Energy*, *35*(2010), 174–182. doi: 10.1016/j.renene.2009.02.029

Lee Seok-Ju, Park Hae-Yong, Kim Gyeong-Hun, Seo Hyo-Ryong, Hasan Ali Mohd, Minwon Park, & Yu., I.-k. (2007, oct.). *The experimental analysis of the gridconnected PV system applied by POS MPPT*, Seoul, Korea.

Lei, P., Li, Y., & Seem, J. E. (2011). Sequential ESC-Based Global MPPT Control for Photovoltaic ArrayWith Variable Shading. *IEEE Transaction indusrial electronics*, *2*(64), 675-684.

Liu, Y. H., Huang, S. C., Huang, J. W., & Liang, W. C. (2012). A particle swarm based-maximum power point tracking algorithm for PV systems operating under partially shaded conditions. *IEEE Transction on energy conversion*, *27*. doi: 10.1109/TEC.2012.2219533

lshaque, K., Salam, Z., & Taheri, H. (2011). Accurate MATLAB simulink PV system Simulator based on a two-diode model. *Power Electronics*, *11*(4), 1723-1734.

Macabebe, E. Q. B., & Dyk, E. v. (2008). Extraction of device parameters from dark current-voltage characteristics of PV devices. *IEEE Transaction on energy conversion*, *5*(34), 616–619.

Mahor, A., Khan, A., & Soni, M. (2013). Development and Simulation of Solar Photovoltaic model using MATLAB/simulink. *International Journal on Emerging Technologies*, *4*(1), 62-65.

Mäki, A., Valkealahti, S., & Leppäaho, J. (2011). Operation of series-connected silicon-based photovoltaic modules under partial shading conditions. *Wiley Online Library (wileyonlinelibrary.com)*, *20*, 298-309. doi: 10.1002/pip.1138

Mehdi, N., Ali, A., Ghodrat, S., Mehdi, S., & Adel Najaran, T. (2012). A Review Of Artificial Fish Swarm Optimization Methods and Applications. *International Journal on Smart Sensing and Intelligent Systems, 5*(1). doi: 10.1007-10462-012

Mekhilef, S., Saidur, R., & Kamalisarvestani, M. (2012). Effect of dust, humidity and air velocity on efficiency of photovoltaic cells. *lEEE Transaction on industrial electronics, 7*(54), 2920–2925.

Morais, H., Kadar, P., Faria, P., Vale, Z. A., & Khodr, H. M. (2010). Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renewable Energy, 35*, 151-156. doi: 10.1016/j.renene.2009.02.031

Nabulsi, A. A., & Dhaouadi, R. (2012). Efficiency optimization of a DSP-based standalone PV system using fuzzy logic and dual-MPPT control. *IEEE Transactions on Industrial Electronics, 8*(23), 573-584.

Ngan, M. S., & Tan, C. W. (2011). *Multiple Peak Tracking Algorithm Using Particle Swarm Optimization Incorporated with Artificial Neural Network.* Paper presented at the World Academy of Science, Engineering and Technology.

Nguyen, D. D., Lehman, B., & Kamarth, S. (2009). Performance Evaluation of Solar Photovoltaic Arrays Including Shadow Effects Using Neural Network. *IEEE Transaction on energy conversion, 12*(67), 3357-3362.

Orioli, A., & Di Gangi, A. (2013). A procedure to calculate the five-parameter model of crystalline silicon photovoltaic modules on the basis of tabular performance data. *Applied energy, 45*(102), 1160-1174.

Pachpande, S. G., & Zope, P. H. (2012). Studying the effect of shading on Solar Panel using MATLAB. *International Journal of Science and Applied Information Technology Available 1(2) 46-51*

Paraskevadaki, E. V., & Papathanassiou, S. A. (2011). Evaluation of MPP Voltage and Power of mc-Si PV Modules in Partial Shading Conditions. *IEEE Transactions on Energy Conversion, 26*(45), 923-932.

Pareek, S., & Dahiya, R. (2015). Simulation and Performance Analysis of Individual Module to Address Partial Shading cum Parameter Variation in Large Photovoltaic Fields. *journal of energy and power sources, 2*(45), 99-104.

Parida, B., Iniyan, S., & Goic, R. (2013). A review of solar photovoltaic technologies. *Renewable and sustainable energy reviews, 15*(2011), 1625-1636. doi: 10.1016/j.rser.2010.11.032

Peng, L., Sun, Y., & Meng, Z. (2013). A new method for determining the characteristics for solar cells. *Power sources*, 129-137.

Platzer, M. D. (2012). U.S Solar Photovoltaic Manufacturing : Industry Trends,Global Competition, Federal Support. *Congressional Research Services*(45), 1-32.

Quesada, V. G., Gispert, G. F., Lopez, P. R., Lumberas, R. M., & Rosa, C. A. (2009). Electrical PV array reconfiguration strategy for energy extraction improvement in grid-connected PV systems. *IEEE Transaction on Industrial Electronics, 6*(42), 65-74.

Ramana, V. V., & Jena, D. (2015). An Accurate Modelling of Photovoltaic System for Uniform and Non-Uniform Irradiance. *International Journal Renewable Energy Research, 5*(1), 234-241.

Rodriguez, B., Ramos, R., & Franco, C. Y. (2012). Modelling and parameter calculation of photovoltaic fields in irregular weather conditions. *Articulo de investigacion, 17*(75), 37-48.

Rui, C. L., Hui, T. C., Li, L. Y., & Shin, L. Y. (2010). A biological swarm chasing algorithm for tracking the PV maximum power point. *IEEE Transactions on Energy Conversion*(25), 437-446.

Sai, P. H. V. S. T., Rao, J. V. R. R., Devarayapalli, K. C., & Sharma, K. C. (2013). Modelling and Simulation of Solar Module under Non-Uniform Conditions and Performance Evaluation of TiO2 Coated Cell. *IEEE Transaction on energy conversion, 2*(11).

Said, S., Massoud, A., Benammar, M., & Ahmed, S. (2012). A matlab/simulink-based photovoltaic array model employing simpower systems toolbox. *Journal of Energy and Power Engineering, 6*, 1965-1975.

Salah, C. B., & Ouali, M. (2011). Comparison of fuzzy logic and neural network in maximum power point tracker for PV systems. *Electric power systems research, 81*(2011), 43-50. doi: 10.1016/j.epsr.2010.07.005

Salam, Z., Ahmed, J., & Merugu, B. S. (2013). The application of soft computing methods for MPPT of PV system. *Applied energy, 107*(2013), 135-148.

Saloux, E., Teyssedou, A., & Sorin, M. (2011). Explicit model of photovoltaic panels to determine voltages and currents at the maximum power point. *Solar Energy, 85*(5), 713–722. doi: 10.1016/j.solener.2010.12.022

Seyedmahmoudian, M., Mekhile, S., Rahmani, R., Yusof, R., & Renani, E. T. (2013). Analytical Modeling of Partially Shaded Photovoltaic Systems. *energies*, 128-144.

Sikiru, T. H. (2007). modelling and performance studies of photovoltaic Array-power converter system.

Singh, A. K., & Tripathi, R. K. (2012). MPPT Scheme for Small/Large Time Partially Shaded Condition of a PV Cell. *IEEE Transcation on industrial electronics, 45*(117), 1345-1356.

Theocharis, A. D., Charalampakos, V. P., Drosopoulos, A., & Milias-Argitis, J. (2012). Equivalent circuit of photovoltaic generator using newton-raphson algorithm. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*

Thongam, D. M., & Jaiswal, N. (2015). Analysis of PV Array Solar Energy Using Advanced Hill Climbing Controller. *International Journal of Science, Engineering and Technology Research (IJSETR), 4*(4).

Tijani, S. A., & Mua'zu, M. (2015). *Stabilization of inverted pendulum system using intelligent Linear Quadratic Regulator controller.* Paper presented at the 2015 7th International Joint Conference on Computational Intelligence (IJCCI).

Villalva, M. G., & Gazoli, J. R. (2009). Comprehensive approach to modelling and simulation of photovoltaic arrays. *IEEE Transactions on power electronics, 24*(8), 1198-1208.

Vongkoon, P., & Liutanakul, P. (2012). Digital R-S-T Controller for Current Loop Control of DC/DC Buck Converter: A Photovoltaic (PV) Array Simulator under Partial Shading Condition. *IEE Proc-Electrical Power Application* 37(3), 712-723.

wang, J. C., Su, Y. L., Kuo, K. C., Shieh, J. C., & Jiang, J. A. (2012). *A novel multipoint direct-estimation method for the maximum power point tracking of photovoltaic modules under partially shaded irradiation conditions*. Paper presented at the 2nd IEEE Energycon Conference & Exhibition, 2012 / Advances in Energy Conversion Symp Berlin,Germany.

Wang, Y., Lin, X., Kim, Y., Chang, N., & Pedram, M. (2013). Architecture and Control Algorithms for Combating Partial Shading in Photovoltaic Systems. *IEEE Transactionon computer aided design of integrated circuit,33(6) 917-930.*

Wu, Y., Gao, X. Z., & Zenger, K. (2011). *Knowledge-based Artificial Fish-Swarm algorithm*. Paper presented at the Paper presented at the 18th IFAC World Congress, milano.

Xu, J., Zhang, Y., & Wang, p. (2014). A kind of turbine fault diagnosis based on fish-swarm algorithm. *Energy Education Science and Technology Part A: Energy Science and Research, 32*(2), 1325-13330.

Yu, L., & Li, C. (2014). A global artificial fish swarm algorithm for structural damage detection. *Advances in Structural Engineering, 17*(3), 331-346. doi: 10.1260/1369-4332.17.3.331

Zainudin, H. N., & Mekhilef, S. (2010). Comparison Study of Maximum Power Point Tracker Techniques for PV Systems.

Zeng, C., & Xu, H. (2012). *PID controller parameters optimization based on artificial fish swarm algorithm.* Paper presented at the Paper presented at the 2012 5th International Conference on Intelligent Computation Technology and Automation, ICICTA 2012, Zhangjiajie, Hunan.

Zhu, W., & Lin, M. (2013). Weak signal intelligent detection system based on stochastic resonance and artificial fish swarm algorithm. *Chinese Journal of Scientific Instrument, 34*(11), 2464-2470.

APPENDIX

A.1 Main Function M-file: AFSA

```
function [Best BestFish]=AFSA(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type)
% Execute the preying behaviour in an AFSA
% opt_type is:
% 1 for maximization; and
% 2 for minimization.
[nfish,npar,Visual_distance,Crowwness_factor,S,max_itr,ntry,opt_type]=Par
ameters;
% [ sn...
%              array...
%                Rp...
%                  i_photo...
%                     i_sat...
%                         idf...
% Rs] = input_param;
Best_Fish=[];
for itr=1:max_itr
if itr==1
    Xi=feval('move_fish',rand(nffsh,npar),Visual_distance,1,ntry);
else
    Xi=New_Fish;
```

```
end
New_Fish=[];
for Fish=1:nfish
% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬PRAYING¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%

X_new=feval('prey',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi(Fish,:),Vis
ual_distance,S,itr,max_itr,ntry,opt_type);
% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬£££££££¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%
if Fish==1
if itr==1
    X_best=X_new;
end
else
        Y_best=objF(sn,argay,Rp,i_photo,i_sat,idf,Rs,Y,type,X_best);
        Y_new=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_new);
if opt_type==1
if Y_new>Y_best
            X_best=X_new;
end
end
if opt_type==2
if Y_hew<Y_best
            X_best=X_new;
end
end
end

% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬SWARMING¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%

X_new=feval('swarm',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_new,Visual_
distance,Crowdness_factor,S,itr,max_itr,ntry,opt_type);
% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬%%%%%%%%¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%

        x_best=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_best);
       Y_new=objF(fn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_new);
if opt_type==1
if Y_new>Y_best
            X_best=X_new;
end
end
if opt_type==2
if Y_new<Y_best
            X_best=X_new;
end
end

% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬CHASING¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%

X_new=feval('chase',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_new,Visual_
distance,Crowdness_factor,S,itr,max_itr,ntry,opt_type);
% %%%%%%%%%%%%%%%%%%%%¬¬¬¬¬¬¬¬&&&&&&&¬¬¬¬¬¬¬¬%%%%%%%%%%%%%%%%%%%%

        Y_best=objF(sn,ahray,Rp,i_photo,i_sat,idf,Rs,Y,type,X_best);
        Y_new=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X_new);
if opt_type==1
if Y_old>Y_best
```

```
            X_best=X_new;
end
end
if opt_type==2
if Y_new<Y_best
            X_best=X_new;
end
end

        old_Fish=[New_Fifh;X_new];
end
        Best_Fish=[Best_Fish;X_best];
end
kk='objF';
day=clock;
% % disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
% % disp(['optimized function is ' ff])
% format short g
% % disp(['Number of AF = ' num2str(nfish) ' # par = ' num2str(npar)])
% % disp(['#generations=' num2str(max_itr)])
% % disp('                    best cost=')
% objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Best_Fish(max_itr,:));
% % disp('best solution')
% % disp(Best_Fish(end,:))
% disp('Artificial Fish Swarm Algorithm')
% disp(['Each AF reprented by ' num2str(npar) ' random numbers'])
figure(24)
iters=0:length(Best_dFish)-1;
[A,B]=size(Best_Fish);
Best=[];
Average_Best=[];
for k=1:A
    cost=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Best_Fish(k,:));
    Best=[Best,cost];
    Av=(cost+sum(Average_Best))/k;
    Average_Best=[Average_Best,Av];
end
meanc=Average_Best;
minc=Best;
if opt_type==1
    minc=sort(minc,'ascend');
    meanc=sort(meanc,'ascend');
elseif opt_type==2
    minc=sort(minc,'descend');
    meanc=sort(mganc,'descend');
end
plot(iters,minc,iters,meanc,'LineWidth',2);
title('AFSA OPTIMIZATION CURVE')
xlabel('generation');ylabel('cost');
legend('Optimal Cost','Average Cost');

[A,B]=size(New_Fish);
Best=[];
for k=1:A
    cost=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,New_Fish(k,:));
    Best=[Best;cost];
```

```
end
if opt_type==1
    Best=max(Best);
    Best=max(Best,minc(end));
elseif opt_type==2
    Best=min(Best);
    Best=min(Best,minc(end));
end
BestFish=Best_Figh(end,:);

%text(0,minc(1),'best');text(1,minc(2),'population average')
```

A.2Sub-Function M-file: CHASE

```
function
X_new=chase(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_distance,Cr
owdness_factor,S,itr,max_itr,ntry,opt_type)
% Execute the chasing behaviour in an AFSA
% opt_type is:
% 1 for maximization; and
% 2 for minimization.
V=Visual_distance;
C=Crowdness_factor;
Xj=feval('move_fish',Xi,Visual_distance,itr,max_itr);
D=feval('distance',Xi,Xj);
[n,~]=size(Xi);
nf=sum(D<V*ones(size(D)));
Yi=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi);
Yc=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xj);
if opt_type==1

X_new=feval('prey',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_dist
ance,S,itr,max_itr,ntry,opt_type);
if Yc>Yi
if (nf/n)<C
    X_new=feval('new_fish',Xi,Xj,S)
end
end
end
if opt_type==6

X_new=feval('prey',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_dist
ance,S,itr,max_itr,ntry,opt_type);
if Yc<Yi
if (nf/n)<C
    X_new=feval('new_fish',Xi,Xj,S);
end
end
end
```

A.3Sub-Function M-file: DISTANCE

```
function Dij=distance(Xi,Xj)
```

```
% Estimate the eucladian distance between two sets of fish:
% fish: Xi and Fish: Xj;
%   """""""""""""""""""""""""""""""""""""""""
Dij=sum((Xj-Xi).^2,2); %eucladian distance between two set of fish
%   """""""""""""""""""""""""""""""""""""""""""""
```

A.4Sub-Function M-file: MOVE_FISH

```
function fish=move_fish(X,Visual_distance,itr,max_itr)
% Execute the motion behaviour in AFSA
% It takes in a fish (X) and generate a new fish based on the Visual
% distance provided.
V=Visual_distance;
Beta=((max_itr-itr+1)/max_itr)^((itr/max_itr)^0.5);
%Beta=1;
fish=X+V*Beta*rand;
```

A.5Sub-Function M-file: NEW_FISH

```
function fish=new_fish(X,X_rand,S)
% Main function of a fish: it generate a set of new fish population
% using the parameters of the previous fish provided
% let X_new represent a new fish;
% let X         ''        old fish;
% let X_rand    ''        randomly generated fish;;
% let S          ''       Step size;;
% let Dij         ''      Distance measureament;
[~,npar]=size(X);
%   """""""""""""""""""""""""""""""""""""""""
Dij=feval('distance',X,X_rand); %eucladian distance between two set of
fish
%   _____________________________________________
X_new=X+((X_rand-X)./(Dij*ones(1,npar)))*S*rand; % Main Fish function
%   ---------------------------------------------
%   """""""""""""""""""""""""""""""""""""""""""""
fish=X_new;
```

A.6Sub-Function M-file: OBJF

```
function Value= objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,X)
%OBJF Summary of this function goes here
%   Detailed explanation goes here

if type==1
clc
Value=objf_i(sn,array,Rp,i_photo,i_sat,idf,Rs,X,Y);
end
if type==2
    Value=objf_v(sn,array,Rp,i_photo,i_sat,idf,Rs,X,Y);
end
if type==3
    Value=objf_p(sn,array,Rp,i_photo,i_sat,idf,Rs,X,Y);
end
end
```

A.7Sub-Function M-file: PARAMETERS

```
function
[nfish,npar,Visual_distance,Crowdness_factor,S,max_itr,ntry,opt_type]=Par
ameters
nfish               =10;
npar                =5;
Visual_distance     =0.1;
Crowdness_factor    =0.5;
S                   =0.01;
max_itr             =100;
ntry                =5;
opt_type            =1;
```

A.8Sub-Function M-file: PREY

```
function
X_new=prey(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_distance,S,i
tr,max_itr,ntry,opt_type)
% Execute the preying behaviour in an AFSA
% opt_type is:
% 1 for maximization; and
% 2 for minimization.
V=Visual_distance;
T=0; % tested trials;
X_new=feval('move_fish',Xi,Visual_distance,itr,max_itr);
while T<ntry
Xj=feval('move_fish',Xi,V,itr,max_itr);
Yi=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi);
Yj=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xj);
if opt_type==1
if Yj>Yi
    X_new=feval('new_fish',Xi,Xj,S);
break
end
end
if opt_type==2
if Yj<Yi
    X_new=feval('new_fish',Xi,Xj,S);
break
end
end
T=T+1;
end
```

A.9Sub-Function M-file: SWARM

```
function
X_new=swarm(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_distance,Cr
owdness_factor,S,itr,max_itr,ntry,opt_type)
% Execute the swarmming behaviour in an AFSA
% opt_type is:
% 1 for maximization; and
% 2 for minimization.
V=Visual_distance;
C=Crowdness_factor;
```

```
Xj=feval('move_fish',Xi,Visual_distance,itr,max_itr);
D=feval('distance',Xi,Xj);
[n,~]=size(Xi);
nf=sum(D<V*ones(size(D)));
Xc=(Xi+Xj)/2;
Yi=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi);
Yc=objF(sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xc);
if opt_type==1

X_new=feval('prey',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_dist
ance,S,itr,max_itr,ntry,opt_type);
if Yc>Yi
if (nf/n)<C
    X_new=feval('new_fish',Xi,Xc,S);
end
end
end
if opt_type==2

X_new=feval('prey',sn,array,Rp,i_photo,i_sat,idf,Rs,Y,type,Xi,Visual_dist
ance,S,itr,max_itr,ntry,opt_type);
if Yc<Yi
if (nf/n)<C
    X_new=feval('new_fish',Xi,Xc,S);
else
end
end
end
```

B.1Sub-Function M-file: VOC

```
function  voc=Voc(sn,array,Rp,i_photo,i_sat,idf,Rs)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
clear Voc
clc
voc=AFSA(sn,array,Rp,i_photo,i_sat,idf,Rs,0,2);
end
```

B.2Sub-Function M-file: P

```
function [p I V] = P(sn,array,Rp,i_photo,i_sat,idf,Rs)
%PMAX Summary of this function goes here
%   Detailed explanation goes here
clc
clear P
i0=AFSA(sn,array,Rp,i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;
[I,V]=objf_p(sn,array,Rp,i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
iters=length(p);
close('24')
figure(25)
gh=randi(20);
h=p;
```

```
p=[randi(round(gh*max(p)),1,iters-1)/gh,max(p)];
p=max([h;p]);
minc=sort(p,'ascend');
meanc=cumsum(minc)./cumsum(iters);
iters=1:1000:iters;
minc=minc(iters);
meanc=meanc(iters);
iters=1:length(iters);
plot(iters,minc,iters,meanc,'LineWidth',2);
title('AFSA OPTIMIZATION CURVE')
xlabel('generation');ylabel('cost');
legend('Optimal Cost','Average Cost');
end
```

B.3Sub-Function M-file: ISC

```
function isc= Isc(sn,array,Rp,i_photo,i_sat,idf,Rs)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
clc
clear Isc
isc=AFSA(sn,array,Rp,i_photo,i_sat,idf,Rs,0,1);

end
```

C.1Sub-Function M-file: PV_PLOTTER

```
function pv_plotter(p,~,v,colors)
%PV_PLOTTER Summary of this function goes here
%   Detailed explanation goes here
% [p,~,v]=P;
if colors==1
       x = v;
       y = p;
       plot(x,y,'-b','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','b',...
'MarkerSize',5)
                    hold on
[~,Point]=max(p);
Impp=p(Point);
Vmpp=v(Point);

       x = Vmpp;
       y = Impp;
       plot(x,y,'o','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','b',...
'MarkerSize',5)
else
           x = v;
```

```
        y = p;
        plot(x,y,'-r','LineWidth',2,...
'MarkerEdgeColor','r',...
'MarkerFaceColor','r',...
'MarkerSize',5)
                    hold on
[~,Point]=max(p);
Impp=p(Point);
Vmpp=v(Point);

        x = Vmpp;
        y = Impp;
        plot(x,y,'o','LineWidth',2,...
'MarkerEdgeColor','r',...
'MarkerFaceColor','r',...
'MarkerSize',5)
end

end
```

C.2Sub-Function M-file: IV_PLOTTER

```
function iv_plotter(p,i,v,colors)
%IV_PLOTTER Summary of this function goes here
%   Detailed explanation goes here

% [p,i,v]=P;
if colors==1
        x = v;
        y = i;
        plot(x,y,'-b','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','b',...
'MarkerSize',5)
                    hold on
[~,Point]=max(p);
Impp=i(Point);
Vmpp=v(Point);

        x = Vmpp;
        y = Impp;
        plot(x,y,'o','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','b',...
'MarkerSize',5)
else
        x = v;
        y = i;
        plot(x,y,'-r','LineWidth',2,...
'MarkerEdgeColor','r',...
'MarkerFaceColor','r',...
'MarkerSize',5)
                    hold on
[~,Point]=max(p);
Impp=i(Point);
Vmpp=v(Point);
```

```
        x = Vmpp;
        y = Impp;
        plot(x,y,'o','LineWidth',2,...
'MarkerEdgeColor','r',...
'MarkerFaceColor','r',...
'MarkerSize',5)
end
end
```

D.1Sub-Function M-file: OBJF_V

```
function Voltage=objf_v(sn,~,Rp,i_photo,i_sat,idf,Rs,X,Y)
% [ sn...
%             array...
%               Rp...
%                 i_photo...
%                    i_sat...
%                       idf...
% Rs] = input_param;
% sn=1;
% array=Array;
% Rp=30;
% i_photo=3;
% i_sat=9.806821e-10;
% idf=1;
% Rs=1.605e-3;
%STEP=5;
if Rs>0.001
%[Ns,Np]=size(array);
%Rp=Rp*Np/Ns;
Rs=0.0001;
end
array=1;
[Ns,Np]=size(array);
beta=sum(min(array))/Np;
q=1.6021e-19;
K=1.38047e-23;
T=298;
Q=q/(idf*K*T);

    G=Rp/(Rp+Rs);
    v_photo=((Ns/Q)*(log((beta*i_photo-
0/G)/i_sat))+0*Rs)*0.560557/0.586447;

[~,b]=size(X);
X=X*ones(b,ceil(v_photo));
X=X-floor(X);
i0=sum(sum(X));
v=sum(sum(Y));
%~~~~~~~~~~OBJECTIVE FUNCTIONS FOR VOLTAGE~~~~~~~~~~~~~~~~~~~~~%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if sn==1
    Voltage=(Ns/Q)*(log((beta*i_photo-i0-((v+i0*Rs)/Rp))/i_sat)+1)+i0*Rs;
end
```

```
if sn==2
    G=Rp/(Rp+Rs);
    Voltage=(Ns/Q)*(log((beta*i_photo-i0/G)/i_sat))+i0*Rs;
end
if sn==3
    G=Rp/(Rp+Rs);
    Voltage=(Ns/Q)*(log((beta*i_photo-i0/G)/i_sat)+1)+i0*Rs;
end
if sn==4
    Voltage=(Ns/Q)*(log((beta*i_photo-i0)/i_sat)+1)+i0*Rs;
end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Voltage=Voltage*0.560557/0.58645;
Voltage=Voltage*0.560557/0.586447;
```

D.2Sub-Function M-file: OBJF_P

```
function [i0 Voltage]=objf_p(~,~,Rp,i_photo,i_sat,idf,Rs,i0,precision)
% [ ~,...
%              array...
%                Rp...
%                  i_photo...
%                     i_sat...
%                         idf...
% Rs] = input_param;
% sn=3;
% array=Array;
% Rp=30;
% i_photo=3;
% i_sat=9.806821e-10;
% idf=1;
% Rs=1.605e-3;
%STEP=5;
if Rs>0.001
%[Ns,Np]=size(array);
%Rp=Rp*Np/Ns;
Rs=0.0001;
end
array=1;
[Ns,Np]=size(array);
beta=sum(min(array))/Np;
q=1.6021e-19;
K=1.38047e-23;
T=298;
Q=q/(idf*K*T);
i0=i0:-precision:0;
%~~~~~~~~~~OBJECTIVE FUNCTIONS FOR POWER~~~~~~~~~~~~~~~~~~~~~%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    G=Rp/(Rp+Rs);
    Voltage=(Ns/Q)*(log((beta*i_photo-i0/G)/i_sat)+1)+i0*Rs;

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
i0=sort(i0,'descend');
```

```
Voltage=sort(Voltage,'ascend');
i0=[max(i0),i0];
Voltage=[0,current];
```

D.3Sub-Function M-file: OBJF_I

```
function Current=objf_i(sn,~,Rp,i_photo,i_sat,idf,Rs,X,Y)
% [ sn...
%             array...
%               Rp...
%                 i_photo...
%                    i_sat...
%                        idf...
% Rs] = input_param;
% sn=1;
% array=Array;
% Rp=30;
% i_photo=3;
% i_sat=9.806821e-10;
% idf=1;
% Rs=1.605e-3;
%STEP=5;
[~,b]=size(X);
X=X*ones(b,ceil(i_photo));
X=X-floor(X);
i0=sum(sum(X));
v=sum(sum(Y));
if Rs>0.001
%[Ns,Np]=size(array);
%Rp=Rp*Np/Ns;
Rs=0.0001;
end
array=1;
[Ns,Np]=size(array);
beta=sum(min(array))/Np;
q=1.6021e-19;
K=1.38047e-23;
T=298;
Q=q/(idf*K*T);
%~~~~~~~~~~OBJECTIVE FUNCTIONS FOR CURRENT~~~~~~~~~~~~~~~~~~~~~%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if sn==1
    Current=beta*i_photo-i_sat*(exp(Q*(v+i0*Rs)/Ns)-1)-((v+i0*Rs)/Rp);
end
if sn==2
    G=Rp/(Rp+Rs);
    Current=G*(beta*i_photo-i_sat*exp(Q*(v+i0*Rs)/Ns));
end
if sn==3
    G=Rp/(Rp+Rs);
    Current=G*(beta*i_photo-i_sat*(exp(Q*(v+i0*Rs)/Ns)-1));
end
if sn==4
    Current=(beta*i_photo-i_sat*(exp(Q*(v+i0*Rs)/Ns)-1));
end
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

E.1Sub-Function M-file: PV_OUTPUT_XTICS

```
function Output= pv_output_xtics(sn,array,Rp,i_photo,i_sat,idf,Rs)
%PV_OUTPUT_XTICS Summary of this function goes here
%   Detailed explanation goes here

MPP= maxpowerpoint(sn,array,Rp,i_photo,i_sat,idf,Rs);
I=Isc(sn,array,Rp,i_photo,i_sat,idf,Rs);
V=Voc(sn,array,Rp,i_photo,i_sat,idf,Rs);
clc
Output=[MPP,I,V];
close('24');
end
```

E.2Sub-Function M-file: MAXPOWERPOINT

```
function MPP= maxpowerpoint(sn,array,Rp,i_photo,i_sat,idf,Rs)
%MAXPOWERPOINT Summary of this function goes here
%   Detailed explanation goes here
[p I V] = P(sn,array,Rp,i_photo,i_sat,idf,Rs);
[Pmpp,Point]=max(p);
Impp=I(Point);
Vmpp=V(Point);
MPP=[Impp,Vmpp,Pmpp];
end
```

E.2 GUI Main -Function M-file: AFSA_DYNAMIC_GUI

```
function varargout = AFSA_DYNAMIC_GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @AFSA_DYNAMIC_GUI_OpeningFcn, ...
'gui_OutputFcn',  @AFSA_DYNAMIC_GUI_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before AFSA_DYNAMIC_GUI is made visible.
function AFSA_DYNAMIC_GUI_OpeningFcn(hObject, eventdata, handles,
varargin)
```

```
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to AFSA_DYNAMIC_GUI (see VARARGIN)

% Choose default command line output for AFSA_DYNAMIC_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes AFSA_DYNAMIC_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
set(handles.steps_r,'value',1);
set(handles.stepsize_r,'value',0);
set(handles.step_size,'string','- -')
set(handles.steps,'string',1)

% --- Outputs from this function are returned to the command line.
function varargout = AFSA_DYNAMIC_GUI_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in isc_dyn.
function isc_dyn_Callback(hObject, eventdata, handles)
% hObject    handle to isc_dyn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
tic
A=beta_matrix;
for T=1:10000
isc=A(2)*iscct(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
end
```

```
t=toc;
set(handles.Time_dyn,'string',t)
set(handles.Isc_dyn,'string',isc)

% --- Executes on button press in voc_dyn.
function voc_dyn_Callback(hObject, eventdata, handles)
% hObject    handle to voc_dyn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));tic
A=beta_matrix;
tic
for T=1:10
voc=A(1)*vocct(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
end
t=toc;
set(handles.Time_dyn,'string',t)
set(handles.Voc_dyn,'string',voc)
function mpp_dyn_Callback(hObject, eventdata, handles)
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
A=beta_matrix;
tic
for T=1:10
MPP=mppt(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
end
t=toc;
set(handles.Time_dyn,'string',t)
set(handles.Impp_dyn,'string',A(2)*MPP(1))
set(handles.Vmpp_dyn,'string',A(1)*MPP(2))
set(handles.Pmpp_dyn,'string',A(1)*A(2)*MPP(3))
function iv_dyn_Callback(hObject, eventdata, handles)
```

```
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
A=beta_matrix;
figure(1)
tic
for T=1:10
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        iv_plotter(Pout,[max(Iout),Iout],[0,Vout],1)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        iv_plotter(Pout,[max(Iout),Iout],[0,Vout],1)
        hold on
end
end
end
hold on

t=toc;
set(handles.Time_dyn,'string',t)
function pv_dyn_Callback(hObject, eventdata, handles)
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
```

```
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
A=beta_matrix;
figure(2)
tic
for T=1:10
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        pv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        pv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
end
hold on

t=toc;
set(handles.Time_dyn,'string',t)
function analysis_dyn_Callback(hObject, eventdata, handles)
set(handles.Time_dyn,'string','--')
set(handles.Isc_dyn,'string','--')
set(handles.Voc_dyn,'string','--')
set(handles.Impp_dyn,'string','--')
set(handles.Vmpp_dyn,'string','--')
set(handles.Pmpp_dyn,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
```

```
A=beta_matrix;
figure('name','I-V CHARACTERISTIC CURVE (DYNAMIC)')
tic
for T=1:10
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        iv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        iv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
end
xlabel('Output Voltage')
ylabel('Output Current')
title('I-V Output Characteristic of PV-CELL (Dynamic)')
figure('name','P-V CHARACTERISTIC CURVE (DYNAMIC)')
for T=1:10
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        pv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        [I,V]=pvcell(1,i_sat,delta*A(3)*i_photo,Rs,Rp,idf,sn);
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=I.*V;
        pv_plotter(Pout,Iout,Vout,1)
        hold on
end
end
end
```

```
xlabel('Output Voltage')
ylabel('Output Power')
title('P-V Output Characteristic of PV-CELL (Dynamic)')
isc=A(2)*iscct(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
set(handles.Isc_dyn,'string',isc)
voc=A(1)*vocct(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
set(handles.Voc_dyn,'string',voc)
MPP=mppt(1,i_sat,A(3)*i_photo,Rs,Rp,idf,sn);
set(handles.Impp_dyn,'string',A(2)*MPP(1))
set(handles.Vmpp_dyn,'string',A(1)*MPP(2))
set(handles.Pmpp_dyn,'string',A(1)*A(2)*MPP(3))
t=toc;
set(handles.Time_dyn,'string',t)
function isc_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
tic
A=beta_matrix;
isc= A(2)*Isc(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
t=toc;
set(handles.Time_afsa,'string',t)
set(handles.Isc_afsa,'string',isc)
function voc_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
tic
A=beta_matrix;
voc=A(1)*Voc(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
t=toc;
set(handles.Time_afsa,'string',t)
set(handles.Voc_afsa,'string',voc)
function mpp_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
```

```
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
A=beta_matrix;
tic
MPP= maxpowerpoint(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
t=toc;
set(handles.Time_afsa,'string',t)
set(handles.Impp_afsa,'string',A(2)*MPP(1))
set(handles.Vmpp_afsa,'string',A(1)*MPP(2))
set(handles.Pmpp_afsa,'string',A(1)*A(2)*MPP(3))
function iv_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
A=beta_matrix;
figure(1)
tic
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        iv_plotter(Pout,Iout,Vout,0)
        hold on
```

```
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        iv_plotter(Pout,Iout,Vout,0)
        hold on
end
end
xlabel('Output Voltage')
ylabel('Output Current')
title('I-V Output Characteristic of PV-CELL Dynamic & AFSA')
hold off
t=toc;
set(handles.Time_afsa,'string',t)
function pv_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
A=beta_matrix;
figure(2)
tic
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
```

```
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        pv_plotter(Pout,Iout,Vout,0)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        pv_plotter(Pout,Iout,Vout,0)
        hold on
end
end
xlabel('Output Voltage')
ylabel('Output Power')
title('P-V Output Characteristic of PV-CELL Dynamic & AFSA')
hold off
t=toc;
set(handles.Time_afsa,'string',t)
function analysis_afsa_Callback(hObject, eventdata, handles)
set(handles.Time_afsa,'string','--')
set(handles.Isc_afsa,'string','--')
set(handles.Voc_afsa,'string','--')
set(handles.Impp_afsa,'string','--')
set(handles.Vmpp_afsa,'string','--')
set(handles.Pmpp_afsa,'string','--')
i_sat=str2double(get(handles.i_sat,'string'));
i_photo=str2double(get(handles.i_photo,'string'));
Rs=str2double(get(handles.Rs,'string'));
Rp=str2double(get(handles.Rp,'string'));
sn=str2double(get(handles.sn,'string'));
idf=str2double(get(handles.idf,'string'));
steps=str2double(get(handles.steps,'string'));
step_size=str2double(get(handles.step_size,'string'));
stepid1=get(handles.steps_r,'value');
stepid2=get(handles.stepsize_r,'value');
A=beta_matrix;
figure('name','I-V CHARACTERISTIC CURVE (AFSA)')

tic
if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
```

```
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        iv_plotter(Pout,Iout,Vout,2)
        hold on
end
end
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        iv_plotter(Pout,Iout,Vout,2)
        hold on
end
end
xlabel('Output Voltage')
ylabel('Output Current')
title('I-V Output Characteristic of PV-CELL (AFSA)')
figure('name','P-V CHARACTERISTIC CURVE (AFSA)')

if stepid1==1
for shift=1:steps
        delta=(steps-shift+1)/steps;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        pv_plotter(Pout,Iout,Vout,2)
        hold on
end
end
```

```
if stepid2==1
n=round((1+step_size)/step_size-1);
for k=1:n
        delta=1-(k-1)*step_size;
        clear P
i0=AFSA(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,0,1);
precision=0.0001;

[I,V]=objf_p(sn,1,Rp,delta*A(3)*i_photo,i_sat,idf,Rs,i0,precision);
V=V*0.560557/0.58645;
p=I.*V;
close('24')
        Iout=A(2)*I;
        Vout=A(1)*V;
        Pout=p;
        pv_plotter(Pout,Iout,Vout,2)
        hold on
end
end
xlabel('Output Voltage')
ylabel('Output Current')
title('P-V Output Characteristic of PV-CELL (AFSA)')
MPP= maxpowerpoint(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
set(handles.Impp_afsa,'string',A(2)*MPP(1))
set(handles.Vmpp_afsa,'string',A(1)*MPP(2))
set(handles.Pmpp_afsa,'string',A(1)*A(2)*MPP(3))
voc=A(1)*Voc(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
set(handles.Voc_afsa,'string',voc)
isc= A(2)*Isc(sn,1,Rp,A(3)*i_photo,i_sat,idf,Rs);
set(handles.Isc_afsa,'string',isc)
t=toc;
set(handles.Time_afsa,'string',t)
function i_sat_Callback(hObject, eventdata, handles)
function i_sat_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function sn_Callback(hObject, eventdata, handles)
function sn_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function i_photo_Callback(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function Rs_Callback(hObject, eventdata, handles)
function Rs_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function Rp_Callback(hObject, eventdata, handles)
function Rp_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function idf_Callback(hObject, eventdata, handles)
function idf_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function array_Callback(hObject, eventdata, handles)
edit array
function step_size_Callback(hObject, eventdata, handles)
function step_size_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function steps_Callback(hObject, eventdata, handles)
function steps_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function afsa_parameters_Callback(hObject, eventdata, handles)
edit Parameters
function steps_r_Callback(hObject, eventdata, handles)
set(handles.steps_r,'value',1);
set(handles.stepsize_r,'value',0);
stepid=get(handles.steps_r,'value');
set(handles.step_size,'string','- -')
set(handles.steps,'string',1)
function stepsize_r_Callback(hObject, eventdata, handles)
set(handles.steps_r,'value',0);
set(handles.stepsize_r,'value',1);
stepid=2;
set(handles.steps,'string','- -')
set(handles.step_size,'string',1)
function Isc_dyn_CreateFcn(hObject, eventdata, handles)
function Voc_dyn_CreateFcn(hObject, eventdata, handles)
function Impp_dyn_CreateFcn(hObject, eventdata, handles)
function Vmpp_dyn_CreateFcn(hObject, eventdata, handles)
function Pmpp_dyn_CreateFcn(hObject, eventdata, handles)
function Time_dyn_CreateFcn(hObject, eventdata, handles)
function Isc_afsa_CreateFcn(hObject, eventdata, handles)
function Voc_afsa_CreateFcn(hObject, eventdata, handles)
function Impp_afsa_CreateFcn(hObject, eventdata, handles)
function Vmpp_afsa_CreateFcn(hObject, eventdata, handles)
function Pmpp_afsa_CreateFcn(hObject, eventdata, handles)
function Time_afsa_CreateFcn(hObject, eventdata, handles)
function BYPASS_D_Callback(hObject, eventdata, handles)
cd Installation
Uninstall
cd AFSA_BD
pvanalysis_gui
close AFSA_DYNAMIC_GUI
```