الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة الشهيد الشيخ العربي التبسي – تبسة
Echahid Cheikh Larbi Tebessi University-Tebessa
Faculty of Science and Technology
Department of Electronics and Communications

Master's Thesis

Submitted for the degree of Academic Master

Field: Electronics

Specialty: Instrumentation

Presented by: - Mr. BELMEKKI MOHAMMED EL HADI

-Mr. BAKHOUCHE NOUR EL ISLEM

Title

Modeling and Identification of Photovoltaic Systems Using Fuzzy Logic Models.

Defended and evaluated on / 06 / 2026 ,by the jury composed of:

First and last name		Rank	Position
Mr.	DHAOUADI GUIZA	MCA	Chair
Mr.	ABDELAZIZ AOUICHE	Prof.	Supervisor
Mrs.	LEYLA CHERIET	MCB	Reviewer

Graduating Class: 2025/2026

ACKNOWLEDGMENTS

ACKNOWLEDGMENTS

First and foremost, all praises and thanks are due to Allah the Almighty for granting me the health, patience, and strength to complete this modest work.

I would like to express my deepest gratitude and sincere appreciation to my esteemed supervisor, **Prof. ABDELAZIZ AOUICH**, for his continuous support, valuable guidance, and insightful remarks throughout the realization of this research. His patience and expertise have been invaluable assets to me.

My profound thanks go to the members of the jury, **MCA.DHAOUADI GUIZA the Chairman** and **MCB.LEYLA CHERIET the Reviewer**, for kindly accepting to read, evaluate, and enrich this work with their constructive criticisms and observations.

I am also highly indebted to the teaching staff and administration of the Department of Electronics and Telecommunications, Faculty of Sciences and Technology at the University of Tebessa, particularly our amiable teachers, for their knowledge and encouragement throughout my academic journey.

I want to express my sincere gratitude to **Dr. BOUTASSETA NADIR** for his knowledge and the help he gave me during my internship at URMM/CRTI Annaba.

I would also like to express my sincere appreciation to my Master 2 Instrumentation classmates. Their unwavering support helped me readjust to the academic environment and was a driving force behind my success, particularly my partner, **BAKHOUCHE NOUR EL-ISLEM.**

Finally, I owe a special debt of gratitude to my beloved family. Their endless prayers, sacrifices, and unwavering support have always been my source of motivation, and I thank them for standing by my side, providing comfort, and continuously believing in me.

By BELMEKKI MOHAMMED EL-HADI.

2026

ACKNOWLEDGMENTS

In the name of Allah, the Most Gracious, the Most Merciful. All praise and thanks are due to Allah, the Almighty, for His countless blessings, guidance, and strength that enabled me to complete this work. Peace and blessings be upon our beloved Prophet Muhammad, his family, and his companions.

First and foremost, I would like to express my deepest gratitude and sincere appreciation to my supervisor, **Prof. ABDELAZIZ AOUICH**, for his valuable guidance, continuous support, patience, and insightful comments throughout the preparation of this thesis.

I am also grateful to the members of the jury, **MCA.DHAOUADI GUIZA the Chair** and **MCB.LEYLA CHERIET the Reviewer**, for accepting to evaluate this work and for their valuable remarks and suggestions.

My heartfelt thanks go to **Dr. BOUTASSETA NADIR**, a person with a brilliant mind and an even bigger heart. Your kindness, wisdom, and encouragement have left a lasting impact on me.

I would also like to thank **Dr. Nezzari, Dr. Moussa and Dr. Bouabida**. You were not only excellent teachers but true friends who supported and guided me with patience and care.

A special thank you to my dear classmates with whom I spent five unforgettable years. In particular, I am deeply grateful to **Redwan Bendjeddah** and **Chettioui Wassim**, who were more than friends they were brothers, redwan please don't restart the USSR and wassim yes Wassim, my head is normal sized. I also thank **Ibrahim**, with whom I faced many difficult and "scrambled" years together. I cannot forget **SamahBayazid**, **Moussa Insaf** with whom I graduated in the third year; and **ChaouchiDouaa,** we helped each other through tough times, and shared many laughs even in our lowest moments. your support, laughter, and help during tough times made the journey much easier.

To my beloved family, thank you from the bottom of my heart for your endless love, patience, sacrifices, and prayers. None of this would have been possible without you.

To my friends — YakoubChettouh, OkbaHarbaoui, Aymen, Maria Traia (our little sister), Feriel, thank you for being in my life blessed to have you .

And especially **AchrafGhaloussi** thank you for standing by me. Achraf, you were there during my lowest moments when almost no one else was. I am truly blessed to have you.

And to **Pedro**… I hope you don't get yourself killed under a rock one day, my friend…

Finally, I would like to give a very special thanks to **Belmekki Mohammed El Hadi**. You appeared as a light at the end of a long and difficult road. I do not know how Allah guided me

to you, but I am forever grateful. You are one of the main reasons I am standing here today. May Allah bless you and your family, ease all your affairs, and grant you success in this life and the hereafter.

Thank you all.

By BAKHOUCHE NOUR EL ISLAM

Abstract

Abstract

Photovoltaic **(PV)** systems are characterized by nonlinear power-voltage characteristics that vary significantly with solar irradiance and cell temperature. Conventional Maximum Power Point Tracking **(MPPT)** techniques often suffer from oscillations, slow response, and reduced efficiency under rapidly changing and partial shading conditions.

This thesis presents the modeling and identification of photovoltaic systems using intelligent techniques. A single-diode equivalent circuit model was studied and implemented in **MATLAB** for the **KC200GT PV** module. The effects of environmental conditions on the I-V and P-V characteristics and the location of the **Maximum Power Point (MPP)** were thoroughly analyzed. Classical **MPPT methods**, including Perturb and Observe **(P&O)**, Incremental Conductance **(IncCond)**, and Particle Swarm Optimization **(PSO)**, were reviewed and their limitations highlighted.

To overcome these limitations, a proposed intelligent models based on **Multi-Layer Perceptron (MLP)** and Adaptive **Neuro-Fuzzy Inference System (ANFIS)** were designed for accurate MPP identification. These models were trained using data generated from the validated photovoltaic model under a wide range of irradiance and temperature conditions.

Simulation results demonstrated that the proposed intelligent models achieve superior tracking accuracy, faster convergence, and significantly reduced oscillations compared to conventional methods, even under dynamic and non-uniform irradiance conditions. The developed approach offers a robust and efficient solution for improving the performance of photovoltaic systems.

Keywords: PV-modul, MPPT, Photovoltaic systems, Single-diode circuit model, ANFIS, MLP, RBF, Fuzzy logic, Intelligent MPPT, MPPT techniques.

ملخص

تمتاز الأنظمة الكهروضوئية بالخصائص اللاخطية و التي تتغير بتغير قيمة الإشعاعات الشمسية و درجة حرارة الخلية الضوئية. تعاني التقنيات التقليدية لتتبع نقطة الطاقة القصوى MPPT عادة من التذبذبات، بطء الإستجابة، وكذا إنخفاض الكفاءة بسبب ظروف التغير السريع و الظل الجزئي.

تقدم هذا الأطروحة نمذجة و تحديد لمنظومة الكهرضوئية بإستخدام تقنيات ذكية. بداية تم دراسة نموذج دائرة مكافئة أحادية الصمام وبرمجته في MATLAB للوحدة الكهروضوئية KC200GT. تم تحليل تأثيرات الظروف البيئية على خصائص I-V وP-V ، و تحديد موقع نقطة الطاقة القصوى (**MPP**) بدقة. تمت مراجعة التقنيات الكلاسيكية لتتبع نقطة الطاقة القصوى (**MPPT**)، بما في ذلك الاضطراب والمراقبة (**O&P**)، والتوصيل التزايدي (**IncCond**)، وتحسين سرب الجسيمات (PSO)، وتم تسليط الضوء على عيوبها.

للتغلب على هذه العيوب، تم إقتراح تصاميم لنماذج ذكية تعتمد على شَبَكِيَّات العَصَب متعدد الطبقات (MLP) و نظام الاستدلال العصبي الضبابي التكيفي (ANFIS) لتحديد نقطة الطاقة القصوى (MPP) بدقة. تم تدريب هذه النماذج باستخدام البيانات الناتجة من النموذج الوحدة الكهروضوئية الذي تم برمجته، في ظل نطاق واسع من ظروف الإشعاع ودرجة الحرارة.

أظهرت نتائج المحاكاة أن النماذج الذكية المقترحة تحقق دقة تتبع فائقة، وتقاربًا أسرع، وتذبذبات أقل بشكل ملحوظ مقارنة بالطرق التقليدية، حتى في ظل ظروف الإشعاع الديناميكية وغير المنتظمة. يقدم النهج المطور حلاً قويًا وفعالًا لتحسين أداء الأنظمة الكهروضوئية.

الكلمات المفتاحية: وحدة الطاقة الشمسية الكهروضوئية، تتبع نقطة الطاقة القصوى، الأنظمة الكهروضوئية، نموذج الدائرة أحادية الصمام الثنائي، نظام الاستدلال العصبي الضبابي التكيفي، شَبَكِيَّات العَصَب متعدد الطبقات، دالة الأساس الشعاعي، المنطق الضبابي، تتبع نقطة الطاقة القصوى الذكي، تقنيات تتبع نقطة الطاقة القصوى.

Table of contents

Table of contents

Acknowledgments

Abstract

List of figures and tables

Acronyms

General Introduction.....1

Chapter 1: Modelling of Photovoltaic Systems Based on the One-Diode Model

1.1. Introduction.....3

1.2. Historical background.....3

1.3. Fundamentals of photovoltaic effect.....3

1.3.1. Principles of Photovoltaic Conversion……..….....3

1.3.2. Fundamentals of Solar Cells……..…….....……..…...5

1.4. Electrical characteristics of PV modules.....6

1.5. Single-diode Equivalent Circuit Model.....8

1.5.1 Photovoltaic module current-voltage relationship…….....…….....9

1.5.2 Basic module's parameters.....9

1.5.3. Generated I-V and P-V curves under different operating conditions.....10

1.6. Modelling and Simulation of the KC200GT Photovoltaic Module.....11

1.7 Influence of temperature and irradiance on PV performance.....13

1.8 Conclusion ...……...……..……..13

Chapter 2: Analysis of Maximum Power Point and MPPT-Techniques

2.1. Introduction.....15

2.2. Definition of Maximum Power Point (MPP).....15

2.3. Effect of environmental conditions on MPP.....16

2.4. MPPT techniques ……..……...17

2.4.1. Fixed Duty Cycle Method .....…….18

2.4.2 Perturb and Observe (P&O) Method.....18

2.4.3 Incremental Conductance (IncCond) Method.....20

2.4.4 Particle Swarm Optimization (PSO) Method.....21

2.5. Motivation for Intelligent MPPT Techniques.....23

2.6 Conclusion……...23

Chapter 3: Fundamental of Neural Networks NN

3.1. Introduction…….....25

3.2. The Neural Networks 25
3.3. The architecture of an Artificial Neural Networks.....26
3.4. Rosenblatt's Perceptron.....29
3.4.1. Definition.....29
3.4.2. Training of Perceptron.....29
3.5. Multilayer perceptron MLP.....30
3.5.1.Definition.....30
3.5.2. Architecture Properties of MLP.....30
3.5.3. Supervised training of MLP.....31
3.6.The Back Propagation Algorithm.....31
3.7. Radial basic neural network RBNN.....37
3.7.1. Definition.....37
3.7.2. Architecture of RBNN.....38
3.7.3. Cover's theorem and the separability of patterns.....38
3.7.4. Radial Basis Function.....38
3.7.5. RBNN algorithm.....39
3.7.6. RBNN training process.....40
3.8. Design and Development of the MLP-based MPPT.....41
3.8.1 Initialized the MLP model.....41
3.8.2 Supervised learning of the Model.....41
3.8.3. Testing the Model.....43
3.9. Conclusion and analyse.....43
Chapter 4: Fundamental of Fuzzy set systems (FSSs)
4.1. Introduction.....45
4.2. Fuzzy Set Systems (FSs).....45
4.2.1. Definitions.....46
4.2.2. Applications of fuzzy logic.....48
4.2.3. Examples of using fuzzy logic system.....49
4.3. Fuzzy inference system (FIS).....49
4.3.1 Mamdani fuzzy model.....50
4.3.2 Tsukamoto fuzzy model.....50
4.3.3 Sugeno fuzzy models.....51
4.4. Adaptive network fuzzy inference system ANFIS.....53

4.4.1. Definition of ANFIS.....53
4.4.2. ANFIS architecture.....53
4.4.3 Hybrid learning algorithm for ANFIS.....55
4.4.4 ANFIS in MPPT.....59
4.5. Design and development of fuzzy –based MPPT.....59
4.5.1. Description of proposed model.....60
4.5.2. Supervised learning of the Model.....61
4.6. Conclusion and analysis.....62
Chapter 5: Simulation Results, Performance Evaluation, and Validation
5.1. Introduction.....64
5.2. MATLAB simulation setup.....64
5.2.1. PV-panel model.....64
5.2.2. MLP-based MPPT model.....66
5.2.3. Fuzzy model for MPPT.....68
5.3. Comparison between fuzzy model and mathematical model.....70
5.4. ANFIS-based MPPT performance analysis.....71
5.5. Efficiency improvement evaluation.....71
5.6. Discussion and validation.....72
5.7. Conclusion.....73
General conclusion.....74
Bibliography
ANNEXES

List of figures

And Tables

List of figures

Chapter 1: Modelling of Photovoltaic Systems Based on the One-Diode Model

Figure1.1 Comparison between PE effect (left) and PV effect in a solar cell (right).....4

Figure1.2 Asymmetrical properties of a P-N junction diode.....4

Figure 1.3 I-V characteristics of a solar cell under Dark and Illuminated conditions.....5

Figure1.4 Association of cells to obtain a photovoltaic module.....6

Figure1.5 Block diagram of a typical photovoltaic (PV) system.....6

Figure1.6.Single-diode model of the theoretical PV cell including series and shunt resistances.....8

Figure1.7 Characteristics of PV Module: (a) Current-Voltage I-V (b) Power-Voltage P-V....10

Figure1.8 I-V Characteristics of PV Module for fixed T=25°C and different irradiations.....10

Figure1.9. Characteristics of PV Module for fixed T=25° and different irradiations values (a) I-V curves (b) P-V curves.....11

Figure.1.10 Characteristics of PV Module for fixed G=1000 W/m² and different temperatures values (a) I-V curves (b) P-V curves.....12

Chapter 2: Analysis of Maximum Power Point and MPPT-Techniques

Figure 2.1: Basic classification of tracking techniques.....17

Figure 2.2: Flowchart of the Perturb and Observe (P&O) algorithm.....19

Figure 2.3: P-V curve of the PV array at shading pattern.....20

Chapter 3: Fundamental of Neural Networks NN

Figure 3.1 Relationship between biological and artificial neuron.....26

Figure 3.2 A single-input artificial neuron.....27

Figure 3.3 Rosenblatt's Perceptron model (multiple-input neuron) 29

Figure 3.4 MLP architecture (a)- With single Hidden layer (b)- With two Hidden layers.....30

Figure 3.5 Signal-flow graph highlighting the details of output neuron j.....32

Figure 3.6 Signal-flow graphs highlighting the details of output neuron **k** connected to hidden neuron**j**.....35

Figure 3.7 Signal-flow graph of a part of the adjoint system pertaining to back-propagation of error signals.....36

Figure 3.8 Architecture of RBNN.....38

Figure 3.9 Flowchart of working procedure of Radial Basis Functions (RBF) 39

Figure 3.10 Diagram of the proposed model.....41

Figure 3.11 Training results.....42

Figure 3.12 Testing results.....43

Chapter 4: Fundamental of Fuzzy set systems (FSSs)

Figure 4.1. (a) crisp set (b) fuzzy set.....46
Figure 4.2. Fuzzy inference processes.....47
Figure 4.3. Some type of Membership functions curves.....48
Figure 4.4 Curves of MFs (a)- Same MFs trapezoidal (b)- mixed MFs.....48
Figure 4.5 Fuzzy logic for direct control.....49
Figure 4.6 Fuzzy logic for helping the PID controller.....49
Figure 4.7 The Tsukamoto fuzzy model 51
Figure 4.8 The Sugeno (TSK) fuzzy model 52
Figure 4.9 Architecture of two-input first-order Sugeno FIS with two rules.....53
Figure 4.10 Flowchart of training ANFIS system.....60
Figure 4.11 Architecture of proposed model.....61
Figure 4.12 Membership function of the Modal using Generalized Bell 61
Figure 4.13 Result of training the ANFIS model.....62
Figure 4.14 Result of testing the ANFIS model.....63

Chapter 5: Simulation Results, Performance Evaluation, and Validation

Figure 5.1 Characteristic curves (a) I-V curve (b) P-V.....66
Figure 5.2 I-V and P-V curves for fixed temperature **25°** with varying irradiations (**W/m²**) **[200, 400, 600, 800, 1000]**.....66
Figure 5.3 I-V and P-V curves for fixed irradiations **1000W/m²** with varying temperature **(°C) [25, 50,75]**.....67
Figure 5.4 Learning Results of the MLP code predicted VS target.....68
Figure 5.5 Case 1 MLP test (T variable, G variable).....69
Figure 5.6 Case 2 MLP test (T fixed, G variable).....69
Figure 5.7 Case 3 MLP test (G fixed, T variable).....70
Figure 5.8 membership functions (a)-3 MFs of T (b)- 3MFs of G.....71
Figure 5.9 Results of training ANFIS Modal.....72
Figure 5.10 Case 1 ANFIS test (T variable, G variable).....72
Figure 5.11 Case2 ANFIS test (T fixed, G variable).....73
Figure 5.12 Case3 ANFIS test (G fixed, T variable).....73

List of tables

Table1.1: Parameters of the KC200GT PV module.....12
Table 3.1: A variety of activation functions.....27
Table 4.1 Two passes in the hybrid learning procedure for ANFIS.....55
Table 4.2 The ANFIS computational complexity.....56
Table 4.3 Comparative studies of ANFIS with other MPPT techniques.....59
Table 5.1 Comparison between PV-modelled and Kyocera KC 200 GT PV panel.....66
Table 5.2 Member functions and there parameters.....71
Table 5.3 Comparison between implemented Models and mathematical Model.....74

Acronyms

Acronyms

AI	Artificial Intelligence
ANFIS	Adaptive Network-Based Fuzzy Inference System
ANN	Artificial Neural Network
BNN	Biological Neural Networks
BP	Back-propagation
BPNN	Back-propagation Neural Network
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FLC	Fuzzy Logic control
FSS	Fuzzy Set System
Isc	short circuit Current
GA	Genetic Algorithm
GMPPT	Global Maximum Power Point
GP	Global Peak
Impp	maximum power point current
IncCond	Incremental Conductance
LMS	Least Mean Squares
LSE	Least Squirt Estimator
MFs	Membership Functions
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLPNN	Multi-Layer Perceptron Neural Network
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
MSE	Mean-Squared Error
P&O	Perturb and Observe.
PSCs	Partial Shading Conditions
PSO	Particle Swarm Optimization
PV	Photovoltaic
RBF	Radial Basis Function
RBNN	Radial basic neural network
RLSE	Recursive Least Squirt Estimator
RMS	Root Mean Square

RMSE	Root Mean Square Error
SD	Steepest descent
STC	Standard Test Conditions
TCT	Total Cross Tied
TSK	Takagi-Sugeno-Kang (or just Sugeno)
Vmpp	maximum power point Voltage
Voc	open circuit Voltage

General introduction

General introduction

In recent decades, renewable and environmentally friendly energies have become critically important, not only to replace conventional fossil fuels or other terrestrial sources, but also to mitigate the catastrophic effects of their irrational and irresponsible use **[1].**

Extensive research has been conducted in this vast field to make this type of energy more cost-effective and controllable by being able to track the maximum power point with accuracy and speed for the best benefit. The major challenge with this energy source is that it is influenced by stochastic and unpredictable conditions, making its control a significant hurdle.

For example, there has been a number of algorithms proposed in literature Blanco et al., 2001, Chen et al., 2006, Grena, 2008]which increase the precision without incurring in high computational efforts**[2].**

Among these sources, solar energy is receiving the most research attention because it is the most promising. The primary research challenge lies in tracking the maximum power point. Numerous methods have been developed, such as Particle Swarm Optimization (PSO), Perturb and Observe (P&O), and Incremental Conductance (IncCond). However, based on reported experiments and studies, researchers have shifted toward hybrid methods. By combining classical methods with artificial intelligence, these hybrid approaches give the system the ability to predict, estimate, and make decisions, ultimately improving its accuracy, speed, and robustness **[3].**

The most powerful hybrid models proposed are based on fuzzy logic because it resembles the way human brains perceive reality, seeking to extract the core value of information rather than focusing on the unnecessary details of the subject itself. It has provided a major boost to enhancing the role of computational and storage systems in decision-making. However, prior to this, machines first needed the ability to learn capability made possible by the creation of Artificial Neural Networks (ANN). These architectures imitated biological neural networks, beginning with the Perceptron and evolving into the multilayer perceptron, which marked the birth of Machine Learning and initiated the broader field of Artificial Intelligence **[4].**

In 1993, Jang proposed a modal combines between Fuzzy logic and neural network, it is the creation of adaptive neural fuzzy inference system (ANFIS), that innovation model gave a great push for developing fuzzy-based models able to estimate or control systems with great accuracy, fast and stable to the dynamic situations **[5]**.

The objective of our work is improving the ability of benefit the most of the energy stored in photovoltaic panels, by following this approach in modeling a photovoltaic system with an intelligent maximum power point tracker based on ANFIS, that hybrid method combines two

of the most powerful intelligent techniques able to mix between the ability of learning (using neurons) and logical inference presented by the fuzzy logic.

To address this issue, we followed the work plan below:

Chapter 1: provides an overview of photovoltaic panels and their energy characteristics, including the plotting of I-V and P-V curves. We then model a photovoltaic system, using the Kyocera KC200GT as a case study, and conclude by analyzing how temperature and irradiation affect the panel's power output.

Chapter 2: In this chapter, we study the classical techniques of Maximum Power Point Tracking (MPPT). Starting with the definition of the maximum power point and the effects of the environment on it, we then describe classical MPPT methods such as **PSO**, **P&O**, and **IncCond,** discuss their limitations, and explain why intelligent control can be more effective.

In Chapter 3, we present a fundamental study on neural networks (**NNs**), covering their descriptions and architectures. We also discuss the Perceptron and two commonly used NN models: the Multilayer Perceptron (**MLP**) and the Radial Basis Function Neural Network (**RBFNN**), also known as the Radial Basis Function (**RBF**). Finally, we conclude the chapter with a practical presentation of the proposed **MLP-based MPPT**.

Chapter 4 presents a fundamental study on fuzzy logic, starting with a description of the fuzzy set system (**FSS**). Then, we move to the Fuzzy Inference System (**FIS**) and its common methods: the Mamdani, Sugeno, and less common Tsukamoto fuzzy models. Following this, we introduce the Adaptive Neuro-Fuzzy Inference System (**ANFIS**), concluding the chapter by presenting our Fuzzy-based **MPPT** using the **ANFIS** model.

Chapter 5 is dedicated to the MATLAB simulation of the proposed MLP-based and ANFIS-based MPPT models. It analyzes their performance and response to dynamic conditions, concluding with a discussion and validation of the results.

Chapter 1:

Modelling of Photovoltaic Systems Based on the One-Diode Model

2026

1.1. Introduction

The first chapter outlines the fundamental concepts associated with photovoltaic (PV) system analysis and modelling. First, the basic principles of photovoltaic operation are described in detail, along with the process by which solar energy is directly converted into electrical energy using semiconductor materials. In addition, the electrical characteristics of photovoltaic modules are explained through the analysis of I–V and P–V curves.

Furthermore, Chapter 1 discusses the one-diode model, which is widely used for the characterization and modelling of solar cells and PV modules. The mathematical modelling of the KC200GT photovoltaic module is carried out using MATLAB software in order to analyze its performance. Finally, the effects of environmental factors such as solar irradiance and temperature on the performance of the PV module are investigated.

1.2. Historical background

The photovoltaic effect was discovered by Alexandre Edmond Becquerel in 1839 during his experiments with metal electrodes suspended in an electrolyte. Becquerel noticed an increase in the electrical conductivity of the device under light conditions, which represented the first observation in the history of photovoltaic panel. The next significant discovery in the field of PV occurred in 1873 when Willoughby Smith discovered the photoconductive properties of selenium. As a result, research was focused on light-sensitive semiconductor materials. The next milestone was accomplished in 1876 when William Grylls Adams and his student Richard Day managed to show that selenium produced electricity when exposed to light**.**

A considerable step forward was made in the twentieth century when research in the area of semiconductor physics gave rise to silicon PV cells. In 1954, the first silicon PV cell with efficiency up to 6% was developed by Bell Laboratories. Since then, various modifications and innovations were introduced into solar cell technology, increasing efficiency and lowering manufacturing costs. Today, photovoltaic systems are widely applied for the generation of electrical power both in residential buildings, industrial facilities, and in outer space **[6]**.

1.3. Fundamentals of photovoltaic effect

1.3.1 Principles of Photovoltaic Conversion

The photovoltaic effect is based on the interaction between light and semiconductor materials. When light falls on a material, the photons excite the electrons to reach a higher energy

state inside the material. This happens particularly if the energy of the light photons exceeds the forbidden band gap (Band Gap) of the semiconductor exposed to the light.

However, the excited electrons do not maintain their energy state and return to their natural or ground state. But in the solar cell or the photovoltaic device, there is a built-in asymmetry resulting from the doping process (Doping), which prevents the electrons from returning to their original state by pulling them away, and directs them to feed an external circuit.

A voltage difference or electromotive force (e.m.f) is generated from the excess energy of the excited electrons, and through this difference, electrical current is generated **[7]**.

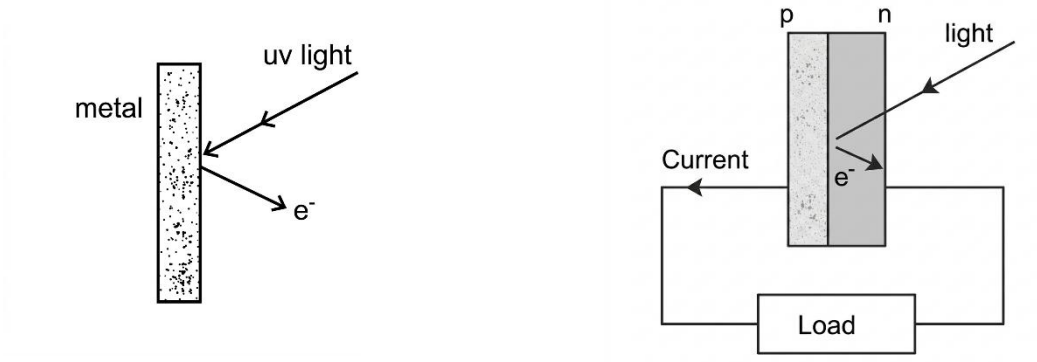


Figure1.1 Comparison between PE effect (left) and PV effect in a solar cell (right**).**

Comparison of the photoelectric effect (left), where uv light liberates electrons from the surface of a metal, with the photovoltaic effect in a solar cell (right). The photovoltaic cell needs to have some spatial asymmetry, such as contacts with different electronic properties, to drive the excited electrons through the external circuit Figure.(1.1).

The solar cell is the key component in photovoltaic devices. In terms of operation, the cell acts as a two-terminal device where illumination from the sun results in photo voltage generation, while dark conditions result in diode conduction.

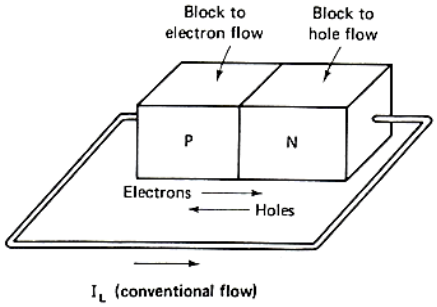


Figure1.2 Asymmetrical properties of a P-N junction diode**.**

These cause a net current flow in an external lead connecting the P-type and N-type regions when the junction Is illuminated.

Under illumination conditions, there will be a current flow within the external circuit that connects the P-region with the N-region.

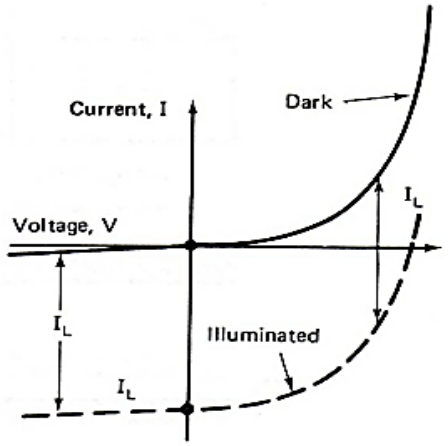


Figure 1.3 I-V characteristics of a solar cell under Dark and Illuminated conditions**.**

This light-generated current is superimposed upon the normal rectifying current-voltage characteristics of the diode. This results in a region in the fourth quadrant where electrical power can be extracted from the device.

The photo-induced current is added to the normal voltage-current relationship of the diode. Energy can be obtained from the device in the area associated with the fourth quadrant **[8]**

1.3.2 Fundamentals of Solar Cells

The solar cell is the basic building block of solar photovoltaic. When illuminated by sunlight, this basic unit generates a dc photo voltage of 0.5 to 1.0V and, in short circuit, a photocurrent of some tens of mA/cm2. Since the voltage is too small for most applications, to produce a useful voltage, the cells are connected in series into modules, typically containing about 28 to 36 cells in series to generate a dc output of 12 V. To avoid the complete loss of power when one of the cells in the series fails, a blocking diode is integrated into the module. Modules within arrays are similarly protected to form a photovoltaic generator that is designed to generate power at a certain current and a voltage which is a multiple of 12 V **[7].**

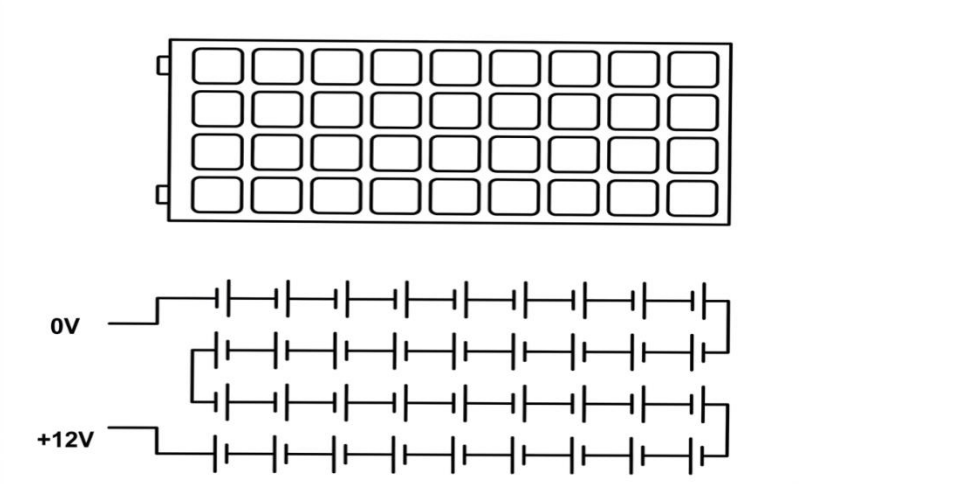


Figure1.4 Association of cells to obtain a photovoltaic module

In a module, solar cells are typically connected in series to obtain a standard DC output voltage of 12 V figure 1.4.

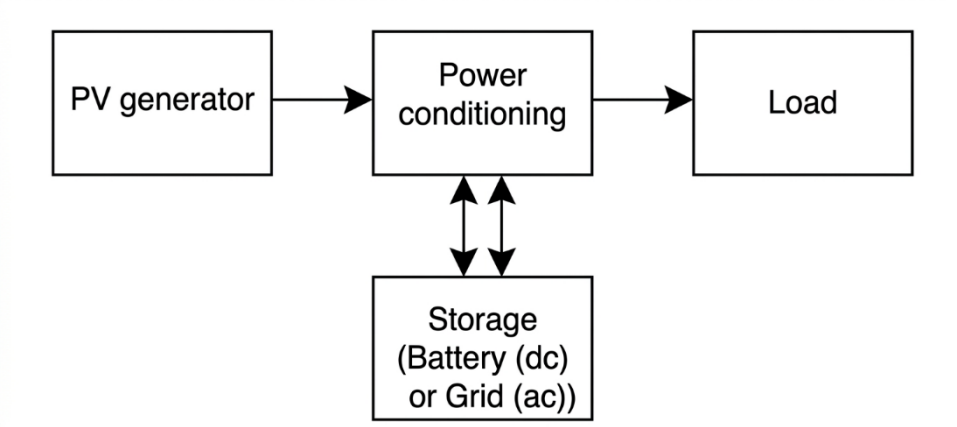


Figure1.5 Block diagram of a typical photovoltaic (PV) system

In most standalone photovoltaic systems, the array is integrated with charge regulation and energy storage components to ensure proper energy management.

1.4. Electrical characteristics of PV modules

Nomenclature

The operation of PV modules is determined by certain basic electrical characteristics that describe their behaviour under specific operating conditions. It is necessary to analyze the above-mentioned electrical characteristics when assessing how efficiently the module transforms the energy received from solar radiation. The following electrical parameters of PV modules should be considered:

- **STC:** Standard Testing Conditions
- **SRC:**Standard Reference Conditions.
- **I:** Current flow through PV panel [A]
- **q:**Elementary charge
- $\mathbf{I_{Ph}}$:Panel photo-generation current [A]
- $\mathbf{I_{SC}}$: Short-circuit current [A]
- $\mathbf{I_{mpp}}$:Peak power current of PV panel[A]
- $\mathbf{I_{irr}}$:the irradiance current or the photo-current,[A]
- $\mathbf{I_{dio}}$**:**current passing through the anti-parallel diode,[A]
- $\mathbf{I_p}$:The shunt current,[A]
- $\mathbf{R_p}$**:**shunt resistance, [Ω]
- $\mathbf{R_s}$:series resistance [Ω]
- $\mathbf{V_{mpp}}$:Peak power voltage [V]
- **Voc**:Open-circuit voltage[V]
- **K**:is the Boltzmann constant,
- **n:**is the ideal constant of the diode,
- **T:**is thetemperature,
- **P:**Delivered power of PV panel [W]
- **Pmpp:** Peak power of PV panel [W]

Some useful definitions:

The performance of a photovoltaic module is mainly described by its current-voltage (I-V) and power-voltage (P-V) characteristics. The most important electrical parameters are defined as follows:

- **Short-Circuit Current (Isc)**: The current produced by the module when its terminals are short-circuited (voltage = 0). It is directly proportional to solar irradiance. figure1.7
- **Open-Circuit Voltage (Voc)**: The maximum voltage produced by the module when no load is connected (current = 0). It is mainly affected by cell temperature. figure1.7 figure1.8.
- **Maximum Power Point (Pmpp)**: The point at which the module delivers its highest power output. figure1.8

- **Voltage at Maximum Power Point (Vmpp)** and **Current at Maximum Power Point (Impp)**: The voltage and current values at which Pmpp occurs figure1.7.
- **Fill Factor (FF)**: A measure of the quality of the solar cell, defined as the ratio of the maximum power to the product of Voc and Isc.

These parameters are typically provided by the manufacturer under **Standard Test Conditions (STC)**: 1000 W/m² irradiance, 25°C cell temperature, and AM 1.5 solar spectrum.

1.5. Single-Diode Equivalent Circuit Model

The electronic component responsible for converting solar energy into electrical energy is known as a solar cell. A photovoltaic (PV) module consists of several solar cells connected in series to provide a higher output voltage. In photovoltaic systems, modules can also be connected in both series and parallel configurations depending on the required voltage and current levels. Most of the mathematical models developed for photovoltaic systems are based on the current–voltage relationship of the single-diode model, as illustrated in the following figure, which assumes that the characteristics of the PV cell can be represented by a single-diode mechanism **[9].**

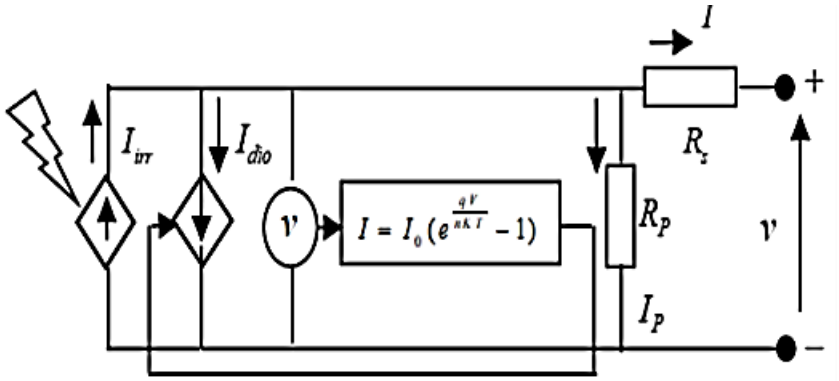


Figure1.6.Single-diode model of the theoretical PV cell including series and shunt resistances.

A single-diode model of the theoretical photovoltaic cell and the equivalent circuit of a practical photovoltaic device including series and shunt resistances.

By the Kirchhoff's law of current.

$$I = I_{irr} - I_{dio} - I_p \qquad (1.1)$$

The solar cell's non-linear properties are produced by the current passing through the anti-parallel diode I_{dio} where

$$I_{dio} = I_o\left(\exp\left(\frac{q(v+IRs)}{nKT}\right) - 1\right) \qquad (1.2)$$

The shunt current Ip given to the branch of the shunt resistor$R$p is given as follow:

$$Ip = \left(\frac{v+IRs}{Rp}\right) \quad (1.3)$$

And then we have:

$$I = I_{irr} - Io\left(\exp\left(\frac{q(v+IRs)}{nKT}\right) - 1\right) - \left(\frac{v+IRs}{Rp}\right) \quad (1.4)$$

1.5.1 Photovoltaic module current-voltage relationship

A photovoltaic (PV) module generally consists of several solar cells connected in series. The parameter N_S represents the number of solar cells connected in series within a single module. When N_S solar cells are connected in series to form a module, the output voltage **V** and output current **I** of the module are expressed by the following relationships **[8]:**

$$I = Iph - I_0\left(\exp\left(\frac{q(V+IRs)}{NsnKT}\right) - 1\right) - \left(\frac{V+IRs}{Rp}\right) \quad (1.5)$$

1.5.2 Basic module's parameters

For the Saturation current at operating temperature I_0:

$$I_0 = I_0 ref\left(\frac{T}{Tref}\right)^3 \times \exp\left(\left(\frac{q \times Egref}{n \times K}\right)\left(\frac{1}{Tref} - \frac{1}{T}\right)\right) \quad (1.6)$$

- T: Cell temperature in its actual state (K)
- Tref: Cell temperature at 25
- $I_{0,ref}$: The third undefined model's parameter is the saturating diode current during cell temperature at SRC.
- Eg: the band gap energy [eV] defined as:

$$Eg = 1.170 - 4.73 \times 10^{-4} \times \left(\frac{T^2}{T+636}\right) \quad (1.7)$$

This is the Varshni equation 1.8 **[10]** applied to silicon, with the standard parameters:

$$Eg = E_{g0} - \left(\frac{\alpha T^2}{T+\beta}\right) \quad (1.8)$$

With E_{g0}=1.170 eV, α=4.73 × 10^{-4}eV/K , β=636 K

There for, I_{0ref} (Saturation current at reference temperature):

$$I_0 ref = \frac{Isc}{\left(\exp\left(\frac{Voc}{nNsVT_{ref}}\right) - 1\right)} \quad (1.9)$$

With:

$$VT_{ref} = \frac{KTref}{q} \qquad (1.10)$$

And for Photo-generated current $I_{ph:}$

$$Iph = Isc\left(\frac{G}{Gref}\right) \times \left(1 + \alpha \times (T - Tref)\right) \qquad (1.11)$$

1.5.3. Generated I-V and P-V curves under different operating conditions

Using the calculated model parameters and the developed mathematical equations of the single-diode model, the current-voltage (I-V) and power-voltage (P-V) characteristics of the photovoltaic panel were generated. By numerically solving the implicit nonlinear equation for different values of solar irradiance and cell temperature, the complete family of I-V and P-V curves was obtained **[11].**

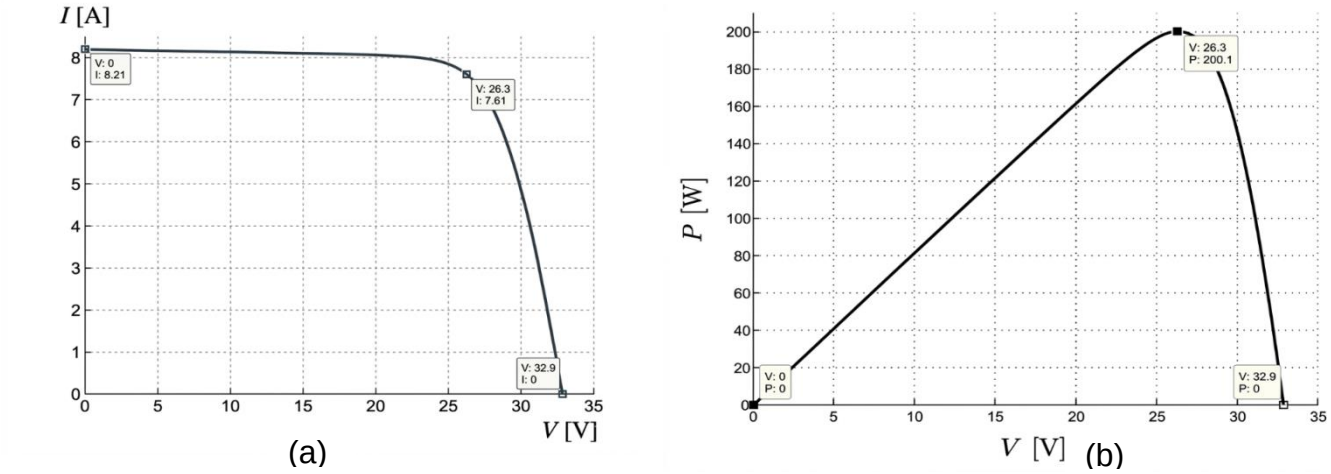


Figure1.7: Characteristics of PV Module: (a) Current-Voltage I-V (b) Power-Voltage P-V

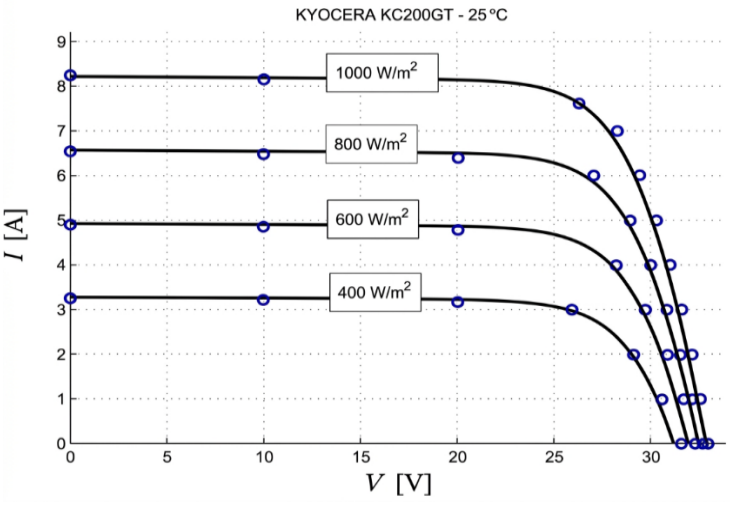


Figure 1.8: I-V Characteristics of PV Module for fixed T=25°C and different irradiations

1.6. Modelling and Simulation of the KC200GT Photovoltaic Module

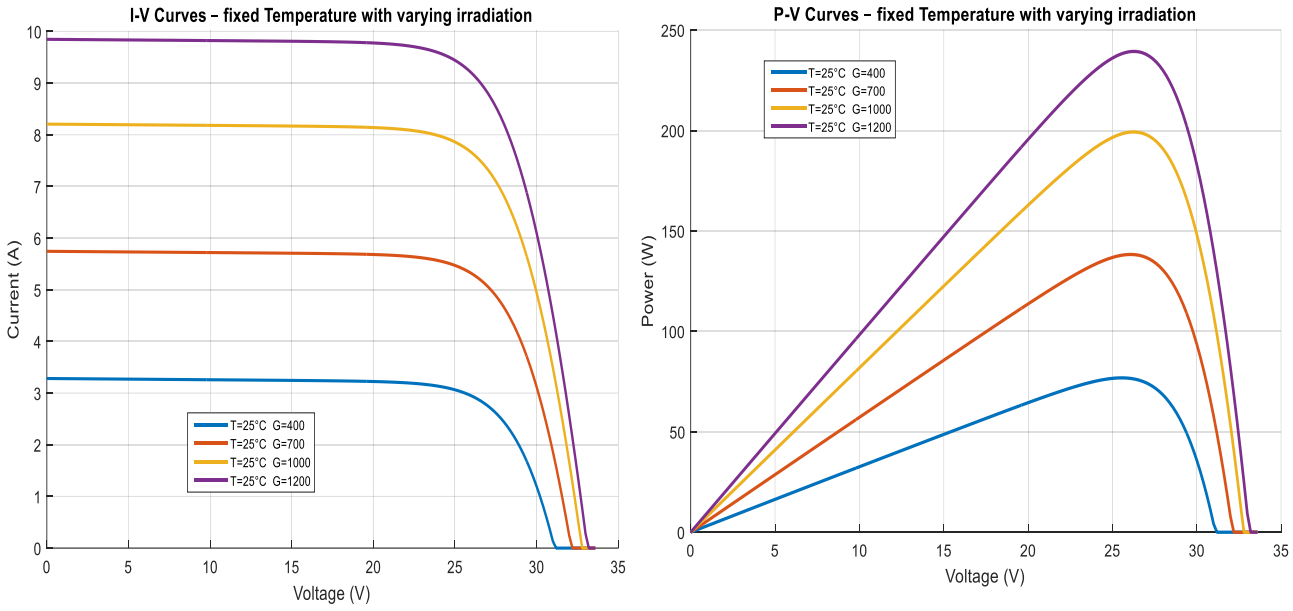
Using the single-diode equivalent circuit model described in the previous section, a MATLAB program was developed to simulate the electrical behaviour of the KC200GT photovoltaic module. The program solves the implicit nonlinear current-voltage equation numerically for each. combination of solar irradiance and cell temperature.

The simulation was performed by defining vectors of different operating conditions:

- Cell temperatures: [25°C, 75°C]
- Solar irradiance levels: [400,1200 W/m²]

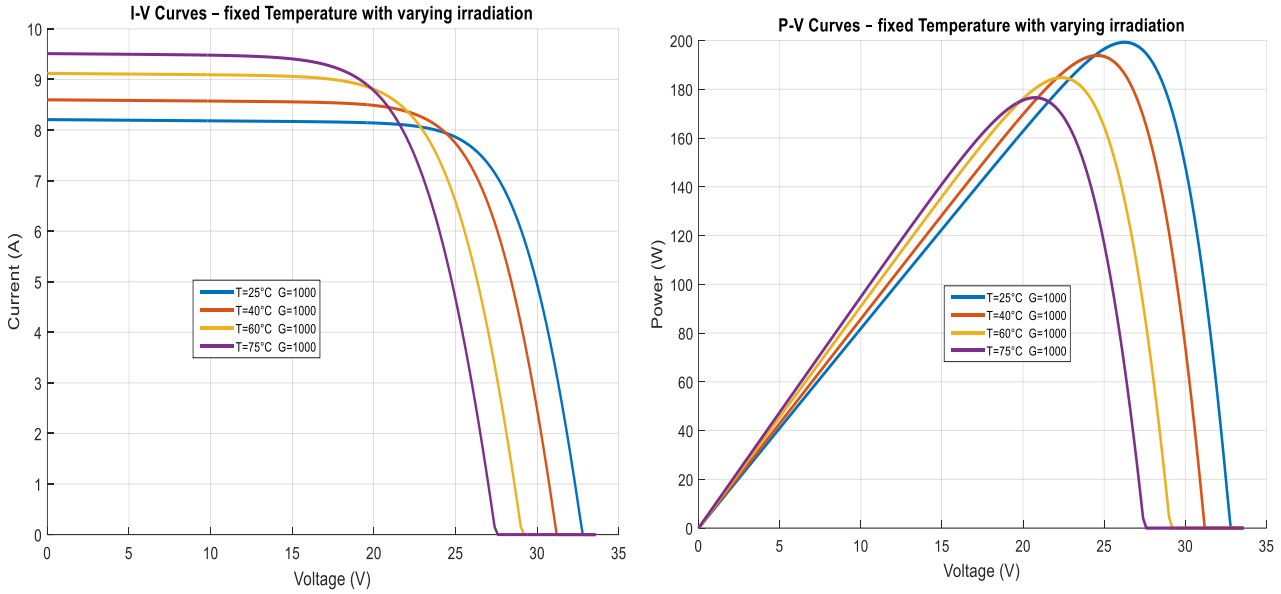
For each pair of temperature and irradiance, the output current I was calculated across a wide range of voltages using a numerical solver (**fsolve**). The corresponding power P=V×I was then computed to generate the complete I-V and P-V characteristic curves.

This numerical approach allows the generation of a comprehensive family of curves that accurately represent the module's performance under various real-world operating conditions. The results are presented in the following figures 1.9 and 1.10:



(a) (b)

Figure1.9. Characteristics of PV Module for fixed T=25° and different irradiations values (a) I-V curves (b) P-V curves.



(a) (b)

Figure.1.10 Characteristics of PV Module for fixed G=1000 W/m² and different temperatures values (a) I-V curves (b) P-V curves.

The simulation was carried out using the following parameters of the KC200GT PV module:

Ns	54
Vmpp	26.3V
Impp	7.61A
Pmax	200W
Isc	8.21A
Voc	32.9V
Rs	415.405Ω
Rp	0.221Ω

Table1.1: Parameters of the KC200GT PV module

We can clearly see that the curves represent the electrical characteristics perfectly of the module of the PV panel KC200GT, we can then import the needed data base to learning our intelligent modals to simulate the panel and tracking the MPP.

In the previous chapters we will study and modelling this intelligent modals.

1.7 Influence of temperature and irradiance on PV performance

From the simulated I-V and P-V characteristic curves, the influence of solar irradiance and cell temperature on the performance of the photovoltaic module can be clearly observed.

It is noticed that increasing the solar irradiance leads to a significant rise in the short-circuit current (Isc) and the maximum power output, while the open-circuit voltage (Voc) increases only slightly. As a result, the I-V curves shift upward and the P-V curves exhibit a notable increase in peak power.

Conversely, the effect of temperature is more pronounced on the voltage. As the cell temperature increases, a clear reduction in the open-circuit voltage and a slight increase in the short-circuit current are observed. This leads to a decrease in the maximum power point, causing the P-V curves to flatten and shift toward lower power values.

These observations confirm that solar irradiance has a dominant positive effect on the generated current and power, while temperature has a negative impact, primarily through the reduction of voltage. The combined variations of both parameters play a crucial role in determining the actual output of the PV module under real operating conditions.

The greatest challenge in implementing these models is reacting to dynamic internal and external conditions—such as sudden changes in irradiance and temperature—with perfect accuracy and speed.

1.8 Conclusion

This chapter has presented the fundamental concepts and mathematical modeling of photovoltaic systems. The basic principles of the photovoltaic effect were explained, starting from the historical development to the physical mechanism of converting solar energy into electrical energy using semiconductor materials. The electrical characteristics of PV modules were analyzed through the I-V and P-V curves. A comprehensive single-diode equivalent circuit model was developed and implemented for the KC200GT photovoltaic module. All essential model parameters were calculated, and the nonlinear governing equations were solved numerically in MATLAB under various operating conditions of solar irradiance and cell temperature. The generated characteristic curves successfully demonstrated the nonlinear behavior of the PV module.

The simulation results clearly showed that solar irradiance has a strong positive influence on the generated current and maximum power, while cell temperature has a negative impact, primarily

by reducing the voltage. These findings confirm the accuracy and reliability of the developed mathematical model.

The mathematical model and simulation results established in this chapter provide a solid foundation for the subsequent analysis of the Maximum Power Point and the design of intelligent identification techniques in the following chapters.

Chapter 2:
Analysis of Maximum Power Point and MPPT-Techniques

2.1 Introduction

The first chapter presented the fundamental principles of photovoltaic systems and developed a comprehensive single-diode mathematical model for the KC200GT photovoltaic module. Through numerical simulation, the I-V and P-V characteristic curves were generated under various irradiance and temperature conditions, providing a clear understanding of the nonlinear behaviour of the PV module.

Building upon this foundation, the power delivered by a photovoltaic (PV) module is highly nonlinear and varies significantly with changes in solar irradiance and cell temperature. Under any specific operating condition, there exists only one unique point on the power-voltage (P-V) characteristic curve at which the module generates its maximum possible power. This optimal operating point is known as the Maximum Power Point **(MPP)**.

Due to the continuous variation of environmental conditions in real-world applications, the location of the MPP keeps shifting. Therefore, it is essential to continuously track this moving point in order to maximize the energy harvested from the PV system. This process is called Maximum Power Point Tracking **(MPPT) [12].**

This chapter provides a detailed analysis of the Maximum Power Point concept. It begins by defining the MPP mathematically, examines the effects of solar irradiance and temperature on its position, reviews the most common classical MPPT techniques, highlights their limitations, and finally establishes the motivation for adopting intelligent techniques.

2.2 Definition of Maximum Power Point (MPP)

The photovoltaic system exhibits nonlinear current–voltage (I–V) and power–voltage (P–V) characteristics that vary according to solar irradiance and temperature conditions. For a given operating condition, there exists a unique operating point on the P–V characteristic curve at which the photovoltaic module delivers its maximum output power. This operating point is known as the Maximum Power Point (MPP). At the MPP, the photovoltaic system operates with maximum efficiency, producing the highest possible electrical power.

$$Pmax = Vmpp \times Impp \qquad (2.1)$$

Where Vmpp and Impp represent the voltage and current at the maximum power point, respectively.

The position of the Maximum Power Point is not constant, since it changes continuously with variations in environmental conditions such as solar irradiation, temperature, and load variations.

Therefore, maintaining the operation of the photovoltaic system at the MPP is considered an essential requirement for improving energy conversion efficiency and maximizing the extracted power from the PV module.

When a photovoltaic array is directly connected to a load, the operating point is determined by the intersection between the I–V characteristic of the PV array and the load line. In most cases, this operating point does not coincide with the Maximum Power Point, resulting in power losses and reduced system efficiency. Consequently, several Maximum Power Point Tracking (MPPT) techniques have been developed to continuously adjust the operating point of the photovoltaic system and ensure operation near the MPP under different environmental conditions **[13][14].**

2.3 Effect of environmental conditions on MPP

Environmental conditions have a significant influence on the performance and operating characteristics of photovoltaic systems. The position of the Maximum Power Point (MPP) is not fixed and changes continuously according to variations in solar irradiance, temperature, and shading conditions. These environmental factors directly affect the output voltage, current, and generated power of the photovoltaic module.

- ⇒ **Solar irradiance** mainly affects the output current of the PV module. An increase in irradiance causes an increase in the photocurrent generated by the solar cells, resulting in higher output power. Conversely, a decrease in irradiance reduces the generated current and shifts the Maximum Power Point to a lower power level.
- ⇒ **Temperature** also has a considerable effect on photovoltaic performance. As the temperature increases, the output voltage of the PV module decreases significantly, while the current increases slightly. Consequently, the overall output power and conversion efficiency of the photovoltaic module decrease at high temperatures. Therefore, photovoltaic systems generally operate more efficiently under moderate temperature conditions.
- ⇒ **Partial shading** is another important factor affecting the Maximum Power Point. Shading may occur due to clouds, nearby buildings, trees, dust accumulation, or other obstacles that block sunlight from reaching certain cells or modules. Under partial shading conditions, the P–V characteristic may exhibit multiple local maximum power points instead of a single MPP. This phenomenon makes the tracking process more complex and may reduce the

overall efficiency of the photovoltaic system. For this reason, advanced Maximum Power Point Tracking (MPPT) techniques are often required to accurately identify the global maximum power point under varying environmental conditions.**[15]**

2.4 MPPT techniques

Maximum Power Point Tracking (MPPT) techniques are widely used in photovoltaic systems to maximize the extracted power from PV modules under varying environmental conditions. Since the Maximum Power Point changes continuously with variations in solar irradiance, temperature, and load conditions, photovoltaic systems cannot maintain operation at the optimal point without a dedicated control technique.

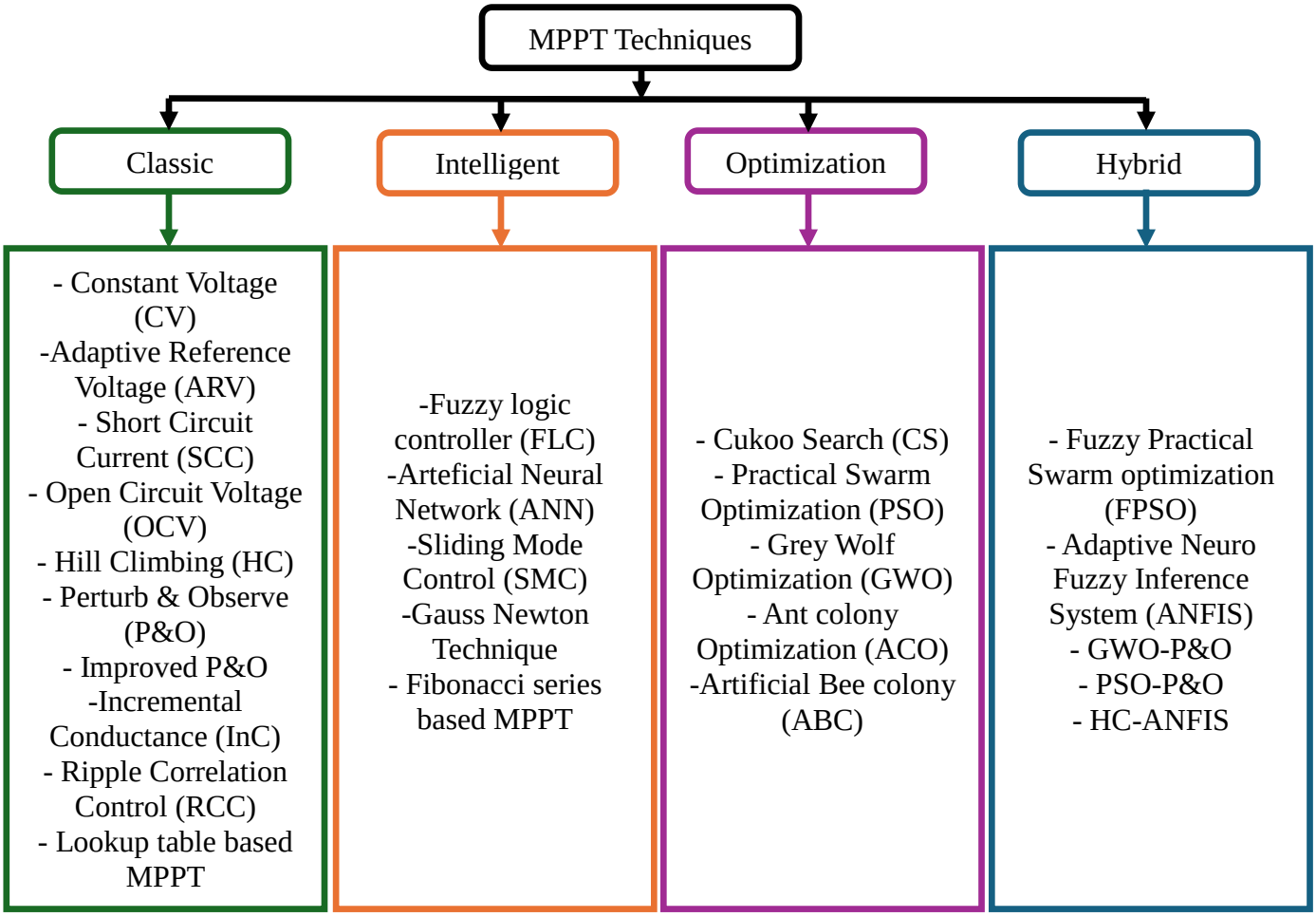


Figure 2.1: Basic classification of tracking techniques **[16].**

MPPT techniques are designed to continuously track and adjust the operating point of the photovoltaic system in order to ensure operation near the Maximum Power Point. These

techniques improve the overall efficiency of the photovoltaic system and reduce power losses caused by environmental variations. Several MPPT methods have been proposed in the literature.

2.4.1 Fixed Duty Cycle Method

The fixed duty cycle technique is the simplest MPPT method. It operates by applying a constant duty cycle to the DC-DC converter, which corresponds to a fixed operating point assumed to be near the Maximum Power Point (MPP) under Standard Test Conditions (STC).

This method does not require any feedback sensors or real-time measurements. The duty cycle is determined once during system design or initial calibration and remains unchanged during operation **[13].**

Limits of the Fixed Duty Cycle Method:

- It cannot adapt to changing environmental conditions (irradiance and temperature).
- The operating point quickly deviates from the actual MPP when weather conditions change.
- It results in significant power loss, especially under varying solar irradiance and temperature.
- Very low efficiency in real outdoor conditions compared to other MPPT techniques **[13].**

2.4.2 Perturb and Observe (P&O) Method

The Perturb and Observe (P&O) algorithm is one of the most widely used MPPT techniques due to its simple structure and ease of implementation. It operates by periodically perturbing the operating voltage of the PV array and observing the resulting change in power.

The algorithm works as follows: the PV array voltage is perturbed by a small step size (ΔV). After each perturbation, the change in power (ΔP) is calculated. If the power increases ($\Delta P > 0$), the perturbation direction is maintained for the next step. Conversely, if the power decreases ($\Delta P < 0$), the direction of the perturbation is reversed. This process continues until the operating point reaches the vicinity of the Maximum Power Point (MPP) **[17][18].**

The basic logic of the P&O algorithm can be summarized as:

- If $\Delta P > 0$ and $\Delta V > 0 \rightarrow$ Continue perturbing in the same direction.
- If $\Delta P > 0$ and $\Delta V < 0 \rightarrow$ Reverse the perturbation direction.
- If $\Delta P < 0$ and $\Delta V > 0 \rightarrow$ Reverse the perturbation direction.
- If $\Delta P < 0$ and $\Delta V < 0 \rightarrow$ Continue perturbing in the same direction.

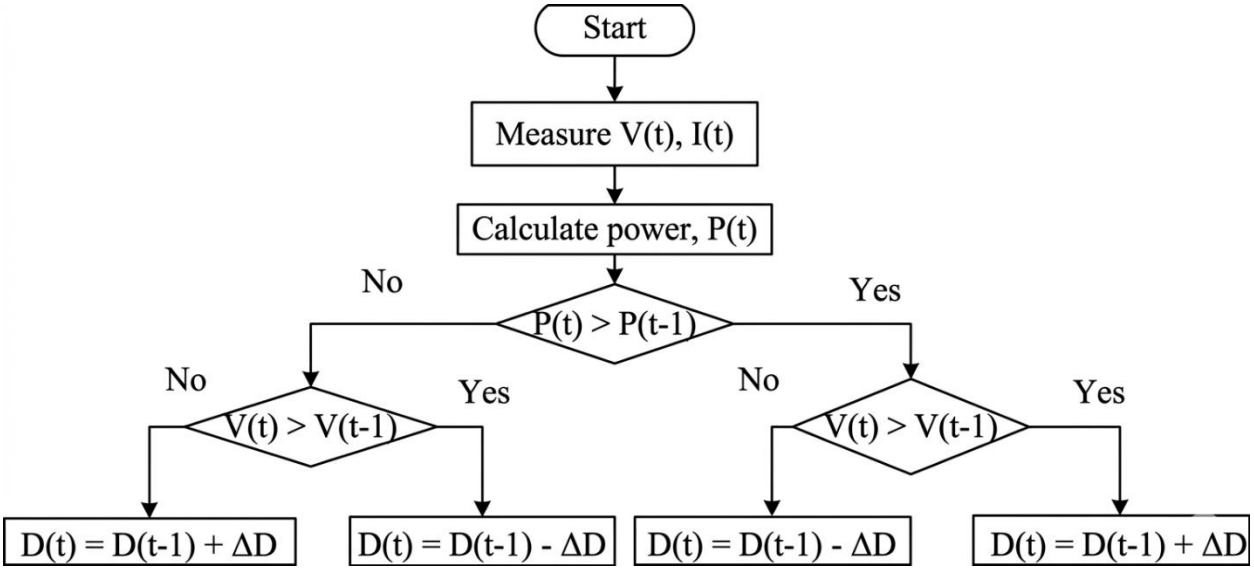


Figure 2.2: Flowchart of the Perturb and Observe (P&O) algorithm **[13]**

Disadvantages of the P&O Method:

Although the P&O algorithm is simple and easy to implement, it has several notable limitations:

- **Steady-state oscillations**: The algorithm continuously oscillates around the MPP even under constant irradiance, resulting in power loss.
- **Poor performance under rapidly changing irradiance**: When solar radiation changes quickly (e.g., on partly cloudy days), the P&O method can lose track of the MPP and move in the wrong direction, significantly reducing tracking efficiency.
- **Trade-off between tracking speed and accuracy**: A large perturbation step size improves tracking speed but increases oscillations, while a small step size reduces oscillations but slows down the response to changing conditions.
- **Inability to distinguish between irradiance change and perturbation effect**: This leads to erratic behavior during fast-changing weather conditions.
- **Partial shading**: Under partial shading, the P-V curve becomes more complex with multiple local maxima (peaks). The conventional P&O algorithm can easily get trapped at a local maximum instead of tracking the global maximum power point. This results in a substantial reduction in the extracted power. Furthermore, during fast-changing environmental conditions (such as passing clouds), the algorithm often fails to distinguish

between the effects of the perturbation and the actual change in irradiance, causing it to diverge from the true MPP **[13].**

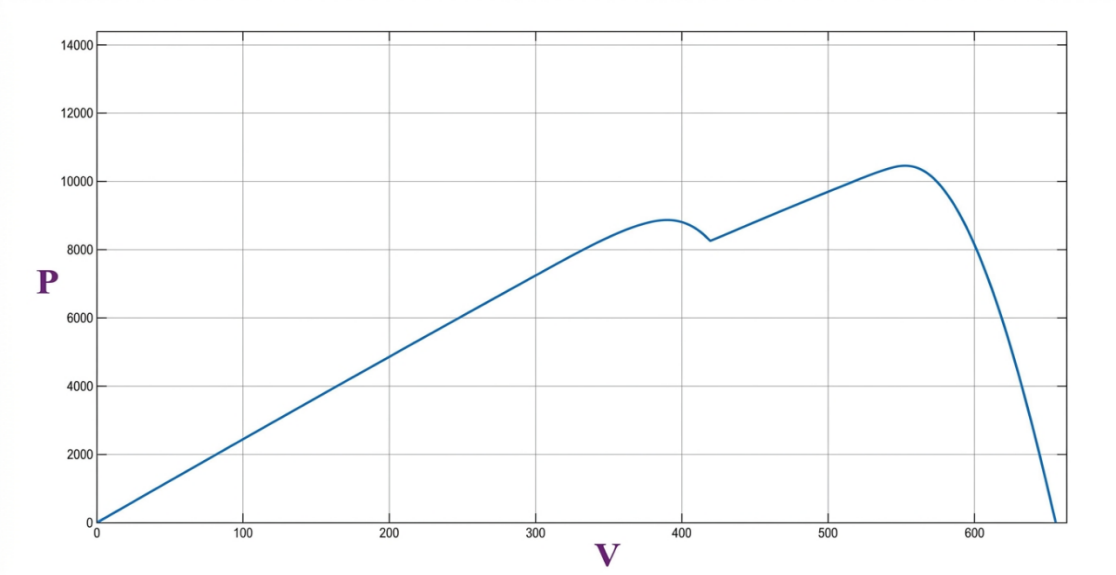


Figure 2.3:P-V curve of the PV array at shading pattern**[15]**

2.4.3 Incremental Conductance (IncCond) Method

The Incremental Conductance (IncCond) algorithm is one of the most widely used classical MPPT techniques. It is based on the fact that the slope of the power-voltage (P-V) curve is zero at the Maximum Power Point (MPP).

The algorithm is derived by differentiating the PV power with respect to voltage and setting the result to zero at the MPP:

$$\frac{dP}{dV} = \frac{d(V \cdot I)}{dV} = I + V\frac{dI}{dV} = 0 \quad \text{at MPP} \qquad (2.2)$$

Which leads to:

$$\frac{dI}{dV} = -\frac{I}{V} \qquad (2.3)$$

Where:

- $\frac{I}{V}$ Is the **instantaneous conductance**
- $\frac{dI}{dV}$ is the **incremental conductance**

At the MPP, these two quantities are equal in magnitude but opposite in sign. This leads to the following decision rules:

$$\frac{dI}{dV} = -\frac{I}{V} \Rightarrow \text{at MPP} \qquad (2.4)$$

$\frac{dI}{dV} > -\frac{I}{V} \Rightarrow$ left of MPP (2.5)

$\frac{dI}{dV} < -\frac{I}{V} \Rightarrow$ right of MPP (2.6)

The IncCond algorithm continuously compares the incremental conductance with the instantaneous conductance to determine the direction in which the operating voltage should be adjusted to reach the MPP. Once the MPP is reached, the algorithm maintains the operating point until a change in irradiance is detected **[13].**

Limits of the Incremental Conductance (IncCond) Method

Despite its advantages in accuracy and fast response under varying conditions, the Incremental Conductance method has several limitations:

- **High computational complexity**: It requires continuous calculation of the incremental conductance ($\frac{dI}{dV}$) and instantaneous conductance ($\frac{I}{V}$), which demands more processing power and memory compared to the simpler P&O method.
- **Need for accurate sensors**: The algorithm is highly dependent on precise and simultaneous measurement of voltage and current. Any noise or error in sensor readings can lead to incorrect decisions.
- **Oscillations around MPP**: Similar to P&O, the IncCond algorithm can cause small oscillations around the MPP due to the discrete nature of the perturbation step size.
- **Poor performance under partial shading**: Like most conventional methods, the standard IncCond struggles with multiple local maxima caused by partial shading conditions and may get trapped at a local MPP instead of the global one.
- **Trade-off between speed and accuracy**: A large step size increases tracking speed but causes more oscillations, while a small step size improves accuracy but slows down the response to rapid changes in irradiance **[13].**

2.4.4 Particle Swarm Optimization (PSO) Method

Particle Swarm Optimization (PSO) is a metaheuristic algorithm, population-based intelligent optimization algorithm inspired by the social behaviour of birds flocking or fish schooling. In the context of MPPT, PSO is used to search for the optimal operating voltage or duty cycle that corresponds to the Maximum Power Point.

In the PSO-based MPPT algorithm, a group of particles (candidate solutions) is initialized with random positions (duty cycles or voltages) and velocities. Each particle represents a potential

operating point. The algorithm iteratively updates the position and velocity of each particle based on two main factors:

- The particle's personal best position (**pbest**)
- The global best position found by the entire swarm (**gbest**)

The velocity and position of each particle are updated using the following equations:

$$v_i^{k+1} = w \cdot v_i^k + c_1 r_1 \left(pbest_i - x_i^k\right) + c_2 r_2 \left(gbest - x_i^k\right) \quad (2.7)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.8)$$

Where:

- v_i^k: Velocity of particle i at iteration k
- x_i^k: Position of particle i at iteration$k$
- w: Inertia weight
- c_1, c_2: Cognitive and social acceleration coefficients
- r_1, r_2: Random numbers between 0 and 1

PSO is particularly effective for MPPT because it can handle **non-linear and multimodal problems**, making it more suitable for partial shading conditions compared to classical methods**[19][20**].

Limits of Particle Swarm Optimization (PSO) Method

Although the PSO algorithm offers good performance in tracking the global Maximum Power Point, especially under partial shading conditions, it has several limitations:

- **High computational complexity**: PSO requires multiple particles and iterative calculations, which increases the processing power and memory requirements compared to classical methods such as P&O and IncCond.
- **Slow convergence speed**: The algorithm may take more time to reach the MPP, especially during rapid changes in environmental conditions.
- **Parameter tuning sensitivity**: The performance of PSO is highly dependent on the proper selection of control parameters (inertia weight w, cognitive coefficient c1 , and social coefficient c2. Poor tuning can lead to slow convergence or premature convergence.
- **Steady-state oscillations**: Similar to other MPPT techniques, PSO can cause small oscillations around the MPP, resulting in some power loss.

- **Implementation complexity**: Due to its population-based nature, PSO is more difficult to implement on low-cost microcontrollers compared to simpler conventional methods.

These drawbacks make PSO more suitable for applications where high accuracy is prioritized over simplicity and low computational cost.

2.5. Motivation for Intelligent MPPT Techniques

Although classical MPPT methods such as Perturb and Observe (P&O), Incremental Conductance (IncCond), and Fixed Duty Cycle are widely used due to their simplicity, they suffer from significant limitations. These include steady-state oscillations around the MPP, slow tracking response under rapidly changing environmental conditions, divergence during sudden irradiance variations, and poor performance under partial shading conditions where multiple local maxima exist.

These drawbacks result in considerable power loss and reduced overall system efficiency. As modern photovoltaic systems are increasingly installed in dynamic and complex environments (urban areas, partially shaded rooftops, or regions with frequent cloud cover), the need for more robust, accurate, and adaptive MPPT techniques has become essential.

To overcome the shortcomings of conventional methods, **intelligent control techniques** have gained considerable attention in recent years. Among these, artificial neural networks (ANNs) particularly Multi-Layer Perceptron (MLP) — and hybrid neuro-fuzzy systems such as **Adaptive Neuro-Fuzzy Inference System (ANFIS)** have shown promising results. These techniques offer excellent nonlinear mapping capabilities, fast response, robustness against parameter variations, and the ability to handle partial shading conditions more effectively than traditional algorithms.

The development of intelligent MPPT techniques based on Multi-Layer Perceptron (MLP) and Adaptive Neuro-Fuzzy Inference System (ANFIS) offers a promising solution to overcome the limitations of conventional methods. These advanced approaches provide superior nonlinear mapping capabilities, enabling higher tracking accuracy, faster convergence speed, and improved energy efficiency under both uniform and non-uniform (partial shading) irradiance conditions.

2.6 Conclusion

This chapter has presented a comprehensive analysis of the Maximum Power Point (MPP) concept and the most common MPPT techniques used in photovoltaic systems. The mathematical

definition of the MPP was introduced, followed by an examination of how solar irradiance and cell temperature significantly affect its location.

Various classical MPPT methods were reviewed, including the Fixed Duty Cycle, Perturb and Observe (P&O), Incremental Conductance (IncCond), and Particle Swarm Optimization (PSO) techniques. While these methods are relatively simple to implement, they exhibit notable limitations such as steady-state oscillations, slow tracking under dynamic conditions, divergence during rapid irradiance changes, and poor performance under partial shading.

These drawbacks highlight the necessity for more advanced and intelligent MPPT approaches. The motivation for adopting intelligent techniques, particularly Multi-Layer Perceptron (MLP) and Adaptive Neuro-Fuzzy Inference System (ANFIS), stems from their ability to handle nonlinearities, adapt to changing environmental conditions, and achieve higher tracking efficiency with reduced oscillations.

The analysis presented in this chapter provides the necessary foundation for the design and development of an intelligent MPPT system based on neural network models and fuzzy logic in the subsequent

Chapter 3 :

Fundamental of Neural Networks NN

3.1. Introduction

In the previous chapter, we studied classical MPPT methods and recognized the need to overcome their limitations. Consequently, the development of artificial intelligence has enabled researchers to create hybrid methods that combine classical approaches with intelligent systems. This improves MPPT performance, making it more accurate, fast, and stable.

Research in the fields of artificial intelligence (AI) and machine learning (ML) has become increasingly prominent, taking on a major role in scientific research and directly impacting our daily lives and decision-making. AI aims to simulate human brain activities at a faster rate, helping to make better and more accurate decisions in a short time. This is achieved by creating complex mathematical models known as Artificial Neural Networks (ANNs), which are derived from the biological neural networks of the human brain.

In this chapter, we will describe ANNs, focusing on two specific models: the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF).

3.2. The Neural Network

The neural networks commonly named artificial neural networks has been motivated right from its inception by the recognition that the human brain computes in an entirely different way from the conventional digital computer.

Of course, the human brain is highly more complex, it has the capability to organize its structural constituents, known as Biological Neural Networks (BNN), so as to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence at least in our day. For example, human vision, which is an information-processing task, It is the function of the visual system to provide a representation of the environment around us and, more important, to supply the information we need to interact with the environment.

In the formative years of neural networks (1943–1958), several researchers stand out for their pioneering contributions, McCulloch and Pitts (1943) for introducing the idea of neural networks as computing machines until Rosenblatt (1958) for proposing the perceptron as the first model for learning with a teacher (i.e supervised learning) **[21].**

S.Haykin (2009) defined ANN as: “A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

3.3. The architecture of an Artificial Neural Networks

The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another; artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes**.**

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Artificial Neural Network primarily consists of three layers:

- **Input Layer**: As the name suggests, it accepts inputs in several different formats provided by the programmer.
- **Hidden Layer (s):** The hidden layer (s) (it can be one or more layers) presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
- **Output Layer:** The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer **[22].**

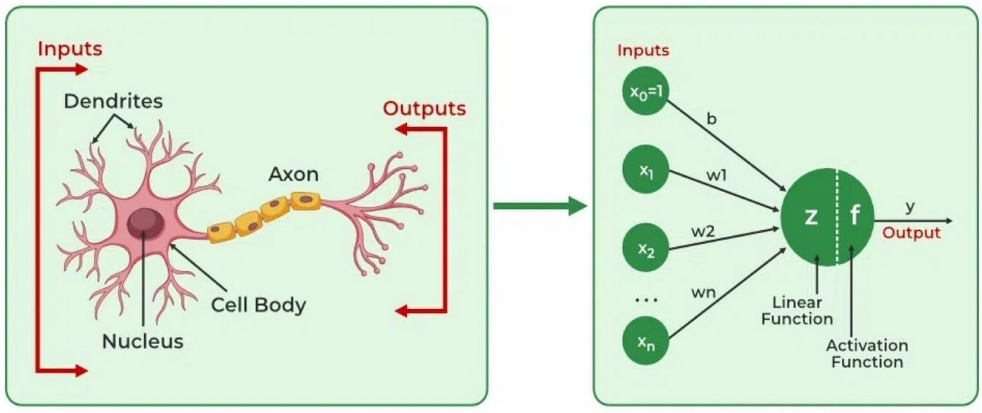


Figure 3.1 Relationship between biological and artificial neuron

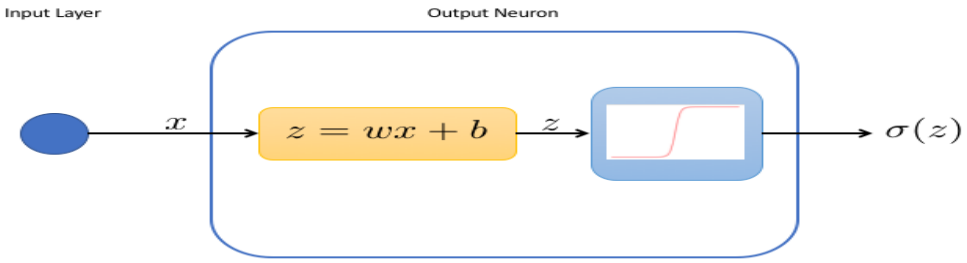


Figure 3.2 A single-input artificial neuron

A single-input neuron (shown in Figure 3.2) is the base of the neuron application with:

x**:** Is the input

w**:** Is the weight

b: Is the bais

z: Is the result of the summing

σ(z): the activation function

The activation function (tab3.1) decides whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose is to introduce non-linearity into the output of a neuron. It's chosen to satisfy some specification of the problem that the neuron is attempting to solve.

Name	Input/output relation	Matlab function
Hard Limit	$\begin{cases} a(n) = 0 & for\ n < 0 \\ a(n) = 1 & for\ n \geq 0 \end{cases}$	hardlim
Symmetrical Hard Limit	$\begin{cases} a(n) = -1 & for\ n < 0 \\ a(n) = 1 & for\ n \geq 0 \end{cases}$	hardlims
Linear	$a(n) = n$	purelin
Saturating Linear	$\begin{cases} a(n) = 0 & for\ n < 0 \\ a(n) = n & for\ 0 \leq n \leq 1 \\ a(n) = 1 & for\ n > 1 \end{cases}$	Satlin
Symmetrical Saturating Linear	$\begin{cases} a(n) = -1 & for\ n < -1 \\ a(n) = n & for\ -1 \leq n \leq 1 \\ a(n) = 1 & for\ n > 1 \end{cases}$	Satlins
Log-Sigmod	$a(n) = \frac{1}{1 + e^{-n}}$	logsig
Hyperbolic Tangent Sigmoid	$a(n) = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$	tansig
Positive Linear	$\begin{cases} a(n) = 0 & for\ n < 0 \\ a(n) = n & for\ n \geq 0 \end{cases}$	poslin

Table 3.1 A variety of activation functions (from MATLAB user)

The most commonly used functions are:

Hard limit activation function $a(n) = \begin{cases} 0 & if\ n < 0 \\ 1 & else \end{cases}$

Linear activation function $a(n) = n$

Log-sigmoid activation function $a(n) = \frac{1}{1-e^{-n}}$

Hyperbolic tangent Sigmoid $a(n) = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$

- **Learning Rate**

The learning rate hyper parameter corresponds to how quickly a neural network model can learn the data patterns. The value of the learning rate ranges from zero to one. If the learning rate is very small, the network operates slowly, and the neurons' weight coefficients are updated slowly. If we set learning rate to a higher value, the network will learn quickly, but too high of a value will make it more unstable, and it may get stuck at local minima. Therefore, this hyper parameter must be optimized. With a suitable learning rate, the loss function (or error) decays with the number of epochs. The nature of decay determines the optimal value of the learning rate. The optimal learning rate depends on the topology of the loss function **[23].**

- **Momentum**

Momentum accelerates the training process. The utilization of momentum in ANN modelling helps to stabilize the test dataset. We must optimize this hyper parameter along with the learning rate. Similar to the learning rate, the momentum value ranges from zero to one. Small values of momentum can dampen the training process. If we assign a small momentum value, the model cannot avoid local minima, and it may get stuck there. Increasing the value of momentum increases the size of the loss function's steps by trying to jump from local minima toward the global minima and makes the training process faster, which is desirable. The higher the value of momentum, the higher the possibility that the network can escape the local minima and reach the global minima. **[24]**

For optimal ANN modelling, momentum and learning rate should have an inverse relationship. If we assign a large value to momentum, we may want to keep the learning rate smaller. If we assign large values to both momentum and learning rate, skipping the global minima with huge steps is possible. In such a case, the loss function will be divergent, and test results will be unstable. It is suggested keeping a lower learning rate and higher momentum in commonly encountered problems in subsurface geosciences.

3.4. Rosenblatt's Perceptron

3.4.1. Definition

A single-input neuron is not useful for problem solution. A modified model of neuron with multiple inputs is used as shown in figure 3.3, it's the modal proposed by **Rosenblatt** in **1958** in the name of **Perceptron.**

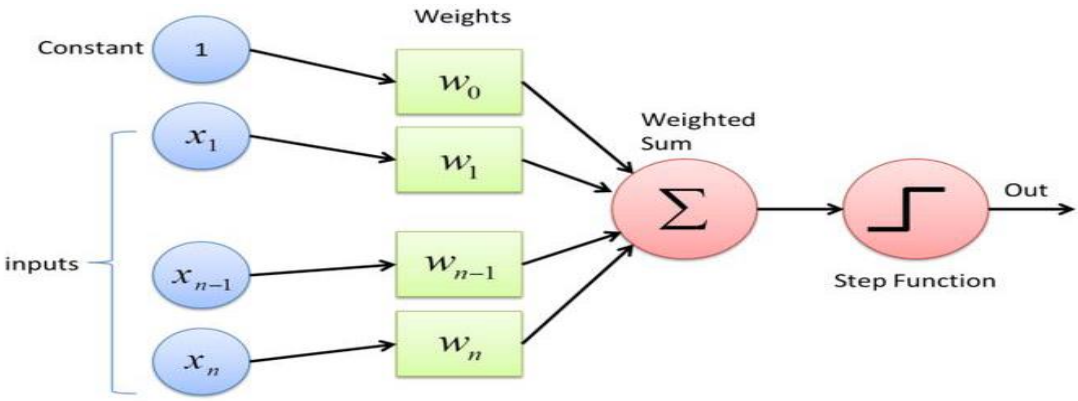


Figure 3.3 Rosenblatt's Perceptron model (multiple-input neuron)

In the case of Perceptron, The mathematic equations become:

$$v = \sum_1^n w_i\, x_i + b \qquad (3.1)$$

$$a = f(v) \qquad (3.2)$$

We could say, that perceptron is a neural network with a single neuron and multiple inputs, it is very useful for linear separable problems **[25].**

3.4.2. Training of Perceptron:

The parameters we want to learn are the weights of the perceptron, not any weight are acceptable; hence, we would want to find a weight vector that makes the perceptron produce the perfect result we need.

There is some sort of mathematical such rule is called the perceptron learning or training rule (3.3) and (3.4).

$$w_i = w_i + \Delta w_i \qquad (3.3)$$

$$\Delta w_i = \eta\,(t - a)\, x_i \qquad (3.4)$$

With **wi**: is the weight of i^{th} input xi, **t**: is the target output.

a: is the Perceptron's output.

Δwi: is the amount that is added to update the old wi.

η: is the learning rate, must be chosen carefully **[26].**

To resume: A perceptron could be used successfully for linear separable problems such as AND, OR and other similar problems.

The main limitation of the perceptron model is that is can only learn linear patterns thus, it was limiting the type of problems to be solved.

3.5. Multilayer perceptron MLP

3.5.1. Definition

A multilayer perceptron (MLP) is a feed-forward neural network with one or more hidden layers. The network consists of an input layer of source neurons, at least of one middle or hidden layer of computational neurons, and an output layer of computational neurons. The input signals are propagated in a forward direction on a layer-by-layer basis.

3.5.2. Architecture Properties of MLP

The MLP is the base of the ANN, it means that the architecture of the ANN is basically is the same (see section 3 of the current chapter) shown in Figure 3.4

Proprieties :

- ⇒ No connections within a layer.
- ⇒ No direct connections between input and output layers.
- ⇒ Fully connected between layers.
  - ➢ Number of output units needs not equal number of input units.
  - ➢ Number of hidden units for each layer can be more or less than input or output units.

A hidden layer "hides" its desired output by minimizing the errors. Neurons in the hidden layer cannot be observed through the input/output behaviour of the network.

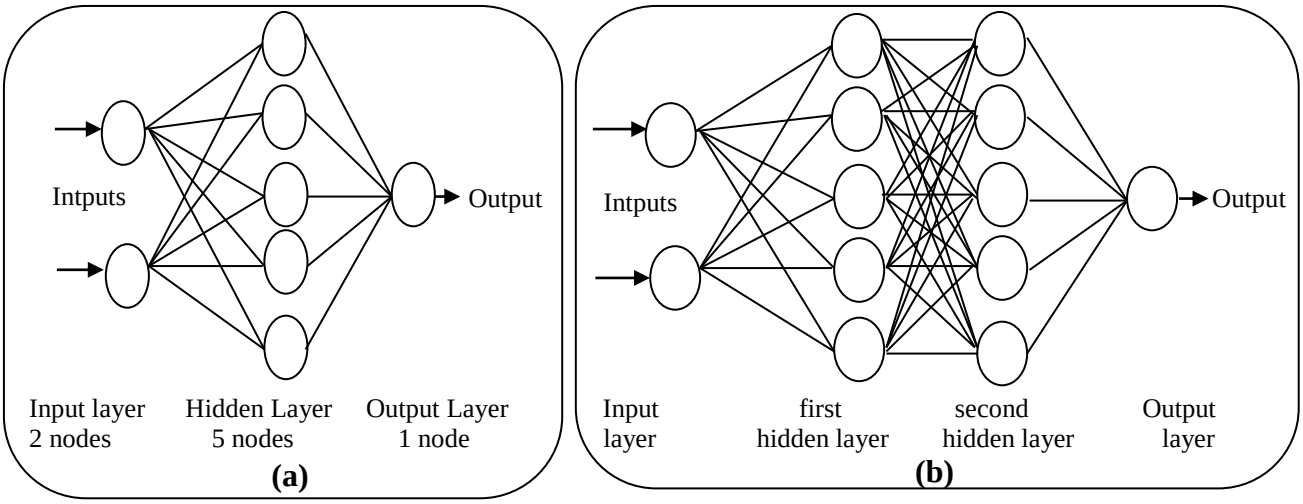


Figure 3.4 MLP architecture (a)- With single Hidden layer (b)- With two Hidden layers.

3.5.3. Supervised training of MLP:

It proceeds the same way as for a perceptron. A training set of input patterns is presented to the network.

The network computes its output pattern, and if there is an error – or in other words, there exist a difference between actual and desired output patterns – the weights are adjusted to reduce this error.

If the results are not convincing and the error is large, the problem is that we don't know what weights are wrong and what is the correct activation for the neurones of the hidden layers, in that case the solution is to use Back propagation algorithm **BP** (Rumelhart, Hinton and Williams, 1986) **[28]**.

3.6.The Back Propagation Algorithm

For training the MLP with Back propagation learning model, we have two Passes to do:

Forward pass:

- Put the input values in the input Layer;
- Calculate activation of the hidden nodes;
- Calculate the activation of the outputs nodes.

Backward pass:

- Calculate the output error.
- Update last layer's weights.
- Propagate error backward, update hidden weights.
- Until first layer is reached.

To resume, we must follow the NN learning steps:

- Weights Initialization.
- Inputs application.
- Sum of inputs- product weights.
- Activation function.
- Weights adaptation.
- Back to step2 **[29].**

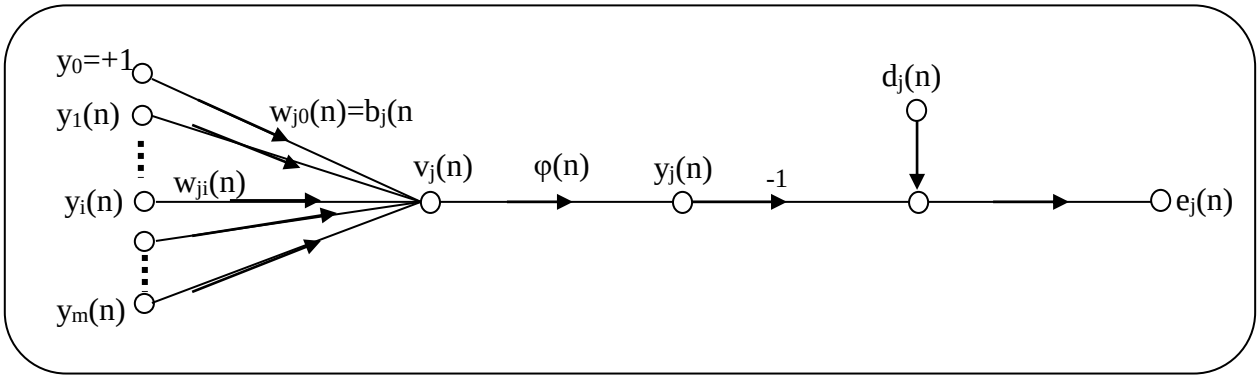


Figure 3.5 Signal-flow graph highlighting the details of output neuron j

The popularity of on-line learning for the supervised training of multilayer Perceptron has been further enhanced by the development of the back-propagation algorithm. To describe this algorithm, consider Figure 3.6, which depicts neuron j being fed by a set of function signals produced by a layer of neurons to its left.

The induced local field v_j(n) produced at the input of the activation function associated with neuron j is therefore:

$$v_j(n)=\sum_{i=0}^{m} w_{ij}(n)y_i(n) \qquad (3.5)$$

With m is the total number of inputs (excluding the bias) applied to the neuron j. The synaptic weight correspondent to the bias $w_{j0}=b_j$ and $y_0=+1$. Hence, the function signal y_j(n) appearing at the output of neuron j at iteration n is:

$$y_j = \varphi(v_j(n)) \qquad (3.6)$$

The back propagation algorithm applies a correction $\Delta w_{ji}(n)$ to the synaptic weight wji(n), which is proportional to the partial derivative . According to **the chain rule** of calculus, we may express this gradient as:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} * \frac{\partial e_j(n)}{\partial y_j(n)} * \frac{\partial y_j(n)}{\partial v_j(n)} * \frac{\partial v_j(n)}{\partial w_{ji}(n)} \qquad (3.7)$$

The error signal produced at the output of neuron *j* is defined by

$$e_j(n) = d_j(n) - y_j(n) \qquad (3.8)$$

Where d_j(n) is the j[th] output target (desired), the instantaneous error energy of neuron j calculated by the LMS method is defined by:

$$E_j(n) = \frac{1}{2}e_j^2(n) \qquad (3.9)$$

E(n) is the total instantaneous error energy of the whole network (in other references is named as average cost**)** as:

$$E(n) = \sum_{j\in C} E_j(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n) \qquad (3.10)$$

Differentiating both sides of equation (3.10) with respect to $e_j(n)$, we get

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \qquad (3.11)$$

Differentiating both sides of equation (3.8) with respect to $y_j(n)$, we get

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \qquad (3.12)$$

Next, differentiating equation (3.6) with respect to $v_j(n)$, we get

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n)) \qquad (3.13)$$

Finally, differentiating equation (3.5) with respect to $w_{ji}(n)$ yields

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \qquad (3.14)$$

The use of equations. (3.11) to (3.14) in equation (3.7) yields

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\, \varphi'\left(v_j(n)\right)\, y_i(n) \qquad (3.15)$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the delta rule, or:

$$\Delta w_{ji}(n) = -\eta\, \frac{\partial E(n)}{\partial w_{ji}} \qquad (3.16)$$

Where η is the learning-rate parameter of back-propagation algorithm, the use of the minus sign in equation (3.16) accounts for gradient descent in weight space (i.e., seeking a direction for weight change that reduces the value of E(n)). Accordingly, the use of equation (3.15) in equation. (3.16) yields:

$$\Delta w_{ji}(n) = \eta\, \delta_j(n)\, y_i \qquad (3.17)$$

Where the local gradient $\delta_j(n)$ is defined by:

$$\delta_j(n) = \frac{\partial E(n)}{\partial v_j(n)} = \frac{\partial E(n)}{\partial e_j(n)} * \frac{\partial e_j(n)}{\partial y_j(n)}\, \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n)\, \varphi'\left(v_j(n)\right) \qquad (3.18)$$

From the eqations. (3.17) and (3.18), we note that a key factor involved in the calculation of the weight adjustment $\Delta w_{ji}(n)$ is the error signal $e_j(n)$ at the output of neuron j. In this context, we may identify two distinct cases, depending on where in the network neuron j is located:

Case 1: Neuron *j* is an Output Node

In this case, it's supplied with a desired response of its own. We may use equation (3.8) to compute the error signal $e_j(n)$ associated with this neuron; see Figure 3.5. Having determined $e_j(n)$, we find it a straightforward matter to compute the local gradient $\delta_j(n)$ by using the equation (3.18).

Case 2: Neuron *j* is a Hidden Node

In that case, there is no specified desired response for that neuron. Accordingly, the error signal for a hidden neuron would have to be determined recursively and working backwards in terms of the error signals of all the neurons to which that hidden neuron is directly connected; this is where the development of the back-propagation algorithm gets complicated. Consider the situation in Figure 3.7, which depicts neuron *j* as a hidden node of the network. According to equation (3.18), we may redefine the local gradient $\delta_j(n)$ for hidden neuron *j* as:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial y_j(n)}\,\varphi'\left(\mathrm{v_j(n)}\right) \qquad (3.19)$$

- Neuron j is hidden layer.

To calculate the partial derivative, we may proceed as follows: From Figure 3.6, we see that:

$$E(n) = \frac{1}{2}\sum_{k\in C} e_k^2(n) \qquad (3.20)$$

- Neuron k is an output node

Differentiating equation (3.20) w.r.t the function signal $y_j(n)$, we get:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \qquad (3.21)$$

Next we use the chain rule for the partial derivative $\partial e_k(n)/\partial y_j(n)$ and rewrite equation (3.21) in the equivalent form:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_{k*}\frac{\partial e_k(n)}{\partial v_k(n)}\frac{\partial v_k(n)}{\partial y_j(n)} \qquad (3.22)$$

However, from figure 3.6, we note that:

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi(v_k(n)) \qquad (3.23)$$

Neuron k is an output node hence:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'\left(v_k(n)\right) \qquad (3.24)$$

We also note from figure 3.6 that for neuron k, the induced local field is

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) * y_j(n) \qquad (3.25)$$

Where m is the total number of inputs (excluding the bias) applied to neuron k. Here again, the synaptic weight $w_{k0}(n)$ is equal to the bias $b_k(n)$ applied to neuron k, and the corresponding input $y_0=1$. Differentiating equation (3.25) with respect to $y_j(n)$ yields

$$\frac{\partial v_k(n)}{\partial y_k(n)} = w_{kj}(n) \qquad (3.26)$$

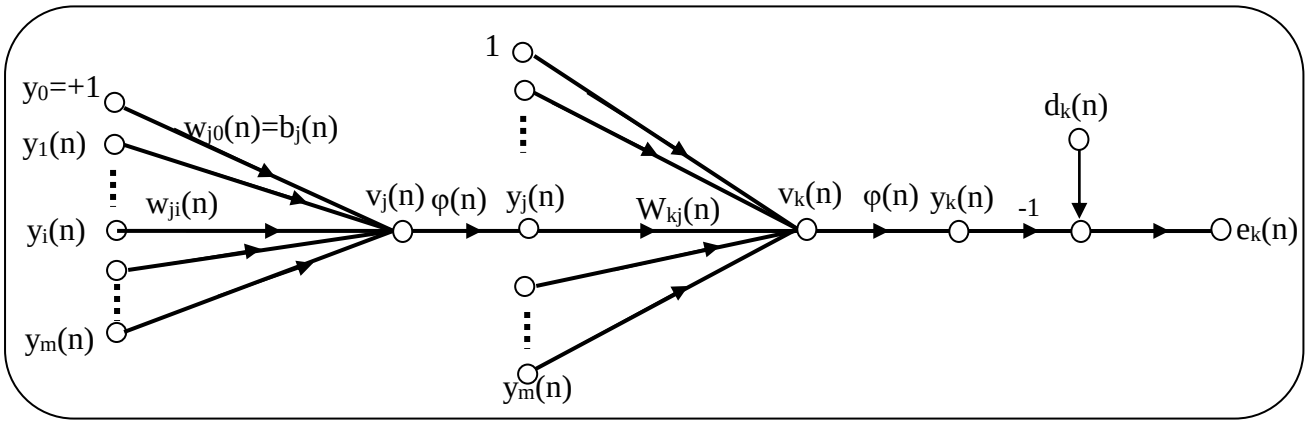


Figure 3.6 Signal-flow graph highlighting the details of output neuron ***k*** connected to hidden neuron ***j.***

By using equations. (3.24) and (3.26) in equation.(3.22), we get the desired partial derivative

$$\frac{\partial E_k(n)}{\partial y_j(n)} = -\sum_k e_k(n)\, \varphi'_k\big(v_k(n)\big)\, w_{kj}(n) = -\sum_k \delta_k(n)\, w_{kj}(n) \qquad (3.27)$$

Where we have used the definition of the local gradient δ_k(n) given in equation (3.18), with the index k substituted for j. Finally, using equation (3.27) in equation (3.19), we get the back-propagation formula for the local gradient $\delta_j(n)$, described by:

$$\delta_j(n) = \varphi'_j\Big(v_j(n)\Big) \sum_k \delta_k(n)\, w_{kj}(n) \qquad (3.28)$$

Figure 3.7 shows the signal-flow graph representation of equation (3.28), assuming that the output layer consists of m L neurons.

The outside factor $\varphi'_k\big(v_k(n)\big)$ involved in the computation of the local gradient $\delta_j(n)$ in equation. (3.28) depends solely on the activation function associated with hidden neuron j.

The remaining factor involved in this computation namely, the summation over *k* depends on two sets of terms. The first set of terms, the $\delta_k(n)$, requires knowledge of the error signals $e_k(n)$ for all neurons that lie in the layer to the immediate right of hidden neuron *j* and that are directly connected to neuron *j*; see Figure.3.6. The second set of terms, the $w_{kj}(n)$, consists of the synaptic weights associated with these connections.

Now, we summarize the relations that we have derived for the back-propagation algorithm.

First, the correction $\Delta w_{ji}(n)$ applied to the synaptic weight connecting neuron i to neuron j is defined by the delta rule equation.3.29

$$\begin{pmatrix} weights \\ correction \\ \Delta w_{ij}(n) \end{pmatrix} = \begin{pmatrix} learning-rate \\ parameter \\ \eta \end{pmatrix} * \begin{pmatrix} local \\ gradiant \\ \delta_j(n) \end{pmatrix} * \begin{pmatrix} input\ signal \\ of\ neuron\ j \\ y_j(n) \end{pmatrix} \qquad (3.29)$$

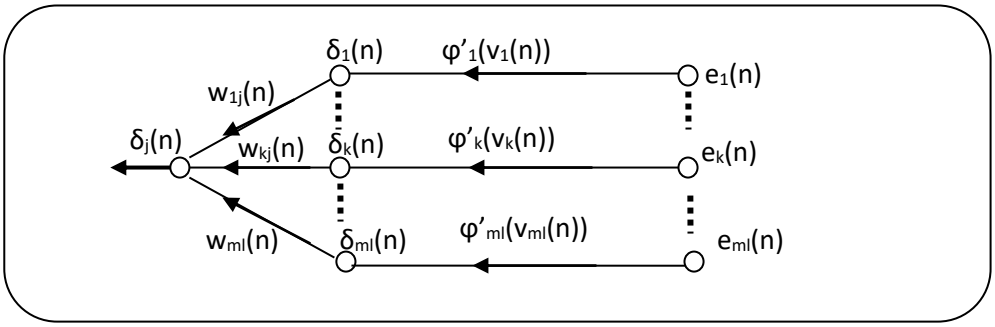


Figure 3.7 Signal-flow graph of a part of the adjoint system pertaining to back-propagation of error signals.

Second, the local gradient $\delta_j(n)$ depends on whether neuron j is an output node or a hidden node:

If neuron j is an output node, $\delta_j(n)$ equals the product of the derivative $\varphi'_j\left(v_j(n)\right)$ and the error signal $e_j(n)$, both of which are associated with neuron j; see equation (3.18).

If neuron j is a hidden node, $\delta_j(n)$ equals the product of the associated derivative $\varphi'_j\left(v_j(n)\right)$ and the weighted sum of the δ_{ks} computed for the neurons in the next hidden or output layer that are connected to neuron j; see equation (3.28**).**

Finally, the parameters of the MLP model (weights and biases) will be updated using a modified delta rule with **momentum** to accelerate the learning process, so that Equation 3.17 becomes:

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \qquad (3.30)$$

The equation 3.30 called generalized delta rule, with assuming that equation 3.17 is a part of it with α=0.

In the end, we have the equation of updating weights and baises:

$$w_{ji}^l(n+1) = w_{ji}^l + \alpha w_{ji}^l(n-1) + \eta \delta_j^l(n) y_i^{l-1}(n) \qquad (3.31)$$

Repeat the process until getting a desired average cost (MSE) **[30]**.

Remember that:

$$\delta_j^l = \begin{cases} e_j^l(n)\varphi'_j\left(v_j^l(n)\right) & if\ the\ neuron\ j\ is\ the\ output\ layer \\ \varphi'_j\left(v_j^l(n)\right)\sum \delta_k^{l+1}(n) w_{kj}^{l+1}(n) & else \end{cases} \qquad (3.32)$$

We can also use the following simple version of back propagation algorithm without using momentum:

- Initialize Network with Random weights and biases;
- For each training sample:
  * Compute Activation for entire Network (feed forward);
  * Compute δ for Neurons in the output Layer using equation (3.18);
  * Compute δ for all neurons in the previous layers using equation (3.28);
  * Compute gradient of cost with respect of each weight and bias for the training sample

    using δ: $\frac{\partial C_x}{\partial w_{jk}^{(l)}} = \delta_j^{(l)} a_k^{(l-1)}$ $\frac{\partial C_x}{\partial b_j^{(l)}} = \delta_j^{(l)}$
- Average the gradient with respect to each Weight and Bias over the entire training set:

$$\frac{\partial C}{\partial w_{jk}^{(l)}} = \frac{1}{n} * \sum \frac{\partial C_x}{\partial w_{jk}^{(l)}} \qquad \frac{\partial C}{\partial b_j^{(l)}} = \frac{1}{n} * \sum \frac{\partial C_x}{\partial b_j^{(l)}}$$

- Update the weights and Biases using gradient descent

$$w_{jk}^{(l)} \to w_{jk}^{(l)} = w_{jk}^{(l)} - \eta * \frac{\partial C}{\partial w_{jk}^{(l)}} \qquad b_j^{(l)} \to b_j^{(l)} = b_j^{(l)} - \eta * \frac{\partial C}{\partial b_j^{(l)}}$$

- Repeat steps 2-4 till cost reduces below acceptable level **[31].**

3.7. Radial basic neural network RBNN:

3.7.1. Definition:

RBNN also known as radial basis function RBF is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

Radial basis networks have many uses, including function approximation, time series prediction, classification, and system control **[32].**

3.7.2. ARCHITECTURE OF RBNN:

RBNN considered as a specific type of MLP, it composes of input layer, just **only one hidden layer (**That is what makes it distinct from the common architecture of MLPs in general.), and output layer. We call this hidden layer as **feature vector**. The following figure3.9 shows general architecture of RBNN. Specifics of RBNN Architecture are:

- Only one hidden layer;
- No weights between inputs and hidden layer;
- The activation function is the same RBF: Φ (r) and it's a vector (for each node there is a function with different parameters), with $r = \|x - t\|$ and t is the center;
- There are linear weights between hidden layer and the output layer **[32]**.

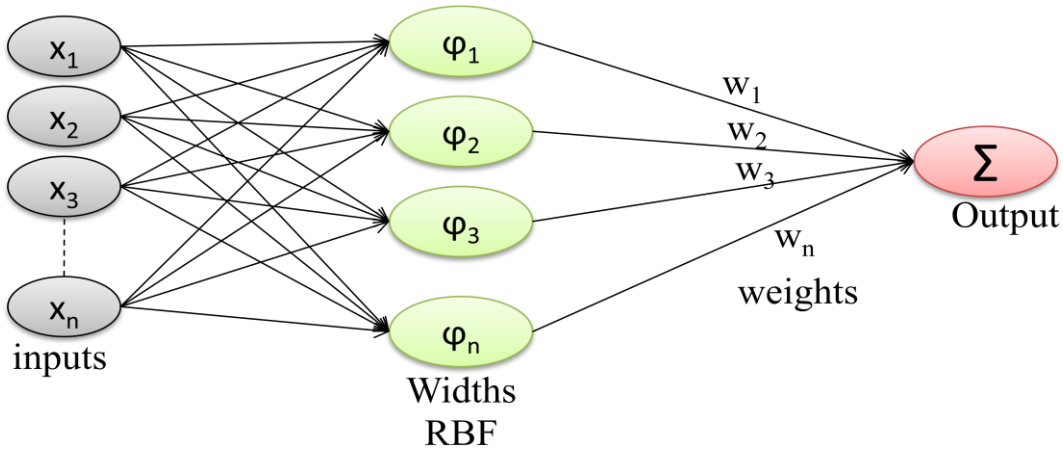


Figure 3.8 Architecture of RBNN.

3.7.3. Cover's theorem and the separability of patterns:

When a radial-basis function (RBF) network is used to perform a complex pattern classification task, the problem is basically solved by first transforming it into a high dimensional space in a nonlinear manner and then separating the classes in the output layer. The underlying justification is found in Cover's theorem on the separability of patterns, which, in qualitative terms, may be stated as follows:

A complex pattern-classification problem, cast in a high-dimensional space non linearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated**.**

It's mean that, when we increase the dimension of the feature vector, the linear separability of feature vector increases. A non-linearity separable problem is highly separable in high dimensional space than it is in low dimensional space **[33]**.

3.7.4. Radial Basis Functions (RBFs):

The difference between RBNN modal and the McCulloch & Pitts modal (MLP modal) is in a way that in RBNN, we measure the distance of an input to a fixed centre and the inputs are not weighted. The centre chosen randomly in the first time, it can be one of the inputs. The distance metric is usually Euclidean.

As we have seen in MLP, at least we need one hidden layer to derive a non-linearity separation. RBNN it does the same task but in different way, it transforms the input signal into another form, which can be then feed into the network to get linear separability.

There are two ways for non linear separability in RBNN:

- Increase the dimensionality.
- Expand the direction horizontally or compress the direction vertically. We can increase the dimensionality by using the cover's theorem.

The mathematical formulation for RBF is the equation 3.29

$$y(x) = f(\sum_{i=1}^{n} \|x_i - t\|) \quad (3.33)$$

Where f(r) is the RB Function , x=[x_1,x_2,.....x_n] is the input vector and t is the centre **[34]**.

Examples of RBFs:

- Multiquadric $f(r) = \sqrt{(r^2 + c^2)}$
- Inverse multiquadric $f(r) = \frac{1}{\sqrt{(r^2+c^2)}}$
- Gaussian function $f(r) = \exp(-\frac{r^2}{2c^2})$

- Where $r=\|x-t\|$ is the distance between the input xi and the centre t.
-c>0 is the radius it determines how diverse the function will be.

3.7.5. RBNN algorithm:

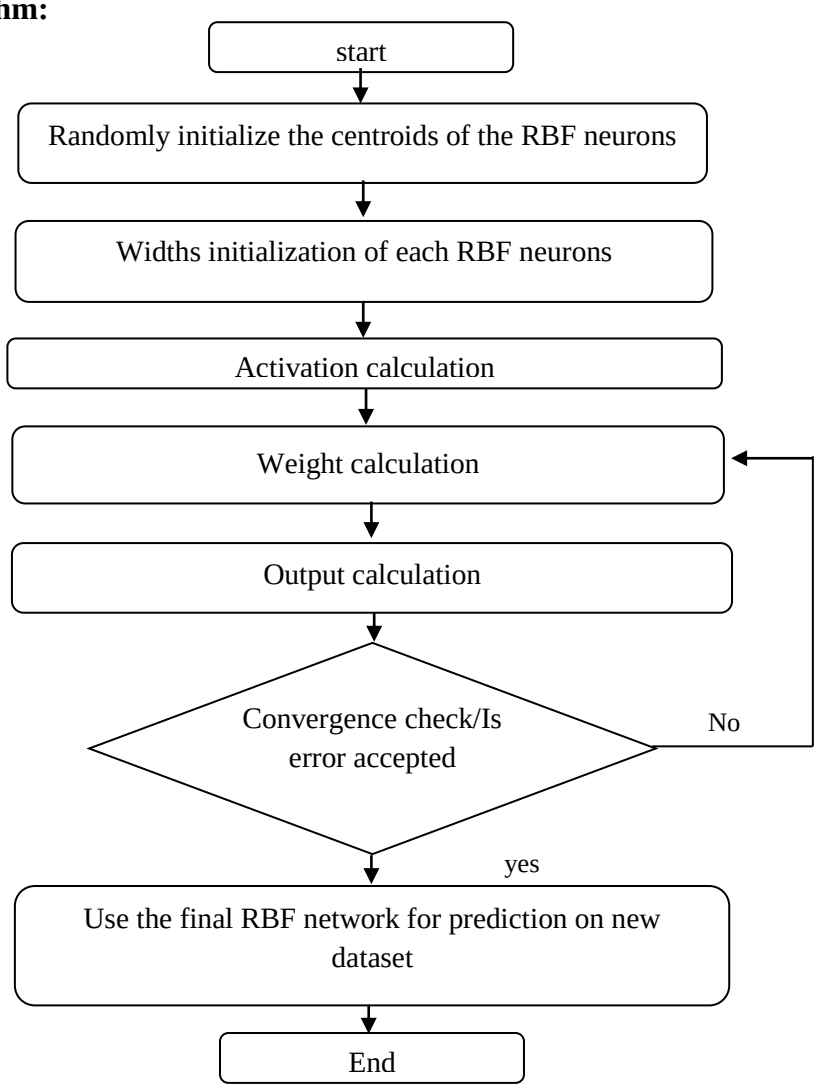


Figure3.9 Flowchart of working procedure of Radial Basis Functions (RBF) **[35]**

According to the architecture of RBNN, there are few differences between MLP algorithm and RBNN algorithm:

- Initialization:

  * Input vector $x=[x_1,x_2,....,x_n]$ and desire output vector $y=[y_1,y_2,....,y_n]$;

  * Assign random weights in the range of [-1,1];

- Feed Forward phase:

  * Step1: In the first layer calculate I^1 and O^1, $I^1=x$ and $O^1=I^1$;

  * Step2: for each node of hidden layer, we find the output with chosen RBF for example with Gaussian;

  * Step 3: compute linear weighted sum of the output of each neuron from the hidden layer.

- Learning phase.

3.7.6.RBNN learning Process:

The adaptive parameters according to the architecture of RBNN are:

- Centres (prototypes) according to the hidden nodes;
- Receptive field widths (parameters of the RBF);
- Weights associated to connections between the hidden and output layers.
- We can distinguish tow variants of learning:

Simultaneous learning of all parameters, that is similar to Back Propagation of the MLPs;

- Separate learning of parameters: centres, widths and weights.
- For the simultaneous learning (same process of MLP's back propagation method)

Separate learning:

- Training set: $\{(x^1,d^1),.....(x^L,d^L)\}$
- Estimation of the centres:
  - If K=L (with k: nbr of centres= nbr of samples)
    $C^k=x^k$ (this corresponds to the case of exact interpolation)
  - K<L the centres are established
    Random select from the training set;
    Systematic selection from the training set by using clustering method.
- Estimating of the receptive fields widths: Heuristic rules
  - $\delta = \frac{d_{max}}{\sqrt{2K}}$ with d_{max}=maximal distance between centres;
  - $\delta_k = \gamma * d(c^k, c^j)$, c^j is the closest centre to c^k , $\gamma \in [0.5,1]$
  - $\delta_k = \frac{1}{m} * \sum_{j=1}^{m} d(c^k, c^j)$ $c^1,....,c^m$: the closest m centre to c^j

- $\delta_k = \frac{1}{q_k} * \sum_{j=1}^{q_k} d(c^k, X^j)$: $X^1 .. X^{q_k}$:input vectors represented by unit k.

➢ Estimating the weights of connections between hidden and output layer.

This is equivalent with the problem of training one layer linear network (see section 6 of the current chapter)**[36].**

3.8. Design and Development of MLP-based MPPT

In this section we we'll present a MLP model for MPP identification implemented by Matlab code, it's very useful for practicing and to compare with a Fuzzy model

3.8.1. Initialized the MLP model

This proposed MLP model has the following architecture figure **3.10**:

⇒ **Input layer:** for modulate the MPPT by using MLP model; we have to import two inputs vector from the mathematic PV panel model described in the first chapter, for learning the MLP model we have to use trainer data.

⇒ **Hidden layers:** we chose two hidden layers with open number of nodes (in first we initialized by 40 nodes for each), with activation function, we used the hyperbolic tangent Sigmoid (mentioned in section 3 of the current chapter).

⇒ **Output layer:** we have just only one output, for the tracking the maximum power point MPPT of the PV panel model.

All weights and biases are initialized randomly.

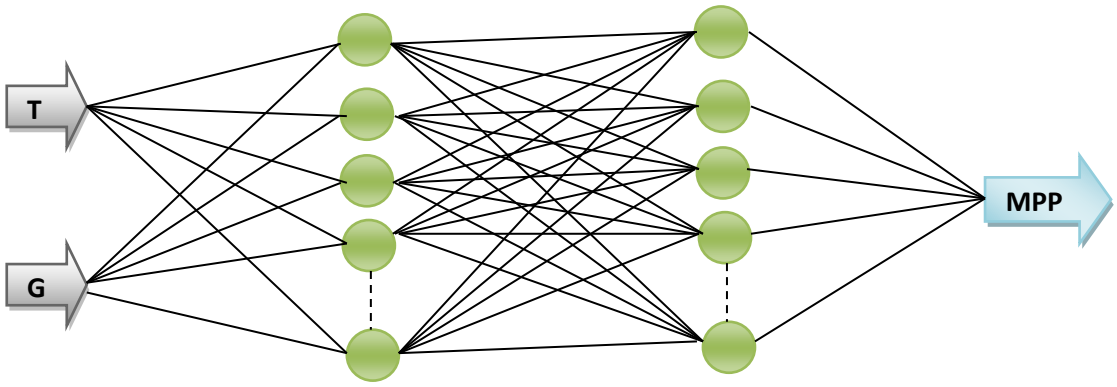


Figure 3.10 Diagram of the proposed model

3.8.2. Supervised learning of the Model

As we have already seen, the common method used to train MLP models is the Back-propagation. This process involves two repeating passes to minimize the cost function until an acceptable value is reached:

⇒ **Forward Propagation**

- **Weighted Sum**: for each neuron in a hidden or output layer, we calculate a weighted sum of its inputs: $\boldsymbol{z = \sum(w_i * x_i) + b}$
- **Activation Function**: we used a non-linear activation function to determine the neuron's output signal in our modal by using hyperbolic tangent Sigmoid $\boldsymbol{a(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}}$

Calculate MSE (the cost)

Once the signal reaches the output layer, the network's prediction is compared to the target value using a Mean squared error $E = mean((d - y)^2)$ where d is the Target and y is the Predicted MPPT.

⇒ **Backward Propagation algorithm**

The algorithm calculates the **gradient descent of the MSE** with respect to each weight and bias in the network by using the **chain rule (**equation 3.7**):**

$$\Delta_{out} = -2 \times (d - a_o),\ \delta_{a_2} = (w_3 - \Delta_{out}).*(1 - a_2^2),\ \delta_{a_1} = (w_2 * \delta_{a_2}).*(1 - a_1^2)$$

This step identifies how much each specific weight and bias contributed to the total error.

Update Parameters weights and biases

$$w_3 = w_3 - \eta \times \Delta_{out} \times a_2; \qquad b_3 = b_3 - \eta \times \Delta_{out};$$

$$w_2 = w_2 - \eta \times \delta_{a_2} \times a_1; \qquad b_2 = b_2 - \eta \times \delta_{a_2;}$$

$$w_1 = w_1 - \eta \times \delta_{a_1} \times x_n; \qquad b_1 = b_1 - \eta \times \delta_{a_1};$$

Repeat until Convergence

Repeat for every sample in the dataset. This process continues for multiple epochs until the error is minimized or the number of epochs is reached.

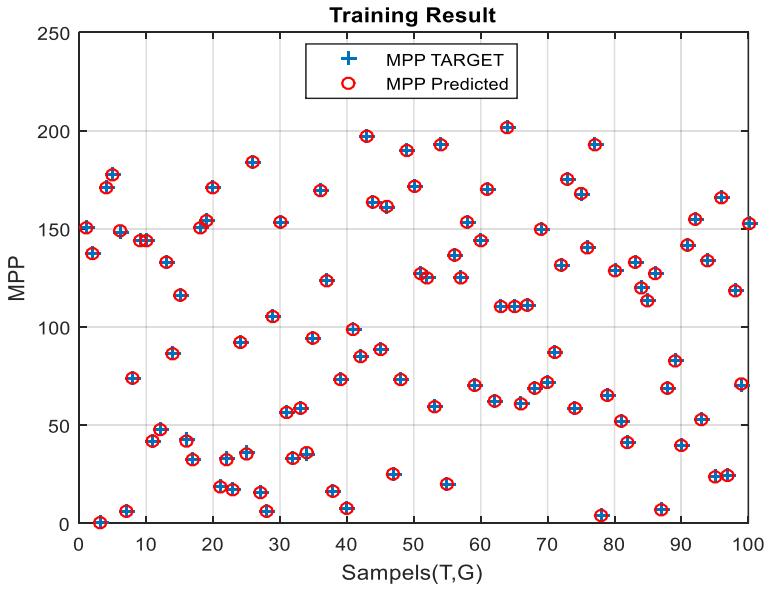


Figure 3.11 Training results

If the results are satisfactory, we can save the optimal parameters, and our model is ready for testing.

3.8.3. Testing the Model

We import the testing data from the PV-panel model, which is different from the training data. The results are showing in Figure 3.13. (This model will be presented in greater detail in the fifth chapter).

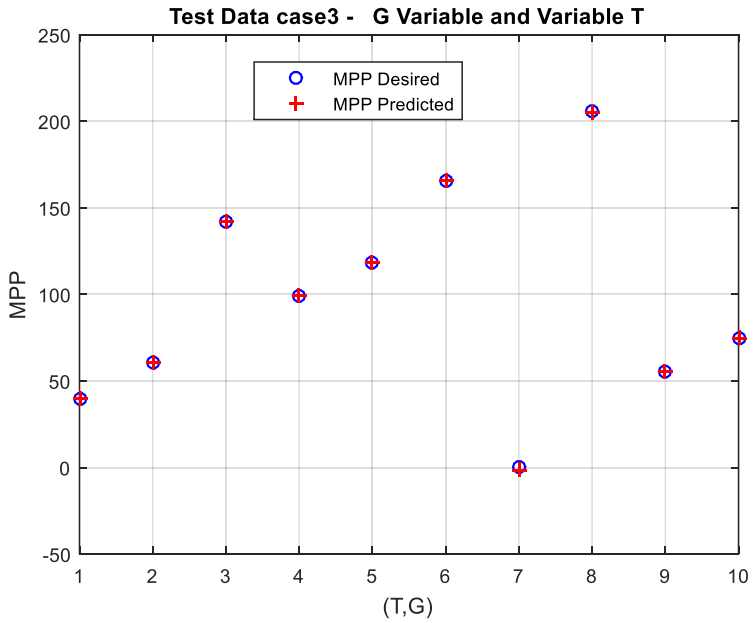


Figure 3.12 Testing results

The results are highly satisfactory, as the MLP-based MPPT detected the MPP accurately and rapidly to validate and using.

3.9. Conclusion and analyse

As the world is moving towards total reliance on artificial intelligence methods in all fields, knowledge of neural networks and learning methods has become a basic and urgent need in order to keep pace with scientific progress.

The NN is very important for its capability of adapt, the challenge now is find the ways to minimise the cost and the time.

The Advantages of MLP

- MLPs can learn and approximate highly complex, non-linear relationships between inputs and outputs, unlike simple linear regression.
- The MLPs are adaptable to a wide range of problems, including both classification (e.g., categorizing data) and regression (e.g., predicting continuous values).

- Through their hidden layers, MLPs can learn representations of data and automatically identify important features without extensive manual engineering.
- Ease of Use with Tabular Data.

Limits of MLP

- **The overfitting:** MLPs have many parameters (weights and biases), making them prone to memorizing training data rather than generalizing well, particularly when datasets are small.
- **Computational Cost:** Training large or deep MLPs requires significant computational power, time, and energy.
- **Lack of Spatial/Temporal Awareness:** Because MLPs rely on "fully connected" layers, they fail to preserve spatial structures (like in images) or sequential dependencies (like in time-series data or text). For these tasks, specialized architectures like CNNs or RNNs are better suited.
- **Black Box Nature:** Like many deep learning models, it is difficult to interpret why a MLP makes a specific decision, making it less than ideal in highly regulated domains where explain ability is required.
- **Hyper parameter Tuning:** Achieving optimal performance requires extensive trial and error to tune hyper parameters like learning rates, number of hidden layers, and layer sizes.

In the next chapter, we will study and describe one of the most powerful methods in the world of AI, and how we can combine it with an ANN to create one of the most **effective** hybrid MPPT methods **available**.

Chapter 4:

Fundamental of Fuzzy set systems (FSSs)

2026

4.1. Introduction

In the previous chapters, we explored neural networks and learning methods for intelligent systems. Although these methods are remarkably powerful, they also have limitations. This is why researchers look forward to improving results by combining ANN methods with other types of logic distinct from classical computing to create something even more powerful.

Fuzzy logic can be identified as a debate between meaningful statements and precision. Albert Einstein famously said, “As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality. Emphasizing the same concept, the founder of fuzzy theory, Lotfi A. Zadeh, noted, “As complexity rises, precise statements lose meaning, and meaningful statements lose precision”**[37].**

The human brain interprets imprecise and incomplete sensory information. For example, we can feel that the weather is cold, but we cannot calculate the exact temperature with complete accuracy. Similarly, fuzzy set theory provides a systematic calculus to process such information linguistically, performing numerical computations using linguistic labels stipulated by membership functions.

Combining fuzzy theory with an Artificial Neural Network (ANN) yields a powerful model known as the Adaptive Network Fuzzy Inference System (ANFIS), formally proposed by Dr. Jyh-Shing Roger Jang in the early 1990s **.**

In this chapter, we will describe Fuzzy Set Systems (FSSs) and explore how to design and develop a fuzzy logic model for Maximum Power Point (MPP) identification. This involves combining ANNs and a Fuzzy Inference System (FIS) by utilizing the ANFIS model, which is capable learning.

4.2. The Fuzzy Set Systems (FSSs)

Many classical sets with crisp boundaries misrepresent the true goal of classification. For example, consider classifying cars based on speed: defining cars travelling at or above 200 km/h as "fast" and those below as "slow" implies that a car travelling at 199 km/h is "slow." This is an unnatural and inadequate way of representing our intuitive concept of a "fast car." Ultimately, this distinction is illogical, as the flaw stems from the sharp, binary transition between inclusion and exclusion in a traditional set.

In contrast to a classical set, a fuzzy set, as the name implies, is a set without a crisp boundary. That is, the transition from “belonging to a set” to “not belonging to a set” is gradual, and this smooth transition is characterized by membership functions that give fuzzy sets flexibility in modelling commonly used linguistic expressions **[38].**

As Zadeh pointed out in 1965 in his seminal paper entitled "Fuzzy Sets": "such imprecisely defined classes play an important role in human thinking, particularly in the domains of pattern recognition, communication of information, and abstraction." **.**

Because conventional computer logic cannot handle subjective or vague data, Zadeh developed fuzzy logic. This system allows computers to evaluate data in shades of gray, effectively imitating the process of human reasoning.

4.2.1. Definitions

- Crisp sets figure 4.1- a **:**

The term crisp means precise. Crisp sets are groups that represent exact values. Standard set theory operations—such as ∪ ∩ ∈ ⊆ ∉

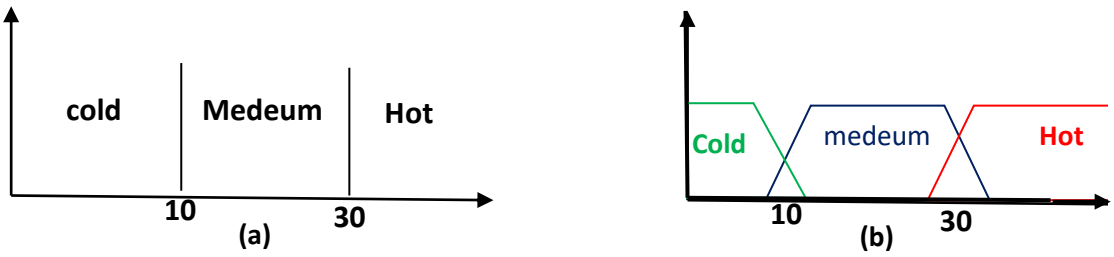


Figure 4.1 (a) crisp set (b) fuzzy set

- Fuzzy set figure 4.1- b**:**

- The idea is that an element belongs to a fuzzy set with a certain degree of membership.
- Hence, a proposition is not strictly true or false, but partially true and partially false to varying degrees**.**
- The degree of membership is usually represented as a real number between 0 and 1[**39]**.

- Fuzzy inference process figure 4.2:

Fuzzy inference is the process of mapping a given input to an output using fuzzy logic, mimicking human reasoning to make decisions or discern patterns. This process is represented by the following steps:

- Fuzzification: Transforms the input from a crisp value to a fuzzy value using a membership function (MF) (see figure 4.4);
- Rule evaluation (the inference engine): Evaluates the fuzzy rules to compute the fuzzy set outputs;
- Defuzzification: Translates the resulting fuzzy set into a useful, crisp output value using a membership function, see figure 4.4.

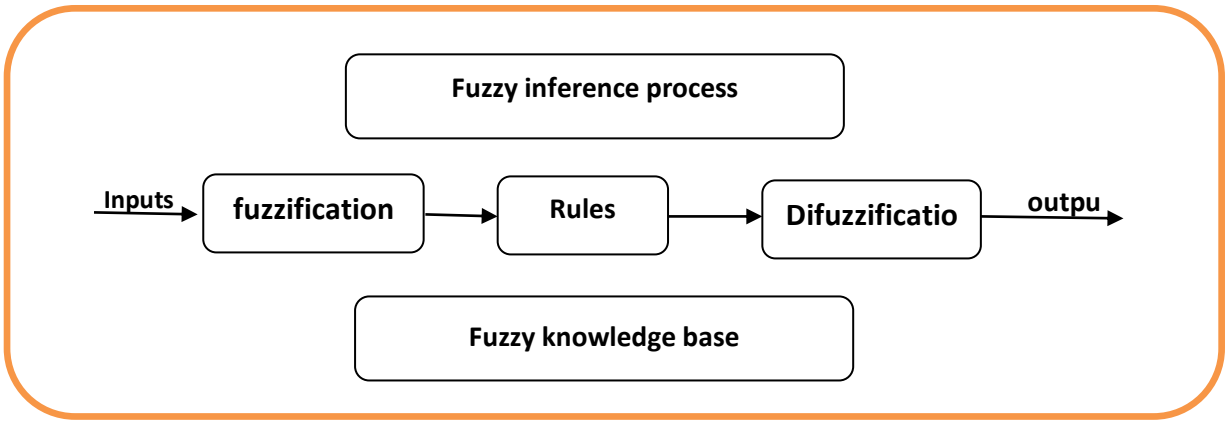


Figure 4.2 Fuzzy inference processes.

- The membership function:

A membership function (MF) is a mathematical formula or a curve that defines the membership's degree of each point in an input space (universe of discourse) is mapped to a value between 0 and 1.

In fuzzy inference we use in general, many types of MF figure 4.3, the most famous ones are:

- Triangle function: is specified by three parameters {a, b, c} as follow

$$\mu_A(x) = \begin{cases} 0 & if\ x \le a \\ \frac{x-b}{b-a} & if\ a \le x \le b \\ \frac{c-x}{c-b} & if\ b \le x \le c \\ 0 & if\ x \ge c \end{cases} \quad \text{or } \mu_A(x) = \max\left(\min\left(\frac{x-b}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{4.1}$$

- Trapezoidal function: is specified by four parameters {a, b, c,d} as follows

$$\mu_A(x) = \begin{cases} 1 & if\ b \le x \le c \\ \frac{d-x}{d-c} & if\ c \le x \le d \\ \frac{x-b}{b-a} & if\ a \le x \le b \\ 0 & otherweise \end{cases} \quad \text{or } \mu_A(x) = \max\left(\min\left(1, \frac{x-b}{b-a}, \frac{d-x}{d-c}\right), 0\right) \tag{4.2}$$

- Gaussian function: is specified by two parameters {δ, c} as follows

$$\mu_A(x) = \exp\left(-\frac{1}{2} * \left|\frac{(x-c)}{\delta}\right|^2\right) \tag{4.3}$$

- Generalise Bell function: is specified by three parameters {a,b, c} as follows

$$\mu_A(x) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{4.4}$$

The type of MF Figure 4.3 normally chosen suits of the application. **[40]**

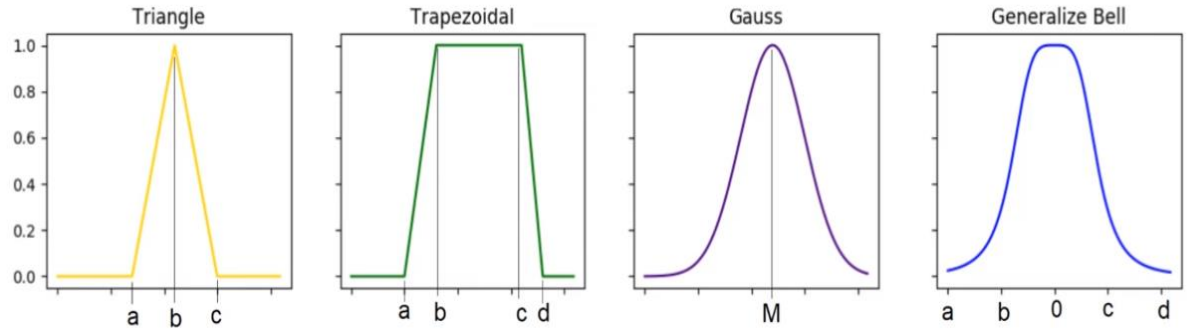
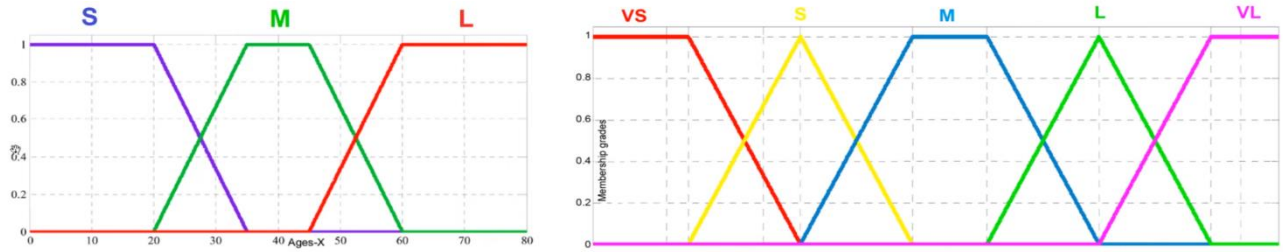


Figure 4.3 some type of Membership functions curves



(a) **(b)**

Figure 4.4 Curves of MFs (a)- Same MF trapezoidal (b)- mixed MFs triangle/trapezoidal

- Rules evaluation:

Inputs are applied to a set of **if/then** control rules

- For Single input system, we use only the IF/Then control rules,

  Eg: If input is MF degree **then** is MF of output

- For multi-input system we use **AND/OR**

  Eg: If input1 is MF1 degree AND/OR input 2 is MF2 Then the output is MF **[41].**

4.2.2. Applications of fuzzy logic:

- Fuzzy logic enables low cost microcontrollers to perform functions by more powerful expensive machines enabling lower cost products to execute advanced features.
- Fuzzy logic controller: automation, Washing machines, Air conditioner;
- Fuzzy clustering;
- Fuzzy mathematical programming;
- Fuzzy graph theory;
- Hybrid system (ANFIS)**[42].**

4.2.3. Example of using fuzzy logic system:

Fuzzy logic control

We can use the fuzzy logic system to control a physical system directly (see Figure 4.5), where the inputs are the error and the rate of error from the system. Alternatively, we can use it as an auxiliary controller to assist the main controller (see Figure 4.6)

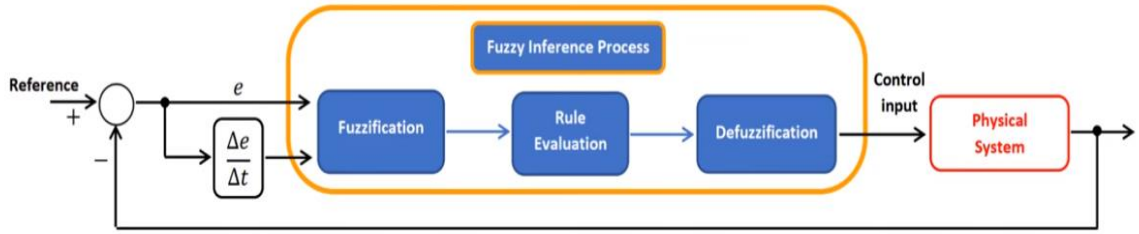


Figure 4.5 Fuzzy logic for direct control**.**

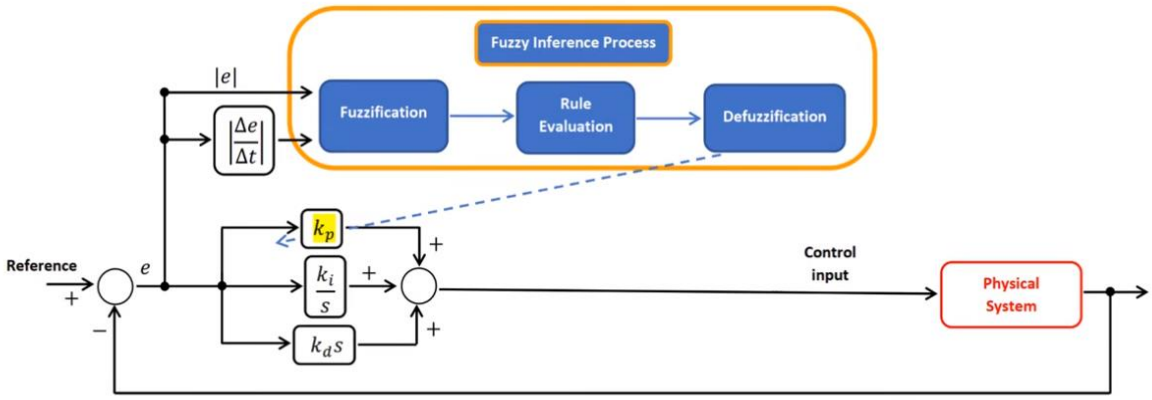


Figure 4.6 Fuzzy logic for helping the PID controller

4.3. Fuzzy inference system (FIS)

Fuzzy inference system called also fuzzy model, based on concept of fuzzy set system, if/then rules and fuzzy reasoning.

The basic structure of a fuzzy inference system consists of three conceptual components:

- **A rule base**, which contains a selection of fuzzy rules;
- **A database**, which defines the membership functions used in the fuzzy rules;
- **A reasoning mechanism**, which performs the inference procedure.

There are three most famous types of fuzzy inference systems that have been widely employed in various applications. The differences between these three fuzzy inference systems FIS lie in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures differ accordingly **[43].**

The most commonly FIS methods used are:

- Mamdani fuzzy models
- Tsukamoto fuzzy models
- Sugeon fuzzy models

4.3.1.Mamdani fuzzy models

The Mamdani fuzzy inference system was proposed as the first attempt in 1975 to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators.

Mamdani-type inference process is completed in **four steps**:

- **Steps 1** Fuzzifier of the inputs variables: translate crisp input into linguistic variables, truth values and membership grade.
- **Steps2** Rule evaluation: Map linguistic variables of the fuzzifier input into linguistic variables related to the defuzzifier output.
- **Step3** Truncation or aggregation of the rule output: apply truncation (mostly for separate, non-overlapping MFs) or truncation + aggregation (for overlapping MFs)
- **Step4** defuzzifier : evaluate the required output value **.**

There is two methods in Mamdani FIS:

- Max –Min inference Method (Rule1 AND)
- Max –Min inference Method (Rule2 OR)

Although, Mamdani fuzzy inference is widely used in many fields, it has limits that make it ineffective in some applications, like:

- Mamdani-type inference requires finding centroid of a two-dimensional shape via integration;
- The process is normally computationally no efficient.
- The process follows high interpretability (advantage) however with low accuracy (disadvantages) **[44]**.

4-3.2.Tsukamoto fuzzy models:

In the Tsukamoto fuzzy models, the consequent of each fuzzy if-then rule is represented by a fuzzy set with a **monotonically** MF, as shown in Figure 4.6. As a result, the inferred output of each rule is defined as a crisp value induced by the rule's firing strength.

The overall output is taken as the weighted average of each rule's output. Figure 4.6 illustrates the reasoning procedure for a two-input two-rule system.

Since each rule infers a crisp output, the Tsukamoto fuzzy model aggregate each rule's output by the method of weighted average and thus avoids the time-consuming process of defuzzifier **[45]**.

However, the Tsukamoto fuzzy model is not used often since it is not as transparent as the Mamdani or Sugeno fuzzy models.

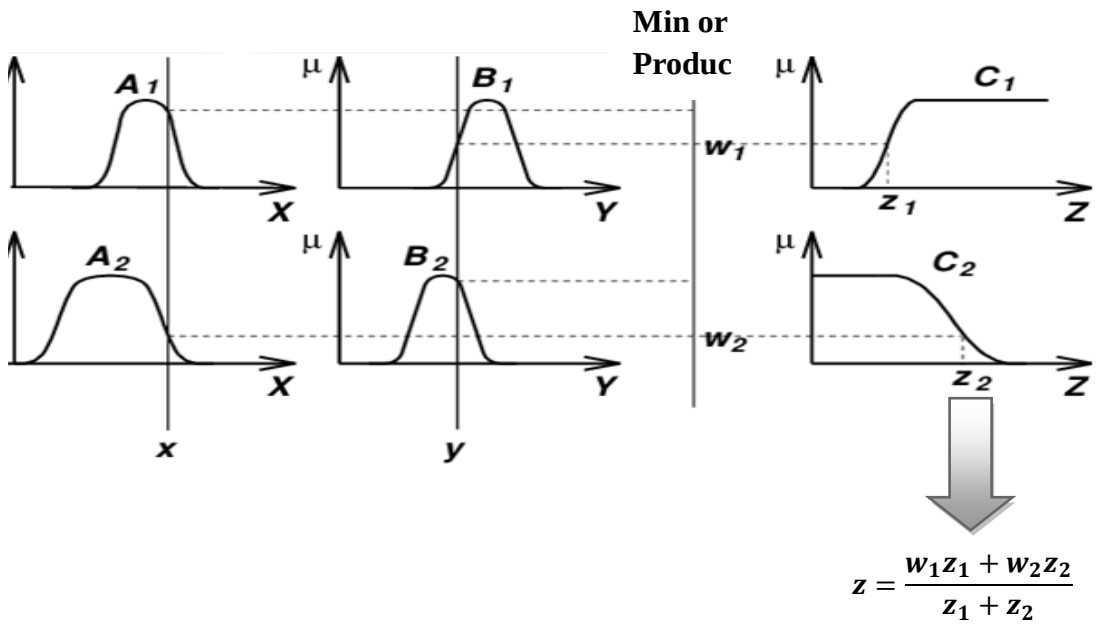


Figure 4.7 The Tsukamoto fuzzy model

4.3.3. Sugeno fuzzy models

The Sugeno fuzzy model (also known as the TSK fuzzy model) is one of the most common fuzzy inference techniques, it was proposed by Takagi, Sugeno, and Kang in 1985 to develop a systematic approach to generating fuzzy rules from a given input-output data set.

The first two steps of the fuzzy inference process are exactly the same of the Mamdani fuzzification of the inputs and using fuzzy operators.

The difference with Mamdani style is that output membership function are only linear, and the crisp output obtained directly.

The typical fuzzy rule of first order Sugeno fuzzy model follows the function:

$$z=f(x,y)= px+qy+r \quad (4.5)$$

Where x, y are the inputs (for example 2), z is the predicted output and p,q and r are the consequent parameters defined arbitrary.

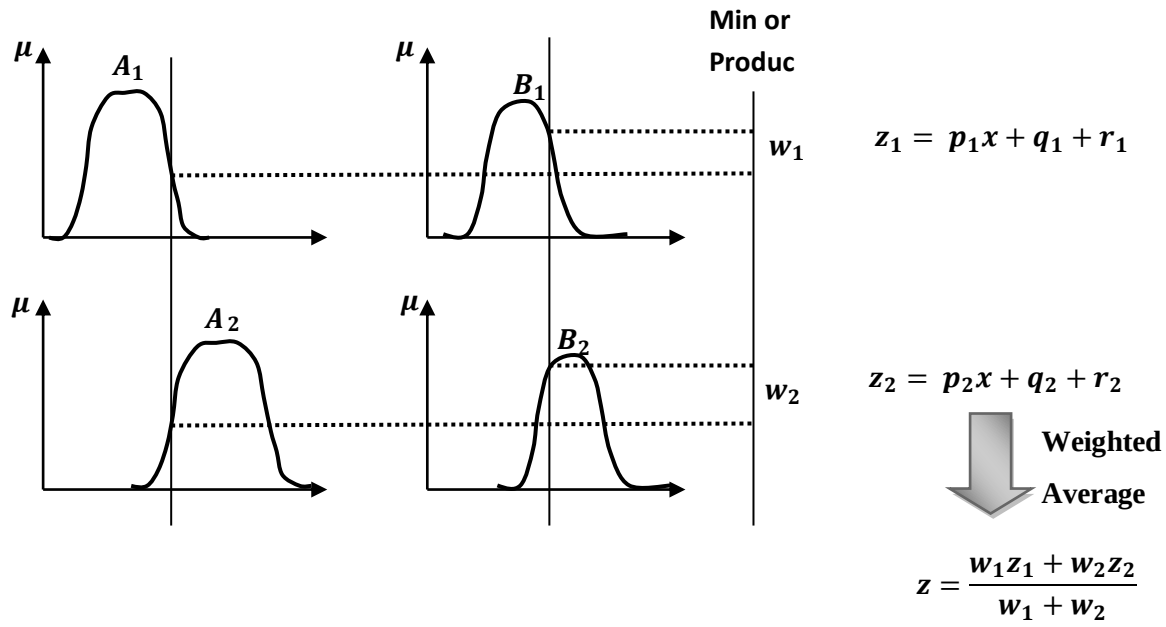


Figure 4.8 The Sugeno (TSK) fuzzy model

The process of the Sugeno fuzzy inference model (shown in Figure 4.7) is as follows:

- **Fuzzification**: Fuzzify the crisp inputs using the chosen membership functions (MF).
- **Rule Evaluation**: Evaluate the fuzzy rules. The total number of rules N_R depends on the number of inputs N_{inp} and the number of membership functions N_{mf} used in fuzzification, calculated as $N_R = N_{mf}$^Ninp. The implication operators generally used are the minimum or product rules.
- **Normalization**: There are two ways to proceed: either normalize the fuzzy sets first or leave them unnormalized until the next step, using the weighted average method.
- **Consequent Calculation**: Calculate the different values of z_i for each rule using equation 4.5, based on the consequent parameters (usually p,q and r).
- **Final Output**: Compute the final crisp output using the weighted average formula in equation 4.6 **[46]**.

$$z = \frac{\sum w_i * z_i}{\sum w_i} \qquad (4.6)$$

Sugeno fuzzy inference follows precise fuzzy modelling:

- This results in high accuracy solution (**advantage**);
- Computationally efficient (**advantage**);
- However with low interpretability (**disadvantage**) **.**

Comparing the two most commonly used FISs, we can clearly observe that, for modelling our photovoltaic system using fuzzy logic, the Sugeno fuzzy model is superior due to its high accuracy, computational efficiency, and, above all, its high precision **[47]**.

4.4. Adaptive network fuzzy inference system ANFIS

4.4.1. Definition of ANFIS

As the name implies, the ANFIS model is composed of two main parts: an adaptive network and a fuzzy inference system (FIS). We have already discussed the FIS in Section 3 of the current chapter.

As for the adaptive network, it is a neural network structure consisting of a number of nodes connected through directional links. Each node represents a processing unit, and the links between nodes specify causal relationships. However, the difference lies in the fact that all or part of the nodes are **adaptive**, meaning the outputs of these nodes depend on modifiable parameters. A learning rule specifies how these parameters should be updated to minimize a prescribed error measure, which is a mathematical expression that measures the discrepancy between the network's actual output and the desired output **.**

4.4.2. ANFIS architecture

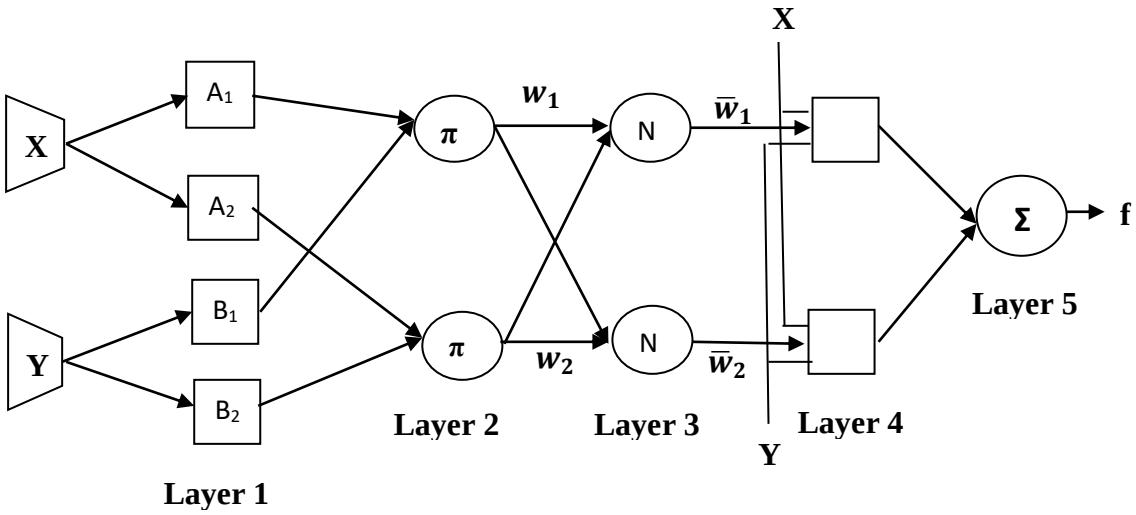


Figure 4.9 Architecture of two-input first-order Sugeno FIS with two rules**.**

For simplicity, we assume the fuzzy inference system under consideration has two inputs (x and y) and one output, as illustrated in Figure 4.8. For a first-order Sugeno fuzzy model, a common rule set consists of the following two fuzzy if-then rules:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$.

Rule 2: If x is A_2 and y is B_2, then $f_2 = P_2x + q_2y + r_2$.

The ANFIS architecture, as shown in Figure 4.8(b), is composed of five layers, with each layer performing a specific function. The nodes in the same layer have similar functions, as described below. (Note that the output of the i^{th} node in layer l is denoted as $O_{l,i}$

- Layer1 "Fuzzification layer": Converts the input data into linguistic values. The parameters, called premise parameters, determine the membership function's shape **[48].**

$$O_{1,i} = \mu_{A_i}(x) \quad for\ i = 1,2 \qquad (4.7)$$

$$O_{1,i} = \mu_{B_i-2}(y) \quad for\ i = 3,4$$

Where x and y are the inputs, μ is the MFs it can be any function and $O_{1,i}$ is the output of the first layer. Here the membership function for A can be any appropriate parameterized membership function introduced in Section 2 of the current chapter, we considered the generalized bell like example equation (4.8).

$$\mu_A(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \qquad (4.8)$$

Where $\{a_i, b_i, c_i\}$ are the premise parameters, it's the parameters of this layer.

- Layer2 "Rules layer": Performs the logical operation "and" by computing the product between the membership functions obtained in the fuzzification layer. The outputs of the product layer act as input weight functions for the next node **[49].**

$$O_{2.i} = w_i = \mu_{A_i}(x) * \mu_{B_i}(y) \quad i = 1,2 \qquad (4.9)$$

Each node output represents the firing strength of a rule.

- Layer3 "The Normalization layer": Calculates the trigger forces belonging to each rule.

In this layer, input weights are normalized to allow the execution of the "and" operation. This layer is labelled as N **[50]**.

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1+w_2} \quad i = 1.2 \qquad (4.10)$$

For convenience, outputs of this layer axe called **normalized firing strengths.**

- Layer4 "Defuzzification layer": It helps obtain the output membership functions using the previously established fuzzy rules **:**

$$O_{4,i} = \overline{w_i}\, f_i = \overline{w_i} \times (p_i \times x + q_i \times y + r_i) \qquad (4.11)$$

Where $\{p_i, q_i, r_i\}$ the parameter set of this node, are referred also as **consequent parameters.**

- Layer5 "Output layer": The summation layer obtains the output by summing the outputs from each rule in the defuzzification layer**:**

$$\boldsymbol{overall\ output} = \boldsymbol{O_{5,i}} = \sum_i \overline{\boldsymbol{w}}_i \boldsymbol{f}_i = \frac{\sum_i \boldsymbol{w}_i \boldsymbol{f}_i}{\sum_i \boldsymbol{w}_i} \qquad (4.12)$$

Next we shall demonstrate how to apply the hybrid learning algorithms developed to identify ANFIS parameters **[51]**.

4.4.3. Hybrid learning algorithm for ANFIS

One of very common method of learning ANFIS is the hybrid learning algorithm, like its name indicates, combines between two methods of estimation and optimization for updating the different parameters of the Model, which are the gradient (or steepest) descent and least-squirt estimator .

ANFIS used two sets of adaptive parameters S_1 and S_2 Such as:

- S_1: **Premise parameters** are the constant of the membership function used in the first layer.
- S_2: **Consequences parameters**: are the constant of the linear regression of the fourth layer $\{p_i, q_i, r_i\}$.

The following table 4.1 show what, when and how theses parameters should be updating.

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least-squirt estimator	Fixed
Signals	Node outputs	Error signal

Table 4.1 Two passes in the hybrid learning procedure for ANFIS**.**

According to the table 4.1, ANFIS uses two passes learning cycle:

- Forward pass: S_1 is fixed and S_2 is computed by Least-squirt estimator algorithm.
- Backward pass: S_2 is fixed and S_1 is computed by using the gradient descent algorithm**.**

Forward pass:

For given values of S1, using K training data, we can rewrite the equation 4.12,(we assume that the ANFIS with two inputs x and y and one output for simplifying) to:

$$f = \bar{w}_1 * (p_1 * x + q_1 * y + r_1) + \bar{w}_2 * (p_2 * x + q_2 * y + r_2)$$

$$f = (\overline{w_1} * x)p_1 + (\overline{w_1} * y)q_1 + \overline{w_1}\, r_1 + (\overline{w_2} * x)p_2 + (\overline{w_2} * y)q_2 + \overline{w_2}\, r_2 \qquad (4.13)$$

We can transform the equation 4.13 to Matrix form:

$$B = AX \qquad (4.14)$$

Where: $B = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_k \end{bmatrix}$ $A = \begin{bmatrix} \overline{w_{11}}x_1 & \overline{w_{11}}y_1 & \overline{w_{11}} & \overline{w_{21}}x_1 & \overline{w_{12}}y_1 & \overline{w_{21}} \\ \overline{w_{12}}x_2 & \overline{w_{21}}y_2 & \overline{w_{12}} & \overline{w_{22}}x_2 & \overline{w_{22}}y_2 & \overline{w_{22}} \\ . & . & . & . & . & . \\ . & . & . & . & . & \\ . & . & . & . & . & . \\ \overline{w_{n1}}x_k & \overline{w_{n1}}y_k & \overline{w_{n1}} & \overline{w_{n2}}x_k & \overline{w_{n2}}y_k & \overline{w_{n2}} \end{bmatrix}$ $X = \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix}$ (4.15)

Where X is the vector of consequents parameters.

This equation solved by $X= (A^T A)^{-1}A^T B$ (4.16)

With $(A^T A)^{-1}A^T$ is the pseudo-inverse of A (we assume that AA^T is non singular)

layer	L-type	No Nodes	No Parameters
L1	Values	p*n	3*p*n=$\|S_1\|$
L2	Rules	p^n	0
L3	Normalize	p^n	0
L4	Linear Func	p^n	(n+1)*p^n= $\|S_2\|$
L5	Sum	1	0

Table 4.2 The ANFIS computational complexity.

Where n is the number of inputs and p is the number of MFs used.

Least-squirt estimator minimizes the error $\|AX - B\|^2$ by approximate X to X^*

For solving the equation 4.16 to find X^*, we can use the recursive least squirt estimator:

$$\left.\begin{aligned} S_{i+1} &= S_i - \frac{S_i a_{(i+1)} a_{(i+1)}^T S_i}{1+a_{(i+1)}^T S_i a_{(i+1)}} \\ X_{i+1} &= X_i + S_{(i+1)} a_{(i+1)} (b_{i+1}^T - a_{(i+1)}^T X_i) \end{aligned}\right\} for\ i = 0,1,\dots,K-1 \quad (4.17)$$

Where:

- Initialized constant $X_0=0$, $S_0=\gamma I$ (Where γ is a large positive number and I is $n \times n$ identity matrix)
- a_i^T is i^{th} line of matrix A, b_i^T is i^{th} value of the vector B and $X^*=X_k$

Backward pass:

We start by computing E_k (for the k^{th} ($1 \le k \le K$) entry of k^{th} training data) with equation.4.18

$$E_k = \sum_{i=0}^{N_L} (t_i - O_{L,i})^2 \quad (4.18)$$

Where : N_L: number of nodes in layer L; t_i: i^{th} component of desired output vector;

$O_{L,i}$=i^{th} component of actual output vector.

Let simplify the error equation (Assuming single dimension output vector):

$$E_k = \frac{1}{2}(t_k - O_{L,k})^2 \quad (4.19)$$

Overall error measure E is $E = \sum_k^K E_k$ (4.20)

For each premise parameter the update formula is:

$$\Delta\beta_i = -\eta \frac{\partial^+ E}{\partial \beta_i} \quad (4.21)$$

Where $\eta = \frac{k}{\sqrt{\sum_i \left(\frac{\partial E}{\partial \beta_i}\right)^2}}$ the learning rate and k is is step size and $\frac{\partial^+ E}{\partial \beta_i}$ is the ordered derivative.

For every parameters of MFs used, we have to calculate the partial derivatives of error function with respect to each parameters of S_1, which need to be tuned, with the following equation:

$$\frac{\partial E_k}{\partial \beta_i} = \frac{\partial E_k}{\partial O_k}\frac{\partial O_k}{\partial \beta_i} = -(t_k - O_k) \times \frac{\partial O_k}{\partial \beta_i} \quad (4.22)$$

Using the chain rule error rate for β_c consequent parameters can be calculated as:

$$\frac{\partial E_k}{\partial \beta_c} = \frac{\partial E}{\partial O_5}\frac{\partial O_5}{\partial O_4}\frac{\partial O_4}{\partial \beta_c} \quad (4.23)$$

The Error rate for premise parameters can be calculated by the chain rue as follows:

$$\frac{\partial E_k}{\partial \beta_p} = \frac{\partial E}{\partial O_5}\frac{\partial O_5}{\partial O_4}\frac{\partial O_4}{\partial O_3}\frac{\partial O_3}{\partial O_2}\frac{\partial O_2}{\partial O_1}\frac{\partial O_1}{\partial \beta_p} \quad (4.24)$$

Where:

$$\frac{\partial E}{\partial O_5} = -(t_k - O_k) \quad (4.25)$$

$$\frac{\partial O_5}{\partial O_4} = \frac{\partial \sum_i f_i \bar{w}_i}{\partial (f_j * \bar{w}_i)} = 1 \quad (4.26)$$

$$\frac{\partial O_4}{\partial O_3} = \frac{\partial (f_j * \bar{w}_i)}{\partial \bar{w}_i} = f_i \quad (4.27)$$

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial \left(\frac{w_i}{\sum_{j=1}^{n} wj}\right)}{\partial w_i} = \frac{\sum_{j=1}^{n} w_j - w_i}{\left(\sum_{j=1}^{n} w_j\right)^2} \quad (4.28)$$

$$\frac{\partial O_2}{\partial O_1} = \frac{\partial \left(\prod_{A_j \in R(m)} A_j\right)}{\partial A_m} = \prod_{A_j \in R(m), A_j \neq A_m} A_j \quad (4.29)$$

Where $A_j \in R(m)$ denote the fuzzy sets, which make the premise part of rule containing fuzzy set Am.

According to the equations from 4.24 until 4.29 we will get:

$$\frac{\partial E_k}{\partial \beta_p} = -(t_k - O_k) \times f_i \times \frac{\sum_{j=1}^{n} w_j - w_i}{\left(\sum_{j=1}^{n} w_j\right)^2} \times \prod_{A_j \in R(m), A_j \neq A_m} A_j \times \frac{\partial O_1}{\partial \beta_p} \quad (4.30)$$

$\frac{\partial O_1}{\partial \beta_p}$ is depend to the MF using, we must computing for each parameters .

If the chosen MFs are Generalize Bell, we get the simplified partial derivatives respect of each premise parameter:

$$\frac{\partial O_1}{\partial a_{ij}} = \begin{cases} \frac{2b_{ij}}{a_{ij}} y_{ij}(1 - y_{ij}) & if\ x \neq c_{ij} \\ 0 & if\ x = c_{ij} \end{cases} \quad (4.31)$$

$$\frac{\partial O_1}{\partial b_{ij}} = \begin{cases} -2ln\left|\frac{x - c_{ij}}{a_{ij}}\right| y_{ij}(1 - y_{ij}) & if\ x \neq c_{ij} \\ 0 & if\ x = c_{ij} \end{cases} \quad (4.32)$$

$$\frac{\partial O_1}{\partial c_{ij}} = \begin{cases} \frac{2b_{ij}}{x-c_{ij}} y_{ij}(1-y_{ij}) & if\ x \neq c_{ij} \\ 0 & if\ x = c_{ij} \end{cases} \quad (4.33)$$

Where $y_{ij} = \frac{1}{1+\left|\frac{x-c_{ij}}{a_{ij}}\right|^{2b_{ij}}}$ note that it's the used MF (4.34)

Now we can calculate the error rate for each premise parameters, and by using the equation 4.21 we can update the parameters by the equation 4.35 **[52]**

$$\beta_{ij} = \beta_{ij} + \Delta\beta_{ij} \quad (4.35)$$

Different ways of combining steepest descent and LSE

The computational complexity of the least-squares estimator (LSE) is usually higher than that of steepest descent (SD) for one-step adaptation. However, for achieving a prescribed performance level, the LSE is usually much faster. Consequently, depending on the available computing resources and required level of performance, we can choose from among at least five types of hybrid learning rules combining SD and LSE in different degrees, as follows:

- **One pass of LSE only**: Nonlinear parameters are fixed while linear parameters are identified by one-time application of LSE.
- **SD only**: All parameters are treated as nonlinear and updated by SD iteratively.
- **One pass of LSE followed by SD**: LSE is employed only once at the beginning to obtain the initial values of linear parameters, and then SD take over to update all parameters iteratively.
- **SD and LSE**: Linear and nonlinear parameters are distinguished first. Each iteration (epoch) of SD used to update the nonlinear parameters is followed by LSE to identify the linear parameters.
- **LSE only**: The outputs of an adaptive network are linearized with respect to its parameters **[53].**

4.4.4. ANFIS in MPPT

Solar energy has established itself as one of the main renewable energy sources for electricity generation. The power generated by a PV system is intrinsically related to weather conditions, particularly solar irradiance, and ambient temperature. These highly variable and nonlinear conditions present significant challenges in harnessing solar energy**.**

A particularly promising approach is using ANFIS, which have demonstrated exceptional performance and significant potential within the AI domain**.** Due to its promising results, the

ANFIS architecture has been widely implemented in PV systems. However, it is relevant to note that there remains a need for additional and comparative research in this domain **[54]**.
ANFIS has emerged as one of the advanced MPPT techniques to address the challenges associated with ANN algorithms and fuzzy approaches. ANFIS offers remarkable advantages, such as fast learning capability, adaptability, and the ability to deal with imperfect sensory information, deficiencies, and abrupt changes in system dynamics **[55]**.

Ref	Year	Dc-Dc	Findings in references
[56]	2016	Boost	ANFIS better tracking as Fuzzy Logic Controller and ANN techniques
[57]	2017	Buck	ANFIS exhibit greater efficiency and dynamic response than IncCond and CV
[58]	2016	Boost	ANFIS can efficienly and accurately track the GP under various partial shading conditions and configurations
[59]	2014	Boost	ANFIS can efficiently and quickly track the unique MPP under dynamic and steady-stat conditions.
[60]	2012	Boost	ANFIS is used to extract the GP of the Total CTC by adjusting the duty cycle of boost converter

Table 4.3.Comparative studies of ANFIS with other MPPT techniques **[61].**

4.5. Design and development of fuzzy –based MPPT

In the previous chapter, we covered the design of the MLP model for MPPT. In this section, we present a proposed fuzzy logic model for MPPT using the ANFIS architecture with hybrid learning. Given the complexity of the model, we chose a single pass of the Least Squares Estimator), as it yields excellent results with reduced complexity and in a shorter time.

4.5.1. Description of the proposed model

The training methodology for the ANFIS system is summarized in Figure 4.10. The process begins by obtaining a training data set (comprising two inputs and one output) and a checking data set (which features the same number of inputs and outputs, but a different number of samples). The objective of the training data is to calibrate the ANFIS system by determining the optimal premise parameters for the membership functions. Additionally, an error threshold between the actual and desired outputs is established.

By using the hybrid method for learning the ANFS, The consequent parameters are found using the least squares method. If this error is larger than the threshold value, then the premise parameters are updated using the gradient decent method **[62]**.

The proposed model is an ANFIS model utilizing Sugeno fuzzy inference system, featuring two inputs (temperature and irradiation) and a single output (the MPP). The desired MPP values for training and testing the model are imported from the mathematical model of the PV panel described in the first chapter.

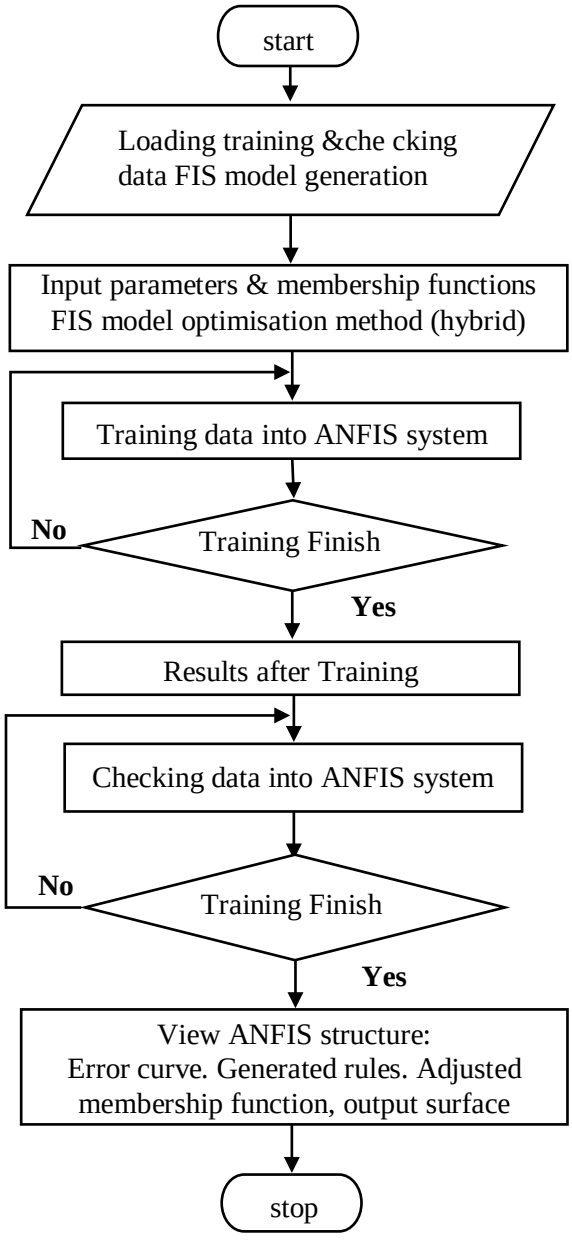


Figure 4.10. Flowchart of training ANFIS system

The architecture of the proposed model:

- Two inputs (n=2) Temperature 'T' and Irradiation 'G';
-3 MFs (p=3) for each input, all of its are Generalization Bell (a, b, c);
- Number of premise parameters = 3*p*n=3*3*2=18;
- Number of nodes 9;
- Number of rules $p^n=3^2=$ 9 rules if-than product;
- Number of consequent parameters $(n+1)*p^n =3*3^2=27$.

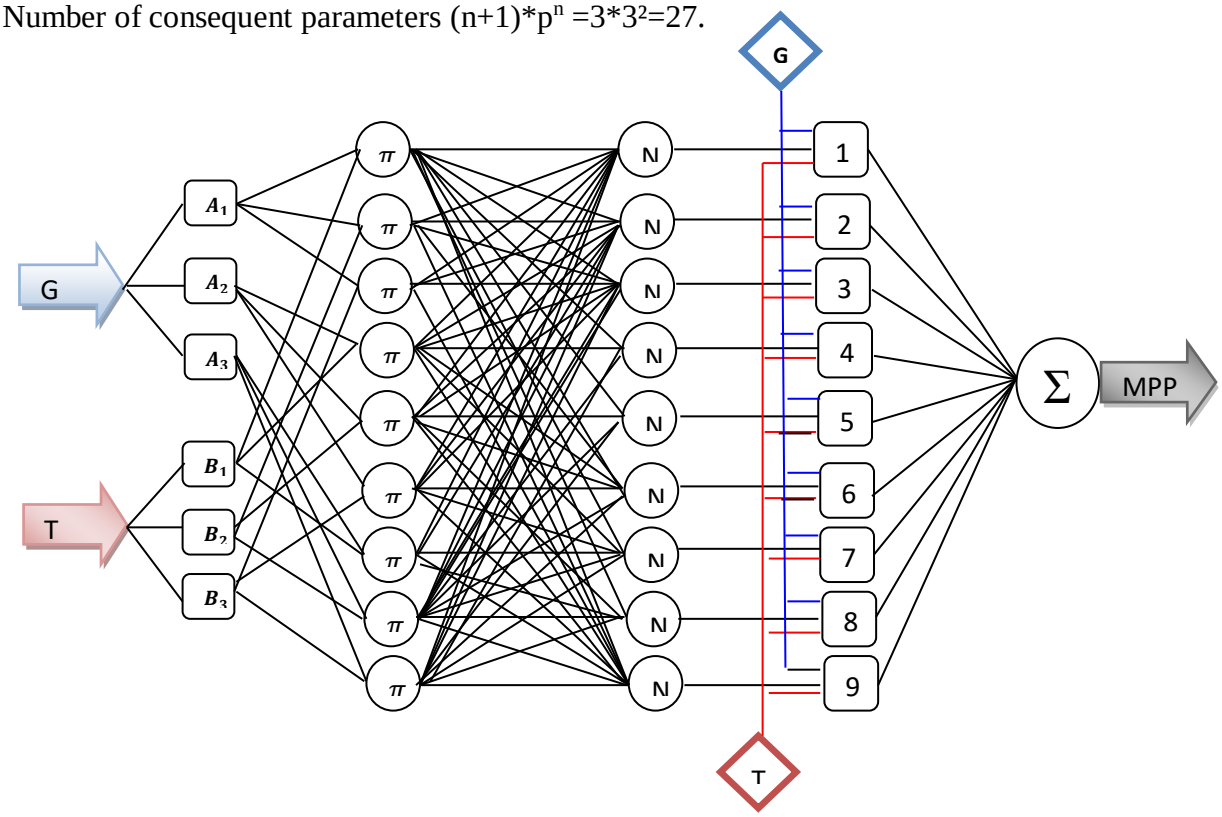


Figure 4.11 Architecture of proposed model

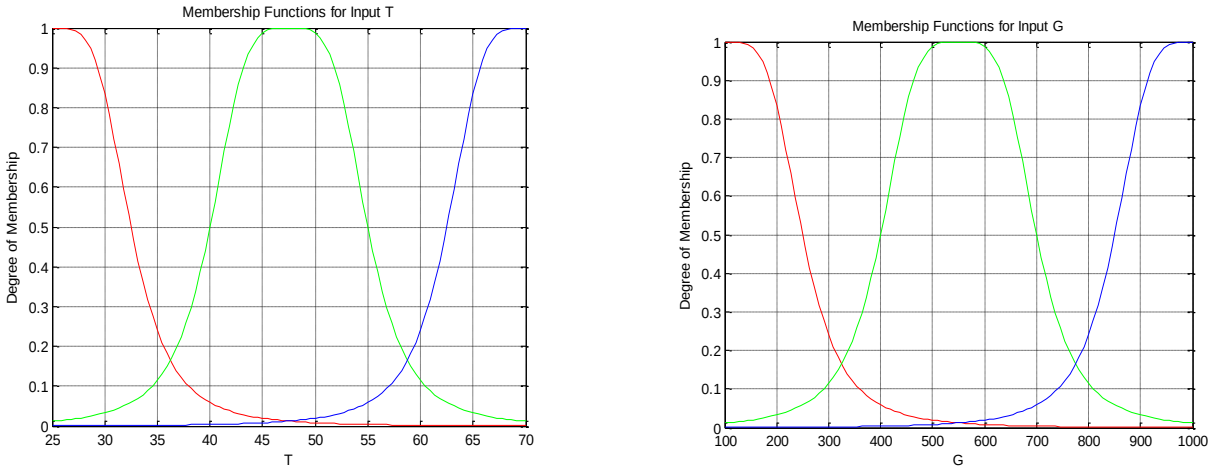


Figure 4.12 membership function of the Modal using Generalized Bell

4.5.2.Supervised learning of the Model

We used hybrid learning to prepare our model, employing only a single pass of the least squares estimator (LSE). This approach is less complex, faster, and yielded highly acceptable results. For the input and training data, we followed the same procedure as the MLP: we used the results from the PV panel model to create the training and test datasets. To update the consequent parameters, we used a R-LSE. The training results are shown in Figure 4.13.

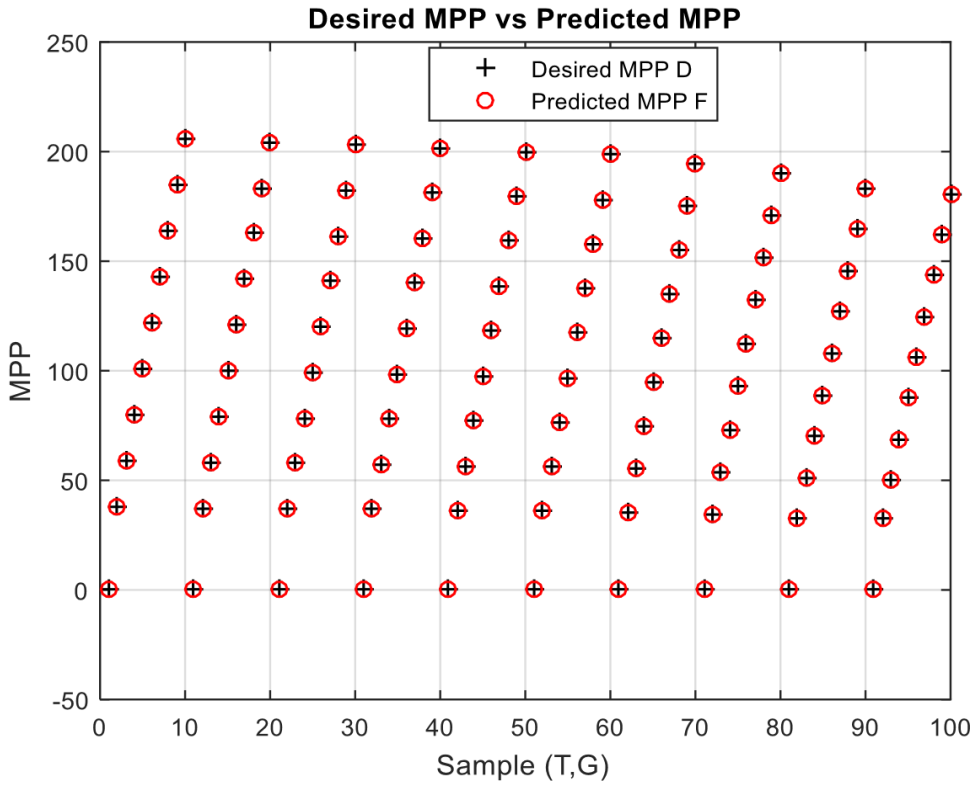


Figure 4.13 Result of training the ANFIS model

According to the Figure 4.12, it can be clearly seen that the predicted values overlap with the desired values for 100 selected samples of temperatures and irradiations, achieving a MSE of 0.00002.

We can deduce that the model is ready to be tested first with different values to ensure its readiness for validation and practical use. Therefore, we save the model parameters (consequent and premise parameters), perform a single forward pass, and observe the resulting outputs. In the next chapter (dedicated to simulation and analysis), we will test our model under different conditions to confirm its accuracy, speed, and stability.

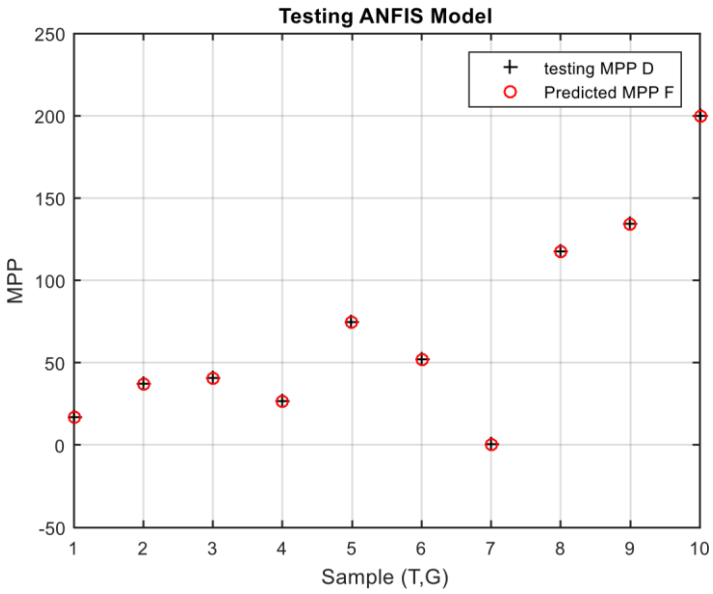


Figure 4.14 Result of testing the ANFIS model

The results of the testing are very satisfactory; we can now say that our model is ready to be validated and used for tracking the MPP.

4.6. Conclusion and analysis

In this chapter, we explored the fuzzy logic, starting by describing the core concepts of this theory and its relation to human reasoning. Then, we detailed the most commonly used methods of fuzzy inference systems. Furthermore, we examined the Adaptive Network-based Fuzzy Inference System (ANFIS), defining its architecture and explaining how it is used to model learning machines. This involved utilizing hybrid learning to update its parameters. Finally, we described an ANFIS model proposed for Maximum Power Point Tracking (MPPT). We can conclude that ANFIS is a powerful hybrid framework that combines the learning capabilities of neural networks with the rule-based reasoning of fuzzy logic.

To summarize, we chose the ANFIS as a fuzzy-based MPPT due to its many advantages, which include:

- Higher efficiency under dynamic conditions specially partial shading conditions (PSC);
- Faster tracking, speed and robustness;
- It combine the benefits of both Fuzzy Logic and Artificial neural network;
- It is simple and does not require much computing or mathematical equations.

This does not preclude the fact that ANFIS also has certain limits, including the following:

- High complexity and expensive;
- Difficult to obtain membership functions and rules;
- More sensors are required;

- Insufficient training on PV array leads to less accuracy [61].

At this point, we have all the tools necessary to move on to the final chapter, which is dedicated to simulation, evaluation, and validation. As seen throughout the preceding chapters, we have progressed from theoretical concepts to the practical implementation of MPPT models. The fifth and final chapter will provide an analysis and summary of all that we have described and studied.

Chapter 5:
Simulation Results, Performance Evaluation, and Validation

5.1. Introduction

In the preceding chapters, we conducted a comprehensive study on photovoltaic (PV) modules, classical MPPT techniques, Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural network models, as well as Fuzzy Inference Systems (FIS) and ANFIS. This foundational background was essential, as it introduced these concepts to our curriculum for the first time. We now consider this knowledge a major asset as we conclude our master's degree.

Additionally, three implemented models are described at the end of each chapter. Although the implementation of the classical MPPT techniques studied in Chapter 2 was not required, we would have liked to include at least one of them had time permitted.

In this chapter, we evaluate the design and simulation of MATLAB-implemented MPPT intelligent estimators. First, we establish the PV panel mathematical model, followed by the behavioral structures of the MLP and Fuzzy Logic MPPT estimators. We then conduct two separate comparative studies, benchmarking the baseline model against the ANFIS-based estimator and the MLP-based neural network model. Concluding the chapter, we analyze the transient response of the proposed models under dynamic operating conditions and assess their overall tracking efficiency.

5.2. MATLAB simulation setup

5.2.1. PV panel model

Using the mathematical equations from Chapter 1, we modelled a Kyocera KC200GT PV module with a maximum power of 200 W.

We first verified its performance by plotting the characteristic PV panel I-V and P-V curves; the results are shown in Figure 5.1.

We can easily see that the two curves accurately represent the electrical performance of the Kyocera KC200GT PV panel under standard conditions (T=25°C and G=1000 W/m²) by comparing the graphs with the datasheet specifications (Annex 1)."

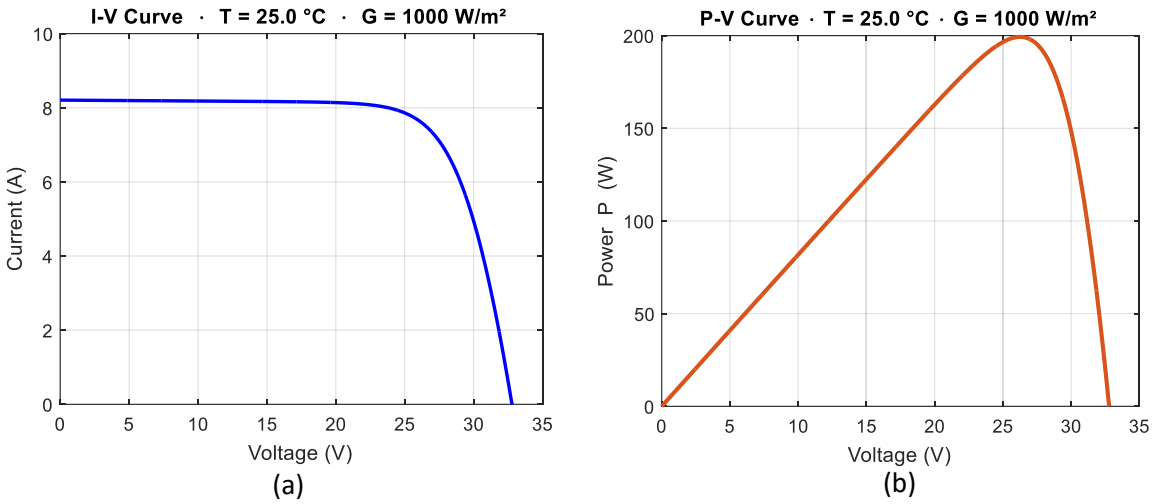


Figure 5.1: Characteristic curves (a) I-V curve (b) P-V curve

	PV modal from the curves	Kyocera KC 200 GT
Maximum power (Pmax) W	200	200
Maximum power Voltage (V_{mpp}) V	26.4	26.3
Maximum power currant (I_{mpp}) A	7.58	7.61

Table 5.1: Comparison between the main results of the proposed mathematical model and the experimental data for the Kyocera KC 200GT PV module.

The following curves showing the I-V and P-V characteristics in such cases. In Figure 5.2**,** the temperature is kept constant while the irradiation varies. Conversely, in Figure 5.3, the irradiation is fixed and the temperature is varied.

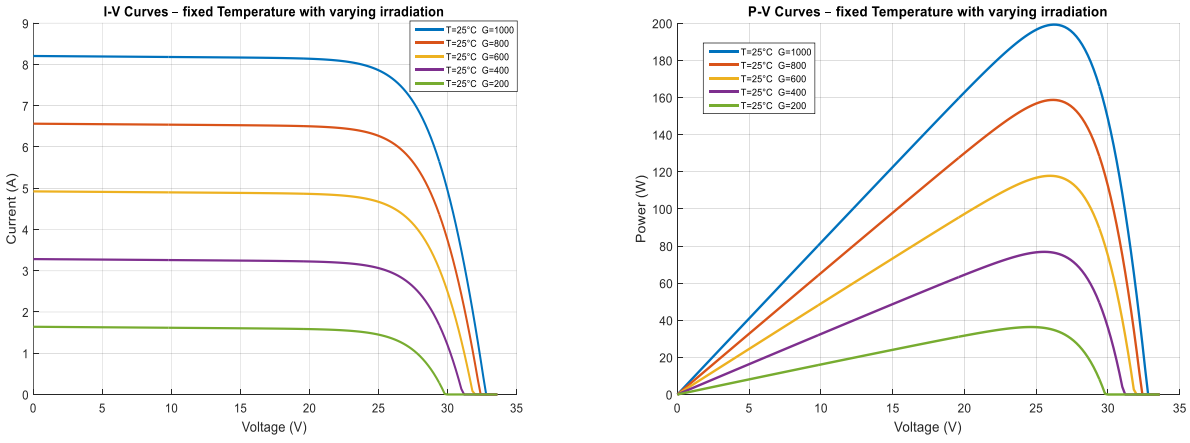


Figure 5.2: I-V and P-V curves for fixed temperature **25°** with varying irradiations (**W/m²**) **[200, 400, 600, 800, 1000]**

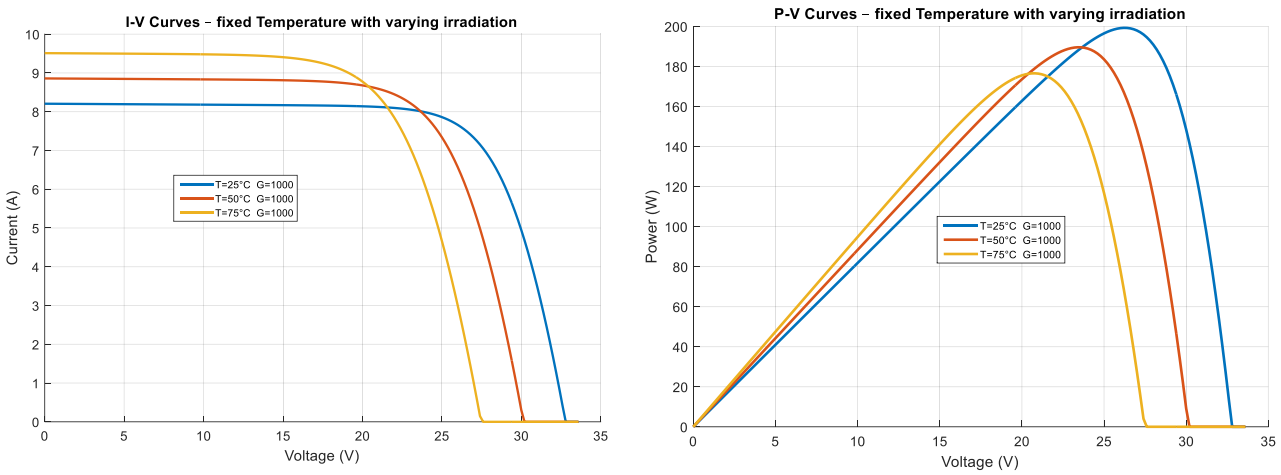


Figure 5.3: I-V and P-V curves for fixed irradiations **1000W/m²** with varying temperature **(°C) [25, 50,75]**

After testing the PV model, we can now create a dataset for training and testing the MLP and ANFIS based MPPT models. This dataset is structured as a matrix that contains the input variables, and one output corresponding power values calculated by the PV module model. These power values will be considered the desired outputs, representing the Maximum Power Point (MPP) to be tracked.

The PV-modal needs two inputs, Temperature T and Irradiation G, to calculate one necessary output, the MPP. Let's take, for example, 100 input samples chosen either randomly or based on prior experience, as the careful selection of samples can aid in the learning process. From these 100 sample pairs (T, G), we obtain 100 corresponding MPP values to use like desired MPP for training the models. The testing data can be created in the exact same way.

5.2.2. MLP for MPPT model

Before moving to the Fuzzy model, we decided to first implement an MLP model to practice with artificial neural networks and compare the results between the two techniques.

We describe this model in section 8 of chapter 3. All of our design choices were the result of extensive experimentation to achieve the best possible results, to improve the model, we followed four steps: **initialization, learning, testing, and validation.**

⇒ **Initialized the model**

For the architecture, we chose an MLP with two hidden layers and variable number of nodes, ultimately determining that 40 nodes per layer yielded satisfactory results. For the activation function, we selected the hyperbolic tangent sigmoid (tansig en MATLAB), defined as

$a(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, this function was chosen because it converges faster and provides a broader output range of [-1, 1], additionally, employing two hidden layers with 40 nodes each makes the tansig function highly effective in hidden layers, as it exhibits strong gradient descent properties that accelerate learning.

Finally, we imported the Maximum Power Point (MPP) target from the PV-panel model for MLP supervised learning**.**

⇒ **Training the model**

We use the database to train our model, applying the back-propagation method to update the model parameters (weights and biases) by minimizing the average cost (see section8, third chapter). The results are shown in Figure 5.4

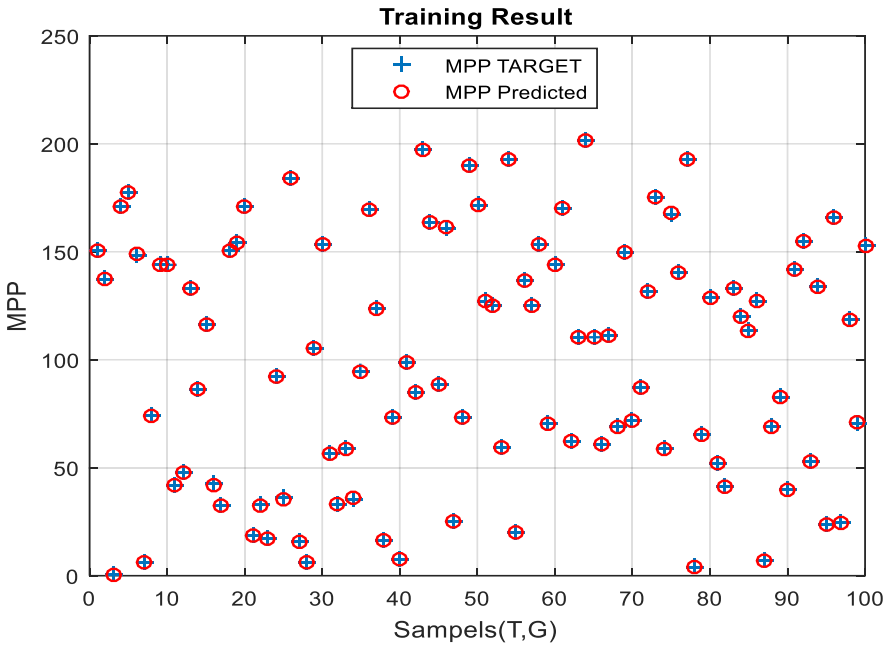


Figure 5.4: Learning Results of the MLP code predicted VS target.

The results are very satisfactory, as the training time was about 6 minutes and 48 seconds. We can now save the updated parameters (weights and biases) to use in the testing phase.

⇒ **Testing the model**

To test the model, we will consider three different cases that mimic the possible conditions it may encounter. We will simulate with 10 samples as example in each case.

Case1- Variable T and Variable G:

T=[33, 05, 28, 41, 50, 63, 39, 0, 70, 58] (°C)

G= [220, 312, 718, 520, 632, 900, 0, 1000, 325, 415] (W/m²)

The results are showing in figure 5.5

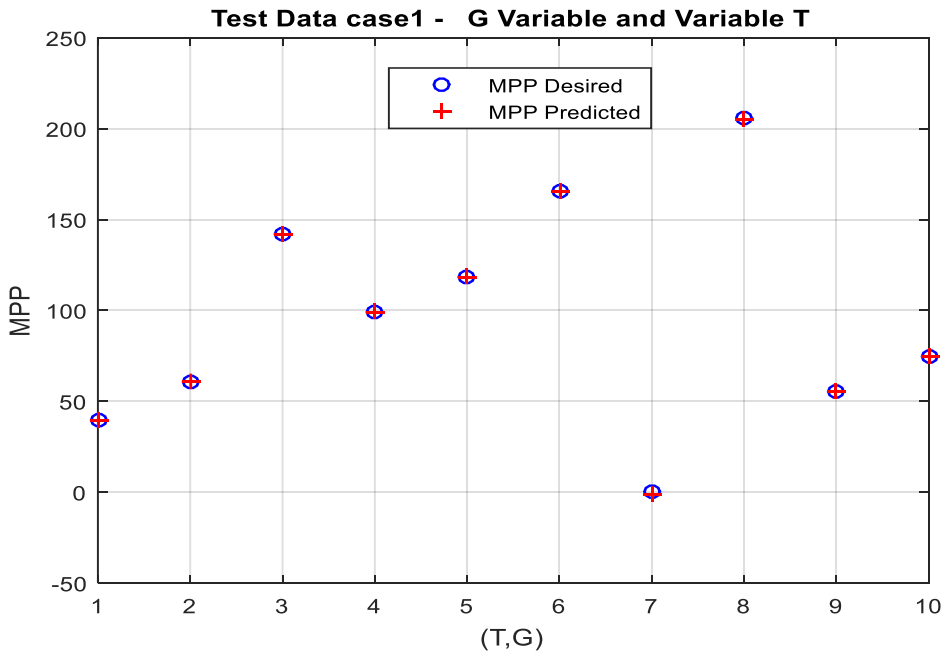


Figure5.5: Case 1 MLP test (T variable, G variable)

➢ **Case2**- Fixed T and Variable G:

T= 35°C G= [250, 470, 620, 870, 1000, 360, 50, 995, 0, 110] (W/m²)

The results are showing in figure 5.6

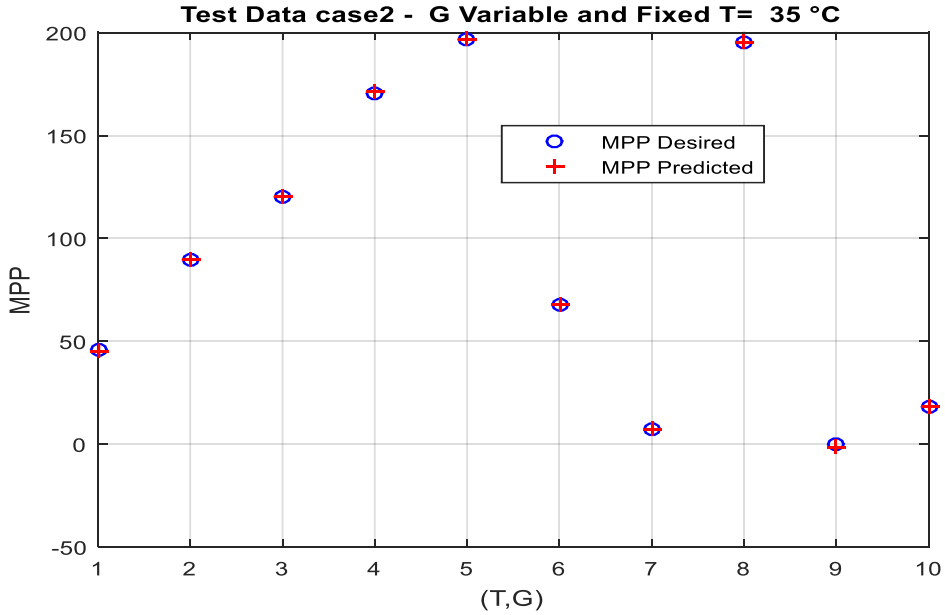


Figure5.6: Case 2 MLP test (T Fixe, G variable)

Case3- Fixed G and Variable T:

G= 800 W/m² T= [15, 22, 45, 62, 53, 70, 35, 29, 49, 75] (°C)

The results are showing in figure 5.7

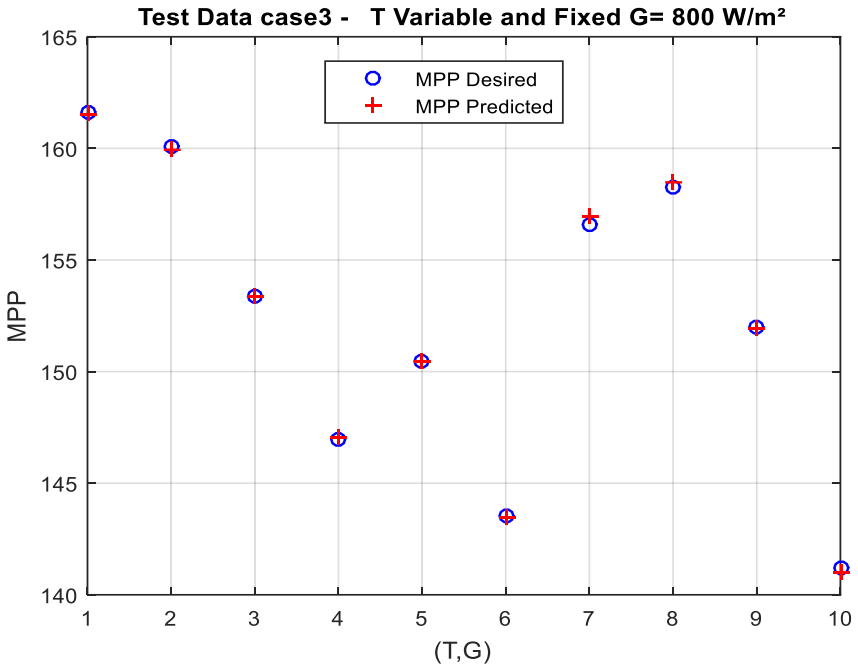


Figure5.7: Case 3 MLP test (G Fixe, T variable)

Analysis: we can resume the results in these points:

- The MLP based MPPT is very faster than mathematical modal;
- The results are very satisfactory;
- The modal resist the dynamic conditions (the change of one or all of the inputs).

⇒ **Validation and using**

Finally, our model is ready for validation. We can simply enter the temperature and irradiation to determine the Maximum Power Point (MPP).

Now, we move on to the ANFIS based MPPT modal.

5.2.3.Fuzzy model for MPPT

As seen in Chapter 4, specifically in Section 5, we described the proposed fuzzy model based on ANFIS. Our model has 3 membership functions (MFs) for each of the two inputs.

The MF used is the generalized Bell shape with 3 premise parameters each, which means we have a total of $S_1 = 3 \times 6 = 18$ premise parameters.

The model also has 9 rules, and in Layer 4, each rule contains 3 consequent parameters. Therefore, we have $S_2 = 3 * 9 = 27$ consequent parameters.

The total numbers of parameters to be updated are $S = S_1 + S_2 = 18 + 27 = 45$ parameters.

This architecture appears complex due to the high number of parameters, which is why we chose to use hybrid learning with only a single pass of the Least Squares Estimator (LSE): the nonlinear parameters are kept fixed, while the linear parameters are identified by a one-time application of the LSE.

⇒ **Initialization of the model**

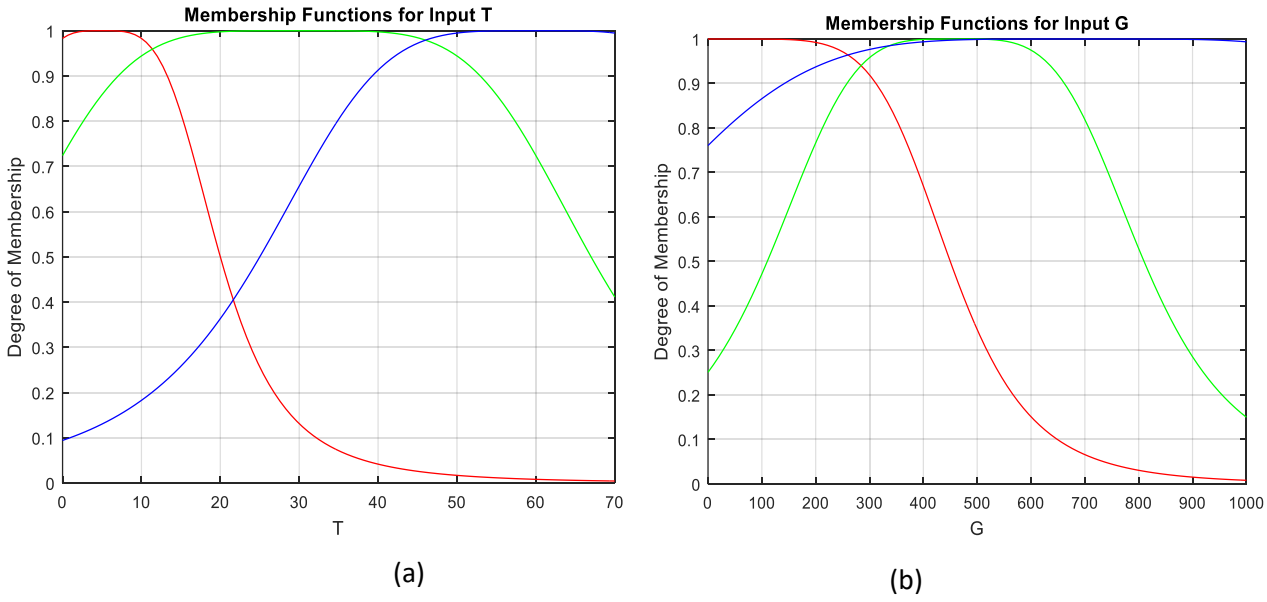
Similar to the MLP model, the inputs and desired output are obtained from the PV panel mathematical model. The premise parameters were initialized manually and will be updated during the learning phase. The consequent parameters will be calculated using a recursive LSE algorithm.

⇒ **Learning of the model**

The chosen input parameters and their respective membership function curves are presented in Table 5.2 and Figure 5.8 respectively. The Figure 5.9 illustrates the training results.

inputs	**Temperature**			**irradiation**		
premise parameters	a	b	c	a	b	c
μ_1	10	1.6	0	450	3	0
μ_2	17.5	1.9	20	350	2	460
μ_3	52	2.19	40	900	2.3	700

Table 5.2 Membership functions and there parameters.



(a) (b)

Figure 5.8 Membership functions curves (a)-3 MFs of T (b)- 3MFs of G

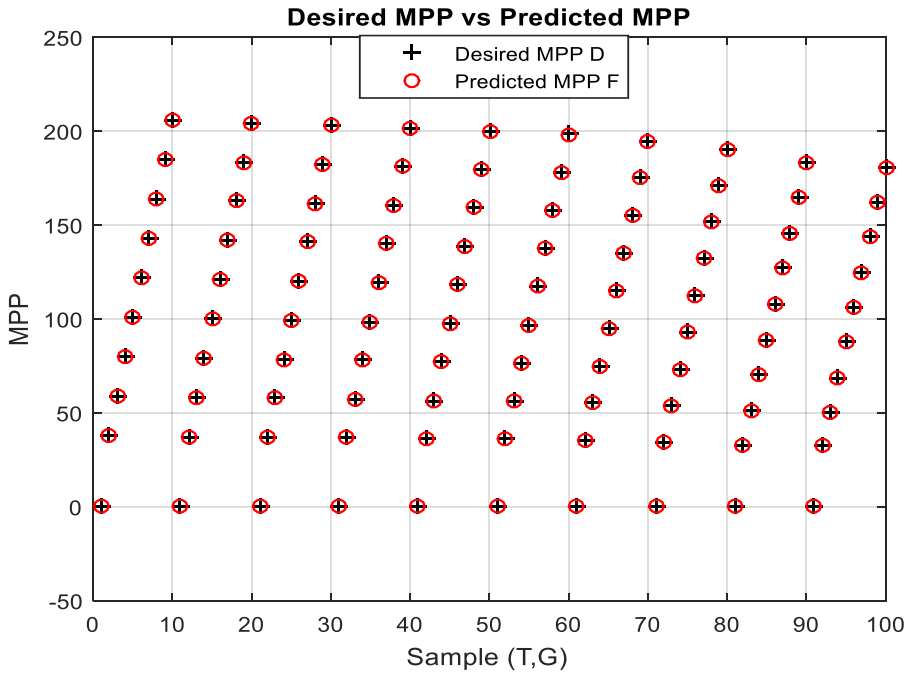


Figure 5.9: Learning Results of the ANFIS code: predicted VS target.

The training results are very satisfactory with average cost MSE=0.00002, and the curve is perfect, we can move to the next step, testing the model after saving the modal parameters.

⇒ **Testing of the model:** We will follow the same steps like we did to testing MLP base modal.

Case1: Variable T(°C) and Variable G(W/m²), for 10 samples

T=[25, 10, 28, 30, 40, 60, 0, 29, 70, 25], G=[100, 200, 220, 150, 400, 300,0,600,750,1000]

The results are showing in figure 5.10

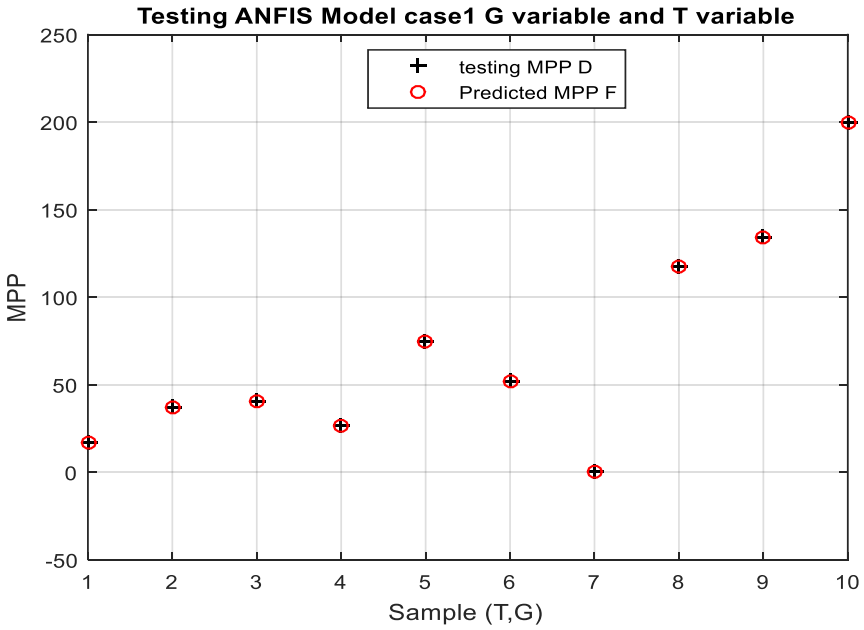


Figure 5.10: Case 1 ANFIS test (T variable, G variable).

Case2: Fixde T(°C) and Variable G(W/m²), for 10 samples

T=39, G= [310, 150, 50, 420, 660, 950, 810, 506 305, 0]

The results are showing in figure 5.11

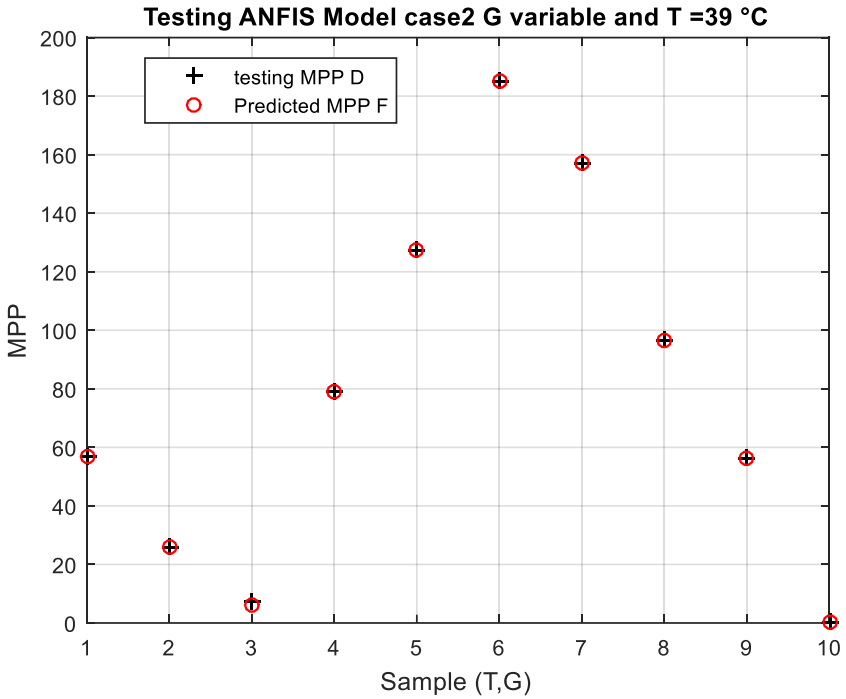


Figure 5.11: Case 2 ANFIS test (T fixed, G variable).

Case3: Variable T(°C) and Fixde G(W/m²), for 10 samples

G=750, T= [33 05 28 41 50 63 39 0 70 58]

The results are showing in figure 5.12

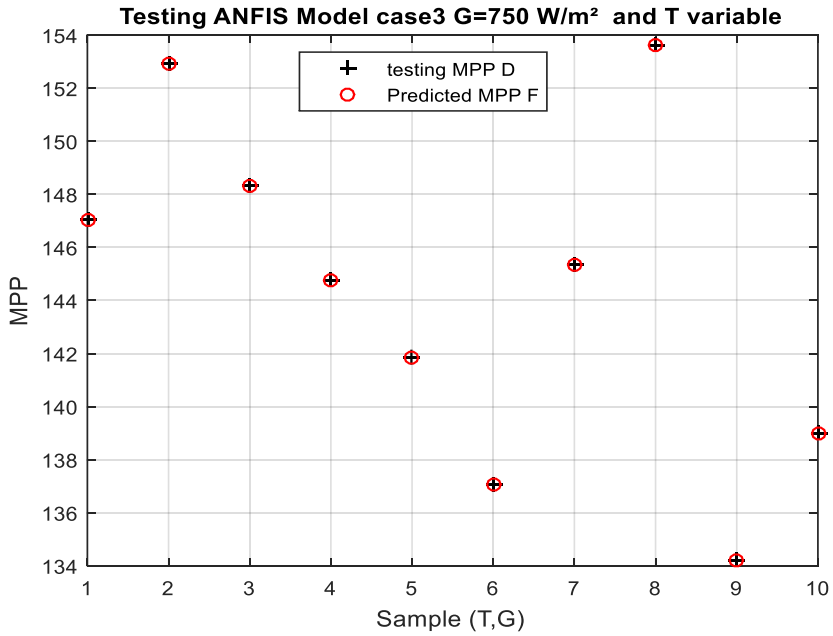


Figure 5.12: Case 3 ANFIS test (G fixed, T variable).

Analysis-The results can be summarized in the following points**:**

- The ANFIS-based MPPT is significantly faster than the MLP-based model.
- The results are highly satisfactory.
- The model is very stable and robust against dynamic conditions (changes in one or all inputs).

⇒ Validation and using

Finally, our model is validated and ready for use. We can simply enter the temperature and irradiation to determine the Maximum Power Point (MPP). Now, we move on to the ANFIS-based MPPT model.

5.3. Comparison between fuzzy model and mathematical model

The following table 5.3 resume the comparison between ANFIS, MLP and Mathematical models “according to our models”

	ANFIS	MLP	Mathematic
Number of samples	100	100	100
Time of learning	1.78 s	30s	/
Number of parameters	45	1801	/
complexity	Middle	High	High
flexibility	High	Low	Low
learning ability	High	Higher	/
MSE	0.000020	0.000004	0

Table 5.3: Comparison between implemented Models and mathematical

The complexity of the fuzzy model is in the implementation in first place, after validation it we’ll be very fast and accuracy.

The ANFIS based MPPT is more powerful than the other models used; it is more accuracy, faster and very stable.

5.4. ANFIS-based MPPT performance analysis

We easily observe that the MPPT tracking performance of the fuzzy logic system is much better compared to classical algorithms (PSO, P&O, incCon...). We can summarize these advantages as follows:

- Offers a much faster response to variations in irradiation and temperature than classical control algorithms like PSO and P&O.
- Maintains higher stability in tracking the MPP by adjusting its parameters in real time, whereas PSO takes a longer time to stabilize and is less optimal.

- Demonstrates high performance because of its ability to react instantaneously to variable conditions.

Thus, ANFIS model is better suitable for all system needs accuracy, rapidity and stability.

5.5. Evaluation of Efficiency Improvement

Although our model has demonstrated its efficacy, it can be further refined and enhanced by the following recommendations:

- Add the gradient descent to update the premise parameters: unfortunately we missed the time to do it.
- Using Kalman filter for **cleaning noisy sensor data** and **adaptively estimating system states**. This makes the MPPT tracking significantly more robust, faster, and free of steady-state oscillations, especially under partial shading.

Research into ANFIS-based MPPT estimators and controllers is receiving significant attention.

Current efforts focus on improving tracking speed, mitigating complex partial shading, and implementing physical hardware in renewable systems. Examples:

- The research introduces a hybrid GMPPT technique namely ANFIS-GA based on ANFIS (Approximate Nonlinear Functional) to model purely nonlinear behaviour of PV and Genetic Algorithm (GA) based controller optimization which yield a higher tracking accuracy under multiple shading cases.
- The development of an intelligent technique of Adaptive-Neuro-Fuzzy Inference System (ANFIS) based on Maximum Power Point Tracking (ANFIS-MPPT) algorithm with PI controller in order to increase the performances of the photovoltaic panel system below change atmospheric circumstances.

5.6. Discussion and validation

The application of ANFIS-based estimators for Maximum Power Point Tracking (MPPT) is a highly promising field of research, and the work presented in this study establishes a strong foundation for future advancements in this area.

In this chapter, three distinct models were developed and simulated using MATLAB:

⇒ PV Panel Modeling: First, the Kyocera KC200GT photovoltaic panel was modeled using the Single-Diode Equivalent Circuit Model (as detailed in Section 4, Chapter 1). The simulated I-V and P-V characteristic curves perfectly aligned with the manufacturer's

datasheet, confirming the model's accuracy. This validated model was crucial for generating a reliable database to train and test the intelligent MPPT algorithms.

⇒ MLP-Based MPPT: Second, a Multilayer Perceptron (MLP) MPPT model was implemented utilizing the widely adopted back-propagation training algorithm. The predictive results were highly acceptable, providing a robust baseline for comparison against the fuzzy logic approach.

⇒ ANFIS-Based MPPT: Finally, the fuzzy model utilizing the Adaptive Neuro-Fuzzy Inference System (ANFIS) was presented, trained, and tested. For the learning phase, a recursive least-squares (RLS) estimator was utilized to update the linear parameters, while the membership function (MF) parameters were adjusted manually until an acceptable Mean Square Error (MSE = 0.00002) was achieved. While the current results are highly satisfactory, implementing the gradient descent method to dynamically update the nonlinear MF parameters would further optimize the system, a step deferred in this study due to time constraints.

Ultimately, the proposed ANFIS model is fully operational and demonstrates high efficacy. It can be further optimized and evaluated by implementing the recommendations outlined in Section 4 of this chapter, and it holds significant potential for broader applications in renewable energy systems.

This work represents a significant milestone in the development of ANFIS-based Maximum Power Point Tracking (MPPT) estimators. In this chapter, we present and analyze three models simulated using MATLAB.

5.7. Conclusion

This chapter successfully detailed the MATLAB implementation of a Kyocera KC200GT PV panel alongside MLP and ANFIS-based MPPT models. The single-diode mathematical model effectively generated a robust database of Maximum Power Point data for training the intelligent networks. Training the Multilayer Perceptron via back-propagation provided a solid comparative baseline with highly acceptable predictive results. The proposed ANFIS model outperformed classical tracking algorithms by delivering superior response speeds, real-time adaptability, and system stability. By utilizing a recursive least-squares estimator, the fuzzy model achieved a highly accurate Mean Square Error of 0.00002.

General conclusion

2026

General conclusion

In this thesis, we successfully designed and evaluated a photovoltaic (PV) system using one of the most powerful fuzzy logic models: the Adaptive Neuro-Fuzzy Inference System (ANFIS) for Maximum Power Point Tracking (MPPT). The proposed model can be generalized to any PV module, as it learns effectively from a robust database of MPP data generated by a mathematical single-diode model.

The development of an ANFIS-based controller is described in detail, with the aim of enabling effective tracking of the maximum power point (MPP) of a photovoltaic (PV) module under different load, temperature and irradiance conditions. The efficacy of this controller in enabling the PV module to reach the maximum power output in real time by taking into account solar irradiance and temperature variations is demonstrated.

Prior to finalizing the ANFIS model, we implemented and tested an alternative MPPT method using an Artificial Neural Network (ANN), specifically a Multilayer Perceptron (MLP) model. This approach also yielded highly satisfactory results in determining the maximum power point.

The findings of these analyses demonstrate that the proposed model is more stable and performs better than any conventional methods, especially under dynamic conditions.

Additionally, we conducted a comprehensive review of ANNs and their role in machine learning. We also studied fuzzy logic systems, highlighting their pioneering contribution to artificial intelligence by granting it greater accuracy, faster response times, and the ability to operate effectively under dynamic environmental conditions.

Because of its powerful control capabilities, artificial intelligence particularly ANFIS and hybrid ANFIS methods is now a focal point in renewable energy research. Current literature frequently identifies these intelligent techniques as the most effective solution for regulating PV panel operations and accurately estimating the MPP.

Our country is among the richest in the world in terms of solar energy potential, making it the ideal sustainable alternative to polluting, finite traditional fuels. Therefore, we strongly proposed for expanding research into solar energy and its control, particularly through the use of ANFIS and ANFIS-hybrid models to optimize energy to improve the cost-effectiveness of solar energy compared to other energies, which will undoubtedly be of general benefit to individuals and the nation.

At the end, we hope that we have achieved the goal of this thesis, which we strived to be comprehensive and purposeful so that any student or researcher can find, at least, the basic principles on which he can rely to enter the world of artificial intelligence and fuzzy logic.

References

References

[1] M.K.G.Deshmukh, M. Sameeroddin, D.Abdul, and M.A.Sattar."Renewable energy in the 21st century: A review".Journal: Materials Today: Proceedings.2023. DOI**:** 10.1016/j.matpr.2021.05.501

[2] E.F. Camacho, M.Berenguel. "Control of Solar Energy Systems" Springer 2012 IFAC Proceedings Volumes https://doi.org/10.3182/20120710-4-SG-2026.00181.

[3] T. Sutikno, A.C.Subrata , G.Pau , A.Jusoh ,K.Ishaque." Maximum power point tracking techniques for low-cost solar photovoltaic applications" Archives of Electrical Engineering.2023 .DOI 10.24425/aee.2023.145410.

[4] Türkay, Y. and Yüksek, A. G. "Investigating the Potential of an ANFIS-Based MPPT Controller for Solar Photovoltaic Systems". IEEE Access, 2025.
DOI 10.1109/ACCESS.2025.3547954.

[5] J.S.R. Jang "ANFIS : Adaptive Network Based Fuzzy Inference System" IEEE Transactions on Systems, Man, and Cybernetics. 1993. DOI: 10.1109/21.256566

[6] "PVresources," pvresources. Accessed: May 26, 2026. [Online]. Available : https://www.pvresources.com/en/introduction/history.php

[7] J. Nelson, "The Physics of solar cells", Reprint 2007. Londres: Imperial College, 2007. DOI 10.1142/p276

[8]. M.A.Green, "Solar cells: operating principles, technology and system applications", Repr. [der Ausg.] Englewood Cliffs, NJ 1982. Kensington, NSW: Univ. of New South Wales, 1998.ISBN: 9780138222693

[9].A. Abdelaziz, A. El Moundher, A. Chaima, and D. Hanane, "Identification of Photovoltaic Panel MPPT Using Neuro-Fuzzy Model," Eur. J. Electr. Eng., vol. 24, no. 5–6, pp. 273–279, Dec. 2022, DOI: 10.18280/ejee.245-606.

[10]. S.M. Sze and Kwok K. Ng, "Physics of Semiconductor Devices", Third Edition. New York: John Wiley & Sons, 2006 DOI 10.1002/0470068329

[11] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays," IEEE Trans. Power Electron., vol. 24, no. 5, pp. 1198–1208, May 2009, DOI: 10.1109/TPEL.2009.2013862.

[12] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, "Optimization of Perturb and Observe Maximum Power Point Tracking Method," IEEE Trans. Power Electron., vol. 20, no. 4, pp. 963–973, Jul. 2005, DOI: 10.1109/TPEL.2005.850975.

[13]. M. A. Eltawil and Z. Zhao, “MPPT techniques for photovoltaic applications,” Renew. Sustain. Energy Rev., vol. 25, pp. 793–813, Sep. 2013, doi: 10.1016/j.rser.2013.05.022.

[14] Hsieh GC, et al. “Variable frequency controlled incremental conductance derived MPPT photovoltaic stand-along dc bus system”. 2008 23 Annual IEEE Applied Power Electronics Conference and Exposition.

[15]. A. R. Jordehi, “Maximum power point tracking in photovoltaic (PV) systems: A review of different approaches,” Renew. Sustain. Energy Rev., vol. 65, pp. 1127–1138, Nov. 2016, DOI: 10.1016/j.rser.2016.07.053.

[16] R. S. Bhatnagar, and al, "Hybrid, Optimal, Intelligent and Classical PV MPPT Techniques: A Review" CSEE Journal of Power and Energy Systems VOL. 7, NO. 1, JANUARY 2021 DOI: 10.17775/CSEEJPES.2019.02720

[17] Teulings WJA, Marpinard JC, Capel A, O’Sullivan D. “A new maximum power point tracking system”. In: Proceedings 24th annual IEEE power electronics specialists’ conference. 1993. p. 833–8.

[18] Xiao W, Dunford WG.’A modified adaptive hill climbing MPPT method for photovoltaic power systems’. In: Proceedings of 35th annual IEEE power electronics specialist’s conference; 2004. p. 1957–63.

[19] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of ICNN’95 - International Conference on Neural Networks, Perth, WA, Australia: IEEE, 1995, pp. 1942–1948.DOI: 10.1109/ICNN.1995.488968.

[20]. Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in 1998 IEEE International Conference on Evolutionary Computation Proceedings
DOI: 10.1109/ICEC.1998.699146.

[21] S.Walczak and N.Cerpa: “Artificial Neural Networks”, Encyclopedia of Physical Science and Technology, 2003 https://doi.org/10.1016/B0-12-227410-5/00837-1.

[22] Luis B. Almeida's "C1.2 Multilayer perceptrons," published in the 1997 Handbook of Neural Computation (pages C1.2:1–C1.2:30) by IOP Publishing and Oxford University Press.

[23]. S. Bhattacharya.”A Primer on Machine Learning in Sub surface Geosciences “.
Springer. Mai 2021. ISBN-13978-3030717681

[24] S.Nissen. "Implementation of a Fast Artificial Neural Network Library (fann)" Rapport technique de 2003 .publié par l'Université de Copenhague.

[25] Frank Rosenblatt: “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms” Springer, Berlin, Heidelberg 1986 https://doi.org/10.1007/978-3-642-70911-1_20.

[26] F. O-Gyimah, K.O.Boateng, P.B.Adu K.Q.Aphetsi "Prediction of Telecommunication Network Outage Time Using Multilayer Perceptron Modelling Approach" 2021 International Conference on Computing, Computational Modelling and Applications (ICCMA) DOI: 10.1109/ICCMA53594.2021.00025

[27] R. Ali, I.Naim2 "Neural Network Based Supervised Rank Aggregation" 2011, International Conference on Multimedia, Signal Processing and Communication Technologies DOI: 10.1109/MSPCT.2011.6150439

[28] T.Islam," Hands-on Deep Learning: Building Models from Scratch" Springer Nature Switzerland 2025 https://doi.org/10.1007/978-3-032-00488-8

[29] Dr ali Ahmed," Lecture 3 Back Propagation Learning Model" YOUTUBE site
Video: https://www.youtube.com/watch?v=H7EnDcZlD4g&t
Pdf : https://drive.google.com/file/d/1v8If...

[30] S.Haykin."Neural Networks and Learning Machines". Prentice Hall. 3ed 2009.
ISBN-13: 978-0-13-147139-9

[31] S.K.Nayar "Backpropagation Algorithm, Neural Networks" YOUTUBE site https://www.youtube.com/watch?v=sIX_9n-1UbM

[32] V.S.Nguyen; Q.D.Nguyen,T.S.Le, H.V.Dang "Fixed-time adaptive RBF neural network controller via minimum learning parameter for ship roll stabilization", Published by Elsevier Ltd.2025 DOI: https://doi.org/10.1016/j.apor.2024.104403.

[33] T.M. COVER "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition" IEEE Transactions on Electronic Computers.1965 DOI 10.1109/PGEC.1965.264137

[34] M.Huddar, "Radial Basis Function RBF | Gaussian RBF Multiquadratic RBF Artificial Neural Network" Youtube site https://www.youtube.com/watch?v=rF4m6g46p-g&t..

[35] W.Jitchaijaroen. S.Keawsawasvong.W.Wipulanusat, D.R.Kumar, P.Jamsawang and J.Sunkpho" Machine learning approaches for stability prediction of rectangular tunnels in natural clays based on MLP and RBF neural networks". Published by Elsevier Ltd DOI.202410.1016/j.rineng.2024.101831

[36] A.Ahmed "Radial basis Neural Networks (RBNN)"
pdf: https://drive.google.com/file/d/1FRIO...

[37] Lotfi A. Zadeh."Fuzzy logic, neural networks and soft computing". One-page course announcement of CS 294-4, Spring 1993, the University of California at Berkeley.

[38] Lotfi A. Zadeh. "Fuzzy Sets" published by Information and Control journal 1965 DOI: 10.1016/S0019-9958(65)90241-X

[39] Z.Wang, Q.Wang, M.Bai, Z.Chen, and Z.Sun," Further exploration on relationship between crisp sets and fuzzy sets" IEEE. 2010 2nd International Conference on Computer and Automation Engineering (ICCAE) DOI: 10.1109/ICCAE.2010.5451688

[40] M.Kabir; M.M.J.Kabir "Fuzzy membership function design: An adaptive neuro-fuzzy inference system (ANFIS) based approach" IEEE 2021 (ICCCI) DOI: 10.1109/ICCCI50826.2021.9402633

[41] E.Portmann, G.Wilke, L.Terán, and S. D'Onofrio "Fuzzy Sets and Systems I: An Introduction with Cases from Business Informatics, Computer Science and Engineering" . Springer 2026. ISBN (eBook): 978-3-031-98472-3.

[42] Hashim A. Hashim Youtube site : "Fuzzy Logic Control 03 Membership Function Linguistic Variables" https://www.youtube.com/watch?v=fGoeNeBhRoE&t,

[43] R. Manimala; B. Vigneshwaran "Comparison of Mamdani and Sugeno Fuzzy Inference Systems to Assess the Health Condition of Power Transformer" IEEE 2024 DOI: 10.1109/ICOECA62351.2024.00096

[44] C.Wang, Y.Li, J.Du, G.Corzo." Mamdani Fuzzy Inference System for Rating the Performance of Sponge City Programme" IEEE 2022 in 2022 8th (ICSAI) DOI: 10.1109/ICSAI57119.2022.10005410

[45] N.Istiqomah; M.Yuliana; T.B.Santoso " Implementation of Fuzzy Tsukamoto Algorithm on Smart Node Sensors for Air Quality Monitoring". IEEE.2021 in International Electronics Symposium (IES) DOI 10.1109/IES53407.2021.9594014.

[46] J.Jassbi, S. H. Alavi, P.J.A.Serra; RA.Ribeiro " Transformation of a Mamdani FIS to First Order Sugeno FIS". 2007 IEEE International Fuzzy Systems Conference by IEEE DOI: 10.1109/FUZZY.2007.4295331

[47] R. Manimala. Dr. B. Vigneshwaran. "Comparison of Mamdani and Sugeno fuzzy inference systems to assess the health condition of power transformer". Published by IEEE 2024 DOI 10.1109/ICOECA62351.2024.00096

[48] V. S. Jigajinni and V. Upendranath, "Comparison of ANFIS and ANN techniques in the simulation of a typical aircraft fuel system health management,"Int.J. Artif. Intell. Appl., vol. 9, no. 4, pp. 43–55, 2018.

[49] J. Sobhani,M. Najimi, A. R. Pourkhorshidi, and T. Parhizkar, "Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models," Construction Building Mater., vol. 24, no. 5, pp. 709–718, 2010.

[50] D.Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey," Artif. Intell. Rev.,vol. 52, no. 4, pp. 2263–2293, 2018.

[51] K. Amara, A. Fekik, D. Hocine, M. L. Bakir, E. B. Bourennane, and A. Malek, "Improved Performance of a PV Solar Panel with Adaptive Neuro Fuzzy Inference System (ANFIS) Based MPPT". IEEE. 2018 7th ICRERA.
DOI: 10.1109/ICRERA.2018.8566818

[52] Dr. R.Ghongade « ANFIS PART2©Dr.R.B.Ghongade ». Youtube site https://www.youtube.com/watch?v=v4R1jzx0bEo&t

[53] Jang J.S., Sun C.T., Mizutani E. Mizutani "Neuro-Fuzzy and Soft Computing A computational approach to learning and machine intelligence". Prentice-Hall, Inc,Upper Saddle River, NewJersey, 1997.

[54] M.I.S. Guerra, F.M.U. deAraújo, J. T. de CarvalhoNeto, and R. G.Vieira, "Survey on adaptative neural fuzzy inference system (ANFIS) architecture applied to photovoltaic systems," Energy Syst., pp. 17–25, 2022.

[55] O.S.Vargas, S.E. De León Aldaco, J.A Alquicira , L.G. Vela-Valdés ,and A.R.L. Núñez , "Adaptive Network-Based Fuzzy Inference System (ANFIS) Applied to Inverters: A Survey" IEEE Transactions on power electronics vol. 39, NO. 1, JANUARY 2024
DOI :10.1109/TPEL.2023.3327014

[56] A. Gupta,Y. K. Chauhan, andR.K. Pachauri, "A comparative investigation of maximum power point tracking methods for solar PV system," Sol.Energy, vol. 136, pp. 236–253, 2016.

[57] A. A. Aldair, A. A. Obed, and A.F. Halihal, "Design and implementation of ANFIS-reference model controller based MPPT using FPGA for photovoltaic system," Renewable Sustain. Energy Rev., vol. 82, pp. 2202–2217,2018.

[58] F. Belhachat and C. Larbes, "Global maximum power point tracking based on ANFIS approach for PV array configurations under partial shading conditions," Renewable Sustain. Energy Rev., vol. 77, pp. 875–889, 2017.

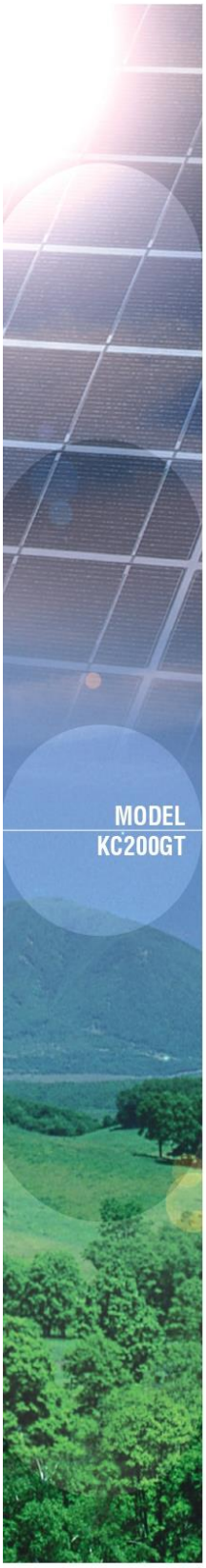
[59] R. K. Kharb, S. L. Shimi, S. Chatterji, andM. F.Ansari, "Modeling of solar PV module and maximum power point tracking using ANFIS," Renewable Sustain. Energy Rev., vol. 33, pp. 602–612, 2014.

[60] D.Radianto, D.A. Asfani, T.Hiyama, and Syafaruddin., "Partial shading detection and MPPT controller for total cross tied photovoltaic using ANFIS," ACEEE Int. J. Elect. Power Eng.,vol. 3, no. 2, pp. 1–5, 2012.

[61] A. M. Eltamaly, H. M. H. Farh, and M. F. Othman, “A novel evaluation index for the photovoltaic maximum power point tracker techniques,” Sol.Energy, vol. 174, pp. 940–956, 2018.

[62] A.Al-Hmouz, J.Shen, R.Al-Hmouz, & J.Yan."Modeling and simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for mobile learning" IEEE Transactions on Learning Technologies published (2012).DOI 10.1109/TLT.2011.36.

ANNEXES



THE NEW VALUE FRONTIER

KYOCERA

KC200GT

HIGH EFFICIENCY MULTICRYSTAL
PHOTOVOLTAIC MODULE



LISTED

HIGHLIGHTS OF KYOCERA PHOTOVOLTAIC MODULES

Kyocera's advanced cell processing technology and automated production facilities produce a highly efficient multicrystal photovoltaic module.
The conversion efficiency of the Kyocera solar cell is over 16%.
These cells are encapsulated between a tempered glass cover and a pottant with back sheet to provide efficient protection from the severest environmental conditions.
The entire laminate is installed in an anodized aluminum frame to provide structural strength and ease of installation.
Equipped with plug-in connectors.

APPLICATIONS

KC200GT is ideal for grid tie system applications.

- Residential roof top systems
- Large commercial grid tie systems
- Water Pumping systems
- High Voltage stand alone systems
- etc.

QUALIFICATIONS

- **MODULE** : UL1703 certified
- **FACTORY** : ISO9001 and ISO 14001

QUALITY ASSURANCE

Kyocera multicrystal photovoltaic modules have passed the following tests.

- Thermal cycling test
- Thermal shock test
- Thermal / Freezing and high humidity cycling test
- Electrical isolation test
- Hail impact test
- Mechanical, wind and twist loading test
- Salt mist test
- Light and water-exposure test
- Field exposure test

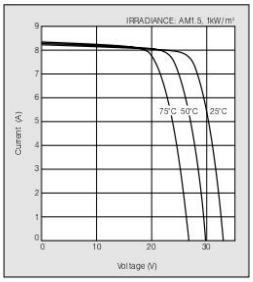
LIMITED WARRANTY

※1 year limited warranty on material and workmanship
※20 years limited warranty on power output: For detail, please refer to "category IV" in Warranty issued by Kyocera

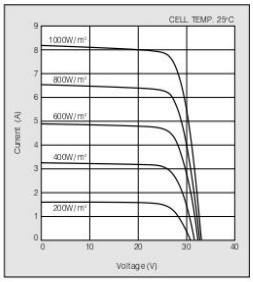
(Long term output warranty shall warrant if PV Module(s) exhibits power output of less than 90% of the original minimum rated power specified at the time of sale within 10 years and less than 80% within 20 years after the date of sale to the Customer. The power output values shall be those measured under Kyocera's standard measurement conditions. Regarding the warranty conditions in detail, please refer to Warranty issued by Kyocera)

ELECTRICAL CHARACTERISTICS

Current-Voltage characteristics of Photovoltaic Module KC200GT at various cell temperatures



Current-Voltage characteristics of Photovoltaic Module KC200GT at various irradiance levels

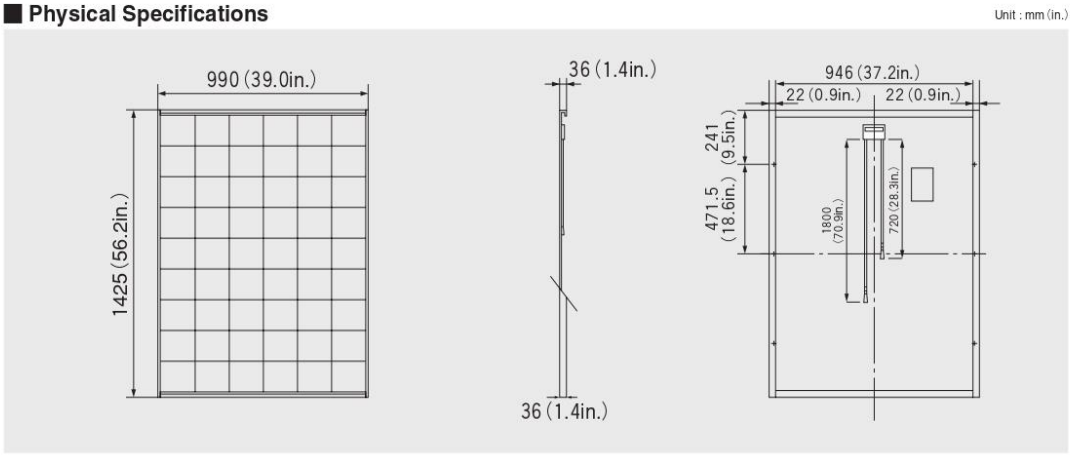


SPECIFICATIONS

KC200GT

■ Physical Specifications

Unit : mm (in.)



■ Specifications

■ Electrical Performance under Standard Test Conditions (*STC)	
Maximum Power (Pmax)	200W (+10%/−5%)
Maximum Power Voltage (Vmpp)	26.3V
Maximum Power Current (Impp)	7.61A
Open Circuit Voltage (Voc)	32.9V
Short Circuit Current (Isc)	8.21A
Max System Voltage	600V
Temperature Coefficient of Voc	-1.23×10^{-1} V/℃
Temperature Coefficient of Isc	3.18×10^{-3} A/℃

*STC : Irradiance 1000W/m², AM1.5 spectrum, module temperture 25℃

■ Electrical Performance at 800W/m², NOCT, AM1.5	
Maximum Power (Pmax)	142W
Maximum Power Voltage (Vmpp)	23.2V
Maximum Power Current (Impp)	6.13A
Open Circuit Voltage (Voc)	29.9V
Short Circuit Current (Isc)	6.62A

NOCT (Nominal Operating Cell Temperature) : 47℃

■ Cells	
Number per Module	54

■ Module Characteristics	
Length × Width × Depth	1425mm(56.2in)×990mm(39.0in)×36mm(1.4in)
Weight	18.5kg(40.7lbs.)
Cable	(+)720mm(28.3in),(-)1800mm(70.9in)

■ Junction Box Characteristics	
Length × Width × Depth	113.6mm(4.5in)×76mm(3.0in)×9mm(0.4in)
IP Code	IP65

■ Reduction of Efficiency under Low Irradiance	
Reduction	7.8%

Reduction of efficiency from an irrandiance of 1000W/m² to 200W/m² (module temperature 25℃)

Please contact our office for further information

KYOCERA

KYOCERA Corporation

■ KYOCERA Corporation Headquarters
CORPORATE SOLAR ENERGY DIVISION
6 Takeda Tobadono-cho
Fushimi-ku, Kyoto
612-8501, Japan
TEL:(81)75-604-3476 FAX:(81)75-604-3475
http://www.kyocera.com

● KYOCERA Solar, Inc.
7812 East Acoma Drive
Scottsdale, AZ 85260, USA
TEL:(1)480-948-8003 or (800)223-9580 FAX:(1)480-483-6431
http://www.kyocerasolar.com

● KYOCERA Solar do Brasil Ltda.
Av. Guignard 661, Loja A
22790-200, Recreio dos Bandeirantes, Rio de Janeiro, Brazil
TEL:(55)21-2437-8525 FAX:(55)21-2437-2338
http://www.kyocerasolar.com.br

● KYOCERA Solar Pty Ltd.
Level 3, 6-10 Talavera Road, North Ryde
N.S.W. 2113, Australia
TEL:(61)2-9870-3948 FAX:(61)2-9888-9588
http://www.kyocerasolar.com.au/

● KYOCERA Fineceramics GmbH
Fritz Muller strasse 107, D-73730 Esslingen, Germany
TEL:(49)711-93934-917 FAX:(49)711-93934-950
http://www.kyocerasolar.de/

● KYOCERA Asia Pacific Pte. Ltd.
298 Tiong Bahru Road, #13-03/05
Central Plaza, Singapore 168730
TEL:(65)6271-0500 FAX:(65)6271-0600

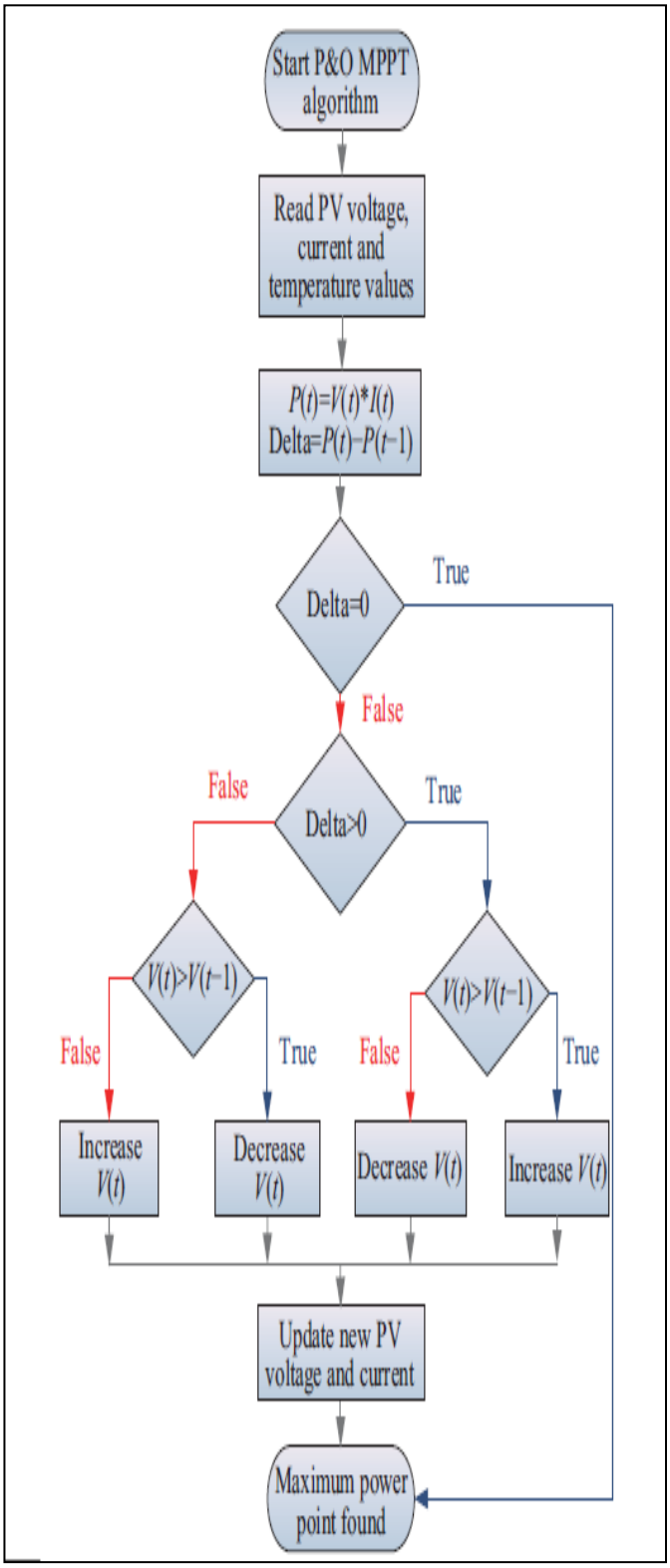
● KYOCERA Asia Pacific Ltd.
Room 801-802, Tower 1 South Seas Centre, 75 Mody Road,
Tsimshatsui East, Kowloon, Hong Kong
TEL:(852)2-7237183 FAX:(852)2-7244501

● KYOCERA Asia Pacific Ltd. Taipei Office
10 Fl., No.66, Nanking West Road, Taipei, Taiwan
TEL:(886)2-2555-3609 FAX:(886)2-2559-4131

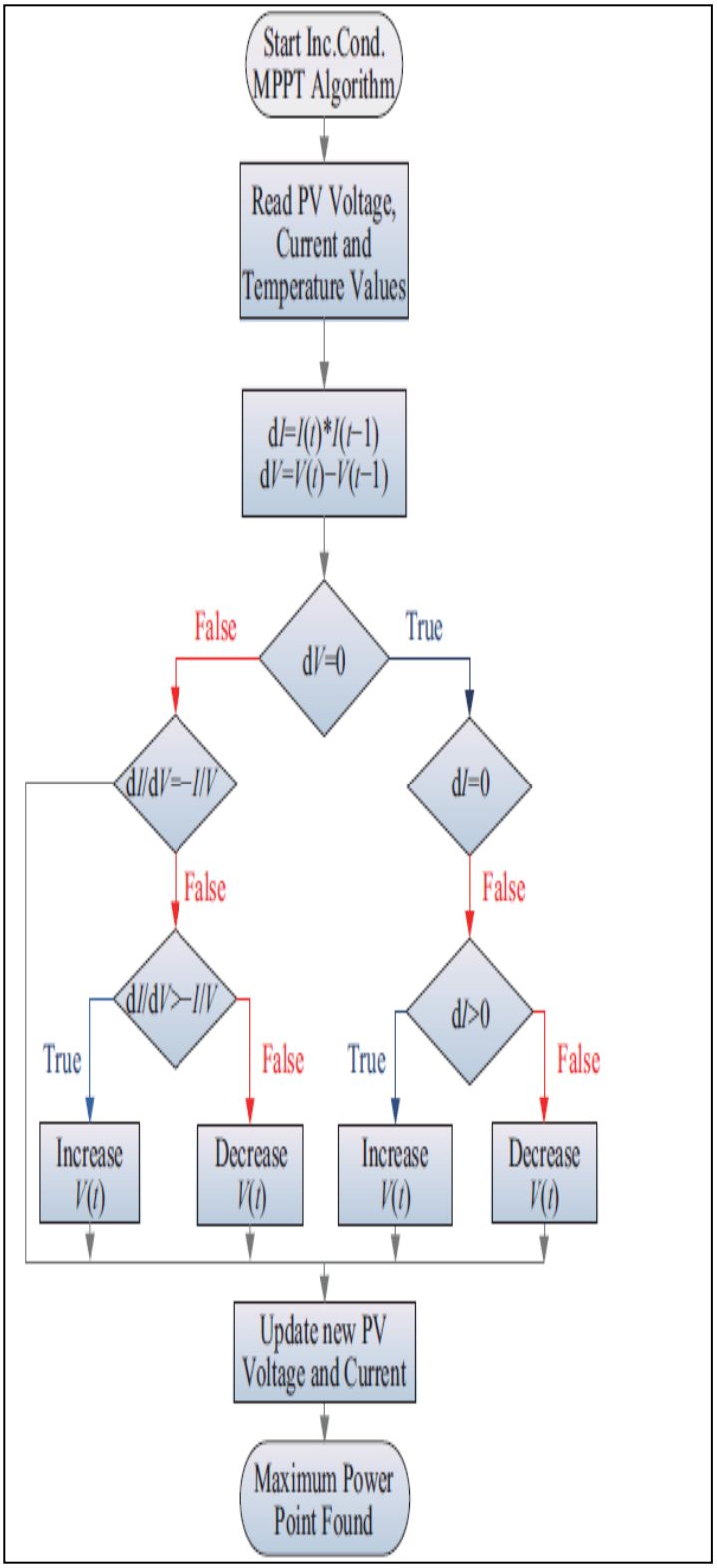
● KYOCERA(Tianjin) Sales & Trading Corporation
19F, Tower C HeQiao Building 8A GuangHua Rd.,
Chao Yang District, Beijing 100026, China
TEL:(86)10-6583-2270 FAX:(86)10-6583-2250

Kyocera reserves the right to modify these specifications without notice

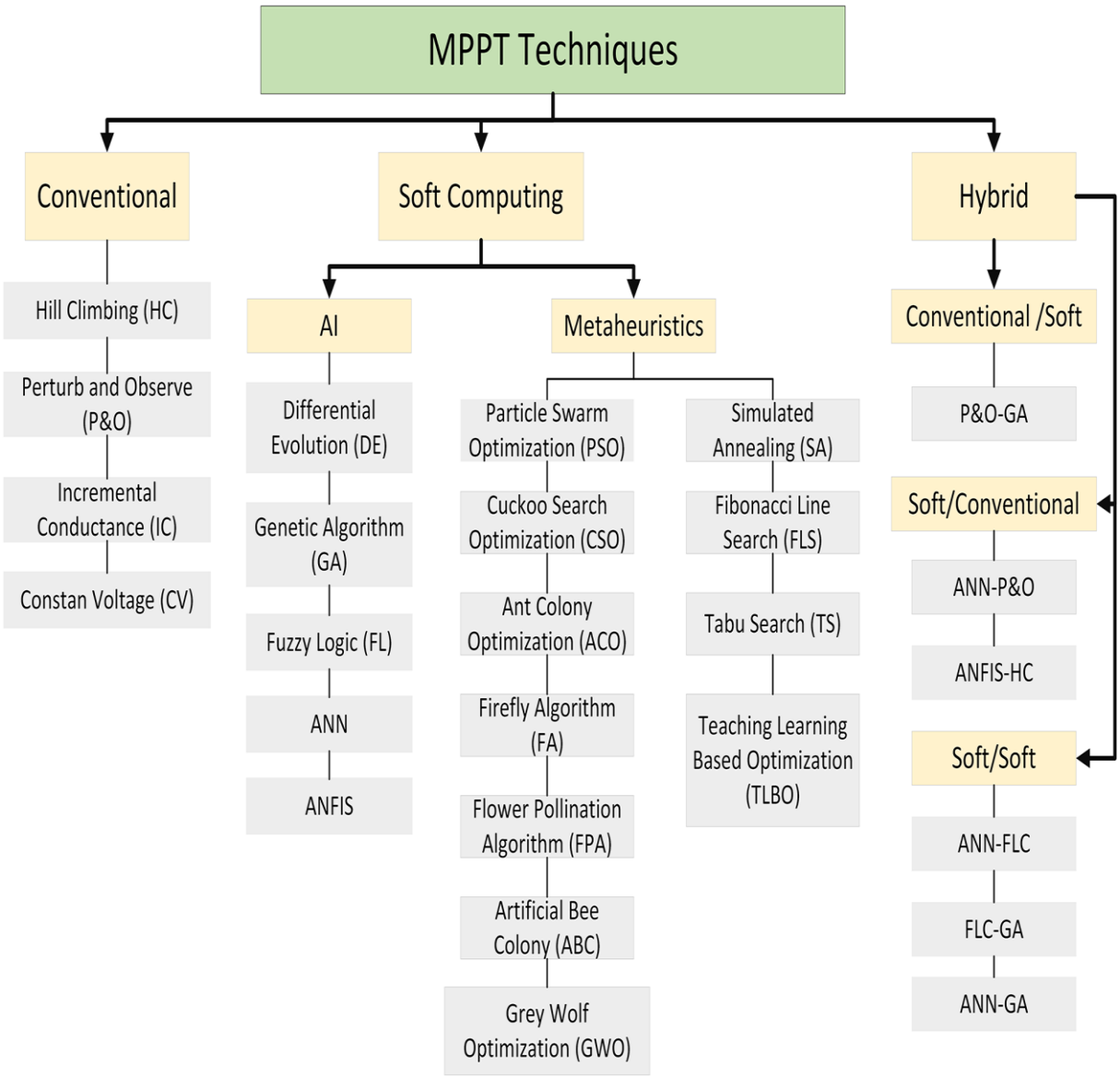
LIE/I09M0703-SAGKM



Flowchart of P&O algorithm [11]



Flowchart of Inc.Con algorithm [11]



Classification of MPPT techniques