Islamic University of Gaza
Faculty of Engineering
Electrical Engineering Department

تصميم متحكم ضبابي من نوع سيجينو بتقنية الخوارزميات الجينية للتتبع أقصى قيمة للقدرة وتتبع الشمس في مصفوفات الخلايا الشمسية

Design of GA- Sugeno Fuzzy Controller for Maximum Power Point And Sun Tracking in Solar Array Systems

By
Khalil T. ElNounou

Supervisor
Dr. Basil Hamed

This thesis is submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

2013 م – 1434هـ

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

تصميم متحكم ضبابي من نوع سيجينو بتقنية الخوارزميات الجينية للتتبع أقصى قيمة للقدرة وتتبع الشمس في مصفوفات الخلايا الشمسية.

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name:	اسم الطالب:
Signature:	التوقيع:
Date:	التاريخ:

صفحة نتيجة الحكم على البحث (نتيجة الحكم من قبل لجنة المناقشة)

DEDICATION

To the memory of my beloved parents, who taught me when I was child they gave me their time love, and attention and to all my family members" big brother, sisters, wife, and lovely kids" who have been a constant source of motivation, inspiration, and support.

Acknowledgement

I thank Allah, the lord of the worlds, for His mercy and limitless help and guidance. May peace and blessings be upon Mohammed the last of the messengers.

I would like to express my deep appreciation to my advisor Dr. Basil Hamed for providing advice, support and excellent guidance.

Special thanks go Dr. Hatem A. Elaydi and Dr. Iyad Abuhadrous –thesis examiners- for their patience, guidance, and generous support during this research.

My thanks go to Islamic University for their help and support, not only during this research but also during the entire period of my Master study.

Special thanks go to the faculty of engineering staff at all levels and positions for their help and assistance that they provided me during my study.

Many thanks go to my best friends for their help specially Eng. Mohammed EL-Moghany and Eng. Hosam Abu Elreesh.

My deep thanks go to my wife for her support and encouragement. I also would like to thank my family and my children for their continuous smiles which gave me a great support.

Finally, thanks for everyone who has raised his hands and prayed ALLAH for my success.

Abstract

Renewable energy sources play an important role in electric power generation. There are various renewable sources, which are used for electric power generation, such as solar energy, wind energy, geothermal etc. Solar energy is a good choice for electric power generation, which is viewed as clean and renewable source of energy for the future since the solar energy is directly converted into electrical energy by solar photovoltaic modules which is suitable for obtaining higher power output.

There are two major approaches: sun tracking and maximum power point tracking to maximizing the power output of the solar arrays. These two methods need efficient intelligent controller such as Fuzzy Logic Controller (FLC). The advantage of the fuzzy logic control is that it does not strictly need any mathematical model of the plant. It is based on operator experience, and it is very easy to apply. Hence, many complex systems can be controlled without knowing the exact mathematical model of the plant. In addition, fuzzy logic simplifies dealing with nonlinearities in systems. The advantage of fuzzy logic control is that the linguistic system definition becomes the control algorithm.

In this thesis: Two Sugeno fuzzy logic controllers are designed to increase the energy generation efficiency of solar cells. These controllers are, sun-tracking controller and maximum power point tracking controller with Genetic Algorithm (GA) to optimize the membership, output gain and inputs gain of the fuzzy controllers.

Solar tracking system uses a stepper motor as the drive source to rotate the solar panel. The position of the sun is determined by using a tracking sensor, the sensor reading is converted from analog to digital signal, and then it passed to a fuzzy logic controller. The controller output is connected to the driver of the stepper motor to rotate PV panel in one axis until it faces the sun. The proposed sun-tracking controller is tested using Matlab/Simulink program, the results show that the Sugeno controller has a good response when compared with Mamdani controller.

Maximum power point tracking system uses dc-to-dc converter to compensate the output voltage of the solar panel to keep the voltage at the value, which maximizes the output power. MPP fuzzy logic controller measures the values of the voltage, and current at the output of the solar panel. Then, calculates the power from the relation (P=V*I) to extract the inputs of the controller, and the crisp output of the controller represents the duty cycle of the pulse width modulation for controlling on/off time of MOSFET switch of a buck converter. The proposed maximum power point tracking controller for grid-connected photovoltaic system is tested using model designed by Matlab/Simulink program and the results show that the Sugeno controller with GA has a response better than Mamdani controller applied to the same system.

ملخص

" تصميم متحكم ضبابي من نوع سيجينو بتقنية الخوارزميات الجينية للتتبع أقصى قيمة للقدرة و تتبع الشمس في مصفوفات الخلايا الشمسية "

تلعب مصادر الطاقة المتجددة دورًا مهماً في توليد الطاقة الكهربائية، مثل: الطاقة الشمسية، طاقة الرياح، الطاقة الحرارية الأرضية وغيرها من المصادر. وتعتبر الطاقة الشمسية هي الخيار الأفضل لتوليد الكهرباء لأنها مصدر طاقة نظيف ومتجدد لا ينضب، وهي تعمل علي تحويل ضوء الشمس إلي طاقة كهربائية مباشرة بإستخدام ألواح الخلايا الشمسية للحصول على أفضل طاقة. وهناك طريقتين رئيسيتان لزيادة القدرة المتحصل عليها من الأنظمة الشمسية وهي: تتبع مكان الشمس وتتبع النقطة التي تكون عندها القدرة أقصى ما يمكن، ونستخدم فيها الخوارزميات الجينية للحصول على أفضل طاقة. وكلا الطريقتان يحتاج إلى متحكم من نوع ذكي مثل المتحكمات الضبابية، حيث أن للمتحكم الضبابي مميزات كثيرة مثل سهولة نسبية في التصميم فهو لا يحتاج معرفة دقيقة عن المعادلات الرياضية التي تصف النظام المراد التحكم به وهو يعتمد علي خبرة المصمم كما يعمل بشكل ممتاز مع الأنظمة اللاخطية.

في هذه الرسالة استخدمنا نوعان من المتحكم الضبابي من نوع (Sugeno) لزيادة كفاءة الطاقة المتولدة من الخلايا الشمسية، فالأول لجعل لوح الخلايا الشمسية يتتبع الشمس والأخر يتتبع النقطة التي تكون عندها القدرة أقصى ما يمكن.

تم تصميم وتطبيق متتبع الشمس والذي يتكون من موتور الخطوة لدوران الخلايا الشمسية، و مجسات تحدد مكان الشمس وتُحول قراءتها من إشارة مستمرة إلي إشارة متقطعة لسهولة التعامل معها وتعتبر كمدخل للمتحكم، وأما مخرجة فهو يعمل علي إبقاء لوح الخلايا الشمسية مواجه للشمس اغلب أوقات النهار. وتم إختبار المتحكم بإستخدام برنامج السيمولينك التابع لبرنامج الماتلاب وكانت النتيجة أن استجابة المتحكم علي الوجه المطلوب.

وتم تصميم نظام تتبع أقصى نقطة للقدرة، والمُكون من لوح خلايا شمسية، ومحول من نوع (Buck) ، و متحكم ضبابي من نوع (Sugeno) وذلك للتحكم في فتح وإغلاق مفتاح الـ(Mosfet) الموجود داخل المحول، و تم تحسين الناتج بإضافة الخوارزميات الجينية مع ذلك النظام وتم أختبار المتحكم بإستخدام برنامج السيمولينك عن طريق تصميم مودل ليحاكي مواصفات الخلايا الشمسية وتُأخذ موصفات وبينات الخلايا الشمسية من المنتج والمصنع للخلايا الشمسية لإختبار سلوك هذه المصفوفة وتأثير المتحكم بها، وكانت النتيجة بإستخدام متحكم ضبابي من نوع (Sugeno) أفضل من النتيجة المتحصل عليها بإستخدام متحكم ضبابي من نوع (Mamdani) وعند إضافة الخوارزميات الجينية مع المتحكم كانت النتيجة أفضل من تصميم المتحكم بدونها وذلك عند تطبيقها علي نفس النظام.

Table of Contents

List of Figures

List of Tables

Nomenclature

Analog to Digital Converter	ADC
Alternating current	AC
Adaptive Fuzzy Logic Control	AFLC
Center of Area	COA
Center of Maximum	COM
Center of gravity	COG
Data Base	DB
Digital to Analog Converter	DAC
Direct Current	DC
Fuzzy Associative Memory	FAM
Fuzzy Logic	FL
Fuzzy Logic Controller	FLC
Fuzzy Logic System	FLS
Fuzzy Rule-Based System	FRBS
Genetic Algorithm	GA
Genetic Fuzzy Rule-Based System	GFRBS
Genetic Fuzzy Systems	GFS
Knowledge Base	KB
Membership Function	MF
Multi-Input-Multi-Output	MIMO
Mean Of Maximum	MOM
Maximum Power Point Tracking	MPPT
Photo Volatic	PV
Perturbation and Observation	P&O
Pulse Width Modulation	PWM
Quality Method	QM
Rule Base	RB
Takagi- Sugeno-Kang	TSK
Single-Input-Single-Output	SISO
Weighted Average method	WAM

CHAPTER 1 INTRODUCTION

1.1. Introduction

Fossil fuels (ex. gas, oil, coal) are limited and produce strong pollutants. In the last 15 years, the price of petroleum has tripled and the previsions on the medium term there are not quite encouraging. The increase of emissions of carbon dioxide, responsible for the global warming and for the greenhouse effect, may have devastating results over the time on the environment. The solution of this problem is the renewable energy, which play an important role in electric power generation. There are various renewable sources, which used for electric power generation, such as solar energy, wind energy, geothermal etc. Solar energy is the best kind of renewable energy, which is a good choice for electric power generation, since the solar energy is directly converted into electrical energy by solar photovoltaic modules. These modules are made up of silicon cells. When many such cells are connected in series, we get a solar PV module. The current rating of the modules increases when the area of the individual cells is increased, and vice versa. When many such PV modules are connected in series and parallel combinations, we get solar PV arrays, that suitable for obtaining higher power output[1].

The applications for solar energy are increased, which need to improve the materials and methods used to harness this power source. Main factors that affect the efficiency of the collection process are solar cell efficiency, intensity of source radiation and storage techniques.
Materials used in solar cell manufacturing limit the efficiency of a solar cell. It is particularly difficult to make considerable improvements in the performance of the cell. Therefore, the increase of the intensity of radiation received from the sun is the most attainable method of improving the performance of solar power.

There are two major approaches for maximizing power extraction in solar systems. They are sun tracking and maximum power point tracking to maximizing the power output of the solar arrays and these methods need intelligent controllers such as fuzzy logic controller. Fuzzy control dose not strictly need any mathematical model of the plant. It is based on plant operator experience, and it is very easy to apply. Fuzzy control gives robust performance for a linear or nonlinear plant with parameter variation [2]. Fuzzy controller is simply to be designed using simulation software such as Matlab software or LabVIEW.

This research propose a system that controls the movement of a solar cell so that it can constantly aligned towards the direction of the sun and controls the system for the maximum power point tracking (MPPT) to maximize the solar cell efficiency.

The Sugeno fuzzy controller will be used in the two methods and Genetic Algorithm (GA) will be applied as an optimization method that optimizes the membership, output gain and inputs gain of the fuzzy controllers.

1.2. Thesis Motivation and Objectives

1.2.1. Motivation

We have noticed a greater academic attention to the study of tracking solar systems the last few years, not only for photovoltaic but also for thermal electricity production. This greater attention seems to be mainly due to the fact that PV systems are getting closer to the level of cost/efficiency that justifies their use on commercial applications. This improvement in cost/efficiency is due to an overall drop in the price of photovoltaic (PV) panels (due to technical developments and greater scales of production) and to increase governmental support to renewable energies in general and to PV systems in particular (due to environmental concerns and to recognition of a future problem in fossil fuels availability**)** [1]. On other hand nowadays, fuzzy logic controllers have an efficient performance over the traditional controller researches especially in nonlinear and complex model systems.

1.2.2. Objectives

The goal of this research is developing a sun tracker approach to maximize the efficiency of solar array systems to provide as much as capability and flexibility as possible in an affordable system that can be implemented.
The main objective is building a FLC to maximize the power output of the solar arrays.

The specific objectives include:

1- Design Fuzzy controllers using Sugeno for (Sun and Maximum power point).
2- Simulate controllers and the system on Matlab.
3- Track the sun all day periodically.
4- Add Genetic Algorithm (GA) to the system to improve the response.

This new technique will be applied in Gaza in developed form. Because we need clean, renewable energy sources do not depend on others. Gaza is power sources do not cover all people requirements, so they have high cost and many daily interruptions. I know that the solar modules devices, which cleanly convert sunlight into electricity, will solve a big part of these problems and offer a practical solution to the problem of power generation in remote areas. On the other hand, the proposed controller can be applied to solar water heater, which found on most houses in Gaza to increase its preheating efficiency.

1.3. Literature review

1.3.1. Sun Tracking Techniques:

1) *In 2007, Louchene, et.al.* [3] designed a solar tracking system with fuzzy, the most important advantage of this method was the short processing time and the decision making manner but they increased the number of membership function.

2) *In 2008, lane* [4] presented means of controlling a sun tracking array with an embedded microprocessor system, but the sensitivity and accuracy of tracking by using a different light sensor not increased enough. Therefore, a phototransistor with

an amplification circuit would provide improved resolution and better tracking accuracy/precision.

3) *In 2011, Usta, et.al.* [5] designed a solar tracking system of Matlab/Simulink environment and a control method was proposed for the system. The results from the proposed FLC were compared with those obtained using PI controller. The results from simulations showed good and acceptable performances for the FLC, but we must use PID Fuzzy controller to get better performance than PI controller.

4) *In 2011, Ozuna, et.al.* [6] proposed and evaluate mathematical simulation of a solar tracking controlled by fuzzy logic in order to achieve the correct positioning of a photovoltaic solar cell and get as much sunlight during the day and therefore producing the most electricity, in this paper the searcher used three member ship so if we increasing the member ship we can improve the result.

5) *In 2011, Elmghany* [2] designed a FLC to maximize the energy received from solar cells by implementing a sun tracker controlled by fuzzy logic controller to keep the PV panel pointing toward the sun by using a stepper motor. The use of stepper motor enabled accurate tracking of the sun, LDR resistors used to determine the solar light intensity. Elmghany's solar tracking power system was able to track the sun light automatically, so it was an efficient system for solar energy collection, but he did not use any optimization tool to get the optimal solution; so I used GA to get the optimal solution of sun tracking.

1.3.2. Maximum Power Point Tracking Techniques:

1) *In 2005, Patcharaprakiti, et.al* [7] presented the AFLC for controlling MPPT of a grid connected photovoltaic system. The simulation results showed that this system was able to adapt the fuzzy parameters for fast response, good transient performance, insensitive in external disturbances but they used FLC and AFLC wich made the system more complex.

2) *In 2007, Cheikh, et.al.* [8] controlled the voltage of the solar panel in order to obtain the maximum power possible from a PV generator but they using try and error in member ship to get the result.

3) *In 2011, Chang and Hsu,* [9] designed an adaptive fuzzy logic control AFLC to improve from scaling FLC and it was mainly to adjust the duty-cycle of the defuzzification of FLC for facing many kinds of external variations, such as loading variation, current of solar cells. Here PV system was composed of solar cell, boost dc/dc converter, and AFLC controller for the goal of MPPT.

4) *In 2011, Elmghany* [2] designed a FLC to maximize the energy received from solar cells by implementing a maximum power point tracker. MPPT controlled by fuzzy logic controller and using buck DC-to-DC converter to keep the PV output power at the maximum point all the time but when using Sugeno control in this case we got better result because this part depended on the carve of the out as Sugeno work.

1.4. Contribution

In this thesis, for the first time we will use an intelligent controller for Sun tracker and MPPT, which is called T.Sugeno fuzzy logic controller.Then by using Matlab/Simulink program we will design the two Sugeno Fuzzy Logic Controllers and apply the GA (Genetic Algorithm) to optimize the memberships and rule base of the fuzzy controller to improve the efficiency of electrical power generated from photovoltaic module.

1.5. Thesis Organization

The remaining chapters of this thesis are organized as follows:
Chapter two, presents a review and introduction to fuzzy logic and its application, fuzzy sets operations, the main concepts in fuzzy sets such as membership functions, and linguistic variable. Chapter three, presents a review and introduces some basic principles of solar energy. Chapter four, presents a review and introduces to genetic algorithm (GA), it's using and main concepts in genetic algorithm such as cross over, mutation, and reproduction. Chapter five, presents the design of the sun tracker controller and simulation results of the fuzzy controller without GA, and the compare results with privious studies. Chapter six, presents the design of the maximum power point tracker controller and simulation results of the fuzzy controller with GA. The final chapter concludes this thesis.

CHAPTER 2 SOLAR ENERGY

2.1. Solar Energy

The sun has produced energy for billions of years. Solar energy is the sun's rays (solar radiation) that reach the Earth. Radiant light and heat from the sun has been harnessed by humans since ancient times using a range of ever-evolving technologies. Solar energy technologies include solar heating, solar photovoltaic, solar thermal electricity and solar architecture, which can make considerable contributions to solving some of the most urgent problems the world now faces.

2.1.1. The energy problem

The energy problem is resulted from the increase of demand for electrical energy and raised of fossil fuel prices. Another problem in the world is the global climate change has increased. As these problems, alternative technologies for producing electricity have received greater attention. The most important solution was in finding other renewable energy resources [10].

2.1.2. Renewable energy

Renewable energy is energy, which comes from natural resources such as sunlight, wind, rain, tides, and geothermal heat, which are naturally replenished [11]. In its various forms, it derives directly from the sun, or from heat generated deep within the earth [1].

Renewable energy replaces conventional fuels in four distinct areas: power generation, water and space heating, transport fuels, and rural and remote areas energy services. Globally, an estimated 193 million households depends on renewable energy systems [11].

The types of renewable energy are solar, wind, hydro power, geothermal and biofuels. Each of these renewable energy sources provides an alternative to traditional energy generation and can be reproduced, reducing our footprint on the environment.

The most important of renewable energy is solar energy however, grid-connected PV increased the fastest of all renewable technologies, with a 60-percent annual average growth rate for the five-year period [11]. Nowadays, solar energy has been widely used in our life, and it's expected to grow up in the next years.

2.1.3. Solar energy can be used for heat and electricity

- Heat water — for use in homes, buildings, or swimming pools
- Heat spaces — inside homes, greenhouses, and other buildings
- Heat fluids — to high temperatures to operate a turbine to generate electricity

2.1.4. Solar energy can be converted to electricity in two ways:

1) **Photovoltaic (PV devices) or "solar cells"** change sunlight directly into electricity. Individual PV cells are grouped into panels and arrays of panels that can be used in a wide range of applications ranging from single small cells that charge calculator

and watch batteries, to systems that power single homes, to large power plants covering many acres.

2) **Solar Thermal/Electric Power Plants** generate electricity by concentrating solar energy to heat a fluid and produce steam that is used to power a generator.

2.2. Photovoltaic

Photovoltaic (PV) is a method of generating electrical power by converting solar radiation into direct current electricity using semiconductors that exhibit the photovoltaic effect. Photovoltaic power generation employs solar panels composed of a number of solar cells containing a photovoltaic material. Materials presently used for photovoltaics include monocrystalline silicon, polycrystalline silicon, amorphous silicon, cadmium telluride, and copper indium gallium selenide/sulfide. Due to the growing demand for renewable energy sources, the manufacturing of solar cells and photovoltaic arrays has advanced considerably in recent years.

Solar panels are made up of photovoltaic cells; it means the direct conversion of sunlight to electricity by using a semiconductor, usually made of silicon. The word photovoltaic comes from the Greek meaning “light” (photo) and “electrical” (voltaic); the common abbreviation for photovoltaic is PV [12].
The need for low cost electric power in isolated areas is the primary force driving the world-wide photovoltaic (PV) industry today. PV technology is simply the least-cost option for a large number of applications, such as stand-alone power systems for cottages and remote residences, remote telecommunication sites for utilities and the military, water pumping for farmers, and emergency call boxes for highways and college campuses [13].

2.3. Solar Cells

Solar cells are converting light energy, to another form of energy, electricity. When light energy is reduced, or stopped as the sun goes down in the evening or a cloud passes in front of the sun, the conversion process stops or slows down. When the sunlight returns, the conversion process immediately resumes, this conversion without any moving parts, noise, pollution, radiation or constant maintenance. These advantages are due to the special properties of semiconductor materials that make this conversion possible. Solar cells do not store electricity; they just convert light to electricity when sunlight is available. To have electric power at night, a solar electric system needs some form of energy storage, like batteries [14].

2.3.1. Principle and Manufacturing of solar cells

A solar cell (also called a photovoltaic cell) is an electrical device that converts the energy of light directly into electricity by the photovoltaic effect. It is a form of (photoelectric cell), when exposed to light, can generate and support an electric current without being attached to any external voltage source.

Solar modules are made of semiconductors that are very similar to those used to create integrated circuits for electronic equipment. The most common type of semiconductor currently in use is made of silicon crystal. Silicon crystals are laminated into n-type and p-type layers, stacked on top of each other. Light striking the crystals induces the ″photovoltaic effect " which generates electricity [15]. As shown in Figure (2.1).

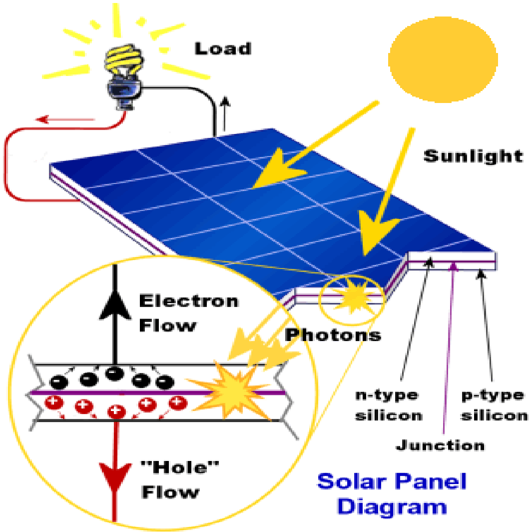


Figure 2.1: Principle of Solar Cells

The electricity produced is called direct current (DC) and can be used immediately or stored in a battery. For systems installed on homes served by a utility grid, a device called an inverter changes the electricity into alternating current (AC), the standard power used in residential homes [15]. As shown in Figure (2.2).

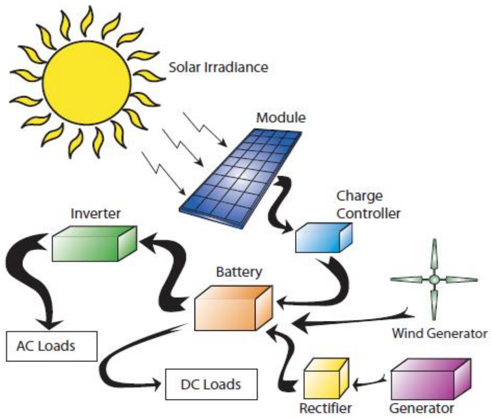


Figure 2.2: How Solar Cells Works

2.3.2. Types of solar panels

There are three basic types of solar panels, which differ in them material, price, and efficiency, since the efficiency is a measurement of the output divided by a certain factor. When talking about solar cell or panel efficiency, we are really looking the percentage of solar energy that is captured and converted into electricity [16].

1. ***Monocrystalline Solar Panels***: It consists of large crystal of silicon, given good power-to-size ratio, have efficiency approximately 18%, typically within the range of 135-170 Watts per m^2, outstanding performance in cooler conditions, excellent life span, however they are the most expensive, see Figure (2.3).

Figure 2.3: Monocrystalline Solar Panels

2. ***Polycrystalline Solar Panels**:* given good power typically 120-150 Watts per m^2, have efficiency approximately 15%. Instead of one large crystal, this type of solar panel consists of multiple amounts of smaller silicon crystals, see Figure (2.4).

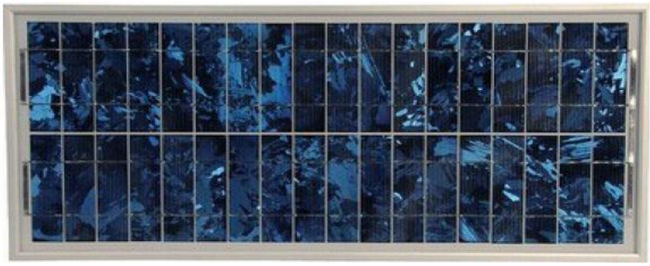


Figure 2.4: Polycrystalline Solar Panels

They are the most common type of solar panels on the market today. They look a lot like shattered glass. They are slightly less efficient than the monocrystalline solar panels and less expensive to produce

3. ***Amorphous Solar Panels**:* given lowest power typically 60-80 Watts/m^2, have efficiency approximately 10%. Consisting of a thin-like film made from molten silicon that is spread directly across large plates of stainless steel or similar material, see Figure (2.5).

Figure 2.5: Amorphous Solar Panels

One advantage of amorphous solar panels over the other two is that they are shadow protected. That means when a part of the solar panel cells are in a shadow the solar panel continues to charge. These types of solar panels have lower efficiency than the other two types of solar panels, and the cheapest to produce. These work great on boats and other types of transportation.

2.3.3. Modeling of PV devices (Solar cell)

a) Ideal PV Cell

As see in Figure (2.6) the equivalent circuit of the ideal PV cell. The basic equation from the theory of semiconductors that mathematically describes the *I–V* characteristic of the ideal PV cell is

$$I = I_{pv,cell} - I_d = I_{pv,cell} - I_{0,cell}\left[e^{qV/akT} - 1\right] \quad (2.1)$$

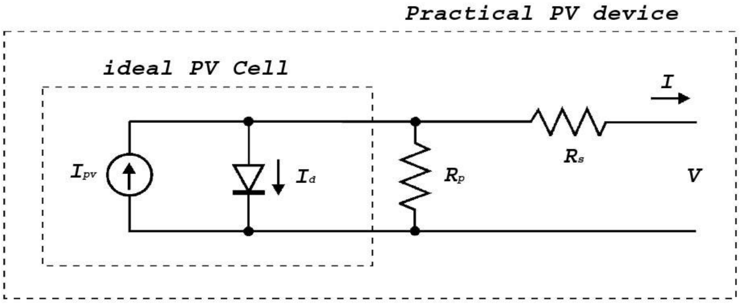


Figure 2.6: Equivalent Circuit of a Practical PV Device

Where $I_{pv,cell}$is the current generated by the incident light (it is directly proportional to the Sun irradiation). I_d is the Shockley diode equation, $I_{0,cell}$ is the reverse saturation or leakage current of the diode, q is the electron charge (1.60217646 × 10^{-19} C), k is the Boltzmann constant (1.3806503 × 10^{-23} J/K), T (in Kelvin) is the temperature of the p–n junction, and a is the diode ideality constant [17].
A shunt resistance and a series resistance component are added to the model since no solar cell is ideal in practice [17].

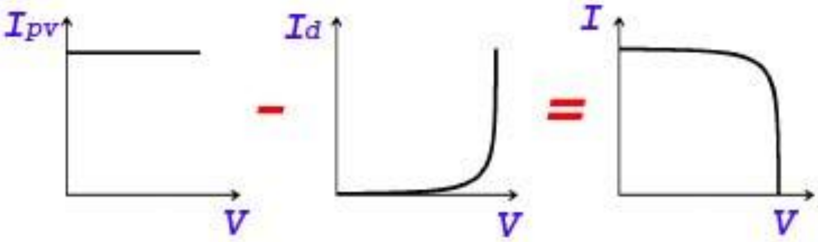


Figure 2.7: Characteristic *I–V* Curve of The PV Cell.

Figure (2.7) show the I-V curve of the PV cell, The net cell current I is composed of the light-generated current I_{pv} and the diode current I_d

b) Modeling the PV Array

The basic equation (2.1) of the elementary PV cell does not represent the *I–V* characteristic of a practical PV array, Practical arrays are composed of several connected PV cells and the observation of the characteristics at the terminals of the PV array requires the inclusion of additional parameters to the basic equation [17].

$$I = I_{pv} - I_0\left[e^{\left(\frac{V+R_S}{V_t\,a}\right)} - 1\right] - \left[\frac{V+R_S I}{R_p}\right] \quad (2.2)$$

Where $I_{pv}$and I_0 are the photovoltaic (PV) and saturation currents, respectively, of the array and V_t = $N_s kT/q$ is the thermal voltage of the array with N_s cells connected in series.
R_s and R_p is the equivalent series and parallel resistance, when Cells connected in parallel increase the current and cells connected in series provide greater output voltages [10].
From equation (2.2) we can originates the I-V curve as shown in Figure (2.8).

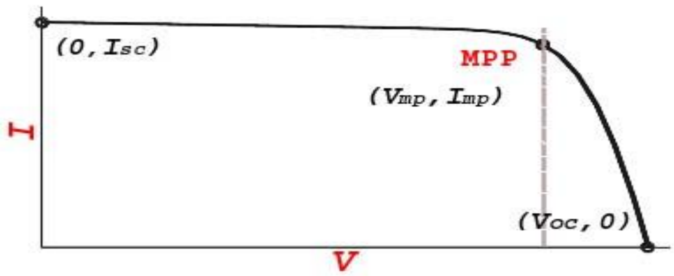


Figure 2.8: Characteristic *I–V* Curve of a Practical PV Device

The light generated current of the photovoltaic cell I_{pv} and saturation current I_0 depend on the temperature according to the following equations:

$$I_{pv} = \left(I_{pv,n} + K_I \Delta_T\right) \frac{G}{G_n} \quad \text{...} \quad (2.3)$$

Where $I_{pv,n}$ is the light-generated current at the nominal condition (usually 25 ◦C and 1000W/m2), $\Delta_T = T - T_n$ (being T and T_n the actual and nominal temperatures [K]), G [W/m^2] is the irradiation on the device surface, and G_n is the nominal irradiation.

c) Improving the Model

The PV model described in the previous section can be improved by:

$$I_0 = \frac{I_{sc,n} + K_I \Delta_T}{\left[e^{\left(\frac{V_{oc,n} + K_V \Delta_T}{aV_t}\right)}\right] - 1} \quad \text{...} \quad (2.4)$$

This modification aims to match the open-circuit voltages of the model with the experimental data for a very large range of temperatures.

2.4. Balance of System (BOS):

PV modules are integrated into systems designed for specific applications. BOS is the components, which added to the module. These components can be classified into four categories [2]:

- Batteries - store electricity to provide energy on demand at night or on overcast days
- Inverters - convert the direct current (DC) output of the array or the battery into alternating current (AC).
- Rectifiers (battery chargers) - convert the AC current produced by a generator into the DC current needed to charge the batteries.
- Controllers - manage the energy storage to the battery and deliver power to the load.

➢ Several electronic devices are used to control and modify the electrical power produced by the photovoltaic array. These include: Battery charge controllers - regulate the charge and discharge cycles of the battery; Maximum power point trackers (MPPT) as it will be explained in section (2.6).

➢ Structure - required to mount or install the PV modules and other components [2]. Many of these plants are integrated with agriculture and some use innovative tracking systems that follow the sun's daily path across the sky to generate more electricity than conventional fixed-mounted systems as explained in the next section

2.5. Principle of Sun Tracker

The sun rises each day from the east, and moves across the sky to the west. When the sun is shining, it is sending energy to us, and we can feel its heat; however, its position varies with the time of day and the seasons. Thus, if we could get a solar cell to turn and look at the sun all day, then it would be receiving the maximum amount of sunlight possible and converting it into electricity. A solar tracker is a device that is used to align a single photovoltaic panel or an array of P.V modules with the sun, so a solar tracker can improve a systems power output by keeping the sun in focus throughout the day; thus improving effectiveness of such equipment over any fixed position [2].
A well-designed system, which utilizes a tracker, will reduce an initial implementation cost, since it needs fewer expensive panels due to increased efficiency. There are two general forms of tracking techniques used: dynamic tracking and fixed control algorithms. The main difference between them is the manner in which the path of the sun is determined. In *the dynamic tracking system*, actively searches for the sun's position at any time of day, light sensors are positioned on the tracker at various locations or in specially shaped holders. If the sun is not facing the tracker directly there will be a difference in light intensity on one light sensor compared to another and this difference can be used to determine which direction the tracker has to tilt in order to be facing the sun. On the other hand, in *the fixed control algorithm systems*, the control system uses no sensing. It does not actively find the sun's position but instead determines the position of the sun through prerecorded data for a particular site. If given the current time, day, month, and year, then the system calculates the position of the sun, so it is called 'open loop trackers'[18]. Common to both forms of tracking is the method of direction control system.

2.6. Principle of Maximum Power Point Tracking

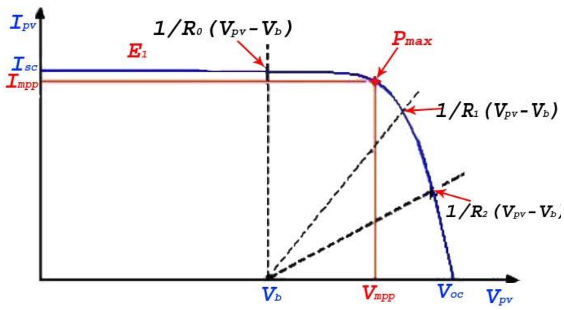


Figure 2.9: Current-Voltage Characteristic of a PV Module

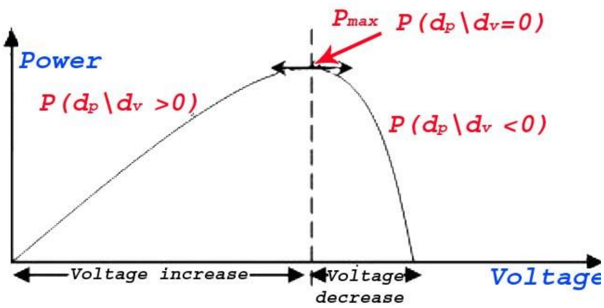


Figure 2.10: Power-Voltage Characteristic of a PV Module

A typical characteristic curve of PV model's current and voltage curve is shown in Figure (2.9) and the power and voltage curve of the module is shown in Figure (2-10). There exists a single maxima power corresponding to a particular voltage and current.

So, when a direct connection is carried out between the source and the load, the output of the PV module is seldom maximum and the operating point is not optimal [8].
To overcome this problem, it is necessary to add an adaptation device, MPPT controller with a DC-DC converter, between the source and the load, Figure (2.11).

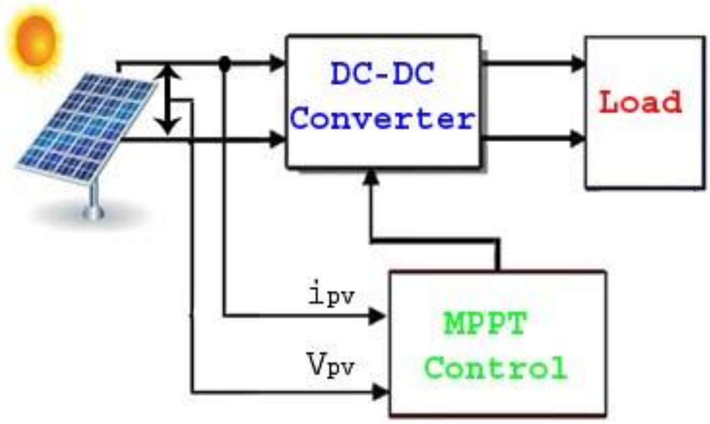


Figure 2.11: Photovoltaic With MPPT System

Furthermore the characteristics of a PV system vary with temperature and insolation, (Figures 2.12 and 2.13) [8]. So, the MPPT controller is also required to track the new modified maximum power point in its corresponding curve whenever temperature and/or insolation variation occurs.

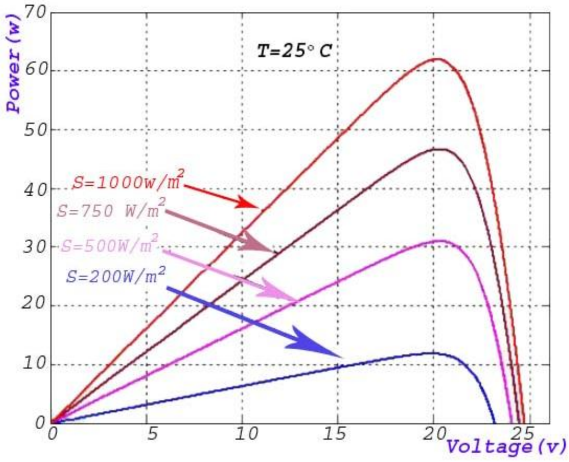


Figure 2.12: Influence of The Solar Radiation for Constant Temperature

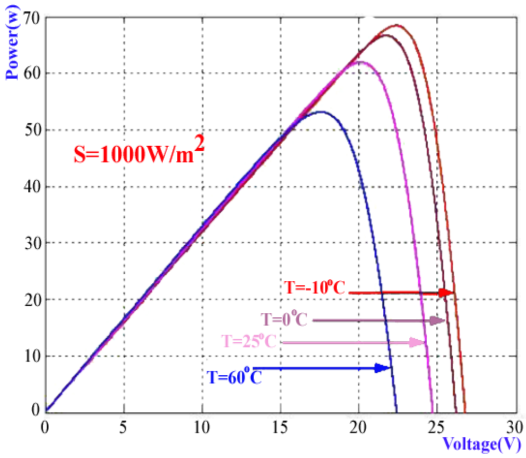


Figure 2.13: Influence of the Temperature Of Junction For Constant Isolation

MPPT is used for extracting the maximum power from the solar PV module and transferring that power to the load. A dc/dc (step up/step down) converter acts as an interface between the load and the module.
The MPPT is changing the duty cycle to keep the transfer power from the solar PV module to the load at maximum point [8].

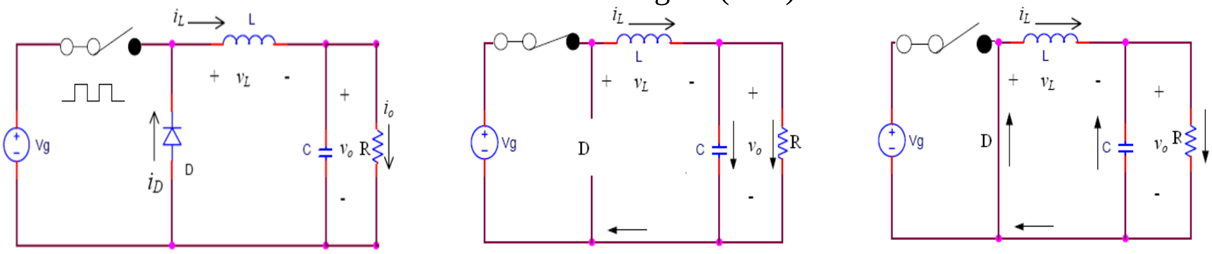
2.7. Dc - Dc Converter

2.7.1. Basic DC - DC converters

- Buck converter (Step-down converter) which using in this research.
- Boost converter (Step-up converter).
- Buck-Boost converter (Step-down/step-up converter).

2.7.2. Buck Converter

A buck converter is a step-down DC-to-DC converter. The operation of the buck converter is simple, with an inductor and two switches (transistor and diode) that control the current of the inductor as shown in Figure (2.14).



(a) **(b)** **(c)**

Figure 2-14: **(a)** Buck- Converter **(b)** Switch On **(c)** Switch Off

Refer to Figure (2-14b) ″On State″, when the switch is connected, L is connected to the switch, which tends to oppose the rising current and begins to generate an electromagnetic field in its core. Diode D is reverse biased and is essentially an open circuit at this point. The inductor current increases, inducing a positive voltage drop across the inductor and a lower output supply voltage in reference to the input source voltage. The inductor serves as a current source to the output load impedance [19].
Refer to Figure (2-14c) ″Off State″, in the off state, the switch is open, diode D conducts and energy is supplied from the magnetic field of L and electric field of C. The current through the inductor falls linearly. When the FET switch is off, the inductor current discharges, inducing a negative voltage drop across the inductor. Because one port of the inductor is tied to ground, the other port will have a higher voltage level, which is the target output supply voltage. The output capacitance acts as a low-pass filter, reducing output voltage ripple as a result of the fluctuating current through the inductor. The diode prevents the current flowing from the inductor when the FET switch is off [19].
Figure (2-15) evolutes the voltages and currents with time in an ideal buck

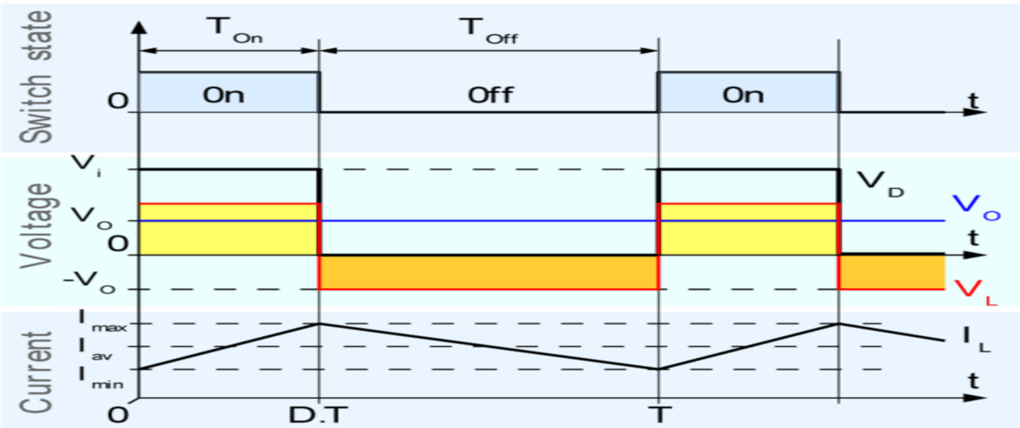


Figure 2-15: The Voltages and Currents
With Time in an Ideal Buck Converter

CHAPTER 3 FUZZY LOGIC CONTROL

3.1. Fuzzy logic history

Lotfi Zadeh conceived the concept of fuzzy Logic (FL), a professor at the University of California at Berkley, who was published the first paper on fuzzy set theory in early 1960's [20], which presented not as a control methodology, but as a way of processing data. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Professor Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control [21].

In 1974; Mamdani published the first paper for fuzzy applications [22]. Mamdani method proposed as an attempt to control a real application in steam engine. The fuzzy inference system proposed by Mamdani; known as the Mamdani model in fuzzy system literature.

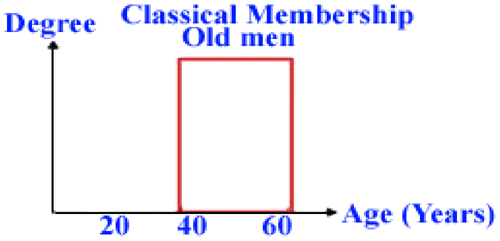
In 1985, Takagi and Sugeno published the paper of fuzzy systems [23]. The fuzzy inference system was proposed by Takagi and Sugeno; known as the T-S model in fuzzy system literature.

There are several advantages of using fuzzy control over classical control methods. As Lotfi Zadeh, who is considered the father of fuzzy logic, once remarked: "In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper" [20]. Japanese were the first to use fuzzy logic in application in 1980's. Japanese and Korean companies are using fuzzy logic to enhance things like computers, air conditioners, automobile parts, cameras, televisions, washing machines, and robotics. In 1994, Japan exported products using fuzzy logic totaling 35 billion dollar. Today, many publications discuss the theoretical background of fuzzy logic; its history, and how to program fuzzy logic algorithms.

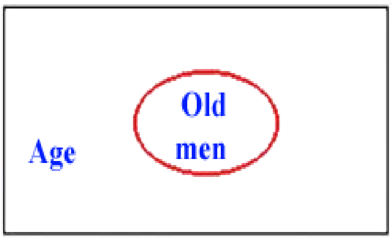
3.2. Fuzzy Logic

3.2.1. Definition

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth. In the (Boolean) logic we see that the results for any operation can be true or false if we refer to true by (1) and the false by (0) then the result may be (1) or (0).



(a)



(b)

Figure 3.1: Classical Sets

Figure (3.1a) shows an example for classical set that has two values true or false. We see that the classical set have crisp boundary. This example shows an age example: the man is old if he between 40 years and 60 years in that interval all age has the same degree (1). In addition, outside of this interval, it has (0) degree. However, there is problem; what about 39 years and 11 months, is the man young! No he old but has degree less than the 40 years, but in the Classical sets there are not degrees there are two values 1 or 0. Therefore, what is the solution; fuzzy sets give the solution [25].

The essential characteristics of fuzzy logic as founded by Lotfi Zadeh are as follows:

- In fuzzy logic, exact reasoning is viewed; as a limiting case of approximate reasoning.
- In fuzzy logic, everything is a matter of degree.
- Any logical system can be Fuzzified.
- In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
- Inference is viewed; as a process of propagation of elastic constraints.

3.2.2. Why Use Fuzzy Logic?

There are many reasons.

1- Fuzzy logic is used to control the complex, and nonlinear systems without making analysis for these systems.
2- Fuzzy control enables engineers to implement the control technique by human operators to make ease of describing the systems [25].
3- Fuzzy logic is flexible with any given systems [26]. If any changes are happening in the system we do not need to start from the first step, but we can add some functions on top of it.
4- Fuzzy logic can be blended with conventional technique to simplify their implementation.

3.2.3. Applications of Fuzzy Logic

The Japanese used fuzzy logic in many applications, such as (subway train and water-treatment control), but in these years there are many more applications of fuzzy logic.

1- Fuzzy logic is used to control the Camcorder to make stabilization in image if there is any rock [25].
2- In washing machine, there is a soft and bad manner clothes, and there are different quantities of laundry. Control of washing cycle is based on these date.
3- Robotics controls, Refrigerators for temperature control.
4- Engine Control in the modern cars.
5- Other usages of fuzzy logic is in image processing, such as image identification, representation, and description.

3.3. Fuzzy Sets

3.3.1. Basic Concepts

In crisp sets, an element in the universe has a well-defined membership or non-membership to a given set. Membership to a crisp set ″A" can be defined through a membership function defined for every element ″x" of the universe as:

$$\mu_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases} \quad \ldots\ldots(3.1)$$

Nevertheless, for an element in a universe with fuzzy sets, the membership function can take any value between ″0" and ″1". This transition among various degrees of membership can be thought of as conforming to the fact that the boundaries of the fuzzy sets are vague and ambiguous. An example of a graphic for the membership function of a crisp set is illustrated in Figure (3.2) [2].

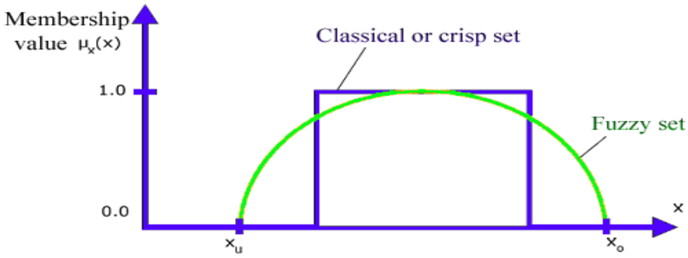


Figure 3.2: Fuzzy and Classical Sets

Fuzzy membership of an element from the universe in this set is measured by a function that attempts to describe vagueness. In fuzzy logic, linguistic variables take on linguistic values, which are words with associated degrees of membership in the set. Thus, instead of a variable temperature assuming a numerical value of 70 C°, it is treated as a linguistic variable that may assume, for example, linguistic values of "hot" with a degree of membership of 0.7, "very cool" with a degree of 0.6, or "very hot" with a degree of 0.92. Each linguistic term is associated with a fuzzy set, each of which has a defined membership function.
Formally, a fuzzy set is defined as a set of pairs where each element in the universe "F" has a degree of membership associated with it:

$$A = \{(x, \mu_A(x)) \mid x \in F, \mu_A(x) \in [0,1]\} \quad \ldots\ldots (3.2)$$

The value $\mu_A(x)$ is the degree of membership of object "x" to the fuzzy set "A" where $\mu_A(x) = 0$ means that x does not belong at all to the set, while $\mu_A(x) = 1$ means that the element is totally within the set [27].

3.3.2. Membership Function (MF)

1) Features of Membership Function:

- Core: comprises of elements "x" of the universe, such that $\mu_A(x) = 1$
- Support: comprises of elements "x" of universe, such that $\mu_A(x) > 0$
- Boundaries: comprise the elements "x" of the universe $0 < \mu_A(x) < 1$
- A normal fuzzy set has at least one element with membership 1

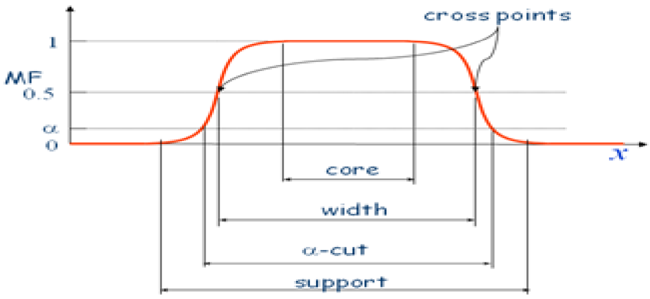


Figure 2.3: MF Terminology

For fuzzy set, if one and only one element has a membership = 1, this element is called as the *prototype* of set, and "A" *subnormal* fuzzy set has no element with membership=1.

2) Types of member ship functions

Every fuzzy set can be represented by its membership function. The shape of membership function depends on the application and can be monotonic, triangular, and trapezoidal or bell shaped as shown in Figure (3.4) [2].

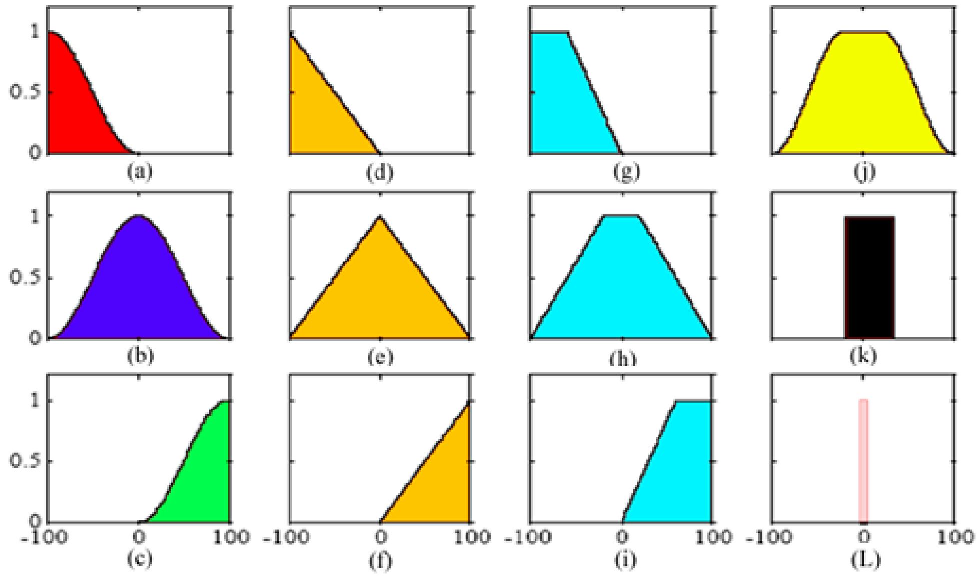


Figure 3.4: Different Shapes Of Membership Functions

(a) s_Function. (b) π_Function. (c) z_Function. (d-f) Triangular versions. (g-i) Trapezoidal versions. (J) Flat π_function. (k) Rectangle. (L) Singleton.

The membership function could be defined as a graphical representation of the quantity of participation of the inputs. It links a value with each of the inputs parameters that are treated, defines functional overlap amongst inputs, and finally defines an output parameter. The rules usually take the input membership parameters as features to establish their weight over the fuzzy output sets of the final output response. Once the functions are deduct, scaled, and combined, they have to be defuzzified into a crisp output, which leads the application. There are some different memberships functions linked to each input and output parameter [28].

3) MF Formulation

- Triangular MF $trimf(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$ ....(3.3)
- Trapezoidal MF $trapmf(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$ ....(3.4)
- Gaussian MF $gaussmf(x; a, b, c) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2}$ ....(3.5)
- Generalized bell MF $gbellmf(x; a, b, c) = \frac{1}{1+\left|\frac{x-c}{b}\right|^{2b}}$ ....(3.6)

3.3.3. Fuzzy Set Operations

Basic operations on sets in crisp set theory are the set complement, set intersection, and set union. Fuzzy set operations are very important because they can describe intersections between variables for a given element "x" of the universe, the following function theoretic operations for the set theoretic operations of complement, intersection, and union are defined [29]:

1) Complement (NOT Operation):

Consider a fuzzy set "A" in universe "X". its complement "A" as shown in Figure (3.5)

$$\mu_{\bar{A}}(x) = NOT(\mu_A(x)) = 1 - \mu_A(x) \quad \text{.......} \quad (3.7)$$

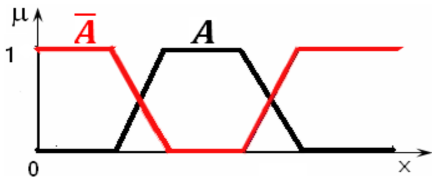


Figure 3.5: Complement Of Fuzzy Sets A

2) Intersection (AND Operation):

Consider two fuzzy sets "A" and "B" in universe "X". as shown in Figure (3.6)

$$\mu_{A\cap B}(x) = \mu_A(x)\ AND\ \mu_B(x) = \mu_A(x) \wedge \mu_B(x) = min\{\mu_A(x), \mu_B(x)\} \forall\ x \in X \quad \text{.....}(3.8)$$

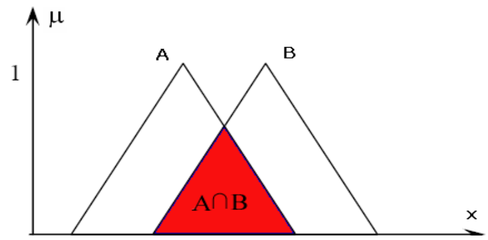


Figure 3.6: Intersection If Fuzzy Sets A and B

3) Union (OR Operation):

Consider two fuzzy sets "A" and "B" in universe "X". $A \cup B$ is the whole area covered by the sets as shown in Figure (3.7)

$$\mu_{A\cup B}(x) = \mu_A(x)\ OR\ \mu_B(x) = \mu_A(x) \vee \mu_B(x) = max\{\mu_A(x), \mu_B(x)\}\ \forall\ x \in X \quad \text{....} \quad (3.9)$$

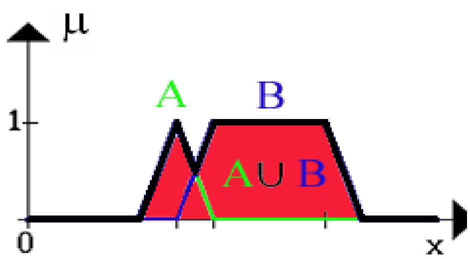


Figure 3.7: Union Of Fuzzy Sets A and B

Table 3.1: Some Properties of Fuzzy Sets Operations

Complement	$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$
Intersection	$\mu_{A \cap B}(x) = \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x))$
Union	$\mu_{A \cup B}(x) = \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x))$
Law of contradiction	$A \cap A' \neq \emptyset$
Law of excluded middle	$A \cup A' \neq X$
De Morgan's laws	$(A \cap B)' = A' \cup B'$ $(A \cup B)' = A' \cap B'$
Commutative	$A \cap B = B \cap A$ $A \cup B = B \cup A$
Associative	$A \cap (B \cap C) = (A \cap B) \cap C$ $A \cup (B \cup C) = (A \cup B) \cup C$
Distributive	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Involution (Double negation)	$A'' = A$
Conjunction	$A \wedge B = C$ "Quality C is the *conjunction* of Quality A and B"
Disjunction	$A \vee B = C$ "Quality C is the *disjunction* of Quality A and B"

3.4. Notion of linguistic rule

When fuzzy sets used to solve the problem without analyzing the system, but the expression of the concepts and the knowledge of it in human communication are needed. Human usually do not use mathematical expression but use the linguistic expression [26].
For example, if you see heavy box and you want to move it, you will say, "I want strong motor to move this box" we see that, we use strong expression to describe the force that we need to move the box. In fuzzy sets we do the same thing we use linguistic variables to describe the fuzzy sets.

The principal idea of fuzzy logic systems is to express the human knowledge in the form of linguistic if-then rules. Every rule has two parts:

- Antecedent part (premise), expressed by if... and
- Consequent part, expressed by then...

The antecedent part is the description of the state of the system, and the consequent is the action that the operator who controls the system must take. There are several forms of if then rules. The general is:

If (a set of conditions is satisfied) then (a set of consequences can be inferred).

Example: If the temperature is high, then the pressure is small.
The general form of this rule is:
Rule: If x is "A" then y is "B".

Temperature "x" and pressure "y" are linguistic variables. "x" represents the state of the system, and "y" is control variable and represents the action of the operator. High "A" and small "B" are linguistic values or labels characterized by appropriate membership functions of fuzzy sets. They are defined in the universe of discourse of the linguistic variables "x" and "y".

Example

Linguistic variable — Linguistic value — Linguistic variable — Linguistic value — Linguistic variable — Linguistic value

If **temperature is cold** ***and*** **oil is cheap** ***then*** **heating is high**

μ_{cold} — μ_{cheap} — μ_{high}

antecedent or premise — consequence or conclusion

Takagi and Sugeno [30] proposed the form which has the fuzzy sets only in the premise part of the rule, and the consequent part is described by a non-fuzzy equation of the input variable.
Example: If velocity is high, then force is k*(velocity) ^{2}
Another form of this rule is:

Rule: If "x" is "A", then "y" is k*x^2.
Or more general,

Rule: If x is A then y is f(x)

3.5. General Structure Of Fuzzy Logic Control "FLC" System

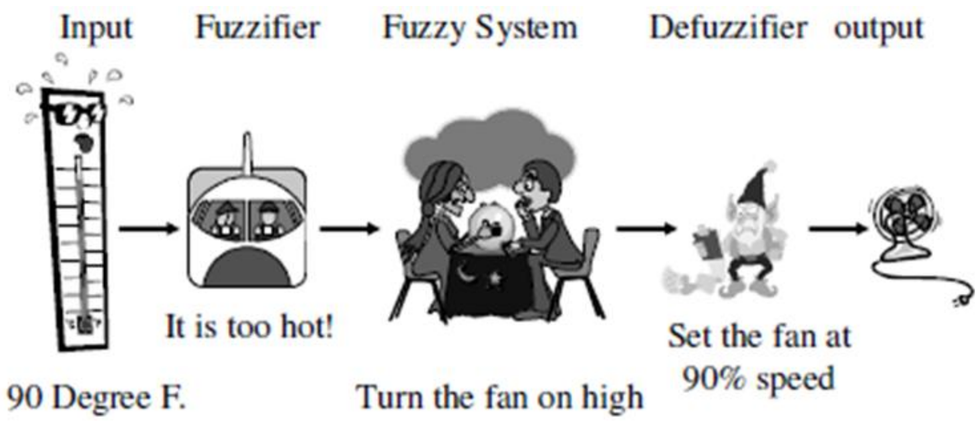


Figure 3.8: General Structure Of Fuzzy Systems

The basic parts of every fuzzy controller are displayed in Figure (3.9). The fuzzy logic controller (FLC) is composed of a fuzzification interface, knowledge base, inference engine, and defuzzification interface.

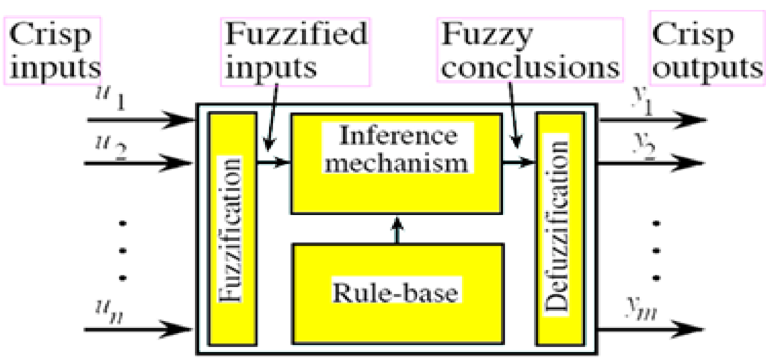


Figure 3.9: Basic Parts Of Fuzzy Logic Controller

The fuzzifier maps the input crisp numbers into fuzzy sets to obtain degrees of membership. It is needed in order to activate rules, which are in terms of the linguistic variables. The inference engine of the FLC maps the antecedent fuzzy (IF part) sets into consequent fuzzy sets (THEN part). This engine handles the way in which the rules are combined. The defuzzifier maps output fuzzy sets into a crisp number, which becomes the output of the FLC [2].

3.5.1. Fuzzification

The first step in fuzzy logic processing the crisp inputs is transformed into fuzzy inputs as shown in Figure (3.10). This transformation is called fuzzification. The system must turn numeric values into language and corresponding domains to allow the fuzzy inference engine to inference to transform crisp input into fuzzy input, membership functions must be first be defined for each input. Once membership functions are defined, fuzzification takes a real time input value, such as temperature, and compares it with the stored membership function information to produce fuzzy input values. Fuzzification plays an important role in dealing with uncertain information that might be objective in nature [2].

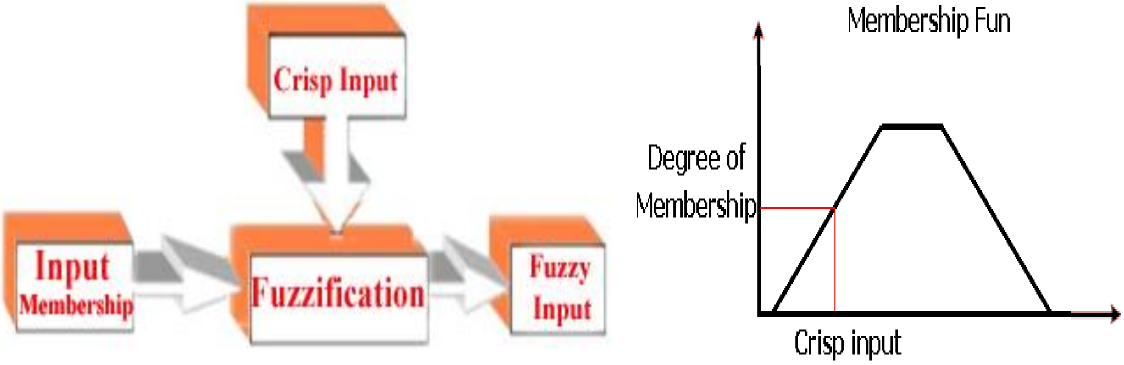


Figure 3.10: Fuzzification

Converts the *crisp input* to a *linguistic variable* using the membership functions stored in the fuzzy knowledge base.

3.5.2. Knowledge Base

Knowledge base is the inference basis for fuzzy control. It defines all relevant language control rules and parameters. The knowledge base is the core of a fuzzy control system. The knowledge base of Fuzzy Logic Controller (FLC) is comprised of two parts [31]:

1. Database.
2. Rule base

There are four principal design parameters in the database for a fuzzy logic controller:

1. Discretization
2. Normalization of universe of discourse
3. Fuzzy partition of input and output spaces
4. Membership functions of primary fuzzy set.

A linguistic controller contains rules in the (if-then) format. The Rule base is the cornerstone of the fuzzy model. The expert knowledge, which is assumed to be given as a number of if-then rules, is stored in a fuzzy rule base. The rules may use several variables in both the condition and the conclusion of the rules.

3.5.3. Fuzzy Interference Engine

There are many inference methods, which deals with fuzzy inference like: Mamdani method, Larsen method, Tsukamoto method, and the Sugeno style inference, or, Takagi-Sugeno_Kang (TSK) method. The most important and widely used in fuzzy controllers are the Mamdani and Takagi-Sugeno methods.

3.5.3.1 Mamdani method

Which is the most commonly used fuzzy inference technique. In 1974, Professor Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators. The Mamdani-style fuzzy inference process is performed in four steps [24]:

1) Fuzzification of the input variables
2) Rule evaluation.
3) Aggregation of the rule outputs
4) Defuzzification

To illustrate the fuzzy inference let's examine a simple two-input one-output problem that includes three rules:

Rule (1) …. IF X is A_3 OR Y is B_1 THEN z is C_1
Rule (2)…. IF X is A_2 AND Y is B_2 THEN z is C_2
Rule (3)…. IF X is A_1 THEN z is C_3

Step 1: Fuzzification

The first step in the application of fuzzy reasoning is a Fuzzification of inputs in the controller, which is to take the crisp inputs, x1 and y1, and determine the degree to which these inputs belong to each of the appropriate fuzzy sets. It means that to every crisp value of input we attribute a set of degrees of membership (mj, j=1,n) to fuzzy sets defined in the universe of discourse for that input.

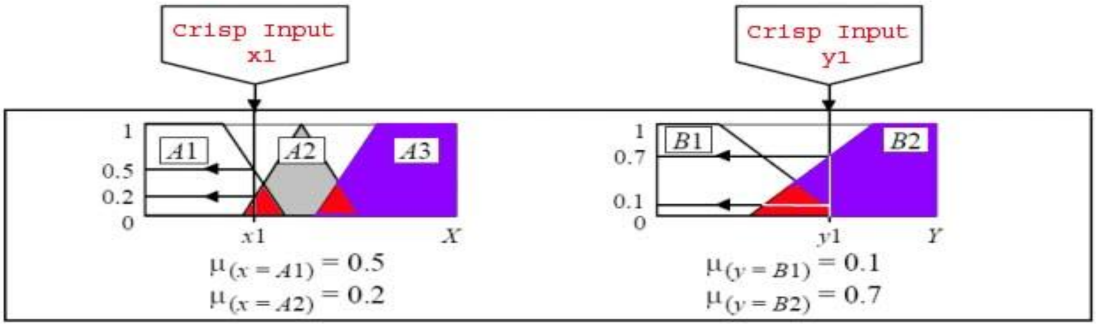


Figure 3.11: Fuzzification Stage

Step 2: Rule evaluation

The second step is to take the Fuzzified inputs, $\mu(x=A1) = 0.5$, $\mu(x=A2) = 0.2$, $\mu(y=B1) = 0.1$ and $\mu(y=B2) = 0.7$, and apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function. To evaluate the disjunction of the rule antecedents, we use the OR fuzzy operation. As shown Operations with fuzzy sets the most used approach for the union is to get the maximum:

$\mu A \cup B(x) = max\ [\mu A(x), \mu B(x)]$ ………………………………………….…… (3.10)

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the AND fuzzy operation intersection which used minimum approach:

$\mu A \cap B(x) = min\ [\mu A(x), \mu B(x)]$ ……………………………….……………..… (3.11)

The rule evaluations are clearly appears in Figure (3.12).

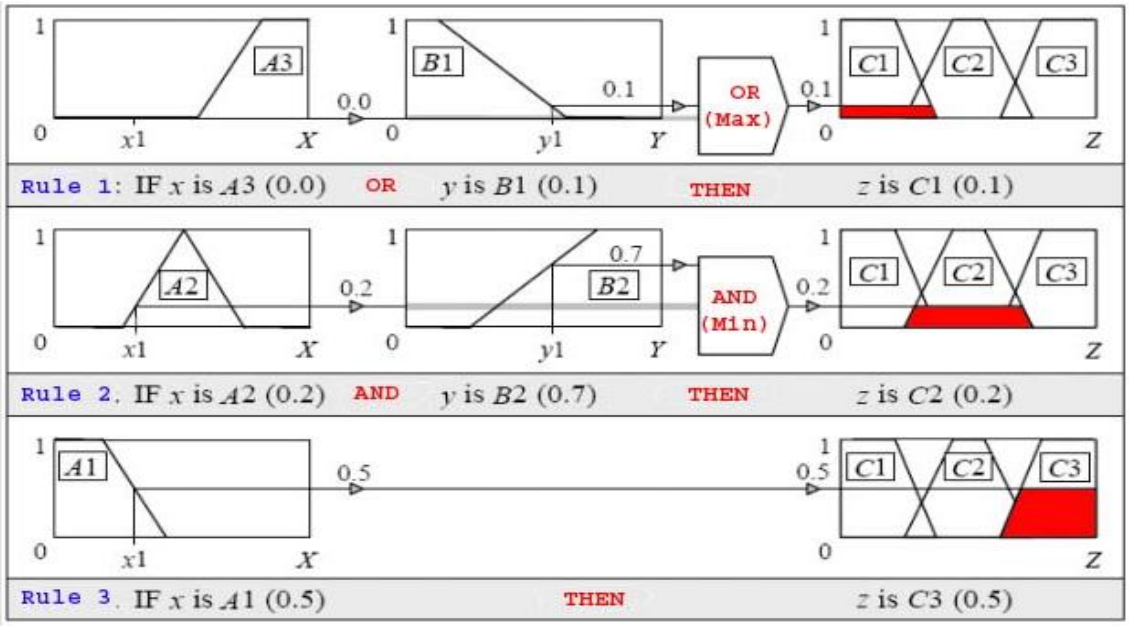


Figure 3.12: Rule Evaluation in Mamdani Method

The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth. This method is called clipping. Since the top of the membership function is sliced, the clipped fuzzy set loses some information. However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to Defuzzify.

Step 3: Aggregation of the rule outputs

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule consequents previously clipped (Max-Min Composition) or scaled (Max-Product Composition) and combine them into a single fuzzy set.

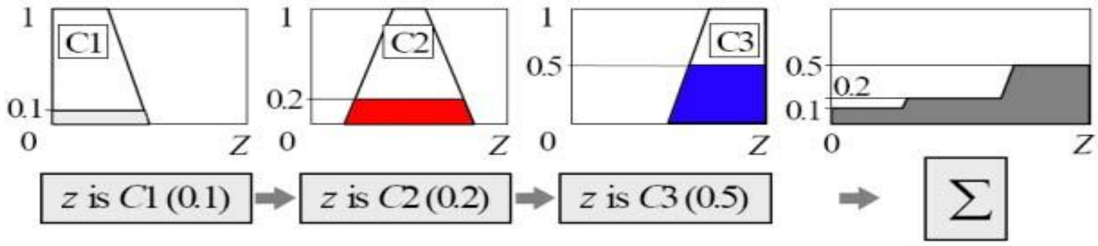


Figure 3.13: Aggregation Stage in Mamdani Method

Step 4: Defuzzification

Defuzzify the aggregate output fuzzy set into a single number. This step will explainin details in section 3.5.4. But in this example we used the COG method to solve the defuzzification as shown in Figure (3.14)

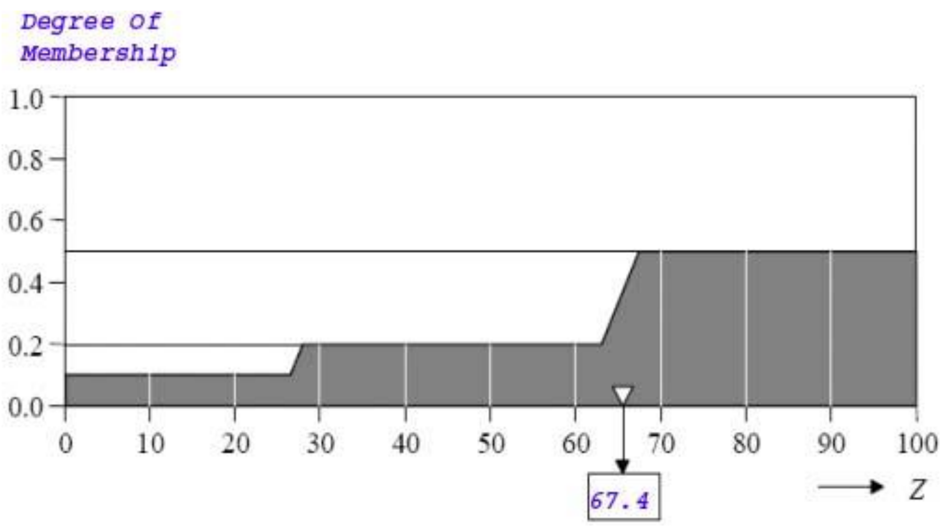


Figure 3.14: COG Approach in Defuzzification Stage

3.5.3.2 Sugeno method

Since Mamdani's pioneering works on fuzzy control motivated by zadeh's approach to inexact, there have been numerous studies on fuzzy reasoning. Most fuzzy controllers have been designed, based on human operator experience and/or control engineer knowledge. It is however often the case that an operator cannot tell linguistically what kind of action he takes in a particular situation. In this respect, it is quite useful to provide a method of modeling the control actions using numerical data. In 1985 Takagi-Sugeno-Kang suggested to use a single spike, a singleton, as the membership function of the rule consequent, and they suggested another approach that using equation consequent in place off singleton consequent. A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. Sugeno-style fuzzy inference is very similar to the Mamdani method. Sugeno changed only a rule consequent. Instead of a fuzzy set, he used a mathematical function of the input variable [24].

The format of the Sugeno-style fuzzy rule is:

IF X is A AND Y is B THEN Z is $f(x, y)$ ………………………………………….…… (3.12)

Where X, Y and Z are linguistic variables; A and B are fuzzy sets on universe of discourses X and Y, respectively; and $f(x, y)$ is a mathematical function.

The most commonly used zero-order Sugeno fuzzy model applies fuzzy rules in the following form:

IF X is A AND Y is B THEN Z is k ……………………………………….……….…. (3.13)

Where k is constant.

In this case, the output of each fuzzy rule is constant. All consequent membership functions are represented by singleton spikes.
The following Figures (3.15, 3.16, and 3.17) illustrate the idea for TSK which likes Mamadni steps.

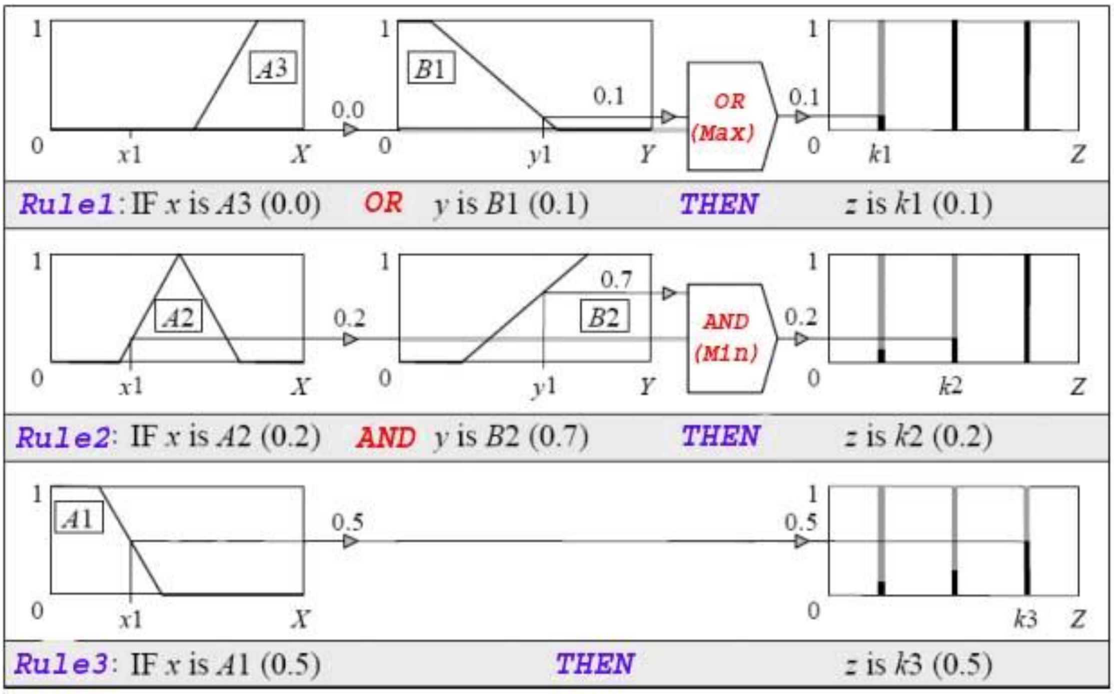


Figure 3.15: Rule Evaluation Stage in TSK Method

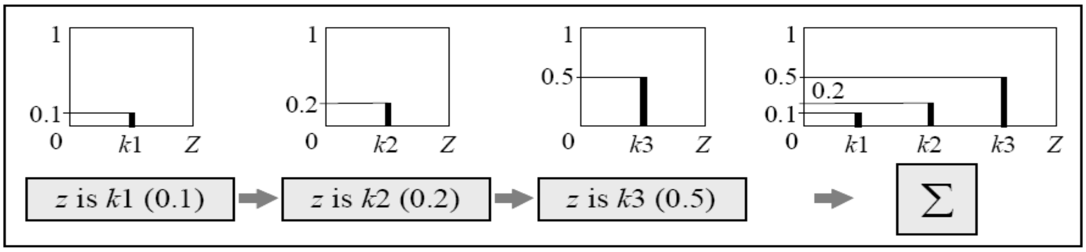


Figure 3.16: Aggregation Stage in TSK Method.

For Defuzzification stage it's better to use Weighted Average method (WA):

$$WA = \frac{\mu(k1) \times k1 + \mu(k2) \times k2 + \mu(k3) \times k3}{\mu(k1) + \mu(k2) + \mu(k3)} = \frac{0.1 \times 20 + 0.2 \times 50 + 0.5 \times 80}{0.1 + 0.2 + 0.5} = 65$$

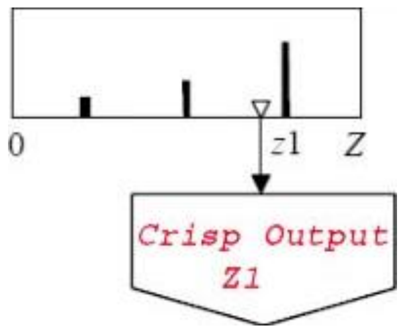


Figure 3.17: (WA) Method in Defuzzification Stage.

The overall fuzzy logic controller "FLC" appear in Figure (3.18)

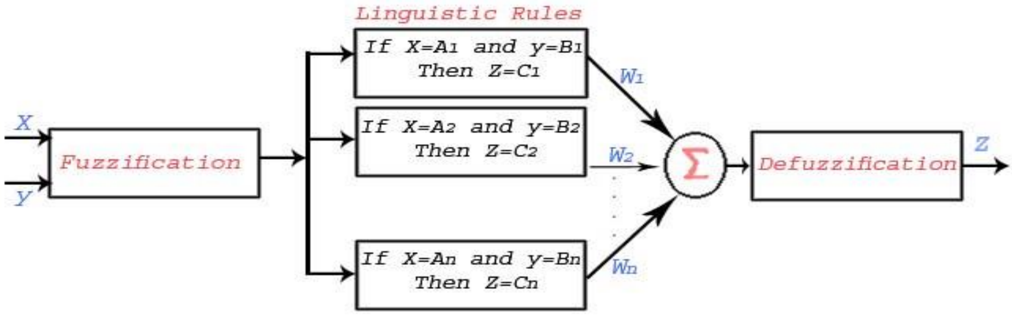


Figure 3.18: General Structure of Fuzzy Logic Control Part of The System

3.5.4. Defuzzification

The last step in the fuzzy inference process is Defuzzification. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the Defuzzification process is the aggregate output fuzzy set and the output is a single number. There are several methods for the Defuzzification, proposed in the literature. Here are some of them [2].

1) The center of gravity method(COG)

It is the best-known defuzzification operator method. A basic general defuzzication method determines the value of the abscissa of the centre of gravity of the area below the membership function in Figure (3.19)

$$COG = \frac{\sum_{x_{min}}^{x_{max}} x.\mu(x)}{\sum_{x_{min}}^{x_{max}} \mu(x)}$$

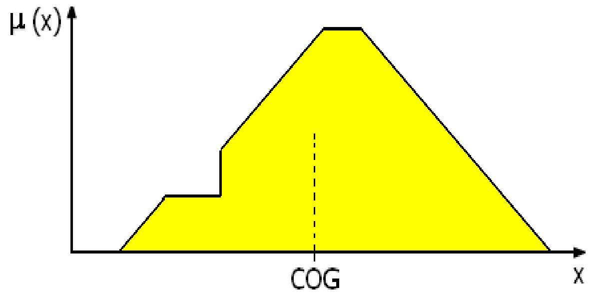


Figure 3.19: (COG) Defuzzification Method

In general, all defuzzification operators can be formulated in discrete form (via $\sum$) as well as in continuous form (via $\int$).

2) The mean of maximum method(MOM)

The mean of maxima method generates a crisp control action by averaging the support values, which their membership values reach the maximum. In the case of discrete universe:

$$MOM = \sum_{x_{min}}^{L} \frac{\mu(x)}{L} \qquad (3.14)$$

Where, L is the number of the quantized x values which reach their maximum memberships.

3) The weighted average method(WAM)

This method is used when the fuzzy control rules are the functions of their inputs. In general, the consequent part of the rule is:

$z = f(x,y)$ If W_i is the firing strength of the rule i, then the crisp value is given by:

$$WAM = \frac{\sum_{i=1}^{n} W_i \, f(x_i, y_i)}{\sum_{i=1}^{n} W_i} \qquad (3.15)$$

Where n is the number of firing rules.

3.5.5. Advantages And Disadvantages Of Two Methods

Advantages of the Mamdani Fuzzy method:

- It is intuitive and simple to build.
- It is widely used for second order systems with both linear and nonlinear characteristics.
- It has widespread acceptance.
- It is well suited to human feeling.

Disadvantages of the Mamdani Fuzzy method:

- It is only suited to the long delay system, such as the temperature control system, since it is too simple to control the process quickly.
- It needs additional device to improve the efficiency, when it controls the high frequent input system.

The advantages of Takagi-Sugeno Model:

- It is computationally efficient.
- It works well with linear techniques (e.g., PID control).
- It works well with optimization and adaptive techniques.
- It has guaranteed continuity of the output surface.
- It is well suited to mathematical analysis.
- It can optimize the parameters of the output to improve the efficiency.

Disadvantages of the Sugeno Fuzzy method

- It is not intuitive.
- When using the higher order Sugeno method, it is complex.

How to make a decision Mamdani or Sugeno?

- Mamdani method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. However, Mamdani-type fuzzy inference entails a substantial computational burden.
- On the other hand, Sugeno method is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

CHAPTER 4 GENETIC ALGORITHM

4.1. Introduction

A genetic algorithm (GA) is a procedure used to find approximate solutions to search problems through application of the principles of evolutionary biology. Genetic algorithms use biologically inspired techniques such as genetic inheritance, natural selection, mutation, and sexual reproduction (recombination, or crossover). Along with genetic programming (GP), they are one of the main classes of genetic and evolutionary computation (GEC) methodologies [32].

Genetic Algorithms are reliable and robust methods for searching solution spaces. Genetic algorithms are typically implemented using computer simulations in which an optimization problem is specified. For this problem, members of a space of candidate solutions, called *individuals* are represented using abstract representations called *chromosomes*. The GA consists of an iterative process that evolves a working set of individuals called a *population* toward an objective function, or fitness function.
Traditionally, solutions are represented using fixed length strings, especially binary strings, but alternative encodings have been developed.

The evolutionary process of a GA starts from a population of individuals randomly generated according to some probability distribution, usually uniform and updates this population in steps called generations. Each generation multiple individuals are randomly selected from the current population based upon some application of fitness, bred using crossover, and modified through mutation to form a new population.

4.2. Basic model of a genetic algorithm

We show in Figure (4.1) a basic model of a genetic algorithm:

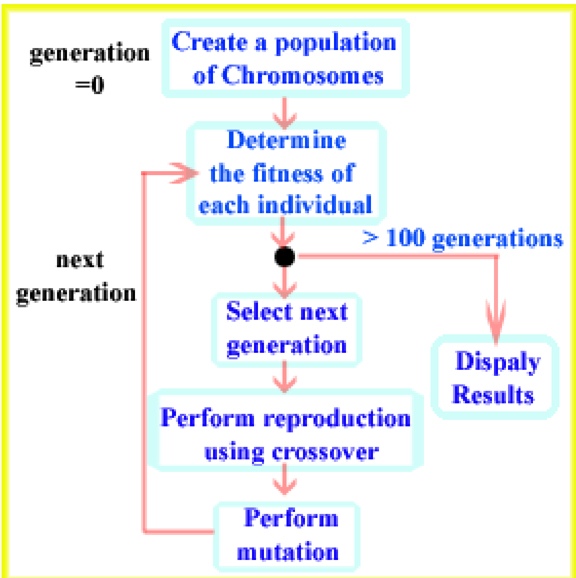


Figure 4.1: The Basic Genetic Algorithm

1- [Start] Generate random population of n chromosome; the individuals of this population represent the possible solutions.
2- [Fitness] Evaluate the fitness f(x) of each chromosome x in the population.
3- [New population] Create a new population by repeating the following steps until the new population is complete.

- [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
- [Crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
- [Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome).
- [Accepting] Place new offspring in the new population.

4- [Replace] Use new generated population for a further sum of the algorithm.
5- [Test] If the end condition is satisfied, stop, and return the best solution in current population.
6- [Loop] Go to step2 for fitness evaluation.

The human designer, which wants to solve optimization problem using GA, must address five issues [20].

1- A genetic representation of candidate solutions
2- A way to create an initial population of solutions
3- An evaluation function, which describes the quality of each individual
4- Genetic operators that generate new variants during reproduction, and
5- Values for the parameters of the GA, such as population size, number of generations and probabilities of applying genetic operators.

4.3. GAs to other optimization methods

The principle of GAs is simple because it depends on emulating genetics and natural selection by software tools. The parameters of the problem are coded most naturally as a DNA-like linear data structure, a vector or a string. Sometimes, the parameters of the problem are coded mostly as a binary code [33].

GA. processes a set called population, of these problem dependent parameter value vectors. To start there is usually a very random population, the values of different parameters generated by a random number generator. Typical population size is from few dozens to thousands. To do optimization we need a cost function or fitness function as it is usually called when genetic algorithms are used. By a fitness function, we can select the best solution candidates from the population and delete the not so good specimens [33].

The nice thing when comparing GAs to other optimization methods is that the fitness function can be nearly anything that can be evaluated by a computer or even something, that cannot! In the latter case it might be a human judgment that cannot be stated as a crisp program, like in the case of eyewitness, where a human being selects among the alternatives generated by GA [33].
Some optimization algorithms can work with discreet and continues problems, and there are algorithms work with one kind of the problems; and so with the constrained and

unconstrained. There are many applications need to work with high speed such control systems, so it needs parallel algorithm. Parallel algorithms are used to speed up the processing. GA differs from conventional optimization techniques in following ways [20]:

1. GA does not deal with data directly but works with encoded data.
2. GA uses least information such as fitness function to solve problems does not need derivation.
3. GA uses probability laws rather than certain laws.
4. GA generate populations of answer not just one answer.
5. Almost all conventional optimization techniques search from a single point but GA always operates on a whole population of points (parallelism).

4.4. Key Elements

The two distinct elements in the GA are individuals and populations. An individual is a single solution while the population is the set of individuals currently involved in the search process.

4.4.1. Individuals

An individual is a single solution. Individual groups together two forms of solutions as given below:

1. The chromosome: which is the raw ″genetic" information (genotype) that the GA deals
2. The phenotype: which is the expressive of the chromosome in the terms of the model.

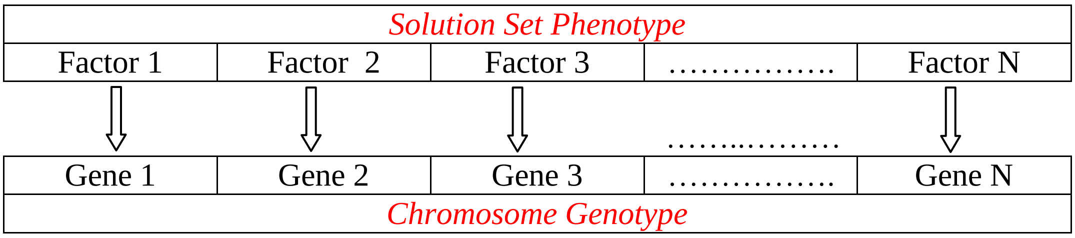


Figure 4.2: Representation of Genotype and Phenotype

A chromosome is subdivided into genes. A gene is the GA's representation of a single factor for a control factor. Each factor in the solution set corresponds to gene in the chromosome [33]. Each chromosome must define one unique solution, but it does not mean that each solution encoded by exactly one chromosome. Bit strings as shown in Figure (4.3) encode chromosomes.

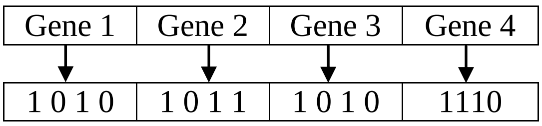


Figure 4.3 Representation of a Chromosome

4.4.2. Population

A population is a collection of individuals. A population consists of a number of individuals being tested, the phenotype parameters defining the individuals and some information about search space. The two important aspects of population used in Genetic Algorithms are:

1. The initial population generation
2. The population size

The population size will depend on the complexity of the problem. It is often a random initialization of population is carried. In the case of a binary coded chromosome this means, that each bit is initialized to a random zero or one. But there may be instances where the initialization of population is carried out with some known good solutions.

In the ideal case the first population must have, large number of individuals to cover all rang of solution space. All possible alleles of each should be present in the population. Sometimes some of the solutions expected can be used to seed the initial population. Thus, the fitness of these individuals will be high which helps the GA to find the solution faster.

Figure (4.4) shows that the population consists of a group of individuals (consists of four chromosomes) [33].

Population	*Chromosome 1*	1 1 1 0 0 0 1 0
	Chromosome 2	0 1 1 1 1 0 1 1
	Chromosome 3	1 0 1 0 1 0 1 0
	Chromosome 4	1 1 0 0 1 1 0 0

Figure 4.4: Population

4.5. Data Structures

Chromosomes, phenotypes, objective function values and fitness values are the main data structures in GA. This is particularly easy implemented when using MATLAB package as a numerical tool. An entire chromosome population can be stored in a single array given the number of individuals and the length of their genotype representation. Similarly, the design variables, or phenotypes that are obtained by applying some mapping from the chromosome representation into the design space can be stored in a single array. The actual mapping depends upon the decoding scheme used. The objective function values can be scalar or vectorial and are necessarily the same as the fitness values. Fitness values are derived from the object function using scaling or ranking function and can be stored as vectors [33].

4.6. Encoding

Encoding is a process of representing individual genes. The process can be performed using bits, numbers, trees, arrays, lists or any other objects. The encoding depends mainly on solving the problem. For example, one can encode directly real or integer numbers.

4.6.1. Binary encoding

The most common way of encoding. This code consists of binary 0 or 1 indicates chromosome gene, often used in numerical problem as in Figure (4.5).

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Figure 4.5: Binary Encoding

4.6.2. Permutation Encoding (Real-encoding)

This encode is used to represents the numbers or symbols, often used in the problems of arrangement type as in Figure (4.6)

Chromosome 1	1 5 3 2 6 4 7 9 8
Chromosome 2	8 5 6 7 2 3 1 4 9

Figure 4.6: Permutation Encoding

4.6.3. Octal encoding

This encoding uses string made up of octal numbers (0-7) as in Figure (4.7)

Chromosome 1	03467216
Chromosome 2	15723314

Figure 4.7: Octal Encoding

4.6.4. Hexadecimal Encoding

This encoding uses string made up of hexadecimal numbers (0-9, A-F) Figure (4.8).

Chromosome 1	9CE7
Chromosome 2	3DBA

Figure 4.8: Hexadecimal Encoding

4.6.5. Value Encoding

In value encoding every chromosome is a string of some values. Values can be anything connected to problem form: numbers, real numbers or chars to some complicated objects.

Value encoding is very good for some special problems. On the other hand, for this encoding is often necessary to develop some new crossover and mutation specific for the problem.

Chromosome 1	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome 2	ABDJEIFDHDIERJFDLDFLFEGT
Chromosome 3	(back), (Right), (Forward), (left)

Figure 4.9: Value Encoding

4.6.6. Tree Encoding

This encoding is mainly used for evolving program expressions for genetic programming. Every chromosome is a tree of some objects such as functions and commands of a programming language.

4.7. Fitness

In order to evaluate how good the different individuals in the population are, a fitness function needs to be defined. A fitness function assigns each chromosome a value that indicates how well that chromosome solves the given problem. A common application of genetic algorithms is optimizing a function [34].

<u>For Example</u>: In Traveling Sales Man (TSM) problem may be the time that the sales man will take it along traveling. So the fitness value can be defined as function of the objective function g(x).

$$Fitness.value = f\left(g(x)\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad (4.1)$$

For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one. When the optimization problem is single criterion, it is simple because there is one goal to achieve, when the problem is multi criterion the optimization problem will be more complex because if the solution is optimal for one criterion it may be worst for another one. The most difficult fitness functions are the ones needed to evaluate non-numerical data [32], as the developer must find other metrics or ways to find a numerical evaluation of non-numerical data.

An example of this is provided by Mitchell, who describes the problem of finding the optimal sequence of amino acids that can be folded to a desired protein structure. The acids are represented by the alphabet {A,…., Z}, and thus no numerical value can be straightforwardly calculated. The used fitness function calculates the energy needed to bend the given sequence of amino acids to the desired protein.

In control applications there are different fitness function that may be used [35].

1. $Fitness.value = \int_0^\infty e^2(t)dt$ sum of squared error … … … … … . … … . … (4.2)

Where (e) is the error signal, this function can track error quickly, but easily gives rise to oscillation.

2. $Fitness.value = \int_0^\infty |e(t)|$ sum of absolute error … … … … … . … . … … … (4.3)

This function can obtain good response, but its selection performance is not good.

3. $Fitness.value = \int_0^\infty te^2(t)dt$ sum of time weighted squared error … . . (4.4)

This function can gives fast tracking and good response.

4.8. Breeding

The breeding process is the heart of the genetic algorithm. This process creates new and hopefully fitter individuals, and consists of three steps:

1. Selecting parents.
2. Crossing the parents to create new individuals (offspring or children).
3. Replacing old individuals in the population with the new ones.

4.8.1. Selection

In this process the developer will choose the pairs of parents that will be crossed. This step is to decide how to perform selection. On other words, who are the individuals of that population will be used to create the next offspring that will be used for next generation. The purpose of this step is to stress the individuals of the population that will be selected with the higher fitness. The problem is how to select the chromosomes that will cross: there are many methods that can be used [20].

4.8.1.1Roulette Wheel Selection.

Roulette selection is one of the traditional GA selection techniques as shown in Figure (4.10). The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. A target value is set, which is a random proportion of the sum of the fitnesses in the population. The population is stepped through until the target value is reached. A fit individual will contribute more to the target value, but if it does not exceed it, the next chromosome in line has a chance, and it may be weak. It is essential that the population not be sorted by fitness, since this would dramatically bias the selection.

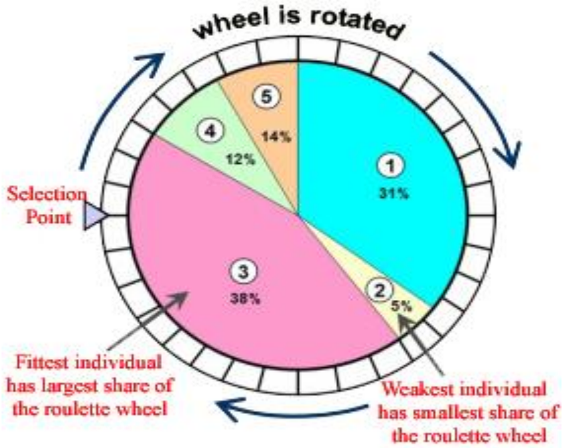


Figure 4.10: Roulette Wheel Selection.

Each individual in a population is allocated a share of a wheel; the size of the share depends on the individual's fitness. The individuals that have higher fitness have big share. The individuals that have lower fitness have small share. That means the lower fitness of the individuals may have no chance to be in the roulette. A pointer is spun (a random number generated) and the individual to which it points is selected. This continues until the requisite number of individuals has been selected [33].

Problem of roulette wheel:
The Roulette wheel will have a problem when the fitness values differ very much. If the best chromosome fitness is 90%, its circumference occupies 90% of Roulette wheel, and then other chromosomes have too few chances to be selected.

4.8.1.2 Random Selection

This technique randomly selects a parent from the population. In terms of disruption of genetic codes, random selection is a little more disruptive on average than roulette wheel selection.

4.8.1.3 Stochastic Universal Sampling

Figure (4.11) shows Stochastic Universal Sampling method in this method the individuals represent a line that is divided into number of Adjacent segments, such that each individual.s segment is equal in size to its fitness exactly as in roulette-wheel selection. Then, create equally space pointers that are placed over the line. The numbers of these pointers (*NPointer*) depends on the number of the individuals that will be selected,; the distance between the pointers is given as 1/NP, and the position of the first pointer is given by a randomly generated number in the range [0, 1/*NPointer*]. [20].

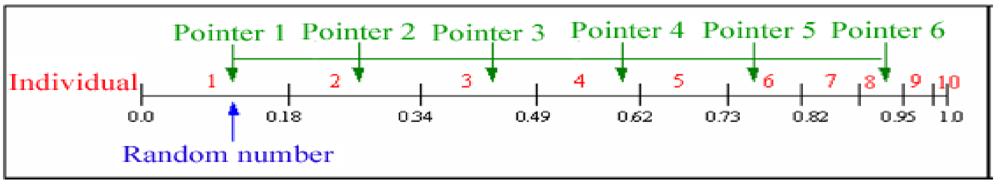


Figure 4.11: Stochastic Universal Sampling

4.8.1.4 Rank Selection

Rank Selection ranks the population and every chromosome receives fitness from the ranking. It results in slow convergence; it also keeps up selection pressure when the fitness variance is low. Here, rank selection is programmed as follow:

I. Select first pair at random.
II. Generate random number R between 0 and 1.
III. If R<r use the first individual as a parent. If the *R*>=*r* then use the second individual as the parent.
IV. Repeat to select the second parent [20].

4.8.2. Crossover (Recombination)

This process is taking about two parents solutions and producing from them a child. After the selection (reproduction) process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. Crossover is a recombination operator that proceeds in three steps [33]:

I. The reproduction operator selects at random a pair of two individual strings for the mating.
II. A cross site is selected at random along the string length.
III. Finally, the position values are swapped between the two strings following the cross site.

The simplest way how to do that is to choose randomly some crossover point and copy everything before this point from the first parent and then copy everything after the crossover point from the other parent, which discussed as follows:

4.8.2.1 Single Point Crossover

The traditional genetic algorithm uses single point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after the cuts exchanged. The cross point is selected random, that means the crossover point at two parents may be changed at other two parents in the same population as shown below[33]:

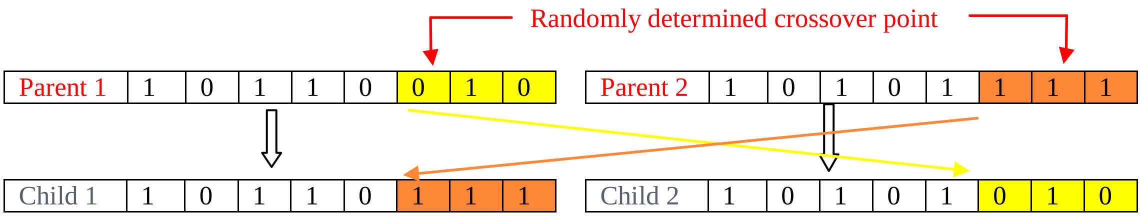


Figure 4.12: Single Point Crossover

4.8.2.2 Two-Point Crossover

That technique is similar to Single-Point Crossover, but the difference is that there are two cut points these points are selected randomly and the part that are between these points are swapped as shown in Figure (4.13).
The disadvantage of this technique is that building blocks are more likely to be disrupted. But ***the advantage*** of it is that the problem space may be searched more thoroughly [20].

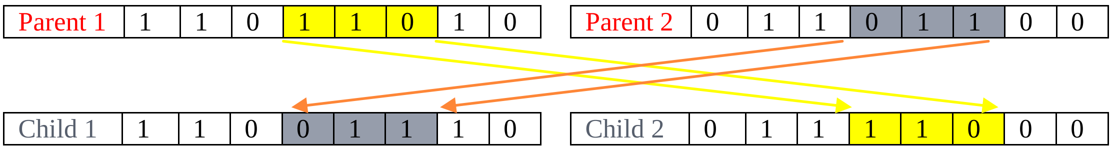


Figure 4.13: Two-Point Crossover

4.8.2.3 Uniform Crossover

Uniform crossover is another crossover technique. In this technique, the random mask is used, and this mask has same length as the chromosome. This mask consists of 1s and 0s. If a bit in the mask is 1 then the corresponding bit in the first child will come from the first parent and the second parent will contribute that bit to the second offspring. If the mask bit is 0, the first parent contributes to the second child and the second parent to the first child as shown in Figure (4.14).

Parent 1	1	0	1	1	0	0	1	1
Parent 2	0	0	0	1	1	0	1	0
Mask	1	1	0	1	0	1	1	0
Child 1	1	0	0	1	1	0	1	0
Child 2	0	0	1	1	0	0	1	1

Figure 4.14: Uniform Crossover

4.8.2.4 Multi-Point Crossover (N-Point crossover)

There are two ways in this crossover. One is even number of cross-sites and the other odd number of cross-sites. In the case of even number of cross-sites, cross-sites are selected randomly around a circle and information is exchanged. In the case of odd number of cross-sites, a different cross-point is always assumed at the string beginning [33].

4.8.3. Mutation

After crossover, the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information [33].
Mutation means swap one bit in binary coding or changes one number if the chromosome consists of numbers. For binary coding, for doing this process there is one

way choosing random number at every individual if the number less than specified number which is chose before by the programmer, then this individual will have mutation process. But how many bits will be changed there are many ways:

1. First choosing random number between 1 and the total numbers of chromosome length and swap the bits which meet that number.
2. Second method is choosing random number between 0 and 1 at every bit of the chromosome if the number less than specified number then this bit will be swapped i.e., if it is a 1 change it to 0 or vice versa.

This mutation probability is generally kept quite low and is constant throughout the lifetime of the GA. However, a variation on this basic algorithm changes the mutation probability throughout the lifetime of the algorithm, starting with a relatively high rate and steadily decreasing it as the GA progresses. This allows the GA to search more for potential solutions at the outset and to settle down more as it approaches convergence.

4.8.4. Replacement

Replacement is the last stage of any breeding cycle. Two parents are drawn from a fixed size population they breed two children, but not all four can return to the population, so two must be replaced. i.e., once offsprings are produced, a method must determine which of the current members of the population if any should be replaced by the new solutions. The technique used to decide which individual stay in a population and which are replaced in on a par with the selection in influencing convergence. Basically, there are two kinds of methods for maintaining the population:

1. Generational updates.
2. Steady state updates.

4.9. Elitism

With crossover and mutation taking place, there is probability that the best solution may be lost as there is no guarantee that these operations will preserve fitness. To combat this elitist models are often used. In these methods, the best individual from a population is saved before any of the operations take place. After the new population is formed and evaluated, it is examined to see if this best structure has been preserved. If not, the saved copy is reinserted back into the population, usually this individuals is reinserted instead of the weakest individual. The GA then proceeds to perform the operations on this population [20].

4.10.Genetic Fuzzy Systems

Fuzzy System (FS) is any fuzzy logic-based system where fuzzy logic can be used either as the basis for the representation of different forms of system knowledge or to model the interactions and relationships among the system variables. FSs have proven to be an important tool for modeling complex systems in which, due to complexity or imprecision, classical tools are unsuccessful.

The fuzzy controller of any system can be found by try and error. Try and error method is not simple in fuzzy control and may take very long time, because there are many parameters have an effect of the fuzzy controller, such as membership shape, rules (number of rules or architecture of rules), inputs gains and outputs gains, and the interval of the membership of the inputs and outputs; so in the recent years Genetic Fuzzy Systems (GFS) are used to solve these problems, and the Hybridization between GA that is one of methods of Evolutionary Algorithms (EA) and FLC so called soft computing. Figure (4.15) shows a family of computing techniques. There are so many ways of how to use GA in fuzzy control. The most extended GFS type is the genetic fuzzy rule-based system (GFRBS), where an EA is employed to learn or tune different components of an FRBS. The objective of a genetic tuning process is to adapt a given fuzzy rule set such that the resulting FRBS demonstrates better performance. The following components of the knowledge base (KB) are potential candidates for optimization [20]:

1. Data base (DB) components: scaling functions and membership function parameters.
2. Rule base (RB) components: "IF-THEN" rule consequents.

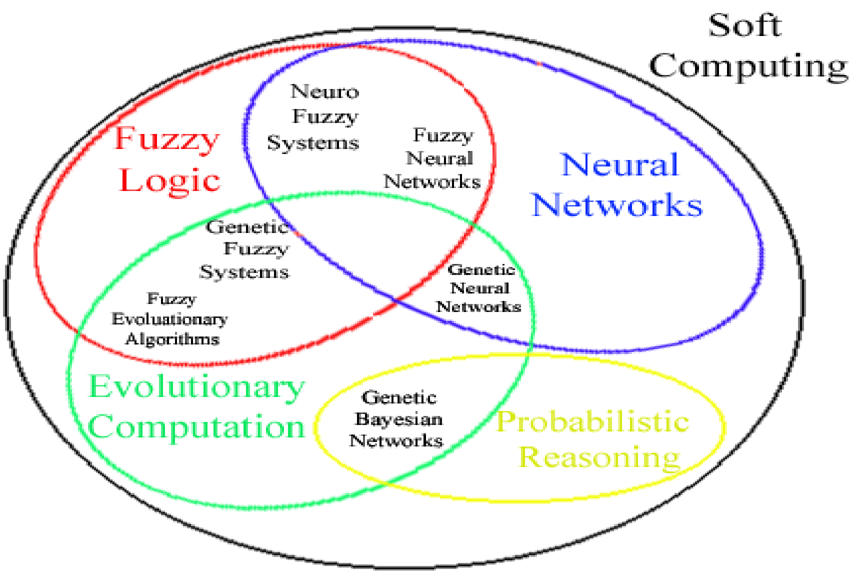


Figure 4.15: Hybridization in Soft Computing

CHAPTER 5 DESIGING SUN TRACKER SYSTEM USING SUGENO FLC

5.1. Sun Tracker and MPPT Systems:

We show in the Figure (5.1) the concept of this system. The system has two controllers, one to control the PV panel position to be aligned to the sun all the day, and another controller for tracking the maximum power point, that to increase the efficiency of PV panel.

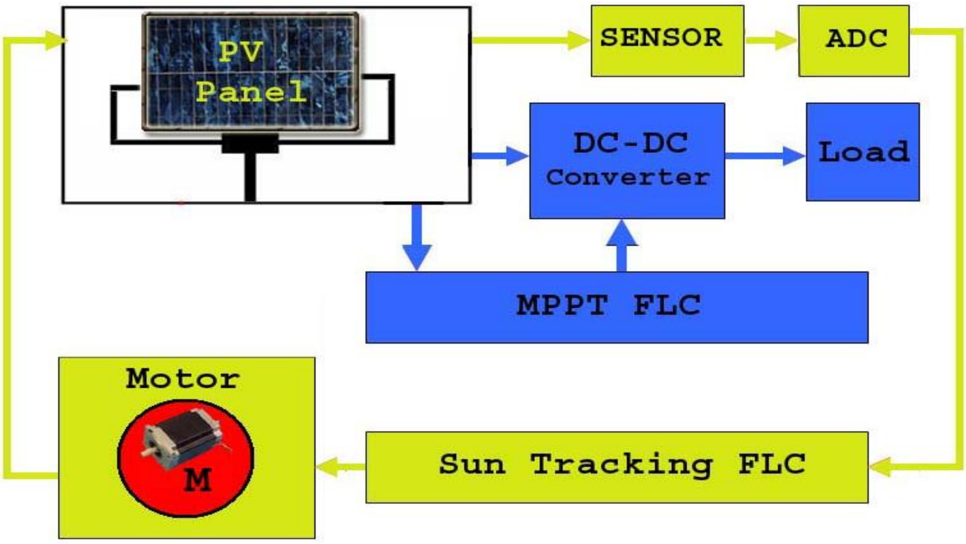


Figure 5.1: Block Diagram For Control System

This chapter presents the process used to design Sun Tracker fuzzy controller for the solar energy system using Try and error method.

5.2. Sun Tracker system Design

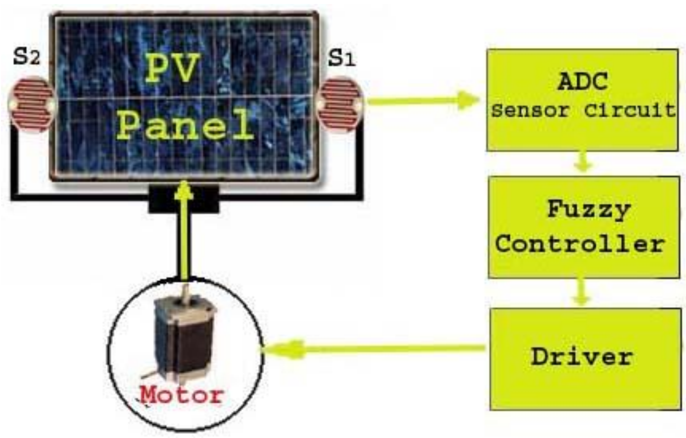


Figure 5.2: Block Diagram For Sun Tracker System

Solar tracking system uses a stepper motor as the drive source to rotate the solar panel. The position of the sun is determined by using a tracking sensor, the reading of sensor is converted from analog to digital signal, and then it passed to a fuzzy logic controller.

The output of controller is connected to the drivers of the motor to rotate PV Panel until it is toward the sun, as shown in the Figure (5.2).

5.2.1. Photovoltaic Panel (PV Panel)

Solar panels are made up of photovoltaic cells, it means the direct conversion of sunlight to electricity by using a semiconductor usually made of silicon. Photovoltaic systems employ semiconductor cells (wafers), generally several square centimeters in size. Semiconductors have four electrons in the outer shell, or orbit, on average. These electrons are called valence electrons.

When the sunlight hits the photovoltaic cells, part of the energy is absorbed into the semiconductor. When that happens the energy loosens the electrons which allow them to flow freely. The flow of these electrons is a current and when you put metal on the top and bottom of the photovoltaic cells, we can draw that current to use it externally. When many such cells are connected in series we get a solar PV module, the current rating of the modules depends on the area of the individual cells. For obtaining higher power output the solar PV modules are connected in series and parallel combinations forming solar PV arrays.

5.2.2. Sensors

5.2.2.1 Photo Sensor

Photoresistors or Light Dependent Resistors (LDR), which change resistance according to light intensity. LDR is used to construct the sensor. The biggest size is used to construct the sensor because the more area of the sensor mean more its sensitivity or less time taken for output to change when input changes. Shown in Figure (5.3)

Figure 5.3: LDR Sensor

5.2.2.2 The Orientation Principle Of Sensor

The tracking principle is based on the input data referring to the position of the Sun on the sky dome. For the highest efficiency, the sunrays have to fall normal on the receiver (i.e. the photovoltaic module) [35]. Therefore, the system must periodically modify its position in order to maintain this relation between the sunrays and the panel. The positions of the Sun on its path represent an input data in designing the tracking system, there are two fundamental ways to track the Sun, by one axis and by two axes, which are located at the east, west, or south, and north to detect the light source intensity.

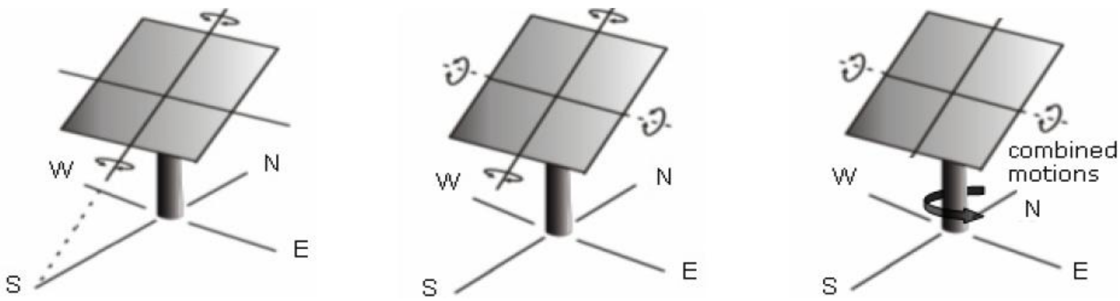


Figure 5.4: Basic Types of Tracking Systems

Two LDR sensors are mounted on the solar panel and placed in an enclosure. It has a response which is similar to the human eye. The east and west LDR sensors compare the intensity of received light in the east and west.

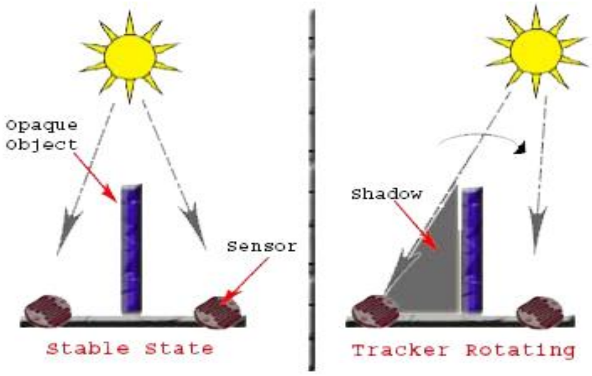


Figure 5.5: How Sensor Work

The right side of the Figure above describes the case when Sun's position shifts, here the light source intensity received by the sensors are different, the system obtains signals from the sensors' output voltage in the two orientations. The system then determines which sensor received more intensive light based on the sensor output voltage value interpreted by voltage type A/D converter. The system drives the step motor towards the orientation of this sensor. If the output values of the two sensors are equal, as the left side of Figure (5.5), the output difference is zero and the motor's drive voltage is zero, which means the system has tracked the current position of the sun [2].

5.2.2.3 How connected LDR

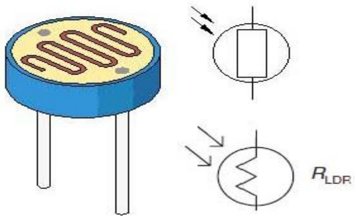


Figure 5.6: The Circuit Diagram Symbol

A typical light dependent resistor (LDR) is pictured above together with (on the right hand side) its circuit diagram symbol.

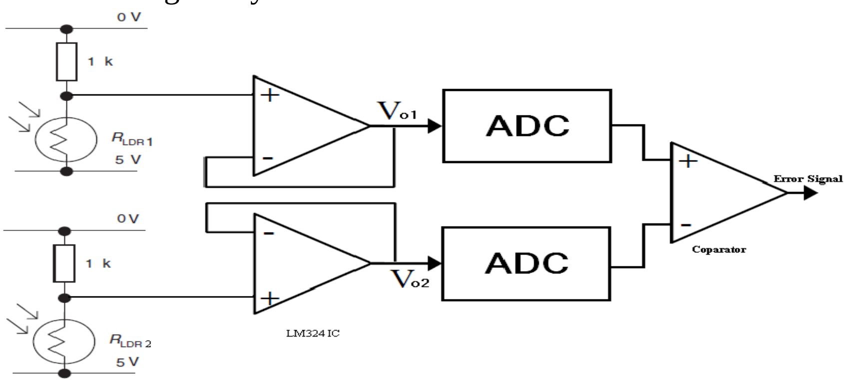


Figure 5.7: LDR Connection

The basic circuit for connecting an LDR is shown in the Figure (5.7). It can be connected in a simple potential divider to give a voltage output buffered by a unity gain amplifier.

5.2.3. Stepper motor driver

In solar tracking application we must select the motor that can carry photovoltaic panel towards the sun, any type of motor can used.

Using induction motor for example is a choice but we will use gear box and complex control circuit to control induction motor.

For that, we will use a stepper dc motor since the application discrete mechanical movements so we prefer using stepper motor.

A stepper motor is an electromechanical device, which converts electrical pulses into discrete mechanical movements. A stepper motor (or step motor) is a brushless DC electric motor that divides a full rotation into a number of equal steps.

The motor's position can be controlled precisely without any feedback mechanism (an open-loop controller), as long as the motor is carefully sized to the application. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical pulses are applied to it in the proper sequence.

5.2.3.1 How Stepper Motors Work?

Stepper motors are formed by coils and magnets and incorporate a shaft that moves when power is applied.

Stepper motors convert digital information into proportional mechanical movement. They are digital and different from DC motors that are controlled by changing the current across them. The electromagnets of a stepper motor are energized by an external control circuit, such as a microcontroller.

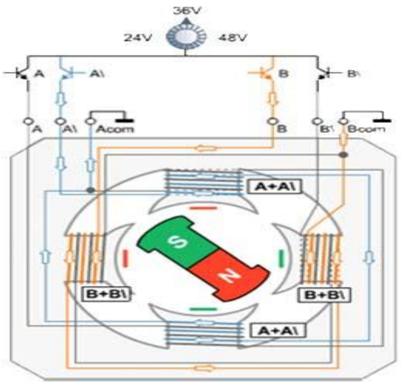


Figure 5.8: Stepper Motor Animation

To make the motor shaft turn, first, one electromagnet is given power, which makes the gear's teeth magnetically attracted to the electromagnet's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. Thus, when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one, and from there the process is repeated as shown in Figure (5.8).

Table 5.1: Step Operation

Stepmode								
F	0		1		2		3	
H	0	1	2	3	4	5	6	7
A	1	0	0	0	0	0	1	1
B	1	1	1	0	0	0	0	0
A\	0	0	1	1	1	0	0	0
B\	0	0	0	0	1	1	1	0
dez	12	4	6	2	3	1	9	8

Each of those slight rotations is called a "step", with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle as shown in table (5.1).

5.2.3.2 Stepper Motors Driver

Stepper motors can be divided into two classes according to the type of winding as bipolar and unipolar steppers.
Unipolar control is the most simple and cost-effective way to drive a stepper motor, but results in approximately 30% less torque in comparison to the nowadays widely used bipolar drivers, since the cost advantage is very small today due to cheap integrated circuits.
Bipolar drives are now used in most new applications. So it is used in the sun tracker system, to rotate the PV panel to face the sun. It has a 1.8-degree for each step, and four wires connect for coils as shown in Figure (5.9).

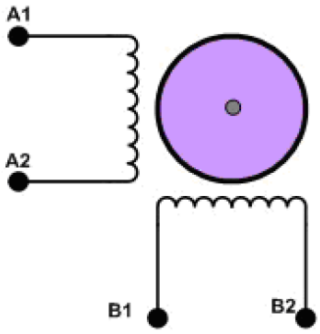


Figure 5.9: Bipolar Stepper Motor

Stepper motors require more power than other components in the circuit, so they are connected with a separate power supply. The voltage is applied to each of the coils in a sequence as shown in table (5.2) to control the stepper [2].

Table 5.2: Full-Step Phase Sequence

Step	A1	A2	B1	B2
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

When the sequence is applied as steps from 1 to 4, the motor will rotate in clockwise direction, and rotate in the other direction if the steps are in reverse order.

5.2.4. Analog-To-Digital Converter (ADC)

An analog-to-digital converter (abbreviated ADC, A/D or A to D) is a device that converts the input continuous physical quantity to a digital number that represents the quantity's amplitude. The conversion involves quantization of the input, so it introduces a small amount of error. The result is a sequence of digital values that have converted a

continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.
An ADC may also provide an isolated measurement such as an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. However, some non-electronic or only partially electronic devices, such as rotary encoders, can also be considered ADCs.

Figure (5.10) shows the ADC conversion characteristic, where the input voltage is represented on the horizontal axis and digital output on the vertical.

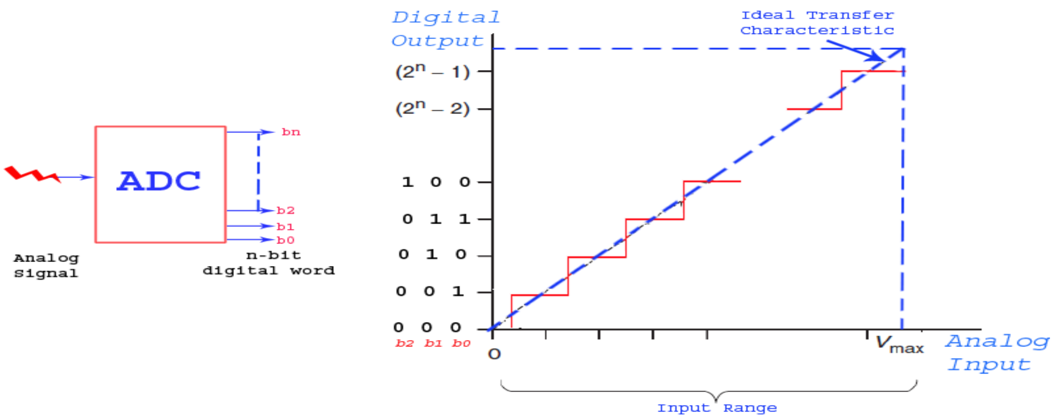


Figure 5.10: The Ideal ADC input\output Characteristic

For an n-bit ADC, the maximum output value will be $(2^n - 1)$. For an 8-bit ADC, the final value will be $(2^8 - 1)$, or 11111111(binary), or 255(decimal). The input range starts from zero and goes up to the value *Vmax* =5V. The horizontal axis is divided into exactly 2^n equal segments, each centered on an output transition. It can be seen intuitively from the diagram that the more the number of output bits, the more will be the number of output steps and the increasing in the resolution.

5.2.5. Fuzzy Logic Controller for Sun Tracker

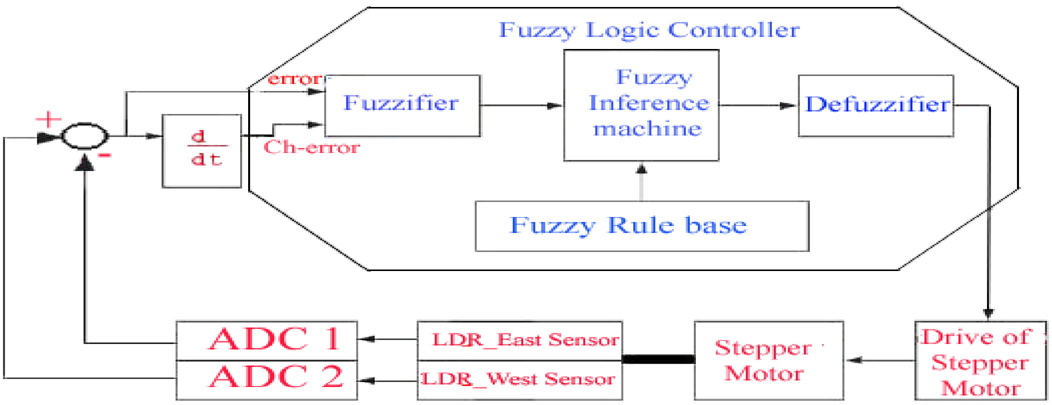


Figure 5.11: FLC Controller for The Sun Tracker System

FLC has been constructed and the block diagram in Figure (5.11) shows the FLC for the sun tracker system.
FLC has two inputs which are: error and the change in error, and one output feeding to the stepper motor driver. The phasor plot in Figure (5.12) explained a method used in

reaching the desired degree value at the equilibrium point to satisfy the stability in the system. For example, at stage A the error is positive (desired degree –actual degree) and the change error (error – last error) is negative which means that the response is going in the right direction; hence, the FLC will go forward in this direction. Using the same criteria at stage B, the error is negative and CE is bigger negative; hence, the response is going in wrong direction so FLC will change its direction to enter Stage C, until reaching the desired degree.

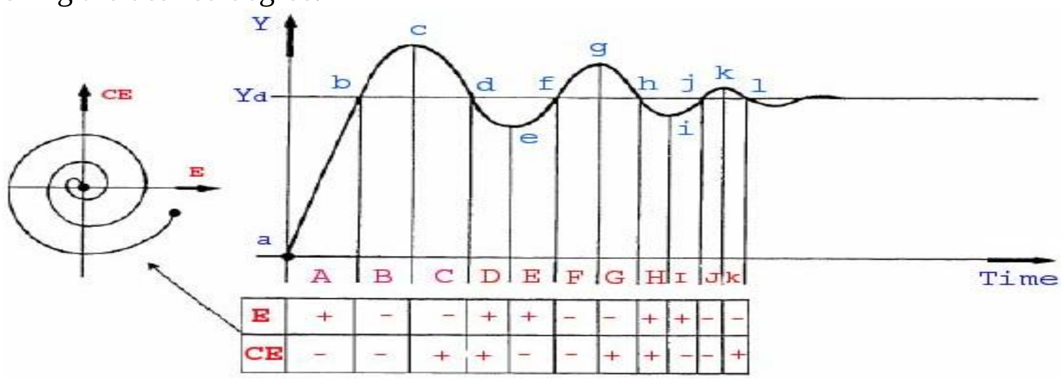


Figure 5.12: Error and Change in Error Approach in FLC

In this thesis, Sugen approach has been used to implement FLC for the sun tracker. FLC contains three basic parts: Fuzzification, Base rule, and Defuzzification.

1- Fuzzification

The fuzzy controller of the system uses Sugeno model. The FLC has two inputs which are error and change of error and the output is the Speed. Figure (5.13- a,b and c) shows the membership functions of fuzzy controller using Fuzzy Toolbox of Matlab software. All have 7 membership functions.

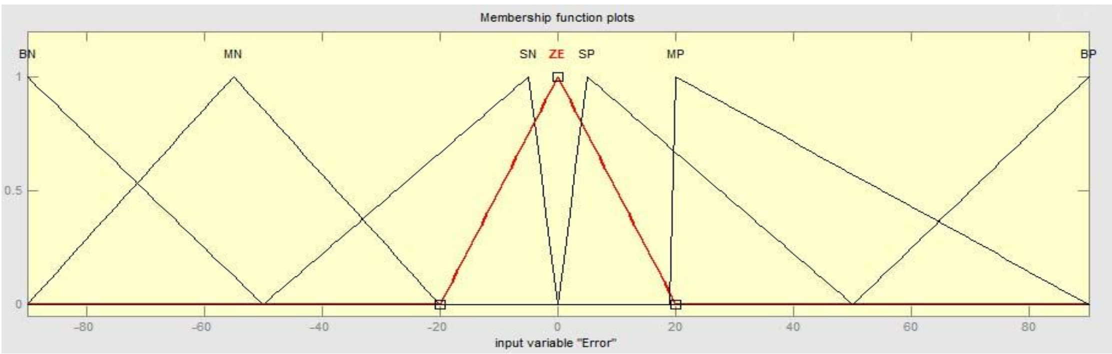


Figure 5.13-a: Membership Function of Input Error e

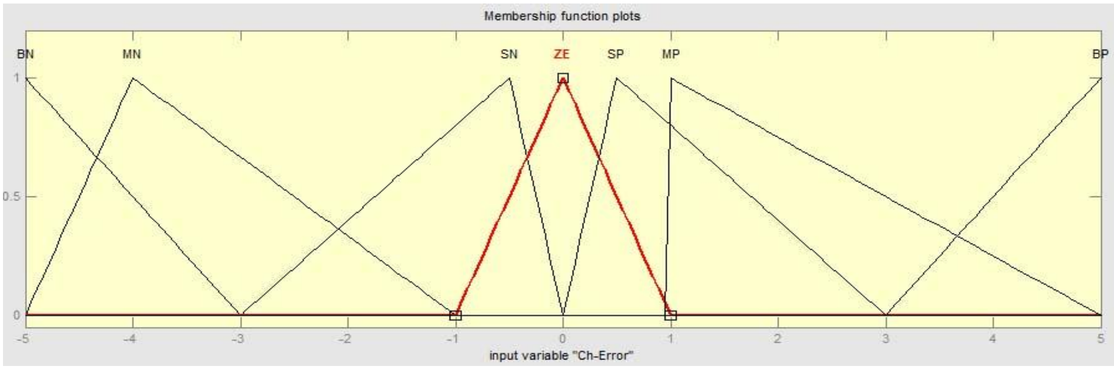


Figure 5.13-b: Membership Function of Input Error Δe

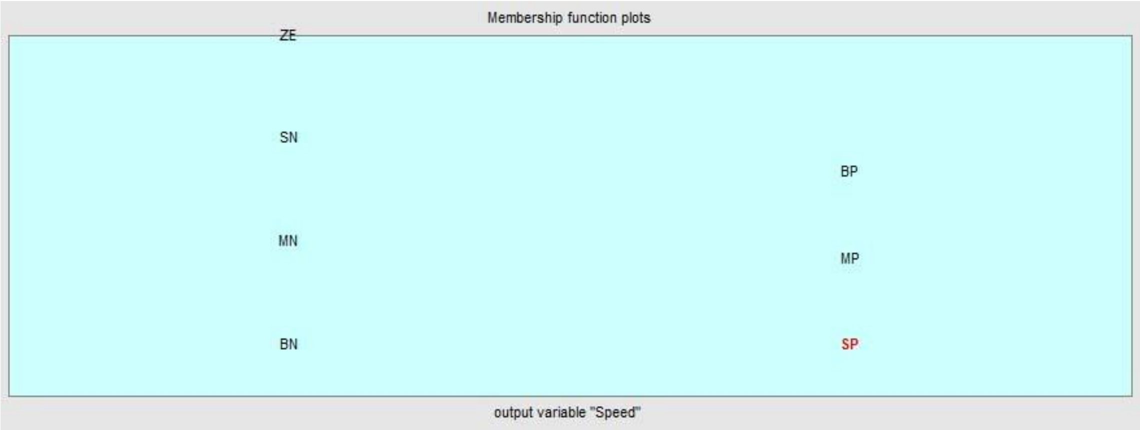


Figure 5.13-c: Membership Function of Output

A typical rule in a Sugeno fuzzy model has the form

- If Input 1= x and Input 2 = y, then Output is $z = ax + by + c$

That's mean If (error) =SP and (Change in error)=MN then output=SN as shown in appendix C-I.

We will show some examples of the output rule which represented as:

a) SN= Parameters [0 70 0.3333]

This equation means:

SN=$0*x+70*y+c = 0*SP+70*MN+0.3333$

b) BN= Parameters [-10 -10 10]

BN= $-10*ZE + -10*BN+10$

2- Base Rule

The knowledge base defining the rules for the desired relationship between the input and output variables in terms of the membership functions illustrated in table (5.4). The control rules are evaluated by an inference mechanism.

Table 5.3: The Linguistic Variables

BN	Big Negative
MN	Medium Negative
SN	Small Negative
ZE	Zero
SP	Small Positive
MP	Medium Positive
BP	Big Positive

FAM table for a two-input (Error and Change in Error), single-output (Speed) fuzzy rule-based system for fuzzy controller as shown in table (5.4):

Table 5.4: Fuzzy Control Rule Base Of Sun Tracker System**.**

Er \ Ch-Er	BN	MN	SN	ZE	SP	MP	BP
BN	BN	BN	BN	BN	MN	SN	ZE
MN	BN	BN	MN	MN	SN	ZE	SP
SN	BN	MN	SN	SN	ZE	SP	MP
ZE	BN	MN	SN	ZE	SP	MP	BP
SP	MN	SN	ZE	SP	SP	MP	BP
MP	SN	ZE	SP	MP	MP	BP	BP
BP	ZE	SP	MP	BP	BP	BP	BP

I have 49 rules in this system as shown in appendix (B-I), but in this paragraph we will show some examples of these rules which represented as a set of:

IF Error is ... and Change of Error is ... THEN the output will ….

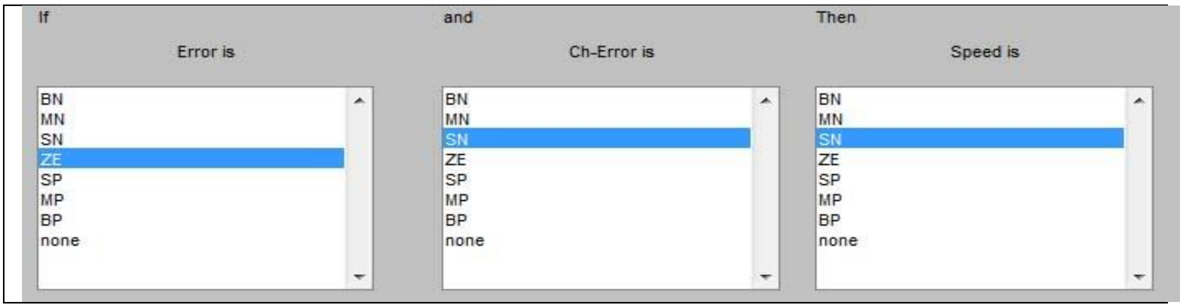


Figure 5.14: General Rule Base.

1. If (Error is BN) and (Ch-Error is BN) then (Speed is BN).
2. If (Error is MN) and (Ch-Error is BN) then (Speed is BN).
3. If (Error is SN) and (Ch-Error is BN) then (Speed is BN).

Figure (5.15) shown the surface of the base rules using in FLC.

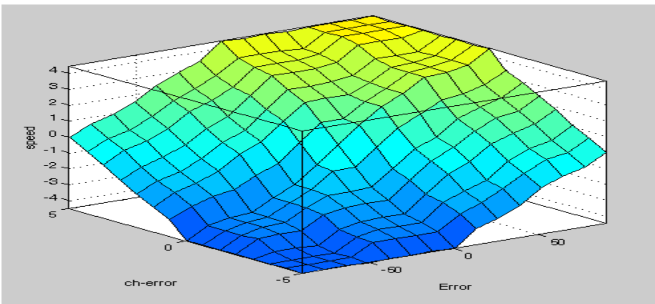


Figure 5.15:The Surface of Fuzzy Controller

3- Defuzzification

Defuzzification method is the final stage of the fuzzy logic control. After the inference mechanism is finished, the defuzzification method converts the resulting fuzzy set into crisp values that can be sent to the plant as a control signal.

The most common method is used the COA method because of the simplicity of implementation

The weighted average of the membership function using COA is written as follows:

$$u = \frac{\sum_{i=1}^{m} \mu(x_i).x_i}{\sum_{i=1}^{m} \mu(x_i)}$$

5.3. Simulation on Matlab\ Simulink For Sun Tracker

Figure (5.16) illustrates the Simulink block diagram for the fuzzy controller for sun tracker system.

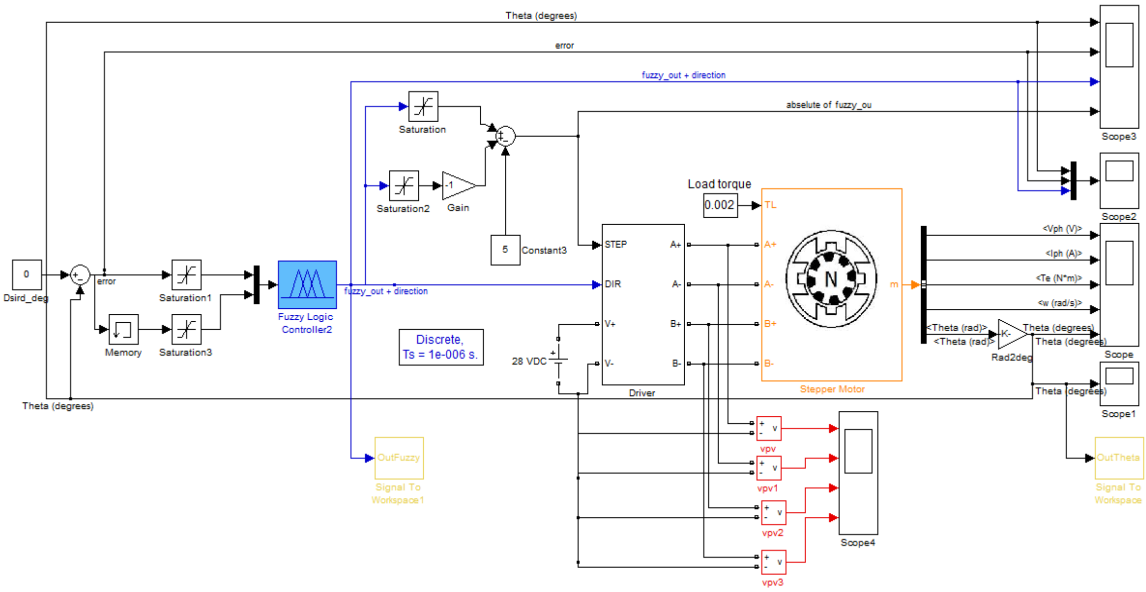


Figure 5.16: Testing The FLC in The Sun Tracker System using Matlab/Simulink.

The controller has been tested using Simulink motor module in MATLAB, by applying the step input and initial degree of the rotor is -10 degree. The output step response is shown in Figures (5.17). The range from -10 to 0 degree takes 5 steps since each step in our motor is 1.8 degree, so (10/1.8)= 5 steps.

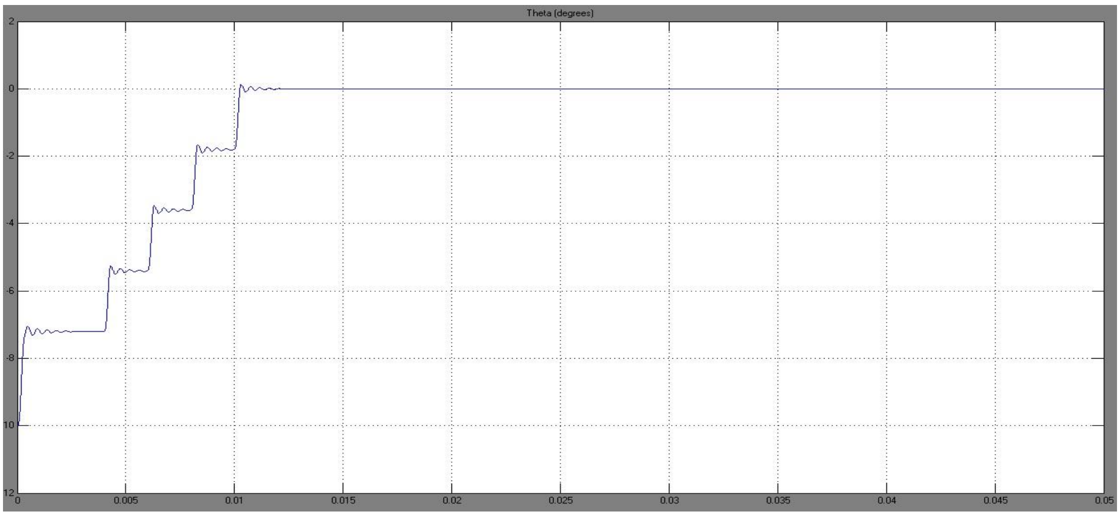


Figure 5.17: Output Degree.

Figure (5.18) is a zoom for one motor step, the overshoot is 1.18%, error steady state is 0.0023 degree, and the settling time is 0.0126 second.

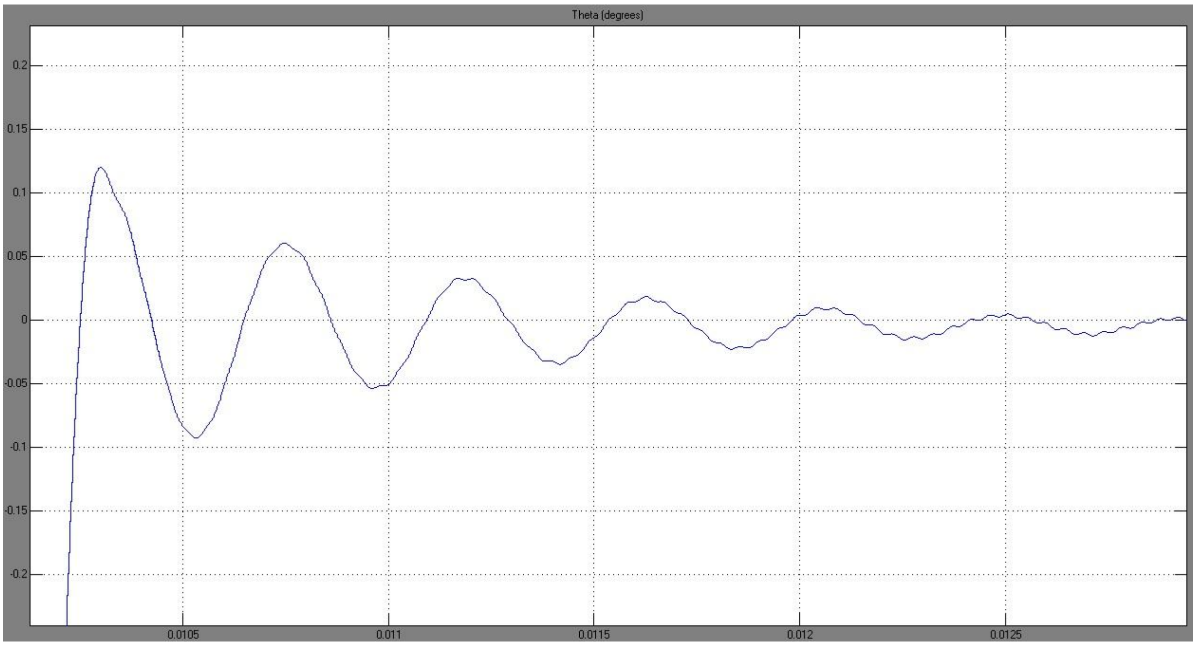


Figure 5.18: a Zoom For One Motor Step.

5.4. Comparison Between Sugeno And Mamdani FLC:

The results of applying the Sugeno FLC to control the PV panel position to be alignment to the sun all day is compared with a Mamdani controller applied on the same system by EL-Moghany [2].
Figure (5.19) shows the effect of the two controllers:

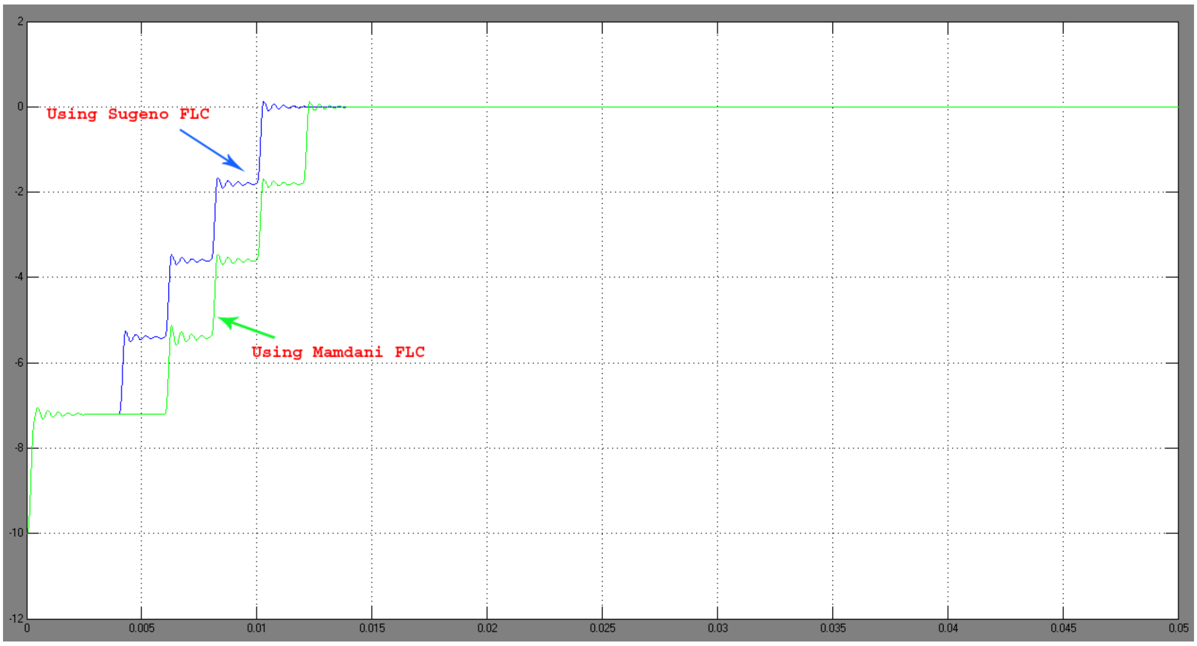


Figure 5.19: The Output Degree of Two Controllers.

Figure (5.20) is a zoom for one motor step between two Mamdani and Sugeno controllers:

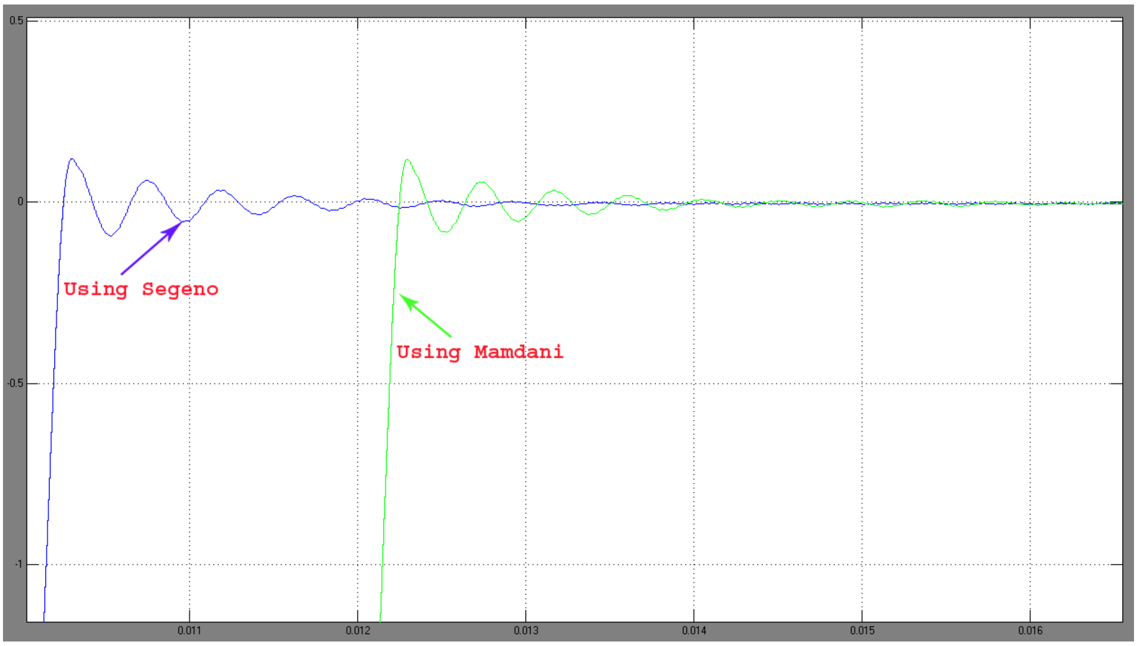


Figure 5.20: A Zoom For One Motor Step Between Two Controllers.

As shown Figure (5.20) we obtain the best result than mamdanias :

	Mamdani	Sugeno
Settling Time	16ms	12.6ms
Overshoot	1.3%	1.18%
Error	0.005	0. 0023

CHAPTER 6 DESINING MPPT SYSTEM USING SUGENO FLC AND GA

6.1. Designing all system

This chapter presents the process used to design MPPT fuzzy controllers for the solar energy system using Try and error method and using Genetic Algorithm to increase the efficiency of PV panel.

6.2. MPPT Designing

Maximum power point tracking system uses dc-to-dc converter to compensate the output voltage of the solar panel to keep the voltage at the value, which maximizes the output power. MPP fuzzy logic controller measures the values of the voltage and current at the output of the solar panel, then calculates the power from the relation (P=V*I) to extract the inputs of the controller. The crisp output of the controller represents the duty cycle of the pulse width modulation to switch the dc-to-dc converter. Figure (6.1) shows the Maximum power point tracker (MPPT) system as a block diagram [2].

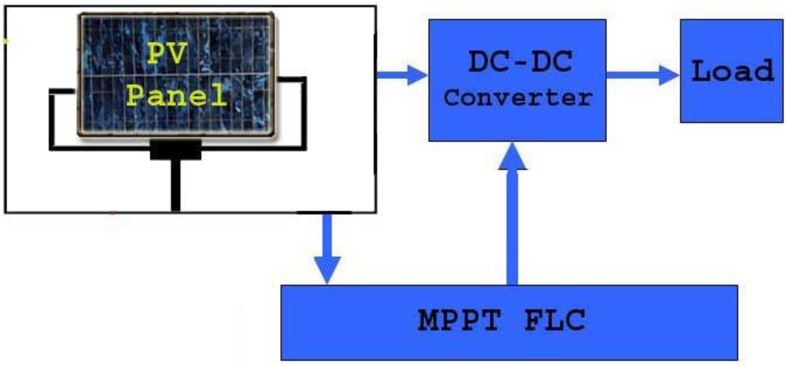


Figure 6.1: Block Diagram for MPPT System

6.2.1. DC-DC Converter (Buck Converter)

The operation of converter is fairly simple with an inductor and two switches (usually a transistor and a diode) and a capacitor that control the current of the inductor. It alternates between connecting the inductor to source voltage to store energy in the inductor and discharging the inductor into the load as shown in Figure (6.2).

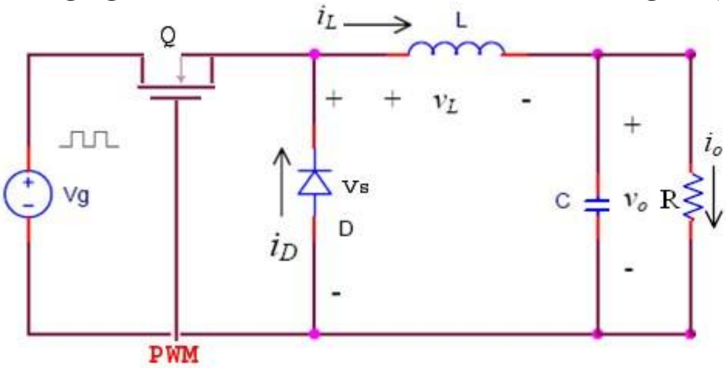


Figure 6.2: Buck Converter

6.2.2. PWM (Pulse Width Modulation

PWM signals are pulse trains with fixed frequency and magnitude and variable pulse width. However, the width of the pulses (duty cycle) changes from pulse to pulse according to a modulating signal as illustrated in Figure (6.3). When a PWM signal is applied to the gate of a power transistor, it causes the turn on and turns off intervals of the transistor to change from one PWM period to another according to the same modulating signal [2].

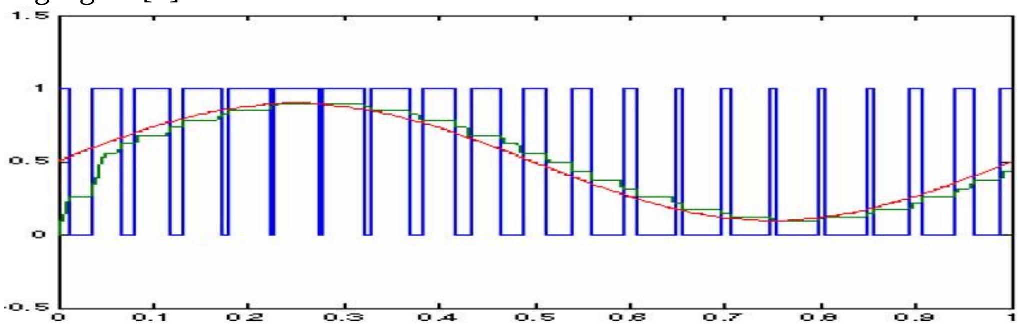


Figure 6.3: Pulse Width Modulation Waveforms

6.2.3. Fuzzy Logic Controller For MPPT

The FLC examines the output PV power at each sample (time_k), and determines the change in power relative to voltage (dp/dv). If this value is greater than zero the controller change the duty cycle of the pulse width modulation (PWM) to increase the voltage until the power is maximum or the value (dp/dv) =0, if this value less than zero the controller changes the duty cycle of the PWM to decrease the voltage until the power is maximum as shown in Figure (6.4) [2].

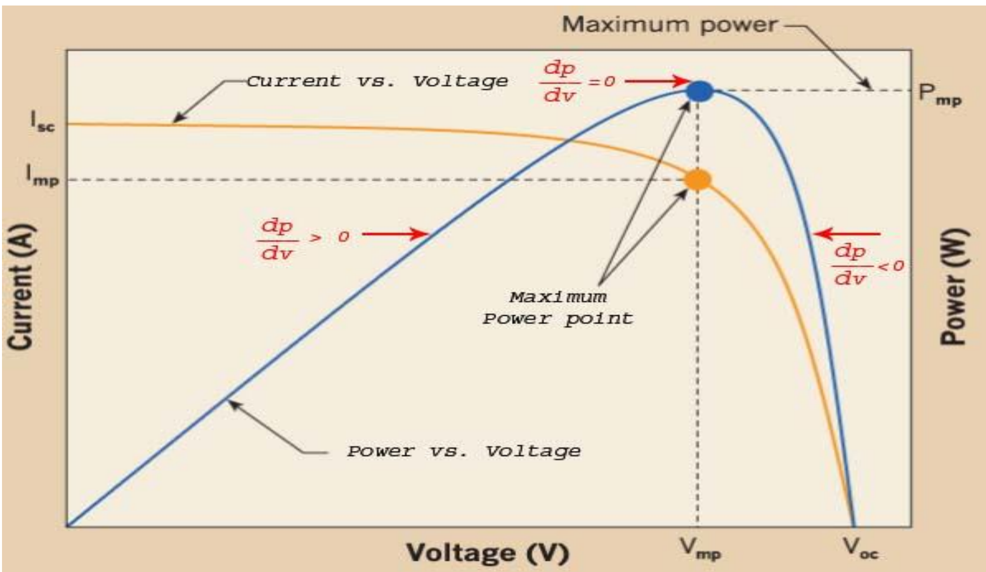


Figure 6.4: Power – Voltage Characteristic of a PV Module

FLC has two inputs which are: error and the change in error, and one output feeding to the pulse width modulation to control the DC-to-DC converter. The two FLC input variables error "e" and change of error "Δe" at sampled times "k" defined by:

$$e\,(k) = \frac{p(k) - p(k-1)}{v(k) - v(k-1)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.1)$$

$$\Delta e\ (k) = \ e(k) - e(k-1) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . \ldots (6.2)$$

Where *P (k)* is the instant power of the photovoltaic generator. The input ″error (k)" shows if the load operation point at the instant k is located on the left or on the right of the maximum power point on the PV characteristic, while the input ″CE (k)" expresses the moving direction of this point [2].

1. Fuzzification

Fuzzification is the first block inside the controller, which scale the input crisp value e and Δe into normalized universe of discourse U then converts each crisp input to degrees of membership function $\mu_A(u)$.

Figure (6.5 -1, 2 and 3) shows the domain, of the error ″e", change in error″Δe", and the output duty ratio ″d" which are covered by seven fuzzy set variables: BN (Big Negative), MN (Medium Negative), SN (Small Negative), ZE (Zero), SP (Small Positive), MP (Medium Positive), and BP (Big Positive).

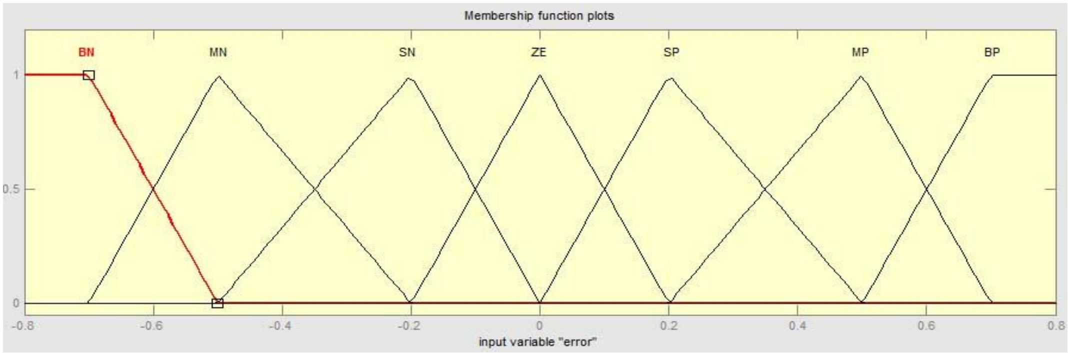


Figure 6.5 – 1: Membership Function of Input Error e

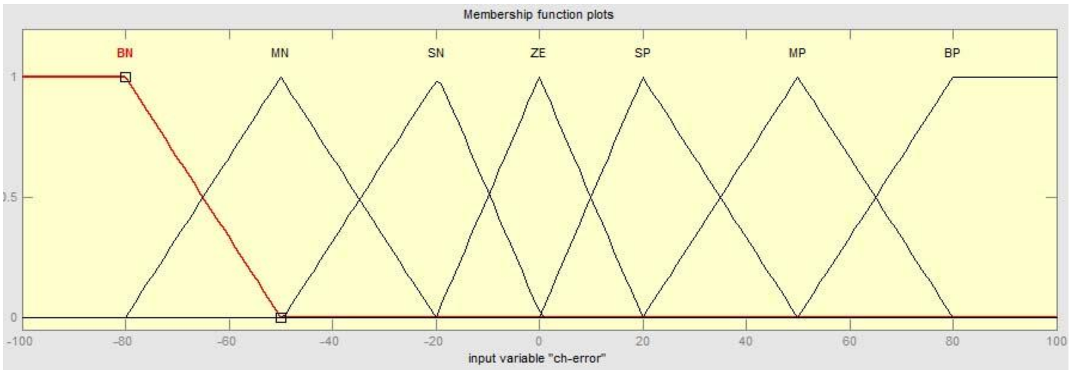


Figure 6.5 – 2: Membership Function of Input Error Δe

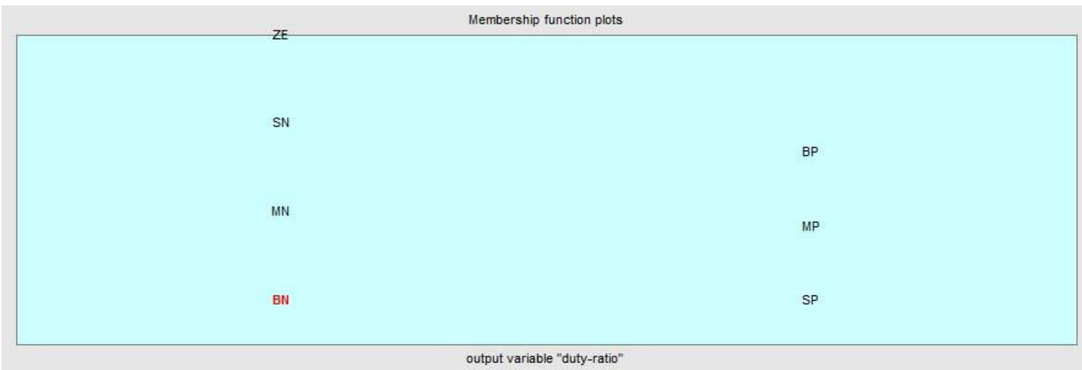


Figure 6.5 – 3: Membership Function of Output of Duty Ratio d

If (error) =MP and (Change in error)= MN then output=SN as shown in appendix C-II.
We will show some examples of the output rule which represented as:

a- MP= Parameters [0.2 0.5 0.7]
This equation means:
MP=*0.2*x+0.5*y +c = 0.2*BN +0.5*ZE +0.7*

b- BN= Parameters [0 -0.7 -0.5]
BN= *-0*SP + -0.7*BN-0.5*

2. Base Rule

Rule base is the core of the FLC; it is a set of rules in the form of IF-THEN statement that describe the state and the behavior of the control system.
Fuzzy control rules express the relation between inputs; it may use several variables usually in the form of conditional statements that have the following form:

$$IF\ e\ is\ A\ \ AND/OR\ \Delta e\ is\ B\ THEN\ d_{ij} = k_i d_i + k_j d_j$$

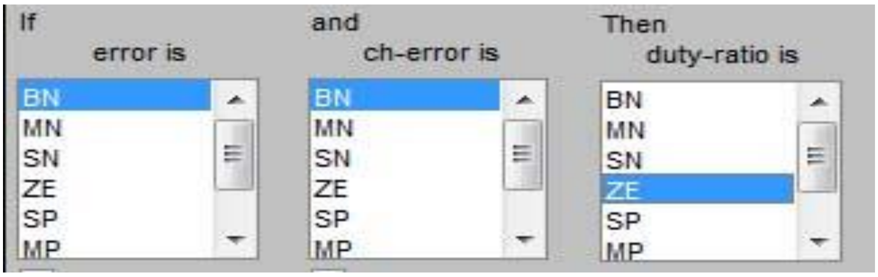


Figure 6.6: General Rule Base

I have 49 rules in this system as shown in appendix (A-II), but in this paragraph we will show some examples of these rules which represented as a set of:

1. If (Error is BN) and (Ch-Error is BN) then (Speed is ZE).
2. If (Error is SN) and (Ch-Error is BN) then (Speed is SN).

The required rules used to control the desired system are shown in above rules and shown in table (6.1).

Table 6.1: Fuzzy Control Rule Base Of MPPT System**.**

Δe \ e	***BN***	***MN***	***SN***	***ZE***	***SP***	***MP***	***BP***
BN	ZE	ZE	ZE	BN	BN	BN	BN
MN	ZE	ZE	ZE	MN	MN	MN	MN
SN	SN	ZE	ZE	SN	SN	SN	SN
ZE	MN	SN	ZE	ZE	ZE	SP	MP
SP	MP	SP	SP	SP	ZE	ZE	SP
MP	MP	MP	MP	MP	ZE	ZE	ZE
BP	BP	BP	BP	BP	ZE	ZE	ZE

Figure (6.7) shown the surface of the base rules using in FLC.

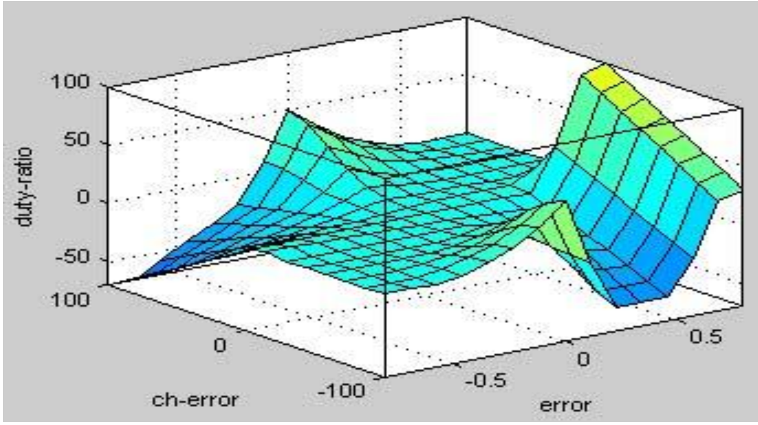


Figure 6.7: The Surface of Fuzzy Controller

3. Defuzzification

As illustrated in the previous section (5.2.5) and chapter 2, the weighted average method is widely used in Sugeno approach which has been selected in this thesis
The weighted average of the membership function using COA to compute the output of FLC which is the duty cycle is written as follows:

$$duty\ cycle = \frac{\sum_{i=1}^{m} \mu(d_j) * k_i}{\sum_{j=1}^{n} \mu(d_j)} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (6.3)$$

6.2.4. Simulation on Matlab\ Simulink For MPPT

6.2.4.1 Modeling Of PV Devices

The basic equation from the theory of semiconductors that mathematically describes the *I–V* characteristic of the ideal PV cell is

$$I = I_{pv,cell} - I_d$$

$$I = I_{pv,cell} - I_{0,cell}\left[e^{qV/akT} - 1\right] \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (6.4)$$

Where

$I_{pv,cell}$	The current generated by the incident light OR the photovoltaic current
I_d	The Shockley diode equation
$I_{0,cell}$	The reverse saturation or leakage current of the diode
q	The electron charge = ($1.60217646 \times 10^{-19}$ C)
k	The Boltzmann constant = ($1.3806503 \times 10^{-23}$ J/K)
T	The temperature of the p–n junction (in Kelvin)
a	The diode ideality constant, Usually $1 \le a \le 1.5$

But the basic equation of the Practical arrays which requires the inclusion of additional parameters to the basic equation is

$$I = I_{pv} - I_0\left[e^{\left(\frac{V+R_S}{V_t\, a}\right)} - 1\right] - \left[\frac{V+R_S I}{R_p}\right] \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (6.5)$$

A parallel resistance (R_p) and a series resistance (R_s) component are added to the model since no solar cell is ideal in practice

Where

I_0 The saturation current of the array which improve in equation (6.7)

V_t The thermal voltage of the array with N_s cells connected in series = $N_s kT/q$
R_s The series resistance
R_p The parallel resistance

Note: when Cells connected in parallel increase the current and cells connected in series provide greater output voltages
The photovoltaic cell I_{pv} and saturation current I_o depend on the temperature according to the following equations:

$$I_{pv} = (I_{pv,n} + K_I \Delta_T)\frac{G}{G_n} \quad (6.6)$$

$I_{pv,n}$	The light-generated current at the nominal condition.
$\Delta_T = T - T_n$	T : The actual temperatures in [K]. T_n : The nominal temperatures in [K].
G	The irradiation on the device surface in [W/m²]
G_n	The nominal irradiation

By using MATLAB/SIMULINK we can simulated the *Ipv* as shown in Figure (6.8)

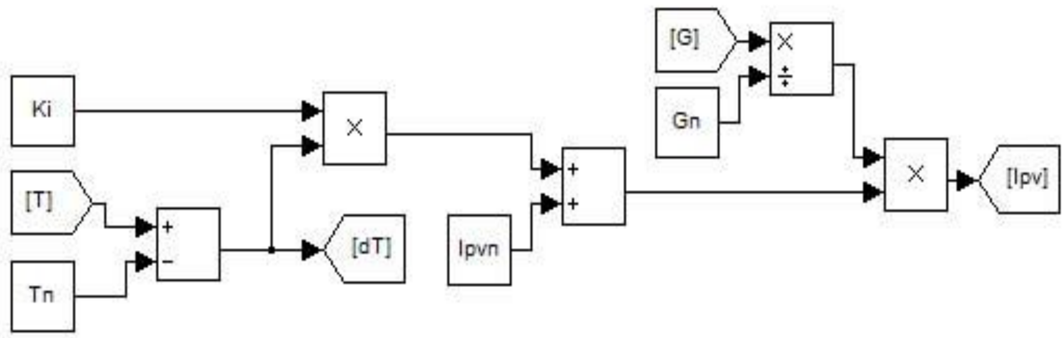


Figure 6.8: The Calculation of i_{pv} (Single Module)

When improving the PV model the I_0 is

$$I_0 = \frac{I_{sc,n} + K_I \Delta_T}{\left[e^{\left(\frac{V_{oc,n} + K_V \Delta_T}{a V_t}\right)}\right] - 1} \quad (6.7)$$

Where

$I_{sc,n}$	The nominal short circuit current
$V_{oc,n}$	The nominal open-circuit voltages
K_V	The open-circuit voltage/temperature coefficient
K_I	The short circuit current/temperature coefficient

This modification aims to match the open-circuit voltages of the model with the experimental data for a very large range of temperatures.

By using MATLAB/SIMULINK we can simulated the *Io* as shown in Figure (6.9)

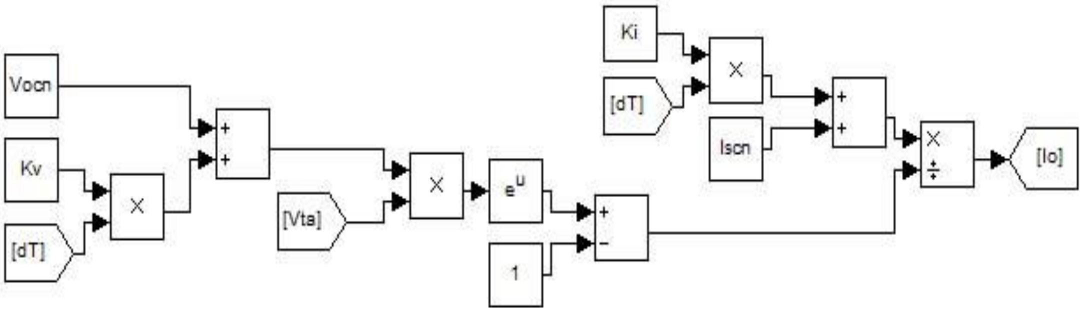


Figure 6.9: The Calculation Of Io (Single Module)

6.2.4.2 SIMULATION OF THE PV ARRAY

The PV array can be simulated with an equivalent circuit model based on the PV model of Figure (3.6), by using one current source (I_m) and two resistors (R_s and R_p) as shown the circuit model in Figure (6.10) [25]:

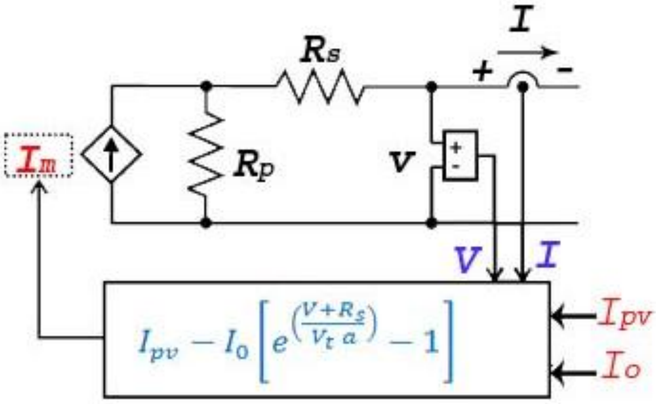


Figure 6.10: PV Array Model Circuit
With a Controlled Current Source, Equivalent Resistors, and the Equation of The Model Current (i_m)

The value of the model current I_m is calculated by the computational block that has V, I, I_0, and I_{pv} as inputs as shown in Figure (6.11) using simulation Matlab:

$$I_m = I_{pv} - I_o\left[\left(e^{\left(\frac{V+R_sI}{V_t a}\right)}\right) - 1\right] \quad (6.8)$$

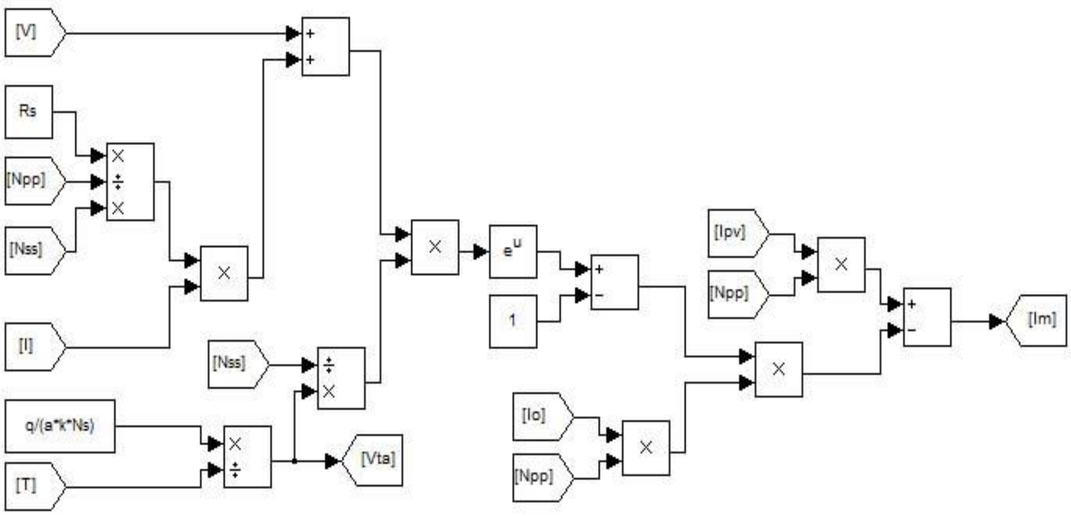


Figure 6.11: The Calculation of i_m =(i_{pv}-i_d)

The Photovoltaic circuit model built with MATLAB/SIMULINK as shown in Figure (6.12)

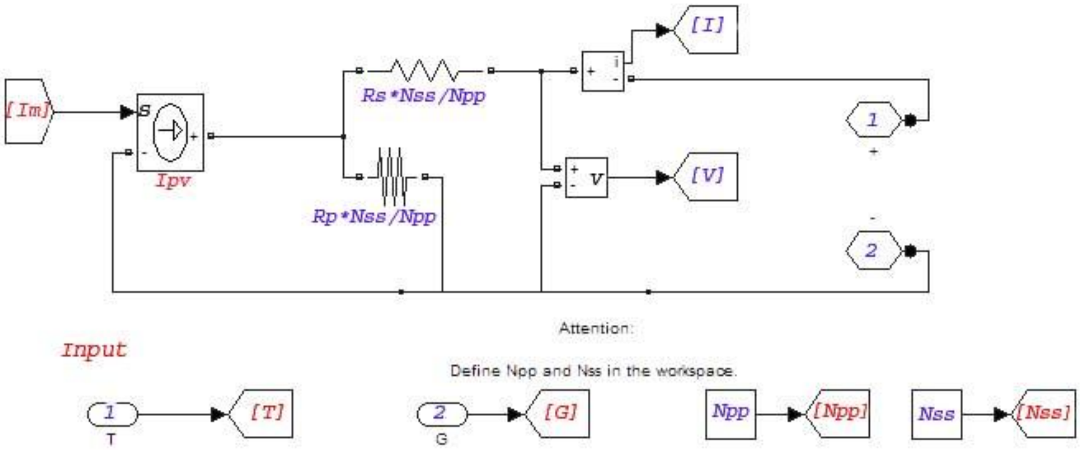


Figure 6.12: Photovoltaic Circuit Model Built With Matlab/Simulink.

Then all of these blocks as shown in Figures (6.9/11/12) are converted to one sub system block with two inputs (Temperature, and Irradiation) and two outputs (I_{pv},V_{pv}) as shown in Figure (6.13).

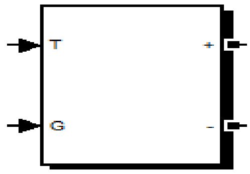


Figure 6.13: PV Model Subsystem

The parameters information of PV array is entered from the KC200GT solar array datasheet and from some (*I-V*) curves which manufacturers provided on datasheet as shown in appendix (A):

I_{scn}:	Nominal short-circuit voltage [A]	= 8.21
V_{ocn}:	Nominal array open-circuit voltage [V]	= 32.9
I_{mp}:	Array current at maximum power point [A]	= 7.61
V_{mp}:	Array voltage at maximum power point [V]	= 26.3
K_v :	Voltage/temperature coefficient [V/K]	= -0.123
K_i:	Current/temperature coefficient [A/K]	= 3.18e-3
N_s:	Nunber of series cells	= 54
G_n:	Nominal irradiance [W/m^2] at 25oC	= 1000
T_n:	Nominal operating temperature [K]	= 25 + 273.15
k:	Boltzmann Constant [J/K]	= 1.3806503e-23
q:	Electron charge Constant [C]	= 1.60217646e-19
a:	Diode ideality constant	= 1.3
R_p:	Parallel resistance [Ω]	= 415.405

R_s	Series resistance [Ω]	= 0.221
I_{mp}	The current at MPP [A]	= 7.61
V_{mp}	The voltage at the MPP [V]	= 26.3
P_{max}	Maximum Power = (I_{mp} *V_{mp}) [W]	= 200.143

6.2.4.3 Generation the Error and Change in error in simulation

In this section, we generate the error and change in error signals as inputs for the fuzzy logic controller as shown in Figure (6.14).

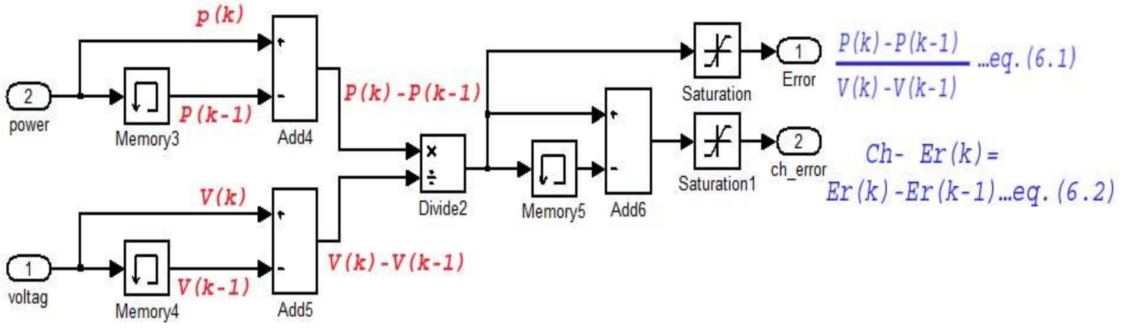


Figure 6.14: Generating The Error and Change in Error Signals

After generating the error and change in error as shown in Figures (6.14), we converted to one sub system block with two inputs (Power, and Voltage) and two outputs (e and Δe) which collected in multiplexer to inter in FLC as shown in Figure (6.15).

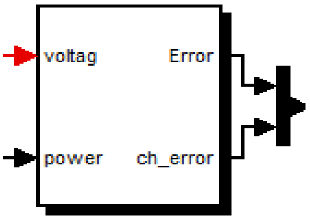


Figure 6.15: Error and Ch_Error Model Subsystem

6.2.4.4 Simulate Pulse Width Modulation (PWM)

The output of the PWM is examined using the scope by changing the values of the FLC and observe the change in the duty cycle of the PWM output.

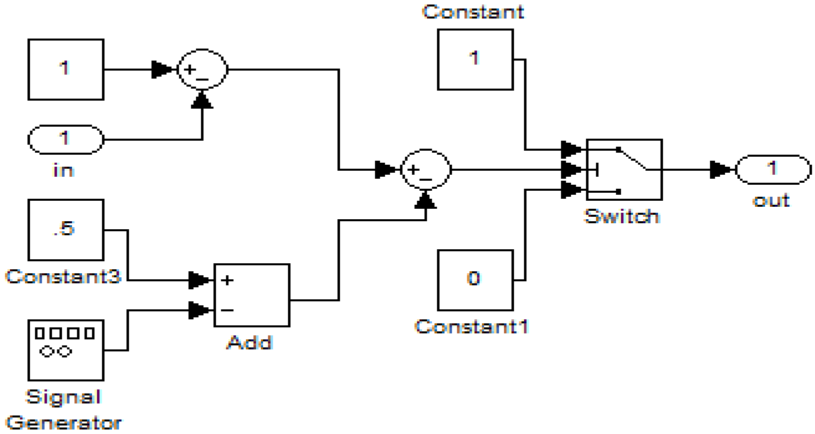


Figure 6.16: The simulation PWM

The simulation PWM converted to one sub system block with input from FLC and the output to the Dc – Dc converter Block as shown in Figure (6.17).

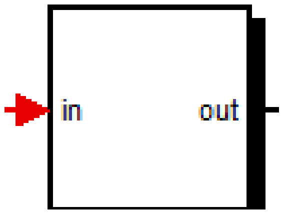


Figure 6.17: PWM Subsystem

6.2.4.5 Implementation the Dc To Dc Converter in simulation

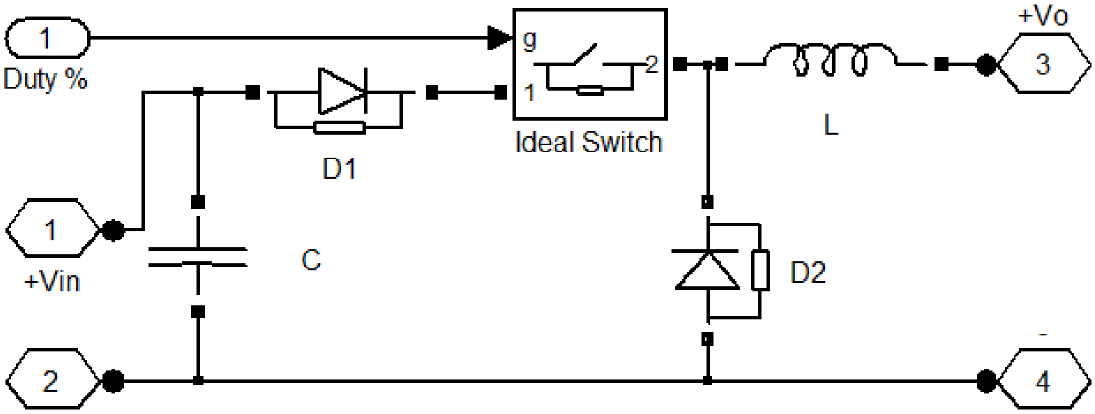


Figure 6.18: The Simulation of DC-DC Converter

This simulation converted to one sub system block as shown in Figure (6.19).

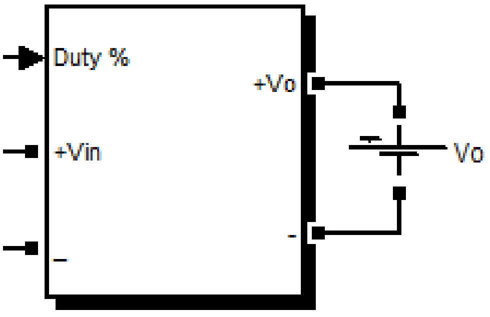


Figure 6.19: DC – DC Converter Subsystem

6.3. Fuzzy Logic controller Simulation

Maximum power point tracking system uses dc-to-dc converter to compensate the output voltage of the solar panel to keep the voltage at the value, which maximizes the output power.

MPP fuzzy logic controller measures the values of the voltage and current at the output of the solar panel, then calculates the power from the relation (P=V*I) to extract the inputs of the controller. The crisp output of the controller represents the duty cycle of the pulse width modulation to switch the dc to dc converter

Figure (6.20) illustrates the Simulink block diagram of the fuzzy controller for MPPT system.

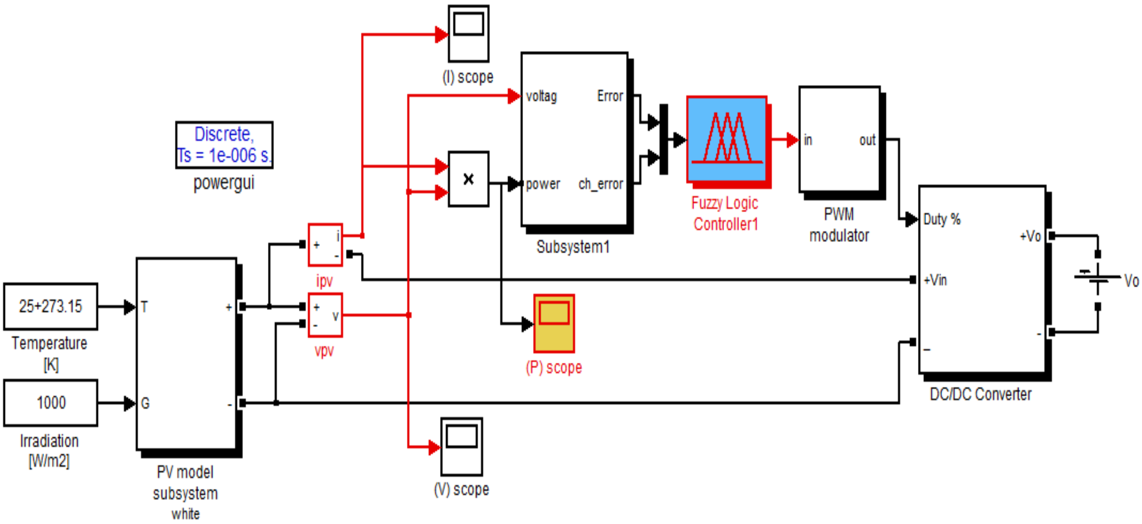


Figure 6.20: Controlling The Power using FLC

As shown above the designed fuzzy controller now connected between PV module and DC-to-DC converter module to tracking the MPP.

We can show the effect of the Sugeno FLC on the PV power in Figure (6.21), since it becomes constant at the maximum value (200.143 W) after a small stilling time.

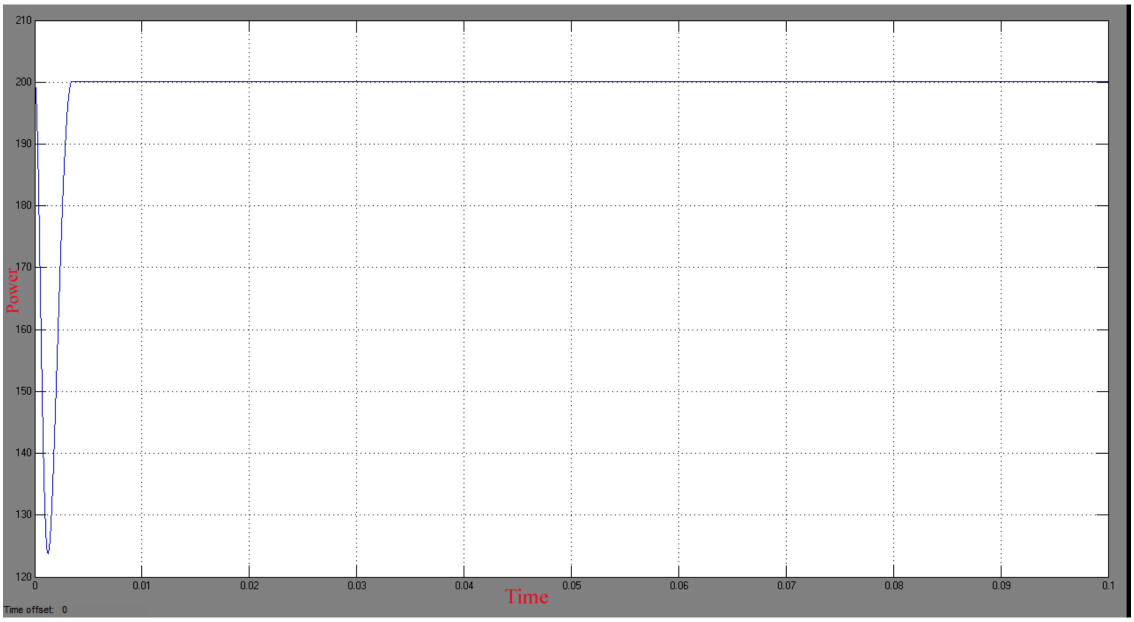


Figure 6.21: The Effect of FLC on the PV Power

The proposed maximum power point tracker is suitable to use to keep the PV output power at the maximum for increasing efficiency of it.

6.4. Comparison between Sugeno And Mamdani FLC without GA:

The results of applying the Sugeno FLC on PV system to track the maximum power point is compared with a Mamdani controller applied on the same system by EL-Moghany [2].

Figure (6.22) shows the effect of two controllers after doing zoom in:

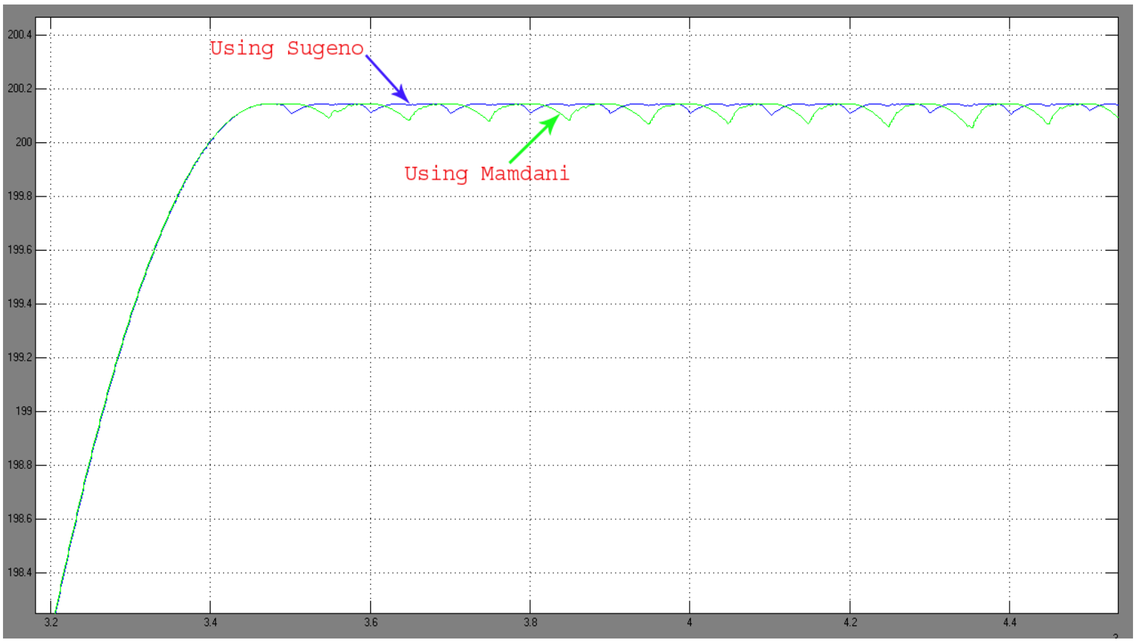


Figure 6.22 Zoom In between two controller
Sugeno and Mamdani FLC on The PV Power

So when we used the Sugeno or Mamdani controllers the proposed maximum power point tracker is suitable to use to keep the PV output power at the maximum for increasing efficiency of it but as shown in Sugeno we improvement the error.

6.5. Comparison Between Sugeno And Mamdani FLC withGA:

There are many parameters effect on the control as shown in Figure (6.23) in addition to the shape of the memberships of the inputs and output of fuzzy controller. These parameters are err_g (error gain) as shown in red triangle, ce_g (change of error gain) as in blue triangle and out_g (fuzzy output gain) as in magenta triangle. These gain parameters can be tuned to give near optimal results.

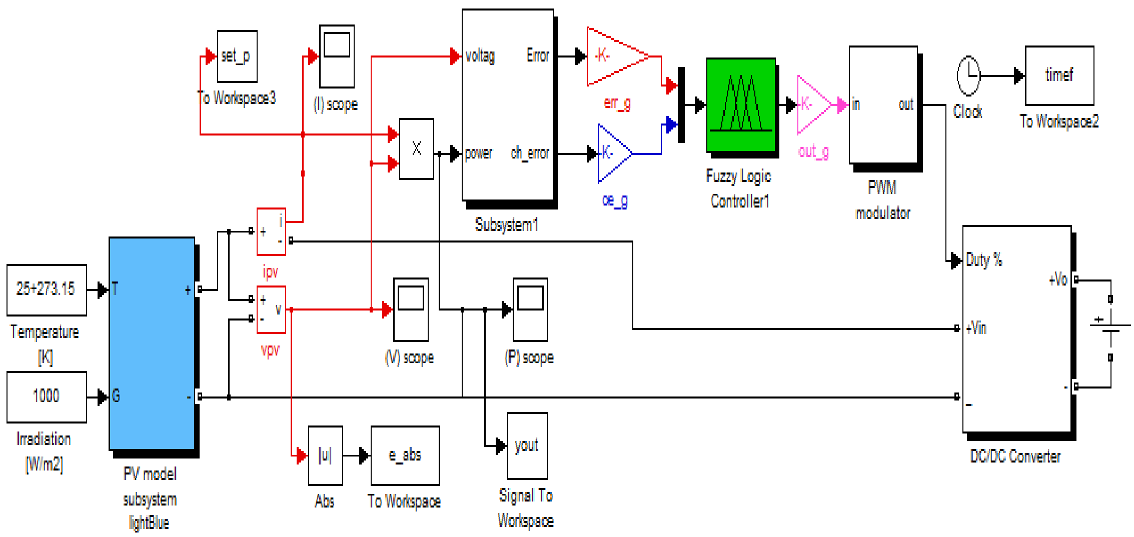


Figure 6.23 Fuzzy Controller on The PV Power

GA is used to optimize these parameters and optimize the shape of memberships of fuzzy controller. There are many ways to have good response, first is to optimize gain parameters only without any change in memberships shape. Second is to optimize the memberships shape without any change in gain parameters. The last way is optimize both gain parameters and memberships shape.
GA Matlab code (Appendix D) is used to optimize the memberships shape and gain parameters. This program is divided into two codes; main code is responsible for performing the GA steps such as selection, crossover, mutation and make new population. The second code is responsible for testing the new population to calculate the fitness function and out the best fitness value. This code is very simple GA codes, so that the memberships shape for inputs and output will change equally. In fuzzy controllers, there are two inputs and one output. Every input and output have seven membership functions, five of these membership functions are triangular memberships, one S membership function and one Z membership function. In triangular membership functions, there are three parameters that can be modified to change its shape, center, left edge, right edge. In Z and S membership functions there are Set Point two parameters, top edge and left or right edges (depending on if it is S or Z membership).

But in this thesis some simplification are used in writing GA code.
1- The interval of the inputs and output will still at [-1 1] range.
2- The symmetric point will still at zero.
3- The inputs and output will have he same shape.

After this simplification the number of variable will be 7 for membership functions and 3 for the gain parameters. The fitness function that will be used is the integral of the absolute values of the error, in the control design the fitness value need to be minimized. The number of individuals per one population will be 50, the number of generation will be 100, the crossover probability will be 0.7. The firs step in matlab program, generate random population consist of 50 individual, every individual consist of 13 variable every variable is coded in 10 bit binary form. Second step, convert the binary code of every variable to real number. Third step, call fuzzy control program and simulink program and find the fitness value of the fitness function (the integral of absolute values of error). Fourth step, implement crossover and mutation, and finally generate the new population. The stop condition can be after exact times or after the exact value of fitness value. In this thesis, the stop condition is after 100 generation.

Figure (6.24) shows the memberships shape of the inputs and output.

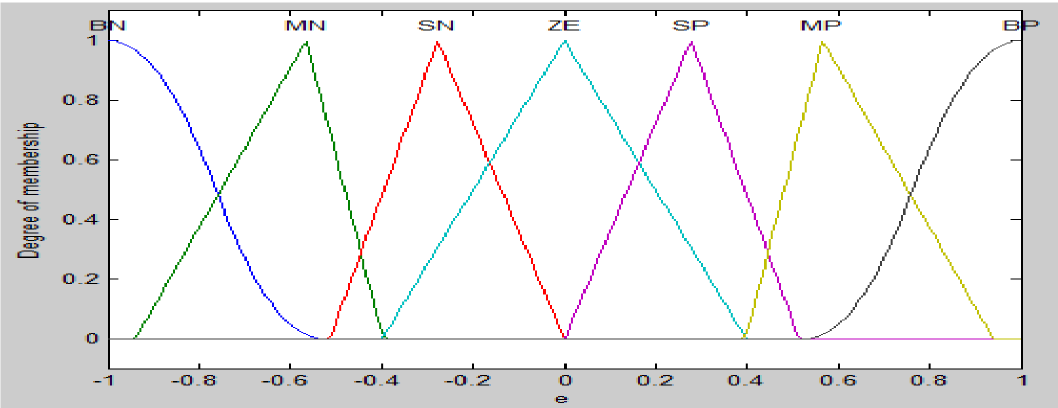


Figure 6.24 – 1: Membership Function of Input Error e

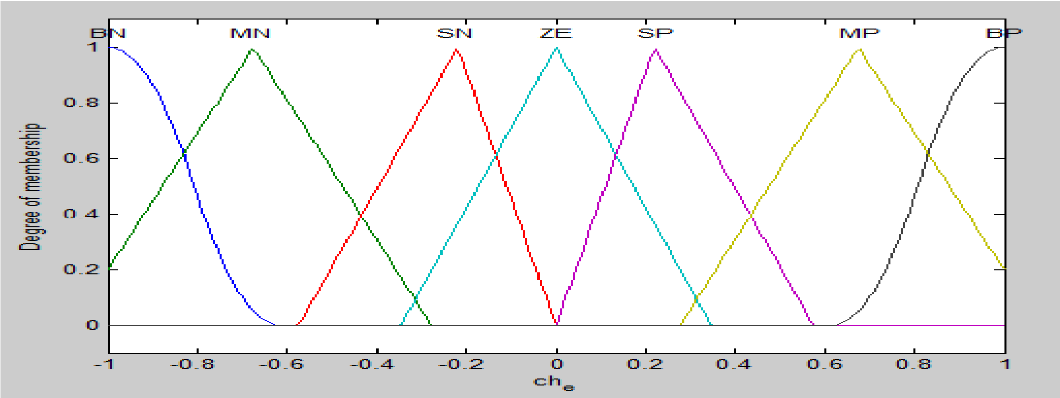


Figure 6.24 – 2: Membership Function of Input Error Δe

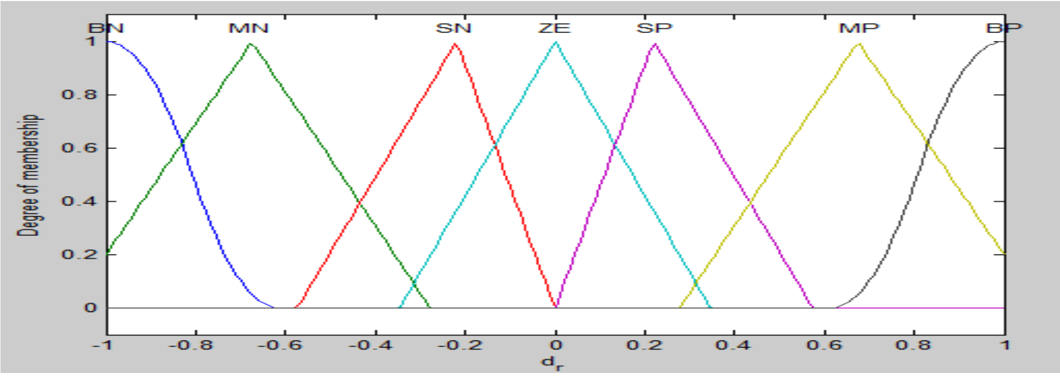


Figure 6.24 – 3: Membership Function of Output of Duty Ratio *d*

The results of applying FLC with GA on PV system to track the maximum power point is compared with a Mamdani and Sugeno controller without GA applied on the same system.

Figure (6.25) shows the effect of FLC controllers with GA after doing zoom in on the PV power:

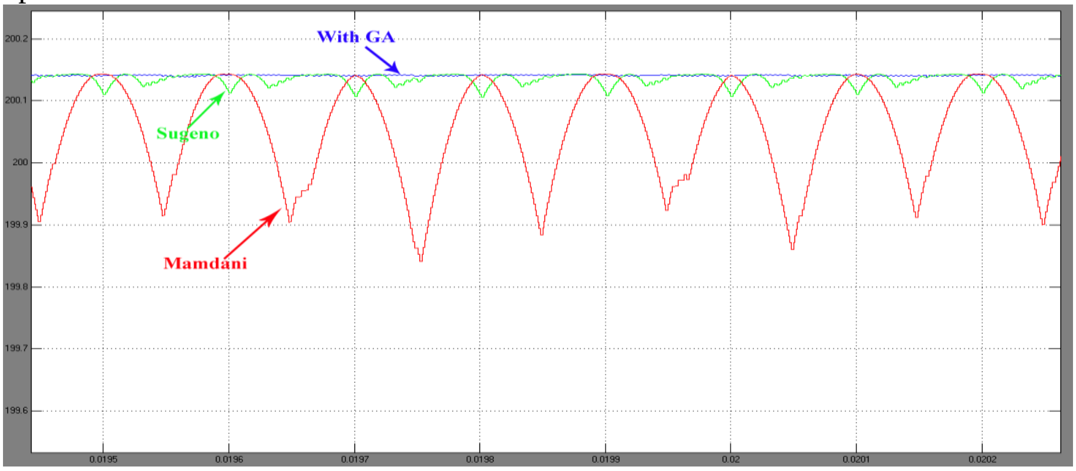


Figure 6.25: The Effect of controllers with - without GA after doing zoom

By comparison, between these results with/without GA optimization can find that GA give better response and improved the error of the system at the same settling time.

	Mamdani	Sugeno	With GA
Error	0.303	0.041	0.005

CHAPTER 7 CONCLUSIONS AND FUTURE RESEARCH

7.1. Conclusion

In recent years, tremendous progress has been made in the techniques of control field. The control field entered almost all areas such as medical applications, military and civilian. Fuzzy control is one of those techniques. This technique in recent years has become involved in many applications and integrated with other technologies to improve their performance. GA is one of these technologies that can be used to optimize the fuzzy controller.

Renewable energy sources are clean and unlimited, which play an important role in electric power generation that can be used to solve electrical power generation problem in Gaza Strip.

In this thesis, Sugeno fuzzy logic controller was designed to maximize the energy received from solar cells by two methods. A sun tracker is the first method controlled by fuzzy logic controller to keep the PV panel pointing toward the sun by using a stepper motor. The use of stepper motor enables accurate tracking of the sun, LDR resistors are used to determine the solar light intensity. This controller has been tested using Matlab/Simulink program. The proposed solar tracking power generation system was able to track the sun light automatically, so it is an efficient system for solar energy collection. The results show that the sun tracking systems by Sugeno FL controller have a better response than a Mamdani FLC. The second method is a maximum power point tracker controlled by fuzzy logic controller with GA and using buck DC-to-DC converter to keep the PV output power at the maximum point all the time. This controller was tested using Matlab/Simulink program, and the results was compared with Mamdani FL controller applied on the same system. The comparison shows that the Sugeno fuzzy logic controller was better in response. It can be said that the proposed sun tracking solar array system and MPPT is a feasible methods of maximizing the energy received from solar cells.

In this thesis, the Sugeno fuzzy logic control demonstrates better performance than Mamdani fuzzy logic control. Furthermore, fuzzy logic offers the advantages of faster design, and emulation of human control strategies. The fuzzy logic control requires complete knowledge of the operation of the system by the designer.

7.2. Future Research

1) Using fuzzy supervised PID techniques to give better results.
2) Change the 8 bits of The ADC and DAC with 12 bit or higher to give more accuracy.
3) We can implement the Sugeno FLC for (Sun T and MPPT) by using the modern technique in FPGA.

REFERENCES

[1] Janssen,R., "*Renewable Energy..into the Mainstream*", The Netherlands, 2002.

[2] Mohammed S. EL-Moghany, Sun and Maximum Power Point Tracking in Solar Array Systems Using Fuzzy Controllers via FPGA, Master Thesis, Islamic University-Gaza, 2011.

[3] A. Louchene, ″Solar tracking system with fuzzy reasoning applied to crisp sets", Revue des Energies Renouvelables Vol. 10 N°2, 2007.

[4] B.Lane, *″Solar Tracker"*, Cleveland State University April 30, 2008.

[5] M. A. Usta, Ö. Akyazı and İ. H. Altas, ″Design and Performance of Solar Tracking System with Fuzzy Logic Controller" International Advanced Technologies Technologies Symposium (IATS'11), Elazığ, Turkey, 2011.

[6] Gustavo Ozuna, Carlos Anaya. Diana Figueroa. Nun Pitalua, ″Solar Tracker of Two Degrees of Freedom for Photovoltaic Solar Cell Using Fuzzy Logic," Proceedings of the World Congress on Engineering, 2011.

[7] N. Patcharaprakiti, S. Premrudeepreechacharn, Y Sriuthaisiriwong , ″Maximum power point tracking using adaptive fuzzy logic control for grid-connected photovoltaic system", Chiang Mai University, Thailand, 2005.

[8] Aït,M. Cheikh,S. C.Larbes, ″Maximum power point tracking using a fuzzy logic control scheme", Revue des Energies Renouvelables Vol. 10 N°3, 2007.

[9] Yuen-Haw Chang and Wei-Fu Hsu, ″A Maximum Power Point Tracking of PV System by Adaptive Fuzzy Logic Control," Proceedings of the international Multi Conference of Engineers and Computer Scientists 2011.

[10] S.Borenstein, The Market Value and Cost of Solar Photovoltaic Electricity Production, Berkeley, This paper is part of the (CSEM), California, 2008.

[11] M. El-Ashry, *Renewables 2010 Global Status Report*, Paris: REN21 Secretariat. Copyright Deutsche (GTZ) GmbH, 2010.

[12] V.Quaschning, *″Understanding Renewable Energy Systems"*, London, Carl Hanser Verlag GmbH & Co KG, 2005.

[13] Photovoltaic Project Analysis. Clean Energy Project Analysis Course. Minister of Natural, Resources Canada, 2004.

[14] Parallax, *″Experiments with Renewable Energy"*, version1, pages 310, 2004.

[15] Principle of solar cells. Retrieved from
http://global.kyocera.com/prdct/solar/spirit/about_solar/cell.html

[16] Photovoltaic Panel Efficiency. Retrieved from http://www.solar-facts.com/panels/panel-efficiency.php/

[17] M.Villalva, J.Gazoli, and E.Ruppert, ″Modeling And Circuit-Based Simulation OF Photovoltaic Arrays", *IEEE Transactions On Power Electronics,* 2009.

[18] Muhammad Saad Rahman, Buck Converter Design Issues, Master Thesis, Linköping Institute of Technology, 2007.

[19] William H. Hsu, *Genetic Algorithms* Kansas State University, USA.

[20] L. A. Zadeh, "Fuzzy sets," Information and control, 1965.

[21] K. M. Passion and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.

[22] L.X. Wang, *A Course in Fuzzy Systems and Controls*, Englewood Cliffs: Prentice-Hall, 1997

[23] E H Mamdani, "Application of Fuzzy algorithms for the control of a dynamic plant," 1974.

[24] Moayed N. EL Mobaied, Fuzzy Logic Speed Controllers Using FPGA Technique for Three-Phase Induction Motor Drives, Master Thesis, Islamic University-Gaza, 2008.

[25] B.Riza, Sheldon,T., ″Fuzzy Systems Design Principles Building Fuzzy IF-THEN Rule Bases", *IEEE* PRESS, 1997.

[26] John Yen & Langari Reza, *Fuzzy Logic Intelligence Control and Information*, Prentic- Hall, Englwood Cliffs, 1999.

[27] P. Basehore, "Fuzzy Logic Outperforms PID Control," 1993.

[28] Funk,P., Online fuzzy case-based individual stress diagnosing system, Master Thesis, Mälardalen University, 2005.

[29] Hosam Abu Elreesh, Design of GA-Fuzzy Controller For Magnetic Levitation Using FPGA, Master Thesis, Islamic University-Gaza, 2011.

[30] H.-J. Zimmermann, *Fuzzy Sets Theory - and Its Applications*, Kluwer Academic Publishers, 1990.

[31] M. Jamshidi, Vadiee,N. and Ross,T., *Fuzzy Logic and Control Software and Hardware Applications, Englewood Cliffs: Prentice Hall, Inc*., 1993.

[32] S.N. Sivanandam and S.N. Deepa, *Introduction to Genetic Algorithms*, Springer, New York, 2008.

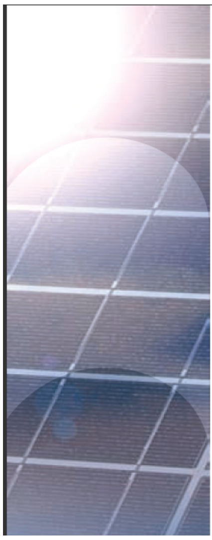
[33] Outi Raiha, Applying Genetic Algorithms in Software Architecture Design, University of Tampere, Department of Computer Sciences Computer Science ,M.Sc thesis, February, 2008.

[34] Hung-Cheng Chen. and Sheng-Hsiung Chang, "Genetic Algorithms Based Optimization Design of a PID Controller for an Active Magnetic Bearing," IJCSNS International Journal of Computer Science and Network Security, 2006.

[35] Cătălin Alexandru, "Aspects Regarding The Mechatronic Tracking Systems Used For Improving The Photovoltaic Converstion", University Transilvania of Braşov, Product Design & Robotics Dept., 2008.

APPENDICES

Appendix A: The Datasheet Of Photovoltaic KC200GT



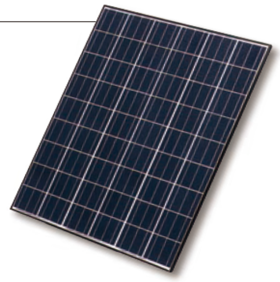
THE NEW VALUE FRONTIER

KYOCERA

KC200GT

HIGH EFFICIENCY MULTICRYSTAL PHOTOVOLTAIC MODULE

UL 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.

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℃

Cells	
Number per Module	54

Module Characteristics	
Length × Width × Depth	1425mm(56.2in)×990mm(39.0in)×36mm(1.
Weight	18.5kg(40.7lbs.)
Cable	(+)720mm(28.3in),(-)1800mm(70.9

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

APPENDIX B-I: The Rule Base Of Sugeno Fuzzy Controller For Sun Tracker System

1. IF (Error is BN) and (Ch-Error is BN) then (Speed is BN).
2. IF (Error is MN) and (Ch-Error is BN) then (Speed is BN).
3. IF (Error is SN) and (Ch-Error is BN) then (Speed is BN).
4. IF (Error is ZE) and (Ch-Error is BN) then (Speed is BN).
5. IF (Error is SP) and (Ch-Error is BN) then (Speed is MN).
6. IF (Error is MP) and (Ch-Error is BN) then (Speed is SN).
7. IF (Error is BP) and (Ch-Error is BN) then (Speed is ZE).
8. IF (Error is BN) and (Ch-Error is MN) then (Speed is BN).
9. IF (Error is MN) and (Ch-Error is MN) then (Speed is BN).
10. IF (Error is SN) and (Ch-Error is MN) then (Speed is MN).
11. IF (Error is ZE) and (Ch-Error is MN) then (Speed is MN).
12. IF (Error is SP) and (Ch-Error is MN) then (Speed is SN).
13. IF (Error is MP) and (Ch-Error is MN) then (Speed is ZE).
14. IF (Error is BP) and (Ch-Error is MN) then (Speed is SP).
15. IF (Error is BN) and (Ch-Error is SN) then (Speed is BN).
16. IF (Error is MN) and (Ch-Error is SN) then (Speed is MN).
17. IF (Error is SN) and (Ch-Error is SN) then (Speed is SN).
18. IF (Error is ZE) and (Ch-Error is SN) then (Speed is SN).
19. IF (Error is SP) and (Ch-Error is SN) then (Speed is ZE).
20. IF (Error is MP) and (Ch-Error is SN) then (Speed is SP).
21. IF (Error is BP) and (Ch-Error is SN) then (Speed is MP).
22. IF (Error is BN) and (Ch-Error is ZE) then (Speed is BN).
23. IF (Error is MN) and (Ch-Error is ZE) then (Speed is MN).
24. IF (Error is SN) and (Ch-Error is ZE) then (Speed is SN).
25. IF (Error is ZE) and (Ch-Error is ZE) then (Speed is ZE).
26. IF (Error is SP) and (Ch-Error is ZE) then (Speed is SP).
27. IF (Error is MP) and (Ch-Error is ZE) then (Speed is MP).
28. IF (Error is BP) and (Ch-Error is ZE) then (Speed is BP).
29. IF (Error is BN) and (Ch-Error is SP) then (Speed is MN).
30. IF (Error is MN) and (Ch-Error is SP) then (Speed is SN).
31. IF (Error is SN) and (Ch-Error is SP) then (Speed is ZE).
32. IF (Error is ZE) and (Ch-Error is SP) then (Speed is SP).
33. IF (Error is SP) and (Ch-Error is SP) then (Speed is SP).
34. IF (Error is MP) and (Ch-Error is SP) then (Speed is MP).
35. IF (Error is BP) and (Ch-Error is SP) then (Speed is BP).
36. IF (Error is BN) and (Ch-Error is MP) then (Speed is SN).
37. IF (Error is MN) and (Ch-Error is MP) then (Speed is ZE).
38. IF (Error is SN) and (Ch-Error is MP) then (Speed is SP).
39. IF (Error is ZE) and (Ch-Error is MP) then (Speed is MP).
40. IF (Error is SP) and (Ch-Error is MP) then (Speed is MP).
41. IF (Error is MP) and (Ch-Error is MP) then (Speed is BP).
42. IF (Error is BP) and (Ch-Error is MP) then (Speed is BP).
43. IF (Error is BN) and (Ch-Error is BP) then (Speed is ZE).
44. IF (Error is MN) and (Ch-Error is BP) then (Speed is SP).
45. IF (Error is SN) and (Ch-Error is BP) then (Speed is MP).
46. IF (Error is ZE) and (Ch-Error is BP) then (Speed is BP).
47. IF (Error is SP) and (Ch-Error is BP) then (Speed is BP).
48. IF (Error is MP) and (Ch-Error is BP) then (Speed is BP).
49. IF (Error is BP) and (Ch-Error is BP) then (Speed is BP).

APPENDIX B-II: The Rule Base Of Sugeno Fuzzy Controller For MPP Tracker System

1. IF (Error is BN) and (Ch-Error is BN) then (Duty ratio is ZE).
2. IF (Error is MN) and (Ch-Error is BN) then (Duty ratio is ZE).
3. IF (Error is SN) and (Ch-Error is BN) then (Duty ratio is SN).
4. IF (Error is ZE) and (Ch-Error is BN) then (Duty ratio is MN).
5. IF (Error is SP) and (Ch-Error is BN) then (Duty ratio is MP).
6. IF (Error is MP) and (Ch-Error is BN) then (Duty ratio is MP).
7. IF (Error is BP) and (Ch-Error is BN) then (Duty ratio is BP).
8. IF (Error is BN) and (Ch-Error is MN) then (Duty ratio is ZE).
9. IF (Error is MN) and (Ch-Error is MN) then (Duty ratio is ZE).
10. IF (Error is SN) and (Ch-Error is MN) then (Duty ratio is ZE).
11. IF (Error is ZE) and (Ch-Error is MN) then (Duty ratio is SN).
12. IF (Error is SP) and (Ch-Error is MN) then (Duty ratio is SP).
13. IF (Error is MP) and (Ch-Error is MN) then (Duty ratio is MP).
14. IF (Error is BP) and (Ch-Error is MN) then (Duty ratio is BP).
15. IF (Error is BN) and (Ch-Error is SN) then (Duty ratio is ZE).
16. IF (Error is MN) and (Ch-Error is SN) then (Duty ratio is ZE).
17. IF (Error is SN) and (Ch-Error is SN) then (Duty ratio is ZE).
18. IF (Error is ZE) and (Ch-Error is SN) then (Duty ratio is ZE).
19. IF (Error is SP) and (Ch-Error is SN) then (Duty ratio is SP).
20. IF (Error is MP) and (Ch-Error is SN) then (Duty ratio is MP).
21. IF (Error is BP) and (Ch-Error is SN) then (Duty ratio is BP).
22. IF (Error is BN) and (Ch-Error is ZE) then (Duty ratio is BN).
23. IF (Error is MN) and (Ch-Error is ZE) then (Duty ratio is MN).
24. IF (Error is SN) and (Ch-Error is ZE) then (Duty ratio is SN).
25. IF (Error is ZE) and (Ch-Error is ZE) then (Duty ratio is ZE).
26. IF (Error is SP) and (Ch-Error is ZE) then (Duty ratio is SP).
27. IF (Error is MP) and (Ch-Error is ZE) then (Duty ratio is MP).
28. IF (Error is BP) and (Ch-Error is ZE) then (Duty ratio is BP).
29. IF (Error is BN) and (Ch-Error is SP) then (Duty ratio is BN).
30. IF (Error is MN) and (Ch-Error is SP) then (Duty ratio is MN).
31. IF (Error is SN) and (Ch-Error is SP) then (Duty ratio is SN).
32. IF (Error is ZE) and (Ch-Error is SP) then (Duty ratio is ZE).
33. IF (Error is SP) and (Ch-Error is SP) then (Duty ratio is ZE).
34. IF (Error is MP) and (Ch-Error is SP) then (Duty ratio is ZE).
35. IF (Error is BP) and (Ch-Error is SP) then (Duty ratio is ZE).
36. IF (Error is BN) and (Ch-Error is MP) then (Duty ratio is BN).
37. IF (Error is MN) and (Ch-Error is MP) then (Duty ratio is MN).
38. IF (Error is SN) and (Ch-Error is MP) then (Duty ratio is SN).
39. IF (Error is ZE) and (Ch-Error is MP) then (Duty ratio is SP).
40. IF (Error is SP) and (Ch-Error is MP) then (Duty ratio is ZE).
41. IF (Error is MP) and (Ch-Error is MP) then (Duty ratio is ZE).
42. IF (Error is BP) and (Ch-Error is MP) then (Duty ratio is ZE).
43. IF (Error is BN) and (Ch-Error is BP) then (Duty ratio is BN).
44. IF (Error is MN) and (Ch-Error is BP) then (Duty ratio is MN).
45. IF (Error is SN) and (Ch-Error is BP) then (Duty ratio is SN).
46. IF (Error is ZE) and (Ch-Error is BP) then (Duty ratio is MP).
47. IF (Error is SP) and (Ch-Error is BP) then (Duty ratio is SP).
48. IF (Error is MP) and (Ch-Error is BP) then (Duty ratio is ZE).
49. IF (Error is BP) and (Ch-Error is BP) then (Duty ratio is ZE).

APPENDIX C-I: Output Rule Base Of Sun Tracker System

BN= [-10 -10 10]
MN= [-70 -35 0]
SN= [0 70 0.3333]
ZE= [9.7 -0.1 -5.485]
SP= [0 0 1.5]
MP= [35 -37.8 0]
BP= [0 0 5.5]

APPENDIX C-II: Output Rule Base Of MPP Tracker System

BN= [0 -0.7 -0.5]
MN= [-0.7 -0.5 -0.2]
SN= [-0.5 -0.2 0]
ZE= [-0.2 0 0.2]
SP= [0 0.2 0.5]
MP= [0.2 0.5 0.7]
BP= [0.5 0.7 100]

Appendix D: GA MATLAB Programs

I. GA FUZZY MATLAB CODE MAIN PROGRAM

```
%----------------------------------------------------------
            % Clear the workspace
%----------------------------------------------------------

clear
clc

global rin yout timef e_abs set_p

Ts=1.000000000000000e-06;
load('MPPT_Parameters')                  % Load data from MAT-file into
workspace
MPPT_MamdaniGA=readfis('MPPT_Mamdani')% Open membership Function in
workspace
MPPT_MamdMOH=readfis('MPPT_MamdMOH')  % Open membership Function in
workspace
MPPT_Sugeno_Com=readfis('MPPT_Sugeno_Com')   % Open membership
Function in workspace

open_system('MPPT_Mam')                   % Open Main Design System
MAXGEN = 100;                             % maximum Number of generations
NVAR = 13;                                % Generation gap, how many new
individuals are created
GGAP = .5;                                % Generation gap, how many new
individuals are created
PRECI = 10;                               % Binary representation
precision
NIND = 50;                                % No. of individuals per
subpopulations

%----------------------------------------------------------
            % First, a field descriptor is set up
%----------------------------------------------------------
FieldD = [rep([PRECI],[1, NVAR]); rep([-0.1;0.1],[1, NVAR]);...
rep([1; 0; 1 ;1], [1, NVAR])];



%----------------------------------------------------------
            % The population is then initialized
%----------------------------------------------------------
FieldD(2,1)=0.9;FieldD(2,2)=14.5;FieldD(2,3)=14.5;FieldD(2,12)=2.4;Fie
ldD(2,13)=0;
FieldD(3,1)=1.1;FieldD(3,2)=15.5;FieldD(3,3)=15.5;FieldD(3,12)=2.8;Fie
ldD(3,13)=0.2;
FieldD;

BsJ=0;

E = crtbp(NIND, NVAR*PRECI);
E(1,:)=[1,0,0,0,0,0,0,0,1,1,0,1,1,0,0,1,0,1,0,0,1,1,0,0,0,1,1,1,1,1,
1,1,0,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,1,0,1,1,0,0,1,0,0,0,1,1,1,1,1,
0,0,0,1,1,1,0,1,0,0,0,1,0,1,0,1,0,1,1,1,0,0,1,1,1,0,0,1,0,0,0,1,0,0,1,
0,0,0,0,0,0,0,0,1,1,1,0,0,1,0,1,0,1,0,1,1,0,1,1,1,0,1,0,1;];
E(2,:)=[1,0,0,0,0,0,0,0,1,1,0,1,1,0,1,0,1,1,0,1,1,0,0,0,0,0,0,0,0,1,
1,1,0,0,1,1,0,1,0,0,0,0,1,0,1,0,1,1,1,0,1,1,1,0,0,0,0,1,0,0,1,0,1,0,0,
```

```
1,0,0,1,1,0,1,1,0,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,1,1,0,0,1,1,0,1,
1,1,1,0,1,0,0,1,0,1,1,0,1,1,0,1,0,1,0,0,1,1,1,0,0,1,0,1,1;];
E(3,:)=[1,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
1,1,1,0,1,0,1,1,0,0,1,1,1,0,0,0,0,0,0,1,0,0,0,0,1,1,1,1,0,1,0,1,0,1,0,
1,0,0,1,0,0,1,1,0,1,0,1,1,1,1,0,0,1,0,1,0,1,0,1,1,1,1,0,0;];
E(4,:)=[1,0,0,0,0,1,1,0,1,0,0,1,1,0,0,0,0,1,1,0,0,0,1,0,1,1,0,1,1,1,
0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,1,1,1,0,0,0,0,1,1,0,0,1,0,1,1,0,0,0,0,1,
1,0,1,1,0,1,1,1,1,0,1,0,0,0,0,0,1,0,1,0,0,1,0,0,0,0,1,1,0,0,0,1,0,1,1,
1,0,0,1,0,1,0,0,1,0,0,1,1,0,0,0,0,1,1,0,1,1,0,1,1,0,0,0,0;];

% Reset counters
Best = NaN*ones(MAXGEN,1); % Defines 'best' in current population
gen = 0;                   % generational counte

%-----------------------------------------------------------
            % Initial population is evaluated
%-----------------------------------------------------------
for  kg=1:1:MAXGEN
     time(kg)=kg;
       Kfuzzy=bs2rv(E,FieldD);

    for s=1:1:NIND

         %****** Step 1 : Evaluate BestJ ******

        Kfuzzyi=Kfuzzy(s,:);

        [Kfuzzyi,BsJ]=fuzzy_MPPT(Kfuzzyi,BsJ);

        BsJi(s)=BsJ;

    end
    [OderJi,IndexJi]=sort(BsJi);
    BestJ(kg)=OderJi(1);
    BJ=BestJ(kg);
    Ji=BsJi+1e-10;

    fi=1./Ji;

    [Oderfi,Indexfi]=sort(fi);     %Arranging fi small to bigger

    Bestfi=Oderfi(NIND);           % Let Bestfi=max(fi)
    BestS=E(Indexfi(NIND),:);      % Let BestS=E(m), m is the Indexfi
belong to max(fi)

    kg
    BJ
    BestS;

%****** Step 2 : Select and reproduct Operation******
    fi_sum=sum(fi);
    fi_Size=(Oderfi/fi_sum)*NIND;

    fi_S=floor(fi_Size);        %Selecting Bigger fi value

    kk=1;
    for i=1:1:NIND
```

```
        for j=1:1:fi_S(i)          %Select and Reproduce
            TempE(kk,:)=E(Indexfi(i),:);
            kk=kk+1;                %kk is used to reproduce
        end
    end

    %************ Step 3 : Crossover Operation ************
    pc=0.99;
    n=ceil(130*rand);
    for i=1:2:(NIND-1)
        temp=rand;
        if pc>temp                   %Crossover Condition
            for j=n:1:130
                TempE(i,j)=E(i+1,j);
                TempE(i+1,j)=E(i,j);
            end
        end
    end
    TempE(NIND,:)=BestS;
    E=TempE;

    %************ Step 4: Mutation Operation **************

    pm=0.001-[1:1:NIND]*(0.001)/NIND;  %Bigger fi, smaller pm

    for i=1:1:NIND
        for j=1:1:3*PRECI
            temp=rand;
            if pm(i)>temp                %Mutation Condition
                if TempE(i,j)==0
                    TempE(i,j)=1;
                else
                    TempE(i,j)=0;
                end
            end
        end
    end
    %Guarantee TempE(Size,:) belong to the best individual
    TempE(NIND,:)=BestS;
    E=TempE;
    %*********************************************************

end

Bestfi
BestS
Kfuzzybest=bs2rv(BestS,FieldD);
[Kfuzzyi,BsJ]=fuzzy_MPPT(Kfuzzybest,BsJ);
Kfuzzyi
Best_J=BestJ(MAXGEN)
Figure(1);
  plot(time,BestJ);
  xlabel('Times');ylabel('Best J');
% Figure(4);
  %plot(timef,set_p,'r',timef,yout,'b');
  %xlabel('Time');ylabel('Power');
```

II. GA FUZZY MATLAB CODE CALL SIMULINK PROGRAM

```
function [Kfuzzyi,BsJ]=fuzzy_gaf(Kfuzzyi,BsJ)
global rin yout timef e_abs set_p


aa=newfis('MPPT_Mamdani');

aa=addvar(aa,'input','e',[-1,1]);            %Parameter e
aa=addmf(aa,'input',1,'BN','zmf',[-1,-0.6-Kfuzzyi(10)]);
aa=addmf(aa,'input',1,'MN','trimf',[-1-Kfuzzyi(9),-0.6-Kfuzzyi(8), -
0.3-Kfuzzyi(7)]);
aa=addmf(aa,'input',1,'SN','trimf',[-0.6-Kfuzzyi(6),-0.3-
Kfuzzyi(5),0]);
aa=addmf(aa,'input',1,'ZE','trimf',[-0.3-
Kfuzzyi(4),0,0.3+Kfuzzyi(4)]);
aa=addmf(aa,'input',1,'SP','trimf',[0,0.3+Kfuzzyi(5),0.6+Kfuzzyi(6)]);
aa=addmf(aa,'input',1,'MP','trimf',[0.3+Kfuzzyi(7),0.6+Kfuzzyi(8),1+Kf
uzzyi(9)]);
aa=addmf(aa,'input',1,'BP','smf',[0.6+Kfuzzyi(10),1]);
Figure(1)
plotmf(aa,'input',1)

aa=addvar(aa,'input','ch_e',[-1,1]);           %Parameter ch_e
aa=addmf(aa,'input',2,'BN','zmf',[-1,-0.6-Kfuzzyi(10)]);
aa=addmf(aa,'input',2,'MN','trimf',[-1-Kfuzzyi(9),-0.6-Kfuzzyi(8), -
0.3-Kfuzzyi(7)]);
aa=addmf(aa,'input',2,'SN','trimf',[-0.6-Kfuzzyi(6),-0.3-
Kfuzzyi(5),0]);
aa=addmf(aa,'input',2,'ZE','trimf',[-0.3-
Kfuzzyi(4),0,0.3+Kfuzzyi(4)]);
aa=addmf(aa,'input',2,'SP','trimf',[0,0.3+Kfuzzyi(5),0.6+Kfuzzyi(6)]);
aa=addmf(aa,'input',2,'MP','trimf',[0.3+Kfuzzyi(7),0.6+Kfuzzyi(8),1+Kf
uzzyi(9)]);
aa=addmf(aa,'input',2,'BP','smf',[0.6+Kfuzzyi(10),1]);
Figure(2)
plotmf(aa,'input',2)

aa=addvar(aa,'output','d_r',[-1,1]);           %Parameter d_r
aa=addmf(aa,'output',1,'BN','zmf',[-1,-0.6-Kfuzzyi(10)]);
aa=addmf(aa,'output',1,'MN','trimf',[-1-Kfuzzyi(9),-0.6-Kfuzzyi(8), -
0.3-Kfuzzyi(7)]);
aa=addmf(aa,'output',1,'SN','trimf',[-0.6-Kfuzzyi(6),-0.3-
Kfuzzyi(5),0]);
aa=addmf(aa,'output',1,'ZE','trimf',[-0.3-
Kfuzzyi(4),0,0.3+Kfuzzyi(4)]);
aa=addmf(aa,'output',1,'SP','trimf',[0,0.3+Kfuzzyi(5),0.6+Kfuzzyi(6)])
;
aa=addmf(aa,'output',1,'MP','trimf',[0.3+Kfuzzyi(7),0.6+Kfuzzyi(8),1+K
fuzzyi(9)]);
aa=addmf(aa,'output',1,'BP','smf',[0.6+Kfuzzyi(10),1]);
Figure(3)
plotmf(aa,'output',1)

%------------------------------------------------------------
                    %Edit rule base
%------------------------------------------------------------

 rulelist=[1 1 1 1 1;
```

```
    1 2 1 1 1;
    1 3 1 1 1;
    1 4 1 1 1;
    1 5 2 1 1;
    1 6 3 1 1;
    1 7 4 1 1;

    2 1 1 1 1;
    2 2 1 1 1;
    2 3 1 1 1;
    2 4 2 1 1;
    2 5 3 1 1;
    2 6 4 1 1;
    2 7 5 1 1;

    3 1 1 1 1;
    3 2 1 1 1;
    3 3 2 1 1;
    3 4 3 1 1;
    3 5 4 1 1;
    3 6 5 1 1;
    3 7 6 1 1;

    4 1 1 1 1;
    4 2 2 1 1;
    4 3 3 1 1;
    4 4 4 1 1;
    4 5 5 1 1;
    4 6 6 1 1;
    4 7 7 1 1;

    5 1 2 1 1;
    5 2 3 1 1;
    5 3 4 1 1;
    5 4 5 1 1;
    5 5 6 1 1;
    5 6 7 1 1;
    5 7 7 1 1;

    6 1 3 1 1;
    6 2 4 1 1;
    6 3 5 1 1;
    6 4 6 1 1;
    6 5 7 1 1;
    6 6 7 1 1;
    6 7 7 1 1;

    7 1 4 1 1;
    7 2 5 1 1;
    7 3 6 1 1;
    7 4 7 1 1;
    7 5 7 1 1;
    7 6 7 1 1;
    7 7 7 1 1];
aa=addrule(aa,rulelist);

aa=setfis(aa,'DefuzzMethod','centroid');      % Defuzzy

writefis(aa,'MPPT_Mamdani');                   % save to fuzzy file
"fuzzy.fis" which can be
```

```
MPPT_MamdaniGA=readfis('MPPT_Mamdani') ;        % Load Fuzzy Inference
System from file


ke=num2str(Kfuzzyi(1));
kce=num2str(Kfuzzyi(2));
ku=num2str(Kfuzzyi(3));

set_param('MPPT_Mam/err_g','Gain',ke);    % Error gain
set_param('MPPT_Mam/ce_g','Gain',kce);    % Change in error gain
set_param('MPPT_Mam/out_g','Gain',ku);    % Output gain


[tt,xx,yy]=sim('MPPT_Mam');
  clear tt;
  clear xx;
  clear yy;
BsJ=0;

ts=1.000000000000000e-06;
 for k=1:1:201
      timef(k)=k*ts;
      Ji(k)=(e_abs.time(k)*e_abs.signals.values(k));
      BsJ=BsJ+Ji(k);
  end


end
% Figure(4)
% plot(yout)
% xlim([0 0.5e4])
% ylim([160 210])
% grid on.
```