EFFICIENCY OPTIMIZATION OF A STANDALONE SOLAR ENERGY SYSTEM USING FUZZY BASED MPPT

A THESIS IN MECHATRONICS

Master of Science in Mechatronics Engineering

Presented to the faculty of the American University of Sharjah

College of Engineering

in partial fulfillment of

the requirements for the degree

MASTER OF SCIENCE

by

AHMAD ABDEL JAWAD AL NABULSI

B.S. 2009

Sharjah, UAE

January 2012

We approve the thesis of Ahmad Abdel Jawad Al Nabulsi

Date of signature

_____ _____

Dr. Rached Dhaouadi
Professor
Department of Electrical Engineering
Thesis Advisor

_____ _____

Dr. Abdul-Rahman Al-Ali
Professor
Department of Computer Science & Engineering
Graduate Committee

_____ _____

Dr. Ali Jhemi
Assistant Professor
Department of Mechanical Engineering
Graduate Committee

_____ _____

Dr. Rached Dhaouadi
Director
Mechatronics Graduate Program

_____ _____

Dr. Hany El Kadi
Associate Dean
College of Engineering

_____ _____

Dr. Yousef Al Assaf
Dean
College of Engineering

_____ _____

Dr. Khaled Assaleh
Director
Graduate Studies

EFFICIENCY OPTIMIZATION OF A STANDALONE SOLAR ENERGY SYSTEM USING FUZZY BASED MPPT

Ahmad Abdel Jawad Al Nabulsi, Candidate for the Master of Science Degree

American University of Sharjah, 2012

ABSTRACT

This work primarily addresses the design and implementation of a standalone rotary sun tracker with dual maximum power point tracking (MPPT). The objective is to improve the efficiency of the PV system by introducing both physical and MPP tracking. A two-axis rotation mechanism is designed to track the sun over both the azimuth and elevation angles. The position of the sun is determined using a time-based approach, in which preprogrammed derived equations are used to calculate the sun's azimuth and elevation angles continuously throughout the day. The calculated angles are used as set points for two closed-loop control systems with PI controllers implemented in the DSP. The panel power is passed through a power converter that implements a fuzzy logic controller (FLC) based perturb and observe (P&O) MPPT algorithm that keeps the system power operating point at its maximum. In the conventional P&O, the converter input reference voltage is perturbed in fixed steps until the maximum power is reached. However, depending on the step size, the system operating point will oscillate around the MPP resulting in a loss of energy. The proposed controller eliminates the resulting oscillations by adaptively modifying the perturbation step using a fuzzy logic controller. The performance of the proposed system is validated through experiments with a 150-Watt PV system.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

Words are completely inadequate to convey my appreciation for the many people who have helped me throughout the process of this work.

This work owes its existence to the support, encouragement, and assistance of my advisor Dr. Rached Dhaouadi. In my five years of research in both undergraduate and graduate studies, he not only supported me technically but also taught me how to strive for perfection and never to compromise.

This thesis would not have been also possible without the support, affection, and prayers of my family and especially my parents to whom I feel completely indebted for their unlimited love and blessings.

Moreover, I would like to thank Dr. Abdul-Rahman Al-Ali and Dr. Ali Jhemi for being part of the graduate committee. In addition, I would like to thank Mr. M.R. Narayanan and Mr. Ricardo De Jesus for their time and help during the work.

This thesis comes as a continuation to previous groups work to whom I am gratefully thankful. I would like also to acknowledge the Emirates Foundation (EF) for its financial support during the early stages of this work.

This work concludes a long chapter of my life that lasted for more than seven years at the American University of Sharjah, so before leaving I would like to express my deepest gratitude to the university's administration, faculty, and staff for giving me this unforgettable experience. Special thanks go to Mrs. Salwa Mohamed and Mrs. Lalitha Murugan for their moral support.

Finally yet importantly, I would like to thank all my friends and colleagues who aided me in both the good and hard times.

DEDICATION

This work is dedicated to my parents Samar and Abdel Jawad to whom I owe everything I am today.

CHAPTER 1 : INTRODUCTION

1.1 MOTIVATION

Energy has been one of the most important driving forces in our fast growing world. Nations are in an endless search for new energy sources. This search passed through many revolutions, from the use of fire and coal to the discovery of fossil fuels such as oil and natural gas. However, this search was faced by many challenges, since many of these resources are expensive, destructive to the environment, or ceasing to exist in the near future.

Significant progress has been made over the last few years in the research and development of renewable energy systems such as wind, sea wave and solar energy systems. Among these resources, solar energy is considered nowadays as one of the most reliable, daily available, and environment friendly renewable energy source [1], [2].

The amount of energy in the sunlight reaching the earth's surface is equivalent to around 10,000 times the world's energy requirements. Consequently, only 0.01 percent of the energy in sunlight would need to be harnessed to cover mankind total energy needs [3]. Given that the sun shines on this part of the globe nearly the entire year, many new projects are intended to make use of solar energy as a backup source to the existing power system.

Another incentive of the spread of such projects is the fact that the cost trend of photovoltaic (PV) systems is descending, while the fuel price is ascending [4]. The reason behind this trend is the growing mass production and market of PV systems. The advancement in solar systems is based on the new technological advances in the industries of photovoltaic cells, power electronics switches, microcontrollers, and

computer-based simulation packages. Solar systems not only can serve as a backup to existing energy systems, but they can also be easily integrated with large grid systems, or can be used as standalone systems serving as independent energy sources.

United Arab Emirates is one of the few Middle East countries that took the initiative and started the journey to become one of the world leaders in renewable energy research and development. One of the first initiatives is building Masdar City the first zero carbon city to be powered entirely by renewable energy. This city will be also hosting the International Renewable Energy Agency (IRENA) headquarters.

The use of Intelligent Controllers as alternate approach to conventional techniques or as components of integrated systems is becoming more and more popular nowadays [5]. These intelligent techniques focus on the nonlinear characteristics of systems, and as a result, provide a good alternative for standard nonlinear controllers [6]. The complexity of these techniques is overlooked while comparing them with the improvements yielded in the system response, steady state value, and transients. Moreover, the implementation of such techniques is becoming easier and less expensive with the advancements in digital signal processing (DSP) devices.

One of the recent applications of such intelligent controllers is the efficiency optimization of photovoltaic energy systems. These techniques are used to maximize the power output of solar systems as well as improve their robustness against external disturbances.

1.2 PROBLEM STATEMENT

In line with the global initiative toward the design and development of PV systems as alternative energy sources, this work will investigate the potential of using solar energy systems in the UAE. One of the drawbacks of PV systems is their low efficiencies compared to their cost. In order to overcome these drawbacks, maximum power should be extracted from these systems.

The aim is to develop an efficient standalone photovoltaic system. This system employs a new digital control scheme using fuzzy-logic and a dual maximum power point tracking (MPPT) controller. The first MPPT controller is an astronomical two-axis sun tracker, which is designed to track the sun over both the azimuth and elevation angles and obtain maximum solar radiation at all times. The second MPPT

algorithm controls the power converter between the PV panel and the load and implements a new fuzzy-logic (FLC) based perturb and observe (P&O) scheme to keep the system power operating point at its maximum. Such systems are exceedingly demanded in remote areas where it is difficult to connect to the grid system.

1.3 CONTRIBUTIONS

This work will explore the effectiveness of intelligent and digital control techniques for PV system efficiency optimization. These techniques combine both physical as well as fuzzy-based perturb and observe MPPT tracking techniques. Furthermore, this work will offer a stability analysis for the FLC based P&O controller.

Though literature is rich with PV systems implementations, no work was based on actual outdoor testing in this part of the world. This work will use experimental data to investigate the potential of solar energy in the gulf region and the effects of the harsh environment on PV systems efficiencies.

1.4 THESIS OUTLINE

In this thesis, a detailed literature review about tracking techniques will be presented in chapter 2. Chapter 3 will focus on PV systems modeling and simulation featuring a MATLAB/Simulink full simulation of the overall standalone PV system. Chapter 4 will discuss the design and implementation of fuzzy logic controller systems. The proposed FLC controller will be fully investigated in chapter 5 highlighting a simulation comparison between the proposed controller and the conventional ones. Chapter 6 will verify the proposed technique effectiveness through presenting the PV system hardware implementation and experimental results. Finally, chapter 7 will conclude the work discussing the work limitations along with the future work.

CHAPTER 2 : RELATED WORK

2.1 MAXIMUM POWER POINT TRACKING TECHNIQUES

MPPT is a real-time control scheme applied to the PV system power converter in order to extract the maximum power possible from the PV panel. The MPPT working principle is based on the maximum power transfer theory. The power delivered from the source to the load is maximized when the input resistance seen by the source matches the source resistance. Therefore, in order to transfer maximum power from the panel to the load the internal resistance of the panel has to match the resistance seen by the PV panel. For a fixed load, the equivalent resistance seen by the panel can be adjusted by changing the power converter duty cycle [7]-[9]. Figure 2.1 illustrates the PV system equivalent circuit and the input resistance concept.

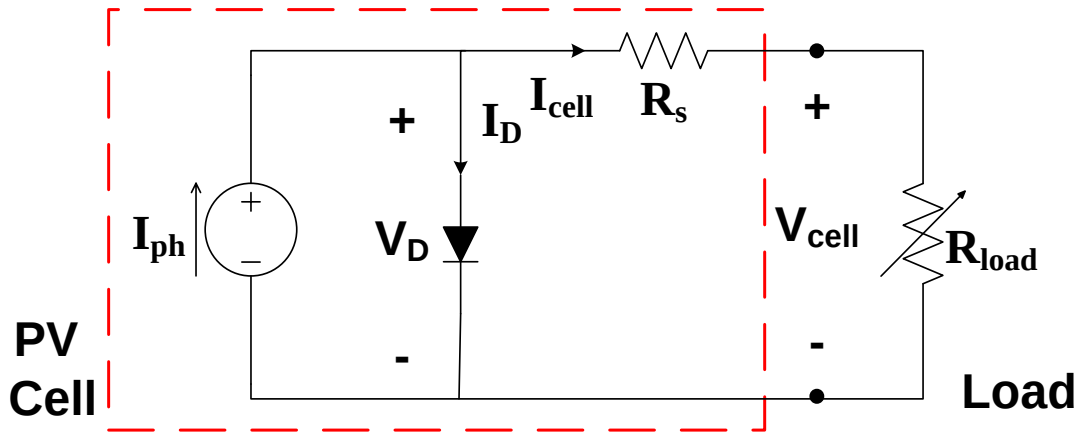


Figure 2.1: PV system circuit model

The literature is rich with various MPPT techniques based on different topologies and with varying complexity, cost, and overall produced efficiency. MPPT techniques can be classified into two categories; basic MPPT techniques and artificial intelligence based techniques.

2.1.1 Basic MPPT Techniques

2.1.1.1 Hill Climbing / Perturb and Observe

Two most commonly used MPPT techniques are the Hill Climbing (HC) and the Perturb and Observe (P&O) [10]-[20]. The working principle of the HC-MPPT technique is summarized as follows. The PV panel voltage and current are measured at fixed sampling intervals and fed to the digital controller to calculate the PV panel power. The power increment $(\Delta P = P(k) - P(k-1))$ is calculated at each sampling time and according to its sign, the duty cycle is incremented or decremented in fixed steps depending on the panel voltage and power values until the maximum power point (MPP) is reached.

The P&O technique shares the same HC concept of operation, but with an additional PI control loop. In the P&O, the converter input reference voltage is the perturbed variable and the duty cycle is computed through an additional PI control loop. The additional control loop results in an increase in the P&O efficiency, as the system demonstrates a faster dynamic performance and better-regulated PV output voltage compared to HC. The operation of the HC and the P&O technique is explained by the flowcharts given in Figure 2.2 and Figure 2.3 respectively.

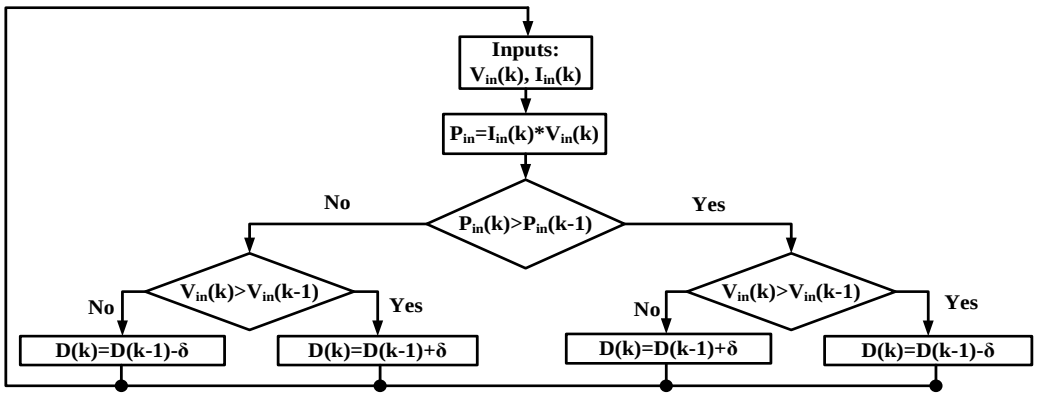


Figure 2.2: HC method flow chart

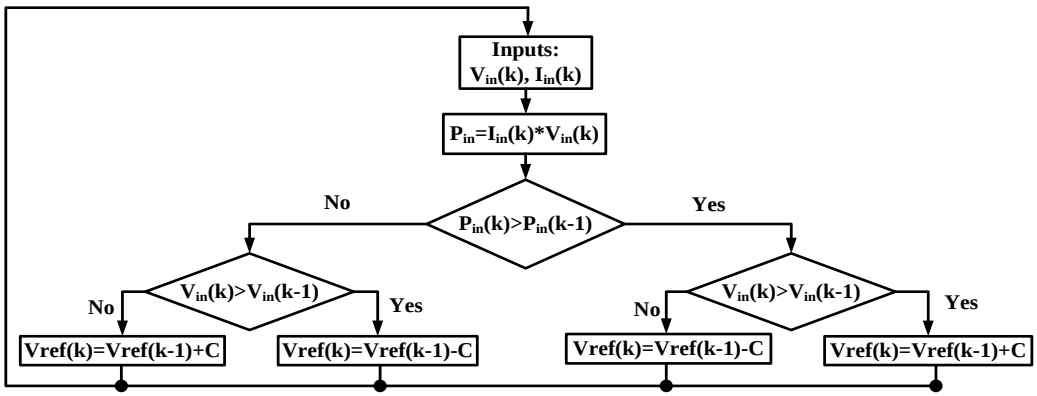


Figure 2.3: P&O method flow chart

Figure 2.4 further illustrates the HC and the P&O methods operations. If the power converter operating point moved from 1 to 2, as the power at 2 is greater than the power at 1 and the voltage at 2 is less than the voltage at 1, the reference voltage will decrease (duty cycle will increase) by a fixed step. This decrease will move the operating point to the left in the direction of the maximum point. The reference voltage will keep decreasing and the operating point will continue climbing the PV curve hill. If the step size is large, the operating point may exceed the MPP when moving from point 4 to reach point 5. The power at point 5 is less than the power at point 4 and the voltage at point 5 is also less than the voltage at power 4 as a result the reference voltage will increase (duty cycle will decrease) with a fixed step C and the operating point will move to the other direction reaching point 6. The same process will keep repeating, as a result the operating point will oscillate around its maximum.

This system is commonly used because of its simplicity and ease of implementation [10]-[13]. Furthermore, P&O (with a small step size) in nominal conditions can have MPPT efficiencies mostly the same like other complex techniques, and still easier implementation [14].

However, the drawback of this technique is that the operating point of the PV array oscillates around the MPP. Therefore, the power loss may increase. Furthermore, when the sun insolation changes rapidly, the P&O method probably fails to track the MPP [6]. Another possible disadvantage is that the MPPT may not be able to locate the MPP as the amount of sunlight decreases, because the PV curve flattens out [12].

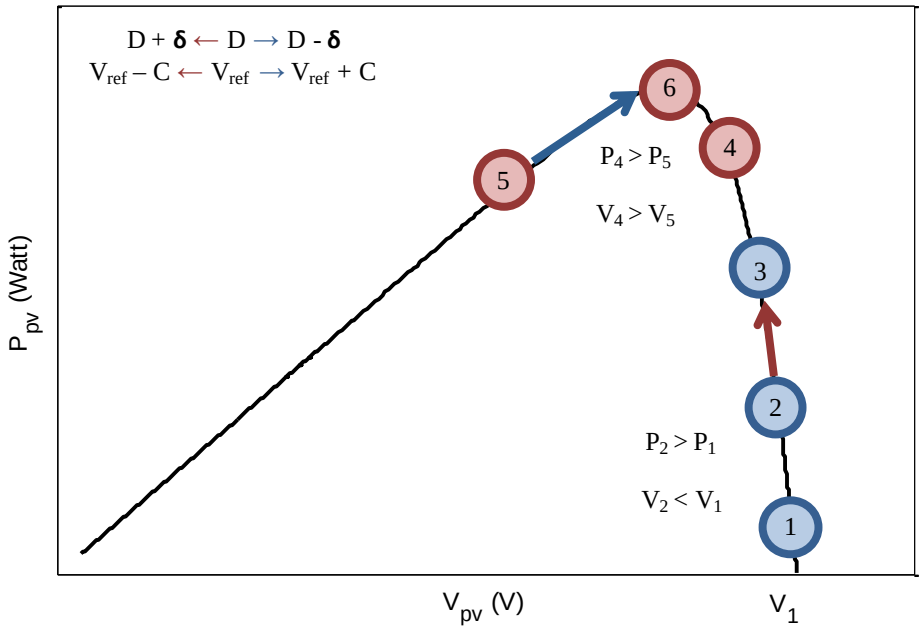


Figure 2.4: HC/P&O method operation

In order to overcome these drawbacks and improve the P&O response, many techniques are suggested and investigated in the literature such as:

- The introduction of three-point weight comparison P&O algorithm [15], [16].
- The use of modified adaptive techniques [17] – [20].

In the three-point weight comparison method, the duty cycle is perturbed in both sides of the operating point and then the power slope at the current operating point is compared to the power slope at the two perturbed points. If the sign of the slope is changed between the two perturbed points then the current operating point is the MPP and the perturbation stops. Although this technique results in less oscillation, it requires an additional point reading resulting in more computations and a slower system.

In the modified adaptive techniques, the step size C of the P&O/HC is modified based on the operating point location. The system starts with a large step size but starts decreasing it till the operating point reaches to the MPP. The main difficulty of this technique is determining the suitable factor by which the step size has to be modified as it depends on the PV system configuration.

2.1.1.2 Incremental Conductance

Another commonly cited MPPT algorithm is the incremental conductance (INC) technique [11]-[14], [21]-[24]. This method tracks the maximum power point

by comparing the solar array incremental and instantaneous conductances, the operation of this technique is explained in the flowchart in Figure 2.5.

The PV panel voltage and current are measured at fixed sampling intervals and fed to the digital controller to calculate the PV panel power. The PV panel incremental conductance is estimated by measuring small changes in array voltage and current. The PV panel instantaneous conductance is calculated by dividing the array current by the voltage. Once these variables are updated, the method tracks the maximum power point by comparing the incremental and instantaneous conductances of the solar array until the maximum power point (MPP) is reached, as illustrated in Figure 2.6.

If the power converter operating point moved from 1 to 2 where the derivative $\Delta P / \Delta V$ is negative, the reference voltage will decrease (duty cycle will increase) by a fixed step C resulting in the operating point to be moving to the left in the direction of the maximum point. The reference voltage will keep decreasing and the operating point will continue incrementing until it reaches point 5. The derivative $\Delta P / \Delta V$ at 5 becomes positive as a result the reference voltage will increase (duty cycle will decrease) with a fixed step C and the operating point will move to the other direction reaching point 6. The same process will keep repeating, and results in the operating point oscillations around its maximum.

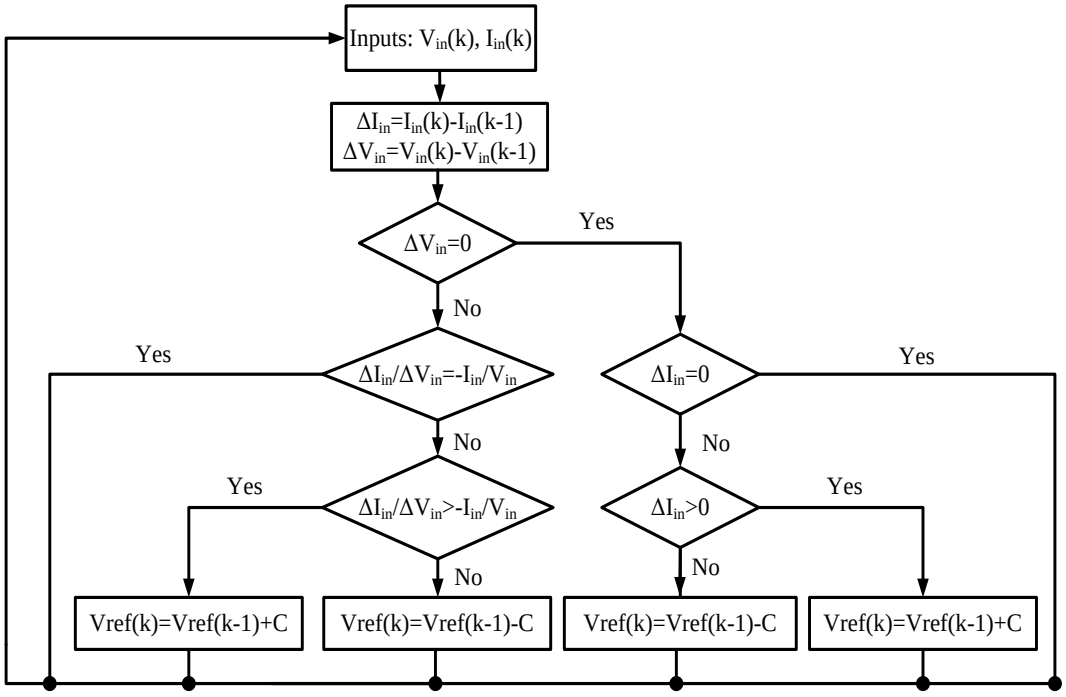


Figure 2.5. INC method flow chart

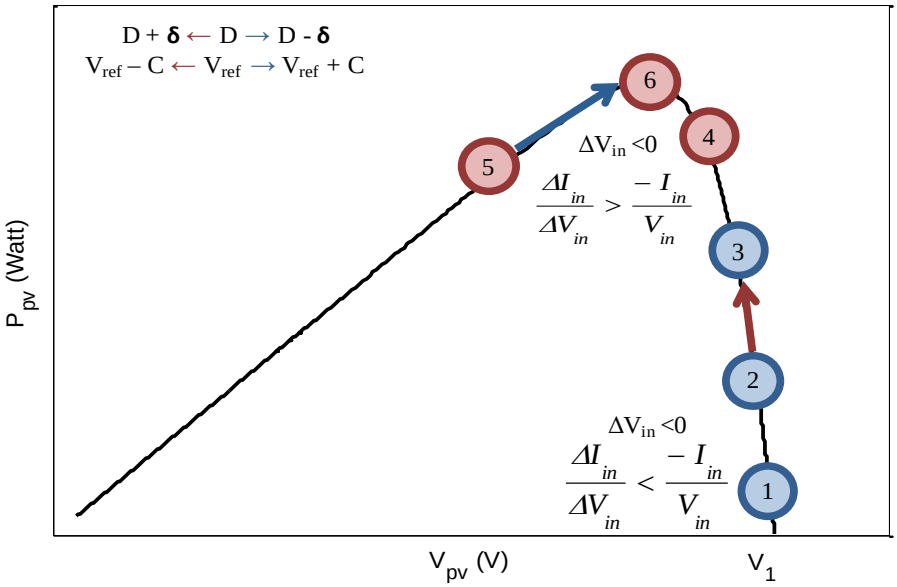


Figure 2.6. INC method operation

The INC method used to have complex analog implementation but it is easier nowadays with DSPs [13]. However, the INC method requires more calculations than P&O method and thus slows down the sampling speed [6], [12], [14]. Finally, the null value of the slope of the PV array power versus voltage curve seldom occurs [6].

Existing improvements in literature of this method focuses on modifying the step size of the algorithm [23], [24]. However, the high complexity of the method requires high sampling accuracy and fast control speed, which may result in a high cost system [6].

2.1.1.3 Extremum Seeking

A third suggested basic technique is the extremum seeking method (ES) [25]-[28]. Unlike the HC/P&O and the INC methods, the perturbation step in ES method is based on a periodic perturbation α sin (ωt), as shown in Figure 2.7.

The main advantage of such periodic perturbation is that it produces an oscillatory behavior around the equilibrium point without previous knowledge of the system parameters. The ES method shows also a better response to rapidly changing weather conditions compared to the previously discussed techniques. However, once it reaches the PV curve extreme point it enters an oscillation cycle around the MPP resulting in a loss of energy. A suggested solution to the oscillation problem is the use of switched control scheme based ES method in which the perturbation step decays exponentially once it has reached the MPP cancelling the steady state oscillations [25].

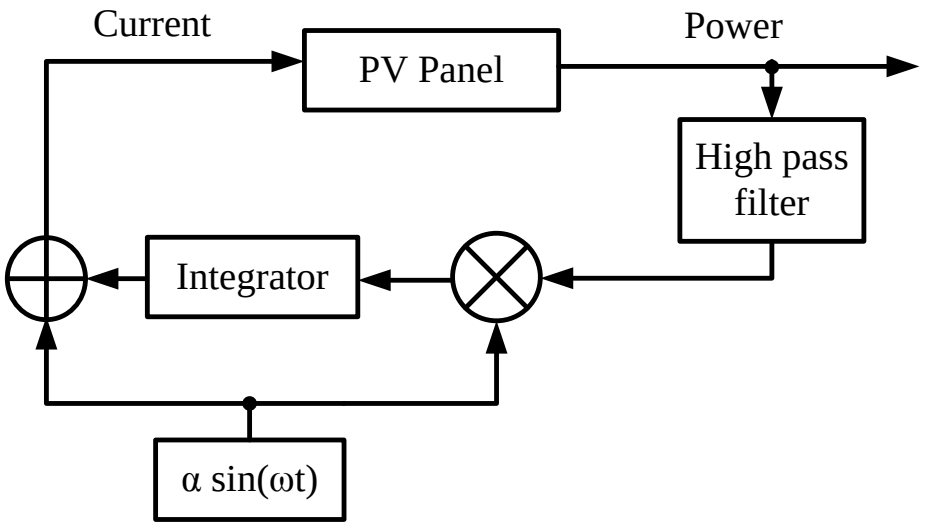


Figure 2.7: ESS method block diagram

2.1.1.4 Constant Voltage Method

Another basic MPPT technique is the Constant Voltage (CV) Method [6], [10], [12]-[14], [29], [30]. In this method, the power converter output voltage is assumed constant. As a result, the ratio $\frac{V_{MPP}}{V_{OC}}$ can be easily calculated. A typical value for this ratio is 76%. In order to reach this value the PV output voltage is increased until it reaches 76% of the open circuit voltage, which is measured after temporarily setting the PV array current to zero. This cycle is repeated again after a specific time. Figure 2.8 further illustrates the algorithm of this technique.

The disadvantages of this method are mainly the energy dissipation by disconnecting the load while measuring the open circuit voltage and the validity of the constant PV output voltage assumption [10]. In order to solve the energy dissipation problem, a pilot PV cell can be added to measure the open circuit voltage but on the expense of a higher system cost [29].

This technique can be also combined with the P&O or the INC technique to give it a smooth start by determining a starting point for the reference voltage that is close to the MPP voltage [30]. However, if the starting point is not selected properly, a system stability problem would occur once the control unit switches between the two controllers [6].

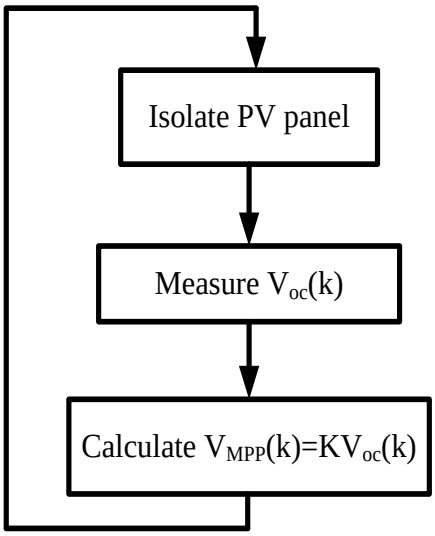


Figure 2.8: CV method flowchart

2.1.2 Artificial Intelligence Based MPPT Techniques

Recently, more focus is given to artificial intelligence based MPPT techniques, as they focus on the nonlinear characteristics of the PV array providing a good alternative for basic MPPT techniques [5]. Two of the mostly commonly used techniques are neural networks and fuzzy logic controllers.

2.1.2.1 Neural Networks

In this technique, trained neural networks (NN) are used to find the solution for the nonlinear PV system model, outputting the MPP reference voltage [5], [6], [11], [13], [31]-[34]. The system reaches this reference voltage using either a fuzzy or a PI controller. The operation of this technique is explained in the block diagram in Figure 2.9. The PV panel voltage, current, and optionally other parameters such as the temperature and insolation are sensed and then fed to the digital controller. Once these parameters are recorded, the trained neural network outputs the MPP reference voltage. The system reaches this reference voltage using either a fuzzy or a PI controller. In this technique, the knowledge of the internal system parameters is not required, so less computational effort is needed [34]. However, NN has to be specifically trained for the PV array for which it will be used. The characteristics of a PV array also changes with time, implying that the neural network has to be periodically trained [6], [12]. One of the improvements to the NN technique is the use of Adaptive Neuro-Fuzzy Inference System (ANFIS), which is designed as a combination of the Sugeno fuzzy model and neural networks [33].

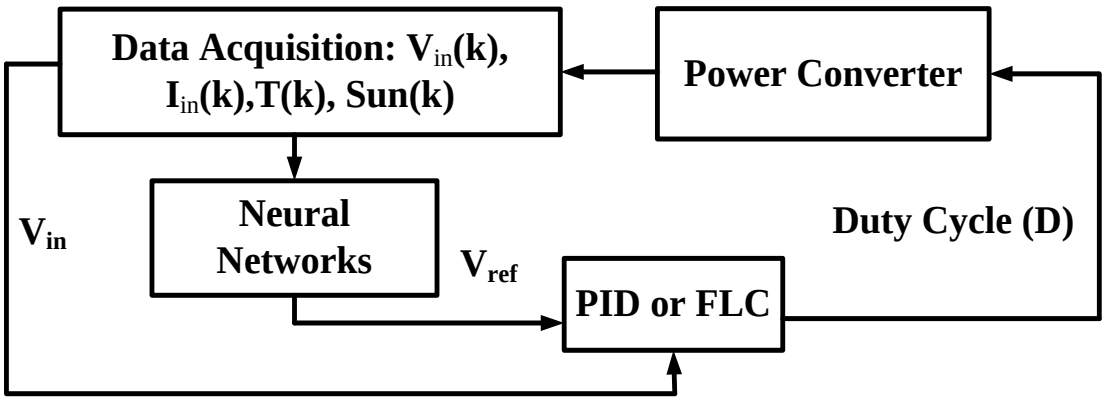


Figure 2.9. NN method block diagram

2.1.2.2 Fuzzy Logic Controller

Fuzzy Logic has also been documented in the literature for its performance in MPPT application [5], [6], [11], [13], [34]-[46]. The operation of this technique is explained in the block diagram shown in Figure 2.10. As shown in Figure 2.10, the PV panel voltage and current are sensed and then fed to the digital controller, which calculates the input power. Next, the inputs of the FLC, which are typically the Error (Er) = $\Delta P/\Delta V$ or $\Delta P/\Delta I$ and the change in Error (CEr), are calculated. The fuzzy system takes Er and CEr as inputs and outputs the duty cycle D.

Fuzzy logic controller (FLC) based MPPT systems have a higher efficiency compared to NN based controllers [34]. Moreover, FLC based systems perform well under varying atmospheric conditions [11]. However, their effectiveness highly depends on the experience of the control engineer in choosing the right inputs and tuning the rule base table [6], [11], [34].

Various techniques are proposed in literature to improve the FLC performance such as the use of non-uniform membership functions (denser at the center in order to provide more sensitivity against variation in the PV terminal voltage) [32], [38].

Another suggested technique is the use of adaptive fuzzy logic controls, which constantly tunes the membership functions and the rule base table so that the optimum performance is achieved [11]. Some of these adaptive fuzzy techniques are the use of PI controllers [39], adaptive tuning factors [40]-[42], genetic algorithms [43], variable universe controllers [44], [45], and self-organizing controllers [46].

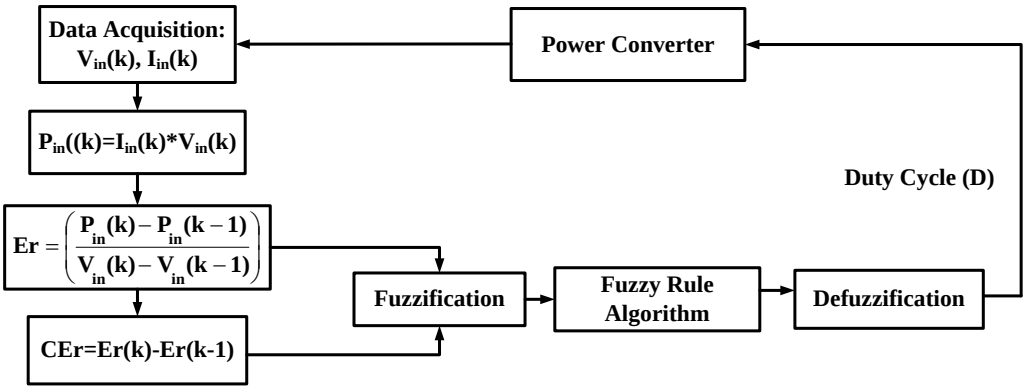


Figure 2.10. FLC method block diagram

2.2 PHYSICAL TRACKING TECHNIQUES

In addition to MPPT, physical tracking can be used to optimize the efficiency of PV systems. Physical tracking involves aligning the PV panels to be orthogonal to the sunrays throughout the day in order to receive maximum solar radiation. This tracking can be done manually or automatically [47].

In manual tracking, the PV panel orientation is manually changed at the beginning of each season to a predetermined angle [48]. In the case of automatic tracking, the PV panel is mounted on a single-axis or multi-axis tracking mechanism and is controlled to follow the sunrays trajectory throughout the day. The tracking mechanisms can be classified into three types namely, passive method, optical method and the astronomical method. The general features that characterize an efficient tracking mechanism are ability to cover a wide range of space, robustness against wind disturbances, and low power consumption.

2.2.1 Passive Method

Passive tracking is based on the heating properties of gas matters such as Freon [49]. As shown in Figure 2.11, the tracker is composed of two cylindrical tubes fitted on the edges of the panel; these tubes are filled with a fluid under partial pressure. Sun heating will increase the gas pressure of the sun side cylinder causing an unbalance in the tracker. As a result, the tracker will tilt towards the sun.

This type of system is relatively cheap and needs little maintenance. However, it has lower efficiency compared to other types of trackers, especially at low temperatures. Moreover, they have not yet been widely accepted by consumers [47].

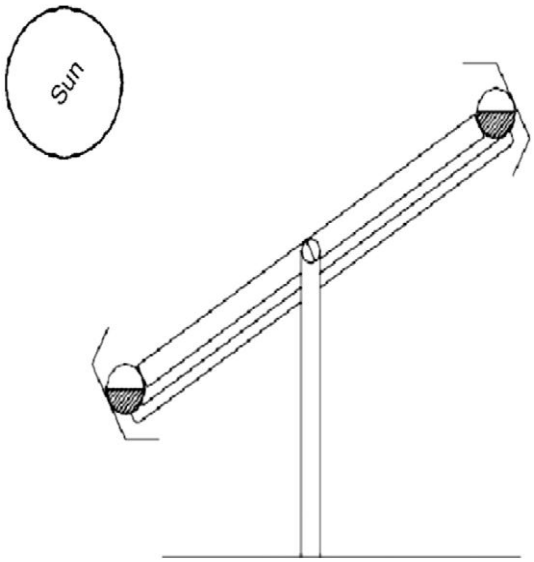


Figure 2.11: Passive tracker

2.2.2 Optical Method

The second method of tracking is the optical or electro-optic trackers. This method uses feedback sensors such as photo-sensors, current, voltage sensors, and auxiliary cells to determine the panel reference position [50]-[53]. A closed loop position control system based on FLC or PID is next used to produce the actuators control commands. The drawback for such a system is that it is very sensitive to atmospheric conditions (clouds shading) and might not be able to continue tracking the sun in a cloudy day. The block diagram of this method is shown in Figure 2.12.

2.2.3 Astronomical Method

The third tracking method is the astronomical method, which employs the longitude and latitude data of a given location to determine the sun current position [54]-[56].

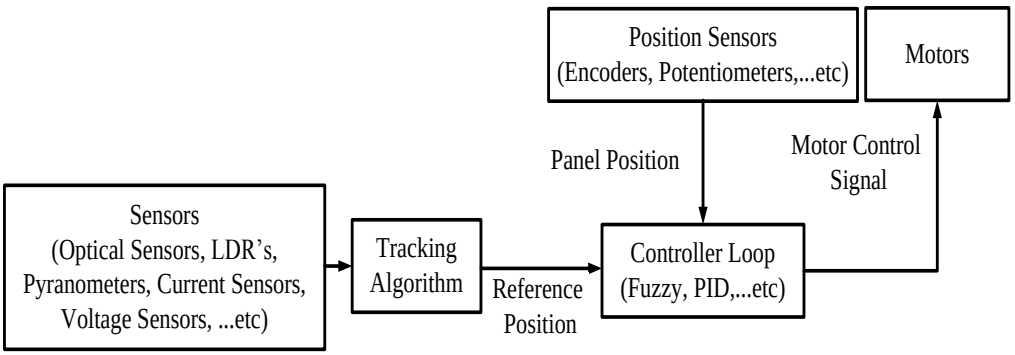


Figure 2.12: Optical tracking method block diagram

The main advantage of this method over the optical method is that it involves simpler programming, reduced implementation cost, and lower power consumption as the need for additional sensors is eliminated. This method also provides high degree of accuracy and is not sensitive to atmospheric conditions. The block diagram of this method is described in Figure 2.13.

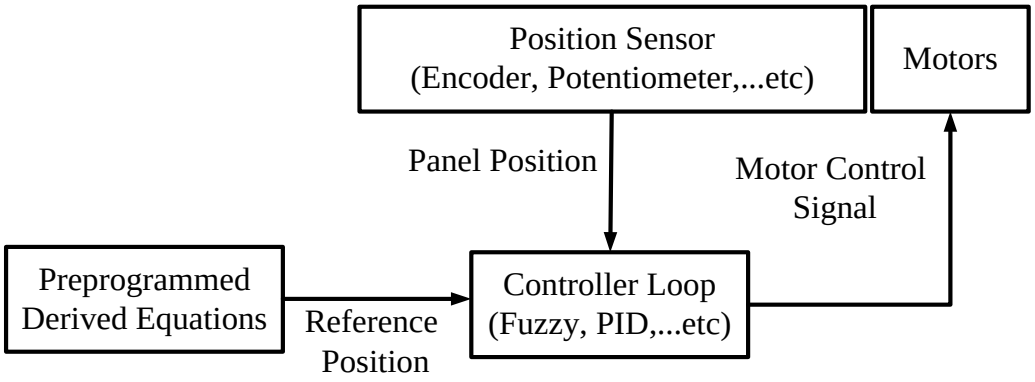


Figure 2.13: Astronomical tracking method flow chart

CHAPTER 3 : PV SYSTEM MODELING

3.1 INTRODUCTION

The proposed standalone photovoltaic system consists of four main blocks: PV panel, solar tracker, power converter, and the MPPT controller. The following sections will describe the modeling of each component.

3.2 SOLAR TRACKER MODEL

One of the essential characteristics of solar systems is that their power is always proportional to the amount of sun irradiance hitting the panels orthogonally. Figure 3.1 shows the major sun position angles [57]. These angles are the solar azimuth angle (*Az*), the solar elevation angle (*Elv*) and the solar zenith angle (θ_z). The solar elevation is the apparent angular height of the sun in the sky as the observer faces it. The solar azimuth angle is the angle between the line from the observer due south and the horizontal projection of the sun's rays. The solar zenith angle is the angle from the zenith (which is the point directly overhead of the observer) to the sun's position in the sky.

For a fixed panel the optimum tilt angle is seasonally calculated based on the PV system location latitude. Table 3-1 shows this angle for Sharjah city given that panels has to be facing true south in Northern hemisphere and true north in the Southern hemisphere [48], [58]. The optimum tilt angel is:

$$\theta_{opt} \text{ (in winter)} = 0.9 \times \text{Latitude} + 29° \quad (3.1)$$

$$\theta_{opt} \text{ (in spring and autumn)} = \text{Latitude} - 2.5° \quad (3.2)$$

$$\theta_{opt} \text{ (in summer)} = \theta_{opt} \text{ (in winter)} - 52.5° \quad (3.3)$$

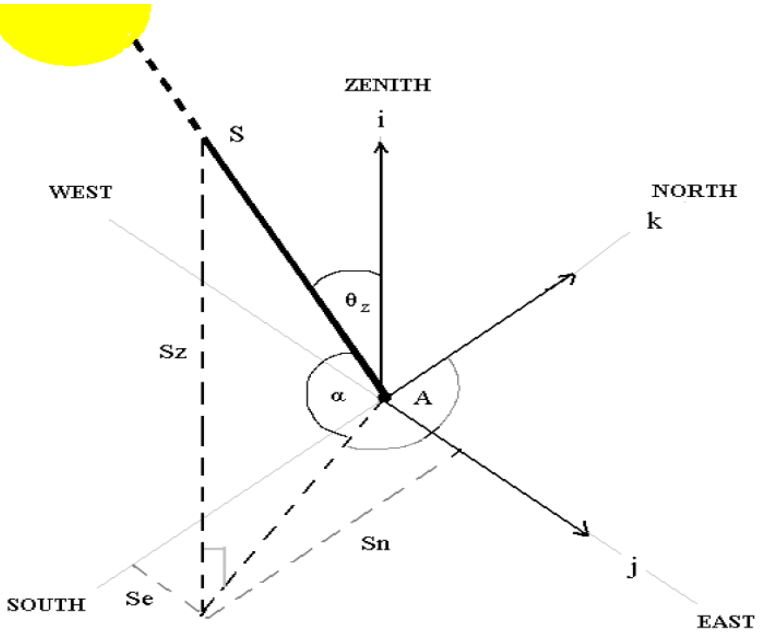


Figure 3.1: Sun position angles

Table 3-1: Optimal seasonal solar panel fixture angles for sharjah city

Sharjah Latitude	25.26 °	Sharjah Longitude	55.21°
θopt (winter)			51.73 °
θopt (spring and autumn)			22.76 °
θopt (summer)			-0.77 °

For a tracking panel the azimuth and elevation angles have to be changing throughout the day as the sun moves from east to west (due to the earth rotation). They also have to change from a day to another as the earth rotates around the sun.

As a result, each day of the year has a unique plot for the azimuth and elevation angle with time. In order to calculate the azimuth and elevation angles at a certain location the following equations are used [59]:

$$\sin Elv = \cos(lat)\cos(hr)\cos(dec) \tag{3.4}$$

$$\cos Az = \frac{1}{\cos\alpha}(\cos(dec)\ \sin(lat)\cos(hr) - \sin(dec)\cos(lat)) \tag{3.5}$$

Where α is the sun's elevation angle, *Az* is the sun's azimuth angle, *lat* is the local latitude, *hr* is the hour angle, *dec* is the sun declination angle throughout the year.

The hour angle (*hr*) of a point on the earth's surface is the angle through which the earth would turn to bring the meridian of the point directly under the sun. Since the earth is rotating, this angular displacement represents time. At solar noon, the hour angle is zero. The hour angle expresses the time of day with respect to solar noon. One hour of time is represented by 360÷24=15 degrees of hour angle. As part of the convention, the hour angle is negative before solar noon and positive after solar noon [59].

The sun's declination angle (*dec*), is the angular distance of a sun's rays north (or south) of the equator. It is the angle between a line extending from the center of the sun to the center of the earth and the projection of this line upon the earth's equatorial plane. The declination is positive when the sun's rays are north of the equator and negative when they are south of the equator [59]. Solar radiation calculations must be made in terms of solar time, therefore, the local clock time has to be converted to solar time. Time is generally expressed on an hour scale from zero to 24 (midnight is 0h and noon is 12h). Local Civil Time is reckoned from the precise longitude of the observer. On any particular meridian, Local Civil Time is more advanced at the same instant than on any meridian further west and less advanced than on any meridian further east. The difference amounts to 1/15 hour (4 minutes) of time for each degree difference in longitude [59].

Clock time generally differs from Civil Time. Clocks are usually set for the same reading throughout an entire zone covering about 15 degrees of longitude. For example, the U.A.E belongs to the time zone (GMT +4) which corresponds to a longitude of 60 degrees, whereas the exact longitude of Sharjah is 55.21 degrees.

Time as measured by the apparent diurnal motion of the sun is called Local Solar Time or Solar Time. Whereas a civil day is precisely 24 hours, a solar day is slightly different due to irregularities of the earth's rotation, obliquity of the earth's orbit and other factors [59].

In order to solve equations (3.4) and (3.5) the hour angle (*hr*) and the sun declination angle (*dec*) has to be determined using the following equations:

$$dec = 23.45 \sin\left[\frac{360}{365}(284 + n)\right] \tag{3.6}$$

$$hr = 15(LTS - 12) \tag{3.7}$$

where LTS is the local solar time

$$LTS = CT + \frac{1}{15}(L_{std} - L_{loc}) + E \tag{3.8}$$

Where CT is the clock time, L_{std} is the standard longitude for the solar time zone, L_{loc} is the actual longitude for the location, *and Tdef* is the difference between the local solar time and civil time.

$$T_{def} = 0.165\sin 2\left(\frac{360(n-81)}{364}\right) - 0.126\cos\left(\frac{360(n-81)}{364}\right) - 0.025\sin\left(\frac{360(n-81)}{364}\right) \tag{3.9}$$

3.3 PV PANEL MODEL

The PV panel model used in our analysis is the single-diode model. This model represents the solar cell in its simplest form as a PN junction followed by a series resistance as shown in Figure 3.2. [60] - [62].

Analyzing the junction cell circuit given that the photocurrent current I_{ph} equals zero gives the following equations:

$$V_{cell} = V_D - IR_s \tag{3.10}$$

$$I_{cell} = -I_D = I_{ph} - I_o(e^{K_{PV}(V_{pv}+I_{pv}R_s)} - 1) \tag{3.11}$$

where V_{cell} is the PV cell terminal voltage, V_D is the diode voltage, I_{cell} is the PV cell terminal current, I_D is the diode current. $K_{PV} = q/pKT$ with the electronic charge $q = 1.6\times10^{-19}$ C, $K = 1.3805\times10^{-23}$ J/K (Boltzmann's constant), T cell temperature, and the ideal p-n junction characteristic factor p=1.3 for monocrystalline solar cells. I_o is the saturation current in diode reverse-biased direction as and R_s is the panel series resistance.

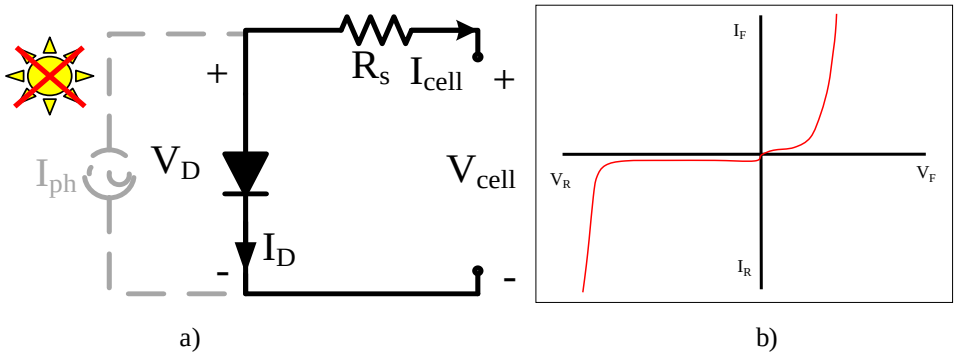


Figure 3.2: a) The un-illuminated junction circuit b) The un-illuminated junction characteristic

$$I_o = I_{rr}\left(\frac{T}{T_r}\right)^3 e^{\left(\frac{qV_g\left(\frac{1}{T_r}-\frac{1}{T}\right)}{pK}\right)} \tag{3.12}$$

where I_{rr} is the reverse saturation current at the reference temperature T_r. V_g is the band-gap voltage of the semiconductor making up the cell, which equals 1.12 eV for Si monocrystalline cells.

$$R_s = -\frac{dV}{dI_{V_{oc}}} - \frac{1}{X_V} \tag{3.13}$$

where:

$$X_V = I_o \times \frac{q}{pKT}\left(e^{\frac{qV_{oc}}{pKT}}\right) \tag{3.14}$$

When the cell is exposed to the sun radiation, the level of the diode characteristic curve is shifted by the magnitude of the photocurrent in the reverse-biased direction as shown in Figure 3.3. As a result, equation (3.11) is altered to include I_{ph} and describe the operation of the illuminated solar cell circuit.

$$I_{cell} = I_{ph} - I_D = I_{ph} - I_o\left(e^{K_{PV}(V_{panel}+IR_s)} - 1\right) \tag{3.15}$$

where I_{ph} is the Photocurrent

$$I_{ph} = (I_{sc} + K_I(T - T_r))\frac{\lambda}{100} \tag{3.16}$$

where I_{sc} is the short-circuit cell current at reference temperature and insolation, K_I in (mA/K) is the short circuit current temperature coefficient, λ is the insolation in (MW/cm), and I_o is the saturation current in diode reverse-biased direction.

The PV panel model can be computed using (2) by taking into consideration the number of cells connected in series N_s and in parallel N_p as follows: The PV panel model can be computed using equation (3.15) by taking into consideration the number of cells connected in series N_s and in parallel N_p.

$$I_{panel} = N_p I_{ph} - N_p I_o \left(e^{\frac{K_{pv}(V_{panel} + IR_s)}{N_s}} - 1 \right) \tag{3.17}$$

The PV panel power is computed by multiplying equation (3.14) with the PV voltage V_{panel} as shown in Figure 3.4.

$$P_{panel} = V_{panel} \times I_{panel} = N_p I_{ph} V_{panel} - N_p I_o V_{panel} \left(e^{\frac{K_{pv}(V_{panel} + IR_s)}{N_s}} - 1 \right) \tag{3.18}$$

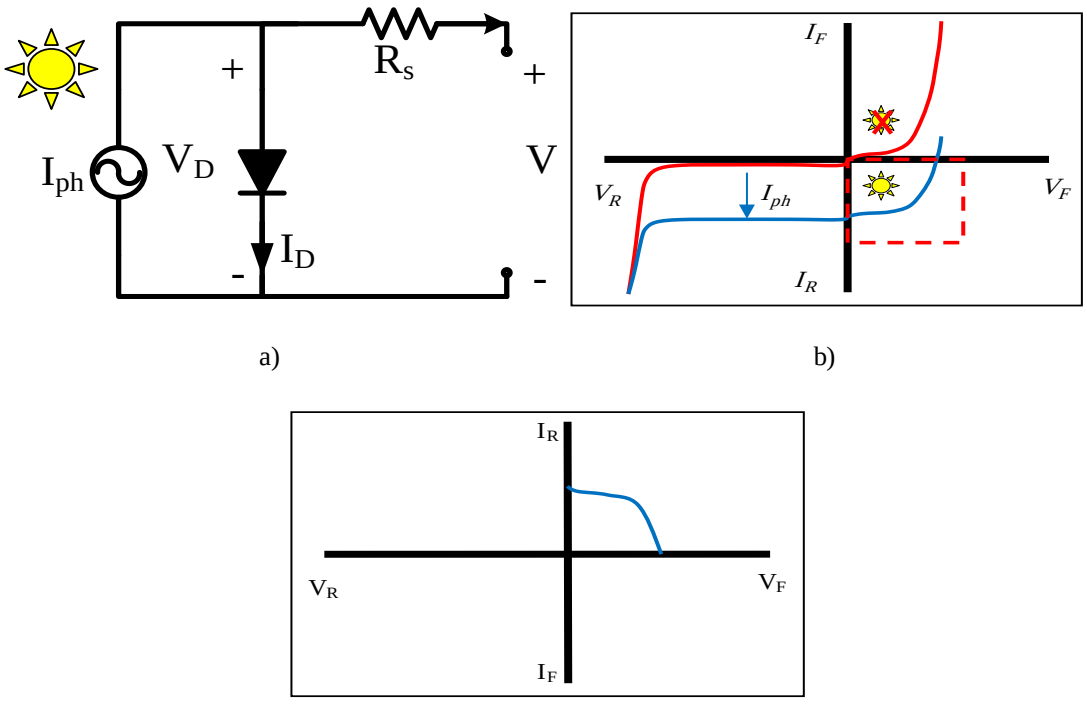


Figure 3.3: a) The illuminated junction circuit b) The illuminated junction characteristic c) PV cell characteristic

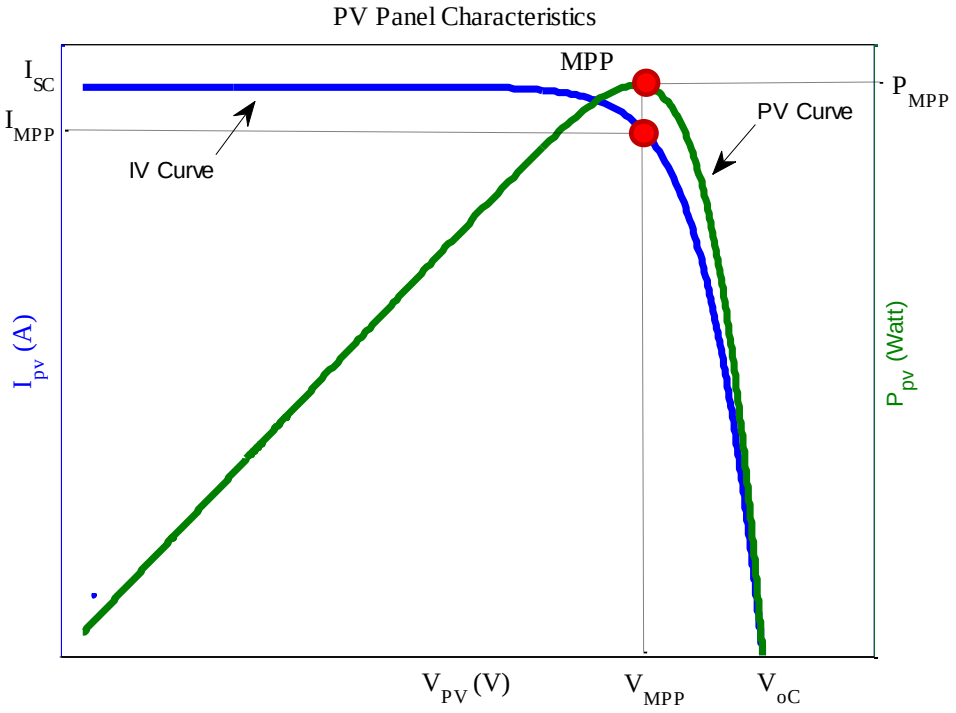


Figure 3.4: PV panel characteristics

This figure illustrates also the location of three important points on the PV panel characteristics: the short-circuit current I_{sc}, the open-circuit voltage V_{oc}, and the maximum power point $P_{MPP}, I_{MPP}, V_{MPP}$. There are two important factors that determine the quality of the PV panel; the panels' fill factor (FF) and the panels' efficiency (η) [7].

The fill factor is a measure of the panel real I-V characteristic and it is defined as the ratio between the MPP power and the multiplication of the panel short circuit current I_{sc} and the panel open circuit voltage V_{oc}. The nearer the value of FF to one, the better the quality of the PV panel.

$$FF = \frac{P_{MPP}}{V_{oc} I_{sc}} \tag{3.19}$$

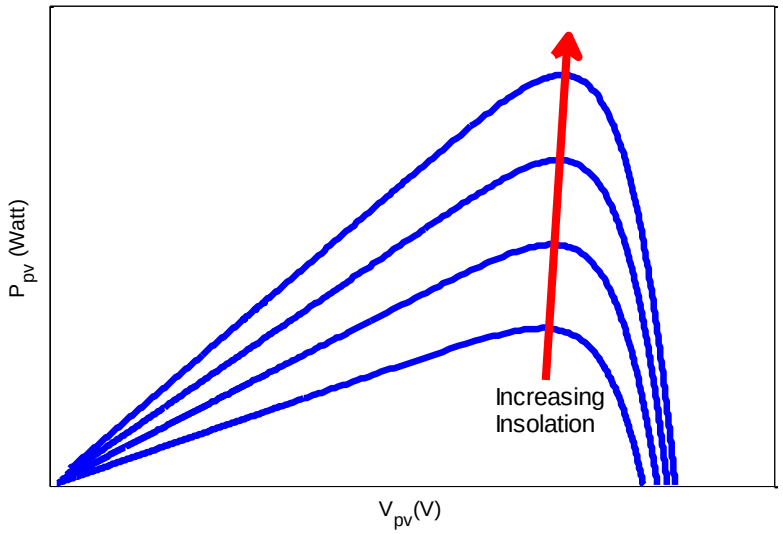
Another important PV panel factor is the efficiency. The PV panel efficiency (η) relates the output power of the PV panel and the power delivered by the sun and is usually reported under standard testing conditions (STC) (T=25^{o}C and Insolation=1000 W/m^{2}).

$$\eta = \frac{P_{MPP}}{Insolation \times Panel\ Area} \tag{3.20}$$

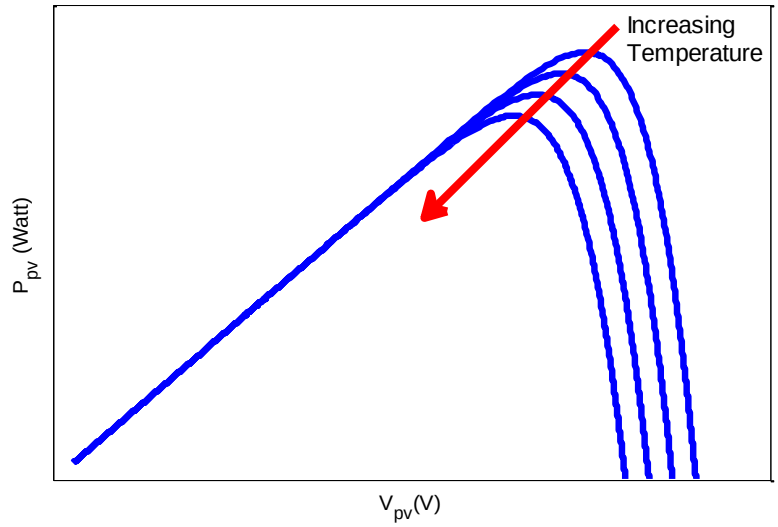
The derived PV panel model also illustrates the effects of temperature and the sun insolation on the PV panel output power as shown in Figure 3.5 [60],[63].

3.4 POWER CONVERTER MODEL

The developed standalone PV system uses a buck converter as a power-processing unit. The buck converter circuit consists of a MOSFET switch and a freewheeling diode, in addition to a rectifier circuit based on a capacitor and an inductor as shown in Figure 3.6. The circuit is controlled through a PWM signal generated by the MPPT controller.



a)



b)

Figure 3.5: PV panel characteristic for different a) insolation and b) temperatures

To design the MPPT control system, it is necessary to model the converter dynamic behavior. In particular, it is of interest to determine how variations in the input current $i_{in}(t)$, the load current $i_{out}(t)$ and the duty cycle $d(t)$ affect the input voltage $v_{in}(t)$. Based on the switch position the circuit goes into two states, in the first state $0 \le t \le t_{on}$, the switch will be ON the equivalent circuit for this mode is shown in Figure 3.7. In the second state $t_{off} \le t \le T$, the switch will be OFF resulting in the circuit shown in Figure 3.8.

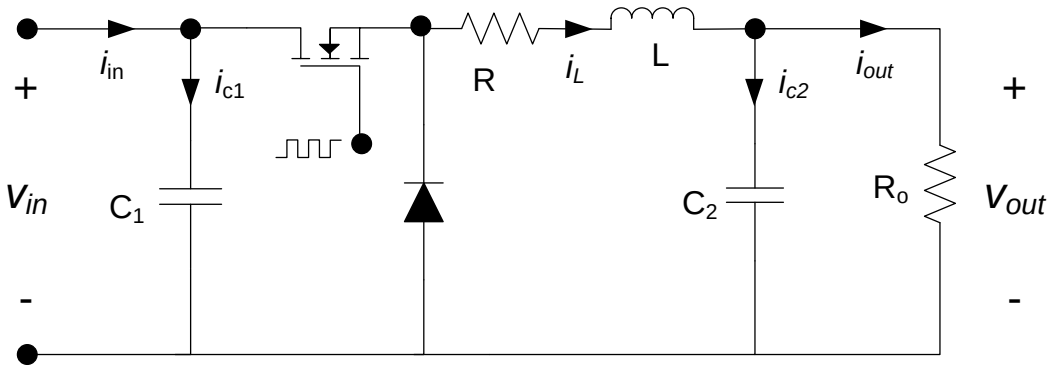


Figure 3.6: Buck converter circuit

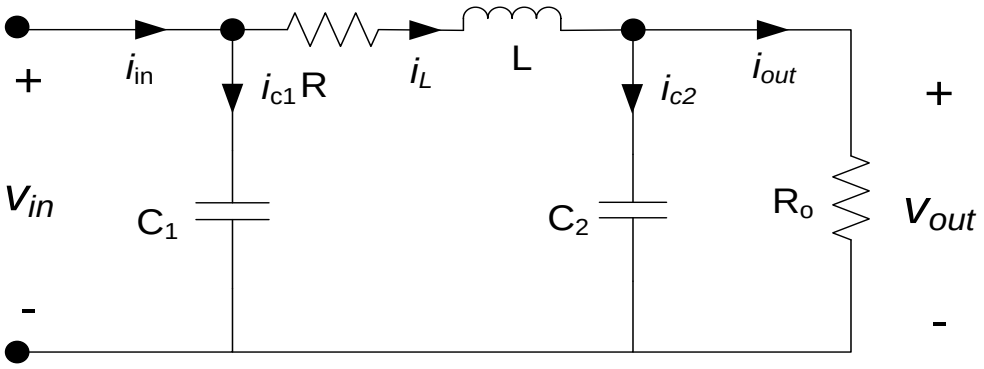


Figure 3.7: Equivalent circuit when switch is ON

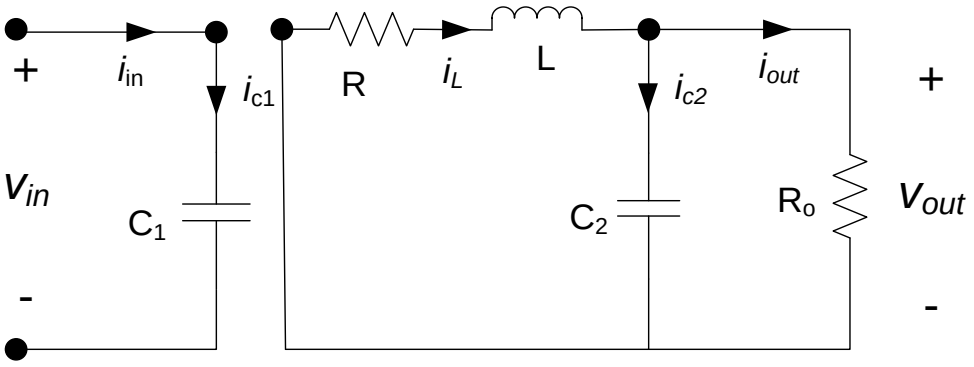


Figure 3.8: Equivalent circuit when switch is OFF

The output voltage of the converter depends on the PWM duty cycle based on the following formulation [64].

$$v_{L\_avg} = \frac{1}{T}\int_{DT} v_{L\_on}\,dt + \int_{(1-D)T} v_{L\_off}\,dt \tag{3.21}$$

$$= \frac{1}{T}\int_{DT} (v_{in} - v_{out})\,dt + \int_{(1-D)T} -v_{out}\,dt \tag{3.22}$$

$$= \frac{1}{T}\left[(v_{in} - v_{out})\,DT - v_{out}(1-D)T\right] \tag{3.23}$$

$$= \frac{1}{T}\left[(v_{in}DT - v_{out}T\right] \tag{3.24}$$

$$= v_{in}D - v_{out} \tag{3.25}$$

However, the average DC component of the inductor voltage has to be zero hence:

$$v_{out} = Dv_{in} \tag{3.26}$$

The state-space averaging technique is used next to generate the low-frequency small-signal AC equations of the PWM DC-DC converter [65]. The buck converter is assumed to operate in a continuous conduction mode. Applying Kirchoff's laws on the converter circuit shown in Figure 3.7 yields:

$$v_{in} = v_L + v_{out} = i_L R + L\frac{di_L}{dt} + v_{out} \tag{3.27}$$

$$i_{out} = i_L - i_{C2} = i_L - C_2\frac{dv_{out}}{dt} \tag{3.28}$$

$$i_L = i_{in} - i_{C1} = i_{in} - C_1\frac{dv_{in}}{dt} \tag{3.29}$$

Taking the inductor current i_L, output voltage v_{out}, and input voltage v_{in} as state variables, these equations can be written in the following state-space form:

$$\begin{cases} \dot{X} = A_1 X + B_1 u \\ y = E_1 X + F_1 u \end{cases} \tag{3.30}$$

where $u = i_{in}$, $y = v_{in}$, and $X = \begin{bmatrix} i_L & v_{out} & v_{in} \end{bmatrix}^t$.

In the second state $t_{on} \le t \le T$, the switch will be OFF. Analyzing the circuit in Figure 3.8 yields

$$i_{in} = i_{C1} = C_1 \frac{dv_{in}}{dt} \tag{3.31}$$

$$i_{out} = i_L - i_{C2} = i_L - C_2 \frac{dv_{out}}{dt} \tag{3.32}$$

$$v_L = i_L R + L \frac{di_L}{dt} = -v_{out} \tag{3.33}$$

These equations can be also written in state space form.

$$\begin{cases} \dot{X} = A_2 X + B_2 u \\ y = E_2 X + F_2 u \end{cases} \tag{3.34}$$

The duty cycle $d(t)$ may now be a time-varying quantity. It is assumed that the natural frequencies of the converter network are much smaller than the switching frequency. This assumption coincides with the small ripple approximation, and is usually satisfied in well-designed converters. It allows the high-frequency switching harmonics to be removed by an averaging process. In addition, the waveforms are linearized about a DC quiescent operating point. The converter waveforms are expressed as quiescent values plus small AC variations as follows [65]:

$$\begin{aligned} x(t) &= X + \hat{x}(t) \\ u(t) &= U + \hat{u}(t) \\ y(t) &= Y + \hat{y}(t) \\ d(t) &= D + \hat{d}(t) \end{aligned} \tag{3.35}$$

The small-signal AC model is then obtained as

$$\begin{cases} \dfrac{d\hat{x}(t)}{dt} = A\hat{x}(t) + B\hat{u}(t) + M\hat{d}(t) \\ \hat{y}(t) = E\hat{x}(t) + F\hat{u}(t) + H\hat{d}(t) \end{cases} \tag{3.36}$$

where

$$\begin{aligned} A &= DA_1 + (1-D)A_2 \\ B &= DB_1 + (1-D)B_2 \\ E &= DE_1 + (1-D)E_2 \\ F &= DF_1 + (1-D)F_2 \end{aligned} \tag{3.37}$$

and

$$\begin{aligned} M &= (A_1 - A_2)X + (B_1 - B_2)U \\ H &= (E_1 - E_2)X + (F_1 - F_2)U \end{aligned} \tag{3.38}$$

Using equations (3.27)-(3.38) yields the following matrices

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} & \frac{D}{L} \\ \frac{1}{C_2} & -\frac{1}{R_o C_2} & 0 \\ -\frac{D}{C_1} & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{C_1} \end{bmatrix}, M = \begin{bmatrix} \frac{DV_{in}}{L} \\ 0 \\ -\frac{DI_L}{C_1} \end{bmatrix}, E = [0 \;\; 0 \;\; 1], F = 0, H = 0.$$

The state-space averaged model that describes the quiescent converter waveforms is

$$\begin{cases} 0 = AX + BU \\ Y = EX + FU \end{cases} \tag{3.39}$$

The steady-state solution of the converter is therefore

$$G_{in} = \frac{V_{in}}{I_{in}} = E(-A)^{-1}B + F \tag{3.40}$$

$$G_{in} = \frac{V_{in}}{I_{in}} = [0 \;\; 0 \;\; 1] \begin{bmatrix} \frac{R}{L} & \frac{1}{L} & -\frac{D}{L} \\ -\frac{1}{C_2} & \frac{1}{R_o C_2} & 0 \\ \frac{D}{C_1} & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{C_1} \end{bmatrix} \tag{3.41}$$

$$G_{in} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{C_1}{D} \\ 0 & R_o C_2 & \frac{R_o C_1}{D} \\ \frac{-L}{D} & \frac{R_o C_2}{D} & \frac{(R + R_o)C_1}{D^2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{C_1} \end{bmatrix} \tag{3.42}$$

$$G_{in} = \begin{bmatrix} -\frac{L}{D} & 0 & \frac{(R + R_o)C_1}{D^2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{C_1} \end{bmatrix} \tag{3.43}$$

This equation yields:

$$G_{in} = \frac{R + R_o}{D^2} \tag{3.44}$$

If the inductor resistance is negligible, the steady-state solution is equivalent to

$$\frac{V_{in}}{I_{in}} = \frac{R_o}{D^2} \tag{3.45}$$

Equation (3.45) represents the DC equivalent input impedance of the buck converter [7], [8]. Referring back to the small signal model (3.36), the control to PV voltage transfer function can be obtained as

$$G_c(s) = \frac{\hat{v}_{in}(s)}{\hat{d}(s)} = E(sI - A)^{-1} M \tag{3.46}$$

$$G_c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s + \frac{R}{L} & \frac{1}{L} & -\frac{D}{L} \\ -\frac{1}{C_2} & s + \frac{1}{R_o C_2} & 0 \\ \frac{D}{C_1} & 0 & s \end{bmatrix}^{-1} \begin{bmatrix} \frac{DV_{in}}{L} \\ 0 \\ -\frac{DI_L}{C_1} \end{bmatrix} \tag{3.47}$$

$$G_c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \frac{DV_{in}}{L} \\ 0 \\ -\frac{DI_L}{C_1} \end{bmatrix} \tag{3.48}$$

where:

$$den_A = C_1C_2LR_os^3 + (C_1L + C_1C_2RR_o)s^2 + (C_2R_oD^2 + C_1R_o + C_1R)s + D^2$$

$$A_{11} = \frac{C_1C_2LR_os^2 + C_1Ls}{den_A}, A_{12} = -\frac{C_1C_2R_os}{den_A}, A_{13} = \frac{C_1C_2DR_os + C_1D}{den_A}$$

$$A_{21} = \frac{C_1LR_os}{den_A}, A_{22} = \frac{C_1C_2LR_os^2 + C_1C_2R_oRs + C_2R_oD^2}{den_A}, A_{23} = \frac{C_1DR_o}{den_A}$$

$$A_{31} = -\frac{DL + C_2DLR_os}{den_A}, A_{32} = \frac{C_2DR_o}{den_A},$$

$$A_{33} = \frac{C_1C_2LR_os^2 + C_1Ls + C_1C_2RR_os + C_1R + C_1R_o}{den_A}$$

As a result,

$$G_c = \begin{bmatrix} A_{13} & A_{23} & A_{33} \end{bmatrix} \begin{bmatrix} \frac{DV_{in}}{L} \\ 0 \\ -\frac{DI_L}{C_1} \end{bmatrix} \tag{3.49}$$

This equation yields:

$$G_c = -(\frac{DL + C_2DLR_os}{den_A})\frac{DV_{in}}{L} - (\frac{C_1C_2LR_os^2 + C_1Ls + C_1C_2RR_os + C_1R + C_1R_o}{den_A})\frac{DI_L}{C_1} \tag{3.50}$$

$$= -(\frac{D^2V_{in} + C_2D^2V_{in}R_os}{den_A}) - (\frac{DI_LC_2LR_os^2 + DI_LLs + DI_LC_2RR_os + DI_LR + DI_LR_o}{den_A}) \tag{3.51}$$

$$-\left(\frac{DI_L C_2 L R_o s^2 + (C_2 D^2 V_{in} R_o + DI_L L + DI_L C_2 R R_o)s + DI_L R + DI_L R_o + D^2 V_{in}}{C_1 C_2 L R_o s^3 + (C_1 L + C_1 C_2 R R_o)s^2 + (C_2 R_o D^2 + C_1 R_o + C_1 R)s + D^2}\right) \quad (3.52)$$

this can be simplified in the following form

$$G_c(s) = \frac{\hat{v}_{in}(s)}{\hat{d}(s)} = -\left(\frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}\right) \quad (3.53)$$

where,

$$b_0 = \frac{DI_L}{LC_1C_2} + \frac{(D^2 V_{in} + DI_L R)}{R_o LC_1C_2},\ b_1 = \frac{1}{C_1}\left(\frac{D^2 V_{in}}{L} + \frac{DI_L R}{L} + \frac{DI_L}{R_o C_1 C_2}\right),$$

$$b_2 = \frac{DI_L}{C_1}$$

$$a_0 = \frac{D^2}{LR_0 C_1 C_2},\ a_1 = \frac{R}{LR_o C_2} + \frac{1}{LC_2} + \frac{D^2}{LC_1},\ a_2 = \frac{1}{R_o C_2} + \frac{R}{L}$$

This equation describes how small AC variations in the duty cycle excite variations in the PV panel voltage.

CHAPTER 4 : FUZZY SYSTEM DESIGN

4.1 INTRODUCTION

Since its proposal by L. Zadeh [66] in 1965, fuzzy logic has been an active research topic. Zadeh defined fuzzy sets as "a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function, which assigns to each object a grade of membership ranging between zero and one" [66].

In general, a fuzzy system is a static nonlinear mapping between sets of inputs and outputs. The inputs and outputs are real numbers, whereas the processes in between consists of fuzzy sets [68]. Fuzzy Logic Control (FLC) is mostly effective when the controlled process is based on human heuristic experience. As a result, it is commonly used to control nonlinear processes where there is no simple mathematical model relating between the process inputs and outputs [68]. A FLC system mainly consists of four main blocks namely, fuzzification, rule base algorithm, fuzzification interface, and defuzzification. The following section will explain the design of each block based on the work of K. Passino and S. Yurkovich [67].

4.2 FUZZY CONTROL SYSTEM DESIGN

The FLC process can be summarized as follows. FLC maps between a set of inputs $u_i \in U_i$;where $i = 1, 2, \ldots, n$ and outputs $y_i \in Y_i$; where $i = 1, 2, \ldots, m$. U_i and Y_i are called the “universes of discourse” for u_i and y_i respectively.

As shown in Figure 4.1, the fuzzification block converts the real number "crisp" inputs to fuzzy sets, and then the inference mechanism uses the fuzzy rules of the rule-base to produce the implied fuzzy sets or the fuzzy conclusions.

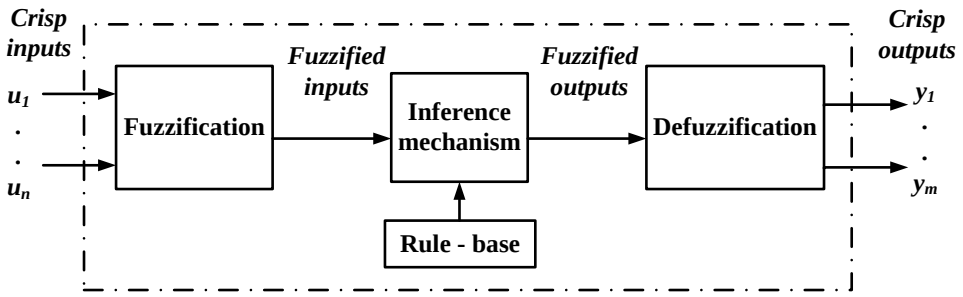


Figure 4.1: FLC block diagram

Finally, the defuzzification block converts these fuzzy conclusions into the crisp real number outputs. The design of a fuzzy control system goes through three main steps: choosing the fuzzy controller inputs and outputs, processing the controller inputs and outputs, and designing each of the four blocks of the fuzzy controller.

While choosing the controller inputs, the designer should make sure that the FLC has enough information available to produce good decisions and efficiently control the process. In general, the choice of the inputs and the outputs will place specific limitations on the FLC design process. In some cases, the designer will be limited to a set of inputs depending on the sensors availability, in such a case observers and estimators can be used to find the needed inaccessible inputs. Choosing the FLC outputs is usually easier than choosing its inputs as the control process itself usually determines them.

4.2.1 FLC Terminologies

Some fuzzy related terminologies have to be defined before going through the FLC design.

4.2.1.1 Linguistic Variables, Values, and Rules

After choosing the FLC inputs and outputs, a linguistic description has to be given to each one of them. For our fuzzy system, we will describe the fuzzy input u_i by the linguistic variable $\tilde{u}_i$, likewise the fuzzy output y_i will be described by the linguistic variable $\tilde{y}_i$. Then a description has to be given to each $\tilde{u}_i$ and $\tilde{y}_i$.

The next step is to give a linguistic value to each linguistic variable. Let $\tilde{A}_i^j$ denote the j^{th} linguistic value of the linguistic variable $\tilde{u}_i$ defined over the universe of discourse U_i. If we assume that there exist many linguistic values defined over U_i,

then the linguistic variable $\tilde{u}_i$ takes on the elements from the set of linguistic values denoted by

$$\tilde{A}_i = \left\{ \tilde{A}_i^j \ : \ j = 1,2,...., N_i \right\}$$

Similarly, let $\tilde{B}_i^j$ denote the j^{th} linguistic value of the linguistic variable $\tilde{y}_i$ defined over the universe of discourse Y_i. The linguistic variable $\tilde{y}_i$ takes on elements from the set of linguistic values denoted by

$$\tilde{B}_i = \left\{ \tilde{B}_i^p \ : \ p = 1,2,...., M_i \right\}$$

Linguistic values are generally descriptive adjectives such as "big", "medium", and “small”.

For an FLC system, a set of rule has to be defined in order to map the inputs to the outputs. These rules usually have the following form

If premise Then consequent (4.1)

The inputs of the fuzzy system are associated with the premise, and the outputs are associated with the consequent. The standard form to represent these rules for a multi-input single-output (MISO) process is:

$$If\ \tilde{u}_1\ is\ \tilde{A}_i^j\ and\ \tilde{u}_2\ is\ \tilde{A}_2^k, and\ ....,\tilde{u}_n\ is\ \tilde{A}_n^l\ Then\ \tilde{y}_q\ is\ \tilde{B}_q^p \quad (4.2)$$

In order to control the system, the designer has to specify an entire set of linguistic rules of this form.

4.2.1.2 Fuzzy Sets, Fuzzy Logic, and the Rule-Base

In order to quantify the meaning of linguistic variables, linguistic values, and linguistic rule, fuzzy sets has to be used. Fuzzy sets are set of membership function defined as follow.

Let U_i denote a universe of discourse and $\tilde{A}_i^j \in \tilde{A}_i$ denote a specific linguistic value for the linguistic variable $\tilde{u}_i$. The function $\mu(\tilde{u}_i)$ associated with $\tilde{A}_i^j$ that maps U_i to [0, 1] is called a membership function (MF). This membership function describes the “certainty” that an element of U_i, denoted u_i,with a linguistic description $\tilde{u}_i$, may be classified linguistically as $\tilde{A}_i^j$. Membership functions are specified and tuned according to the experience or intuition of the designer. These

MF's can be triangular, trapezoidal, or Gaussian, each shape will provide a different meaning for the linguistic values that they quantify. The illustration and the mathematical characterization of the triangular membership function are given in Figure 4.2.

$$\mu^L(u) = \begin{cases} 1 & \text{if } u \le c^L \\ \max\left\{0,\ 1 + \dfrac{c^L - u}{w^L}\right\} & \text{otherwise} \end{cases} \tag{4.3}$$

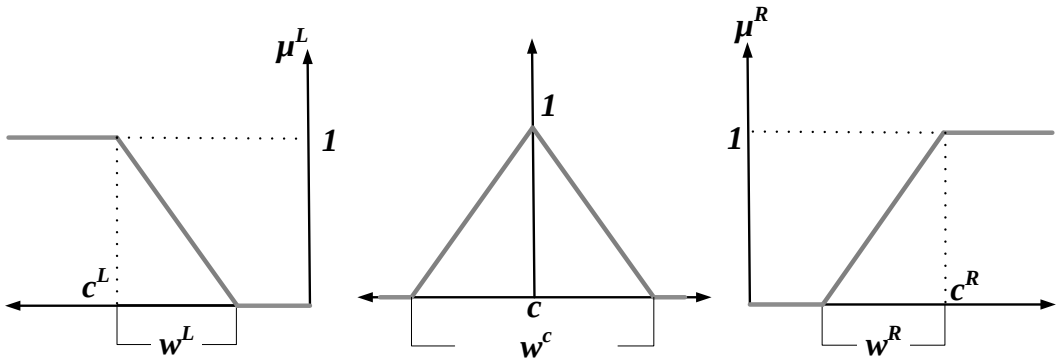
$$\mu^C(u) = \begin{cases} \max\left\{0, 1 + \dfrac{u - c}{0.5w^c}\right\} & \text{if } u \le c^C \\ \max\left\{0, 1 + \dfrac{c - u}{0.5w^c}\right\} & \text{otherwise} \end{cases} \tag{4.4}$$

$$\mu^R(u) = \begin{cases} \max\left\{0, 1 + \dfrac{u - c^R}{w^R}\right\} & \text{if } u \le c^R \\ 1 & \text{otherwise} \end{cases} \tag{4.5}$$

where c^L is the left saturation point, w^L is the width of the non-unity and non-zero part of μ^L. c^R is the right saturation point, w^R is the width of the non-unity and non-zero part μ^R. c is the center of the triangle, w^c is the triangle base-width.

Given a linguistic variable $\tilde{u}_i$ with a linguistic value $\tilde{A}_i^j$ defined on the universe of discourse U_i, and membership function $\mu_{A_i^j}(u_i)$ that maps U_i to [0, 1], a "fuzzy set" denoted with A_i^j is defined as:

$$A_i^j = \left\{(u_i, \mu_{A_i^j}(u_i)) : u_i \in U_i\right\} \tag{4.6}$$



a) b) c)

Figure 4.2: Triangular MF illustrations a)Left, b)Center, c)Right

As a result, a fuzzy set is simply a crisp set of pairings of elements of the universe of discourse coupled with their associated membership values.

4.2.2 Fuzzification

Fuzzification is the process through which the fuzzy system converts its numerical crisp inputs $u_i \in c$ into fuzzy sets. Let U_i^* denote the set of all possible fuzzy sets that can be defined on U_i. Given $u_i \in U_i$, fuzzification transforms u_i to a fuzzy set denoted $\tilde{A}_i^{fuz}$ defined on the universe of discourse U_i. This transformation is produced by the fuzzification operator F defined by

$$F : U_i \rightarrow U_i^* \tag{4.7}$$

where

$$F(u_i) = \tilde{A}_i^{fuz} \tag{4.8}$$

4.2.3 The Inference Mechanism

Two basic tasks are performed by the Inference Mechanism namely, matching and inference step. Matching determines the degree to which each rule applies to the situation characterized by the FLC inputs u_i, whereas the inference step uses the FLC input u_i together with the information in the rule-base to draw the fuzzy conclusions.

The first step in matching is to combine inputs with rule premises through finding fuzzy sets. The second step involves finding which rules are ON. To perform inference, we must first quantify each of the rules with fuzzy logic. To do this, we first quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input.

Through fuzzification, we quantified the meaning of the linguistic input terms via the membership functions. In order to quantify the "and" operation we form membership values $\mu_i(u_1, u_2, ..., u_n)$ for the i^{th} rule's premise $\mu_{premise}$ that represent the certainty that each rule premise holds for the given inputs. Considering the general FLC rule given in (4.2) for a two inputs, single output process

$$\text{If } \tilde{u}_1 \text{ is } \tilde{A}_i^j \text{ and } \tilde{u}_2 \text{ is } \tilde{A}_2^k \text{ Then } \tilde{y} \text{ is } \tilde{B}^p \tag{4.9}$$

$\mu_{premise}$ can be calculated as follow:

- Minimum: Define $\mu_{premise} = \min\left\{\mu_{\tilde{A}_2^j}(u_1), \mu_{\tilde{A}_2^k}(u_2)\right\}$ that is, using the minimum of the membership values.
- *Product:* Define $\mu_{premise} = \mu_{\tilde{A}_2^j}(u_1) * \mu_{\tilde{A}_2^k}(u_2)$, that is, using the product of the membership values.

Both ways of quantifying the "and" operation in the premise indicate that you can be no more certain about the conjunction of two statements than you are about the individual terms that make them up ($0 \le \mu_{premise} \le 1$ for either case). If we consider all possible $\tilde{u}_1$ and $\tilde{u}_2$ values, we will obtain a multidimensional membership function $u_{premise}(\tilde{u}_1, \tilde{u}_2)$ that is a function of $\tilde{u}_1$ and $\tilde{u}_2$ for each rule. The value of this function represents how certain we are that the rule in (4.2) is applicable for specifying the output to the plant. As $\tilde{u}_1$ and $\tilde{u}_2$ change, the value of $u_{premise}$ changes and we become less or more certain of the applicability of this rule.

In general, we will have a different premise membership function for each of the rules in the rule-base, and each of these will be a function of $\tilde{u}_1$ and $\tilde{u}_2$ so that given specific values of $\tilde{u}_1$ and $\tilde{u}_2$ we obtain a quantification of the certainty that each rule in the rule-base applies to the current situation.

The second step of matching is to determine which rules are ON to find out which rules are relevant to the current situation. We say that a rule is "ON at time *t*" if its premise membership function $\mu_{premise}(\tilde{u}_1, \tilde{u}_2) > 0$. In the next step, the inference mechanism will seek to combine the recommendations of all the rules to come up with a single conclusion.

4.2.4 Inference Step

The inference step determines which conclusions should be reached when the rules that are on are applied to decide what the output should be. To do this, we will first consider the recommendations of each rule independently. Then later we will combine all the recommendations from all the rules to determine the output. The minimum operation will be used to consider the conclusion reached by each rule; in general, we will never have more than four rules on at one time. Each ON rule will result in a membership function for the conclusion, which we denote by, $\mu_{(rule\ No.)}$ that is given by:

$$\mu_{(rule\ No.)}(y) = \min\left\{\mu_{\tilde{B}^i}(y), \mu_{premise_{(rule\ No.)}}\right\} \tag{4.10}$$

This membership function defines the "implied fuzzy set" for the ON rule. Notice that the membership function $\mu_{(rule\ No.)}(y)$ is a time-varying function of y that quantifies how certain the rule will apply and that the minimum operation will generally, "chop off the top" of the $\mu_{\tilde{B}^i}(y)$ membership function to produce $\mu_{(rule\ No.)}(y)$. While the input to the inference process is the set of rules that are on, its output is the set of implied fuzzy sets that represent the conclusions reached by all the rules that are on.

4.2.5 Defuzzification

A number of defuzzification strategies exist, each provides a means to choose a single output (which we denote with y_q^{crisp}) based on the implied fuzzy sets.

- Center of gravity (COG): A crisp output y_q^{crisp} is chosen using the center of area and area of each implied fuzzy set, and is given by

$$y_q^{crisp} = \frac{\sum_{i=1}^{R} b_i^q \int_{y_q} \mu_{\tilde{B}_q^i}(y_q)dy_q}{\sum_{i=1}^{R} \int_{y_q} \mu_{\tilde{B}_q^i}(y_q)dy_q} \tag{4.11}$$

where R is the number of rules, b_i^q is the center of area of the membership function of B_i^q associated with the implied fuzzy set $\tilde{B}_i^q$ for the *ith* rule (*j, k, . . ., l*; *p, q*)*i*, and $\int_{y_q} \mu_{\tilde{B}_q^i}(y_q)dy_q$ denotes the area under $u_{\mu\tilde{B}_q^i}(y_q)$. The COG can be easy to compute since it is often easy to find closed-form expressions for the integral, which is the area under a membership function.

- Center-average: A crisp output y_q^{crisp} is chosen using the centers of each of the output membership functions and the maximum certainty of each of the conclusions represented with the implied fuzzy sets, and is given by

$$y_q^{crisp} = \frac{\sum_{i=1}^{R} b_i^q \sup_{y_q}\left\{\mu_{\tilde{B}_q^i}(y_q)\right\}}{\sum_{i=1}^{R} \sup_{y_q}\left\{\mu_{\tilde{B}_q^i}(y_q)\right\}} \tag{4.12}$$

where "sup" which is the "supremum" or the least upper bound, which can often be thought of as the maximum value.

4.3 FUZZY CONTROL SYSTEM DESIGN EXAMPLE

There is no general systematic procedure to design a high-performance fuzzy controller. However, going through the following general example will clarify all the steps described in section 4.2.

Suppose that at some point fuzzification produces $\tilde{A}_1^{fuz}, \tilde{A}_2^{fuz}$ as fuzzy sets representing the crisp inputs u_1, u_2 respectively and that $u_1 = -0.375$ and $u_2 = 0.375$ are the crisp inputs of our system. Figure 4.3 shows the membership functions for the inputs and indicates with thick yellow vertical lines the values $\mu_{\tilde{A}_1^2}(0.375) = 0.75$, $\mu_{\tilde{A}_1^3}(0.375) = 0.25$, $\mu_{\tilde{A}_2^3}(0.375) = 0.25$, and $\mu_{\tilde{A}_2^4}(0.375) = 0.75$ for $\tilde{u}_1$ and $\tilde{u}_2$.

The next step is to define the rule-base that maps the inputs u_1, u_2 to the output y, which is defined by the fuzzy set $\tilde{B}^{fuz}$ shown in Figure 4.4. For our discussed example, we will consider the rules in Table 4-1.

Table 4-1: FLC example system rule-base

$\tilde{u}_1$ \ $\tilde{u}_2$	$\tilde{A}_2^1$	$\tilde{A}_2^2$	$\tilde{A}_2^3$	$\tilde{A}_2^4$	$\tilde{A}_2^5$
$\tilde{A}_1^1$	$\tilde{B}^5$	$\tilde{B}^5$	$\tilde{B}^5$	$\tilde{B}^4$	$\tilde{B}^3$
$\tilde{A}_1^2$	$\tilde{B}^5$	$\tilde{B}^5$	$\tilde{B}^4$	$\tilde{B}^3$	$\tilde{B}^2$
$\tilde{A}_1^3$	$\tilde{B}^5$	$\tilde{B}^4$	$\tilde{B}^3$	$\tilde{B}^2$	$\tilde{B}^1$
$\tilde{A}_1^4$	$\tilde{B}^4$	$\tilde{B}^3$	$\tilde{B}^2$	$\tilde{B}^1$	$\tilde{B}^1$
$\tilde{A}_1^5$	$\tilde{B}^3$	$\tilde{B}^2$	$\tilde{B}^1$	$\tilde{B}^1$	$\tilde{B}^1$

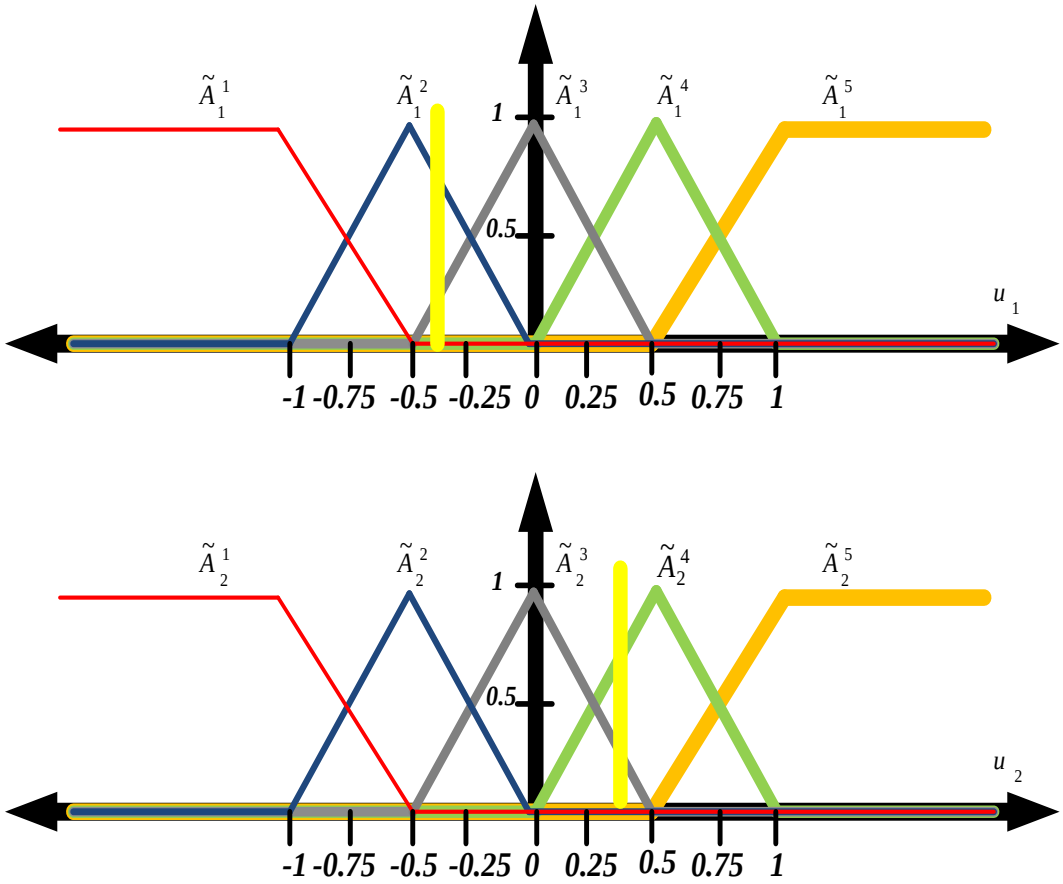


Figure 4.3: FLC example system input membership functions

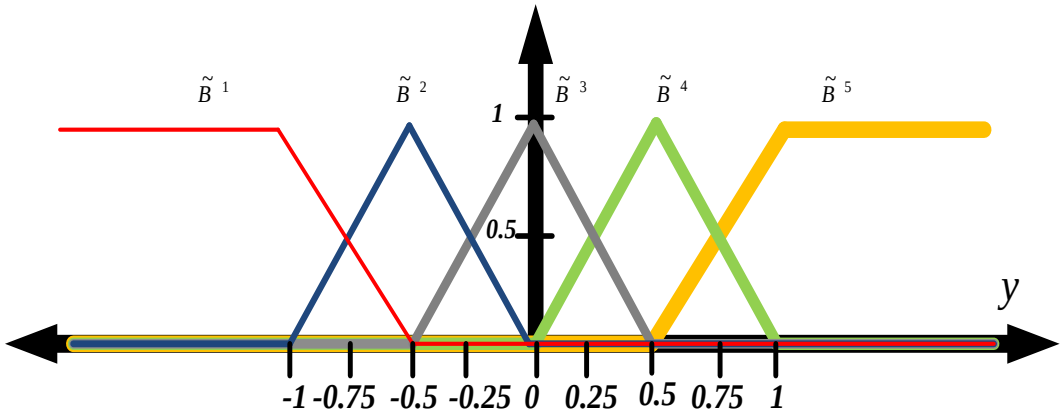


Figure 4.4: FLC example system output membership function

After that, the inference will be performed through quantifying the applicable premises. Therefore, form Figure 4.3 and Table 4-1 the ON rules are:

$$If\ \tilde{u}_1\ is\ \tilde{A}_1^2\ and\ \tilde{u}_2\ is\ \tilde{A}_2^3\ Then\ \tilde{y}\ is\ \tilde{B}^4$$

$$If\ \tilde{u}_1\ is\ \tilde{A}_1^2\ and\ \tilde{u}_2\ is\ \tilde{A}_2^4\ Then\ \tilde{y}\ is\ \tilde{B}^3 \tag{4.13}$$

$$If\ \tilde{u}_1\ is\ \tilde{A}_1^3\ and\ \tilde{u}_2\ is\ \tilde{A}_2^3\ Then\ \tilde{y}\ is\ \tilde{B}^3$$

$$If\ \tilde{u}_1\ is\ \tilde{A}_1^3\ and\ \tilde{u}_2\ is\ \tilde{A}_2^4\ Then\ \tilde{y}\ is\ \tilde{B}^2$$

Then we will start considering the conclusion reached by the four rules. Using the minimum to represent the premise:

$$\begin{aligned}
\mu_{premise_{(1)}} &= \min\left\{\mu_{\tilde{A}_1^2}(u_1) = 0.75, \mu_{\tilde{A}_2^3}(u_2) = 0.25\right\} = 0.25 \\
\mu_{premise_{(2)}} &= \min\left\{\mu_{\tilde{A}_1^2}(u_1) = 0.75, \mu_{\tilde{A}_2^4}(u_2) = 0.75\right\} = 0.75 \\
\mu_{premise_{(3)}} &= \min\left\{\mu_{\tilde{A}_1^3}(u_1) = 0.25, \mu_{\tilde{A}_2^3}(u_2) = 0.25\right\} = 0.25 \\
\mu_{premise_{(3)}} &= \min\left\{\mu_{\tilde{A}_1^3}(u_1) = 0.25, \mu_{\tilde{A}_2^4}(u_2) = 0.75\right\} = 0.25
\end{aligned} \tag{4.14}$$

The previous four premises show that we are 0.25 certain that the first, third and fourth rules apply, where as we are 0.75 certain that the second rule applies to the current situation.

The next step is to find the membership function for the conclusion reached by each rule. Rule (1) will produce the function $\mu_{(1)}(y)$, which is shown in Figure 4.5 and is given by

$$\mu_{(1)}(y) = \min\left\{\mu_{\tilde{B}^4}(y), 0.25\right\} = 0.25 \tag{4.15}$$

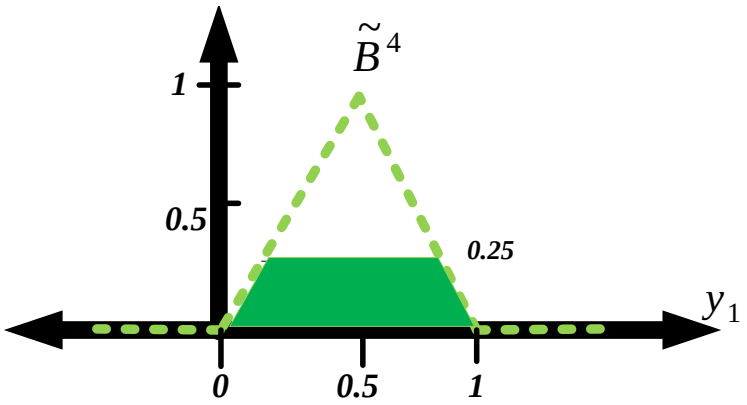


Figure 4.5: Implied fuzzy set with membership function $\mu_{(1)}$(y) for rule (1)

Rule (2) will produce the function $\mu_{(2)}(y)$, which is shown in Figure 4.6 and is given by

$$\mu_{(2)}(y) = \min\left\{\mu_{\tilde{B}^3}(y), 0.75\right\} = 0.75 \tag{4.16}$$

Rule (3) will produce the function $\mu_{(3)}(y)$, which is shown in Figure 4.7 and is given by

$$\mu_{(3)}(y) = \min\left\{\mu_{\tilde{B}^3}(y), 0.25\right\} = 0.25 \tag{4.17}$$

Rule (4) will produce the function $\mu_{(4)}(y)$, which is shown in Figure 4.8 and is given by

$$\mu_{(4)}(y) = \min\left\{\mu_{\tilde{B}^2}(y), 0.25\right\} = 0.25 \tag{4.18}$$

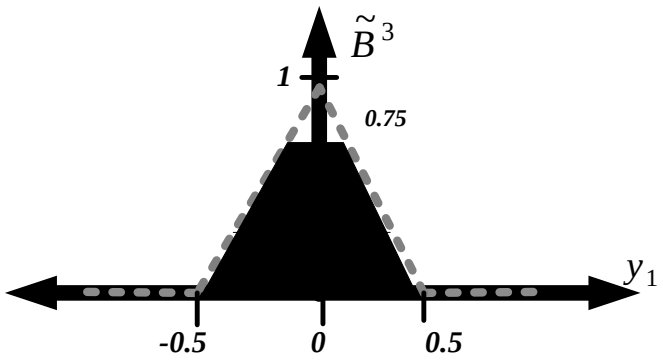


Figure 4.6: Implied fuzzy set with membership function $\mu_{(2)}$(y) for rule (2)

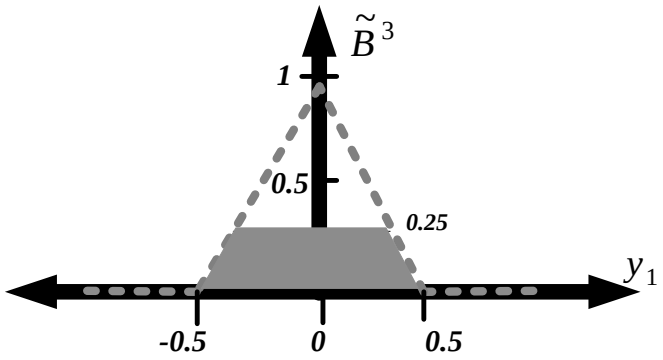


Figure 4.7: Implied fuzzy set with membership function $\mu_{(3)}$(y) for rule (3)

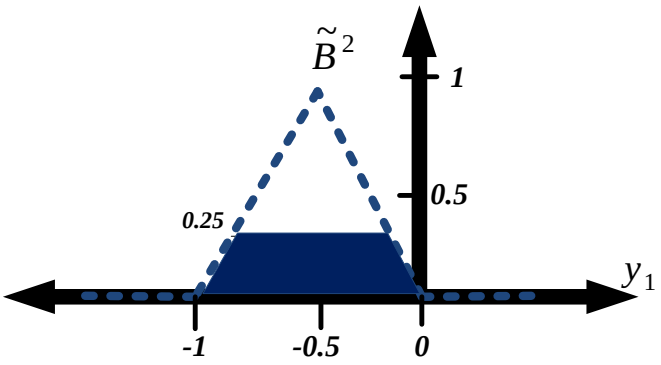


Figure 4.8: Implied fuzzy set with membership function $\mu_{(4)}(y)$ for rule (4)

The last step is to perform defuzzification, for which we will draw all the implied fuzzy sets on one axis as shown in Figure 4.9. In order to find the output y^{crisp} that best represents the conclusions of the fuzzy controller we will consider both COG and average-center techniques.

From Figure 4.9, we can clearly see that the center of the fuzzy implied set is $b_1 = 0.5$ for the first rule, $b_2 = 0$ and $b_3 = 0$ for the second, and third rules, and finally $b_4 = -0.5$ for the fourth rule.

From (4.11) COG method computes y^{crisp} to be

$$y^{crisp} = \frac{\sum_{i=1}^{4} b_i \int \mu(i)}{\sum_{i=1}^{4} \int \mu(i)} \tag{4.19}$$

$\int \mu(i)$ can be calculated through simple geometry. As the output MFs consists of symmetric triangular functions that peak at one and have a base width *w*, the area under a triangle chopped off at a height of $\mu_{premise}$ is

$$\int \mu(i) = w\left(\mu_{premise} - \frac{\mu_{premise}^{\,2}}{2}\right) \tag{4.20}$$

Using Equations (4.19) and (4.20) with Figure 4.9 we have

$$\int \mu(1) = \int \mu(3) = \int \mu(4) = 1\left(0.25 - \frac{(0.25)^2}{2}\right) = 0.21875 \tag{4.21}$$

$$\int \mu(2) = 0.5\left(0.75 - \frac{(0.75)^2}{2}\right) = 0.46875$$

$$y^{crisp} = \frac{(0.5)(0.21875) + (-0.5)(0.46875) + (0)(0.21875) + (0)(0.21875)}{0.21875 + 0.46875 + 0.21875 + 0.21875}$$

$$= -0.11111$$

However, if we use the center-average defuzzifier, shown in (4.12) y^{crisp} will equal

$$y^{crisp} = \frac{\sum_{i=1}^{4} b_i \ \sup(\mu(i))}{\sum_{i=1}^{4} \sup(\mu(i))} \tag{4.22}$$

Where $\sup(\mu(i))$ is the height of $\mu_{premise}$

$$\sup(\mu(1)) = \sup(\mu(3)) = \sup(\mu(4)) = 0.25$$

$$\sup(\mu(2)) = 0.75 \tag{4.23}$$

$$y_1^{crisp} = \frac{(0.5)(0.25) + (-0.5)(0.75) + (0)(0.25) + (0)(0.25)}{0.25 + 0.75 + 0.25 + 0.25} = -0.1667$$

COG is more accurate than center-average. However, the advantages of the center-average defuzzification method are its simpler computations compared to the COG and has less memory requirements as the output membership functions are easy to store since the only relevant information they provide are their center values (b_i).

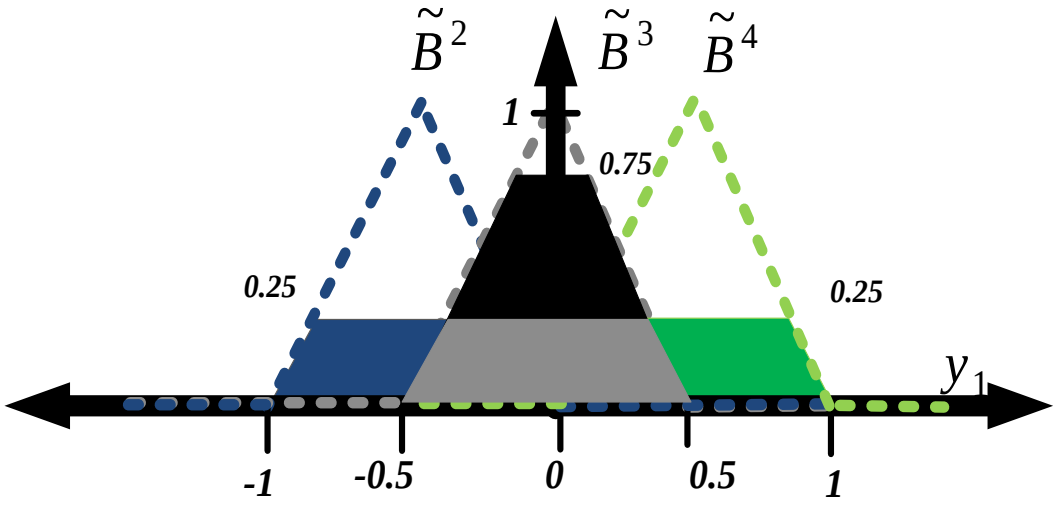


Figure 4.9: Implied fuzzy sets

CHAPTER 5 : PROPOSED SYSTEM DESIGN & SIMULATION

5.1 INTRODUCTION

The proposed DSP based standalone photovoltaic system consists of two main subsystems; an astronomical two-axis tracker combined with an MPPT based buck converter. Both subsystems are controlled through two TMS320F28335 eZdsp boards.

Figure 5.1 shows the overall system architecture. The proposed MPPT controller builds upon the simplicity of the P&O technique but eliminates the resulting steady state oscillations by adaptively modifying the reference voltage perturbation step-size C using a fuzzy logic controller.

5.2 FLC BASED P&O

As shown in Figure 5.1, the proposed control scheme takes the absolute power slope S_a of the PV panel curve and the old voltage perturbation step C_{old} as its inputs and outputs the change in the new P&O step size ΔC.

In order to design the FLC we will repeat the steps given in the example of section 4.3. So first, we will fuzzify the two input variables by using normalized fuzzy sets each of three triangular membership functions (MFs): Small, Medium, and Large as shown in Figure 5.2.

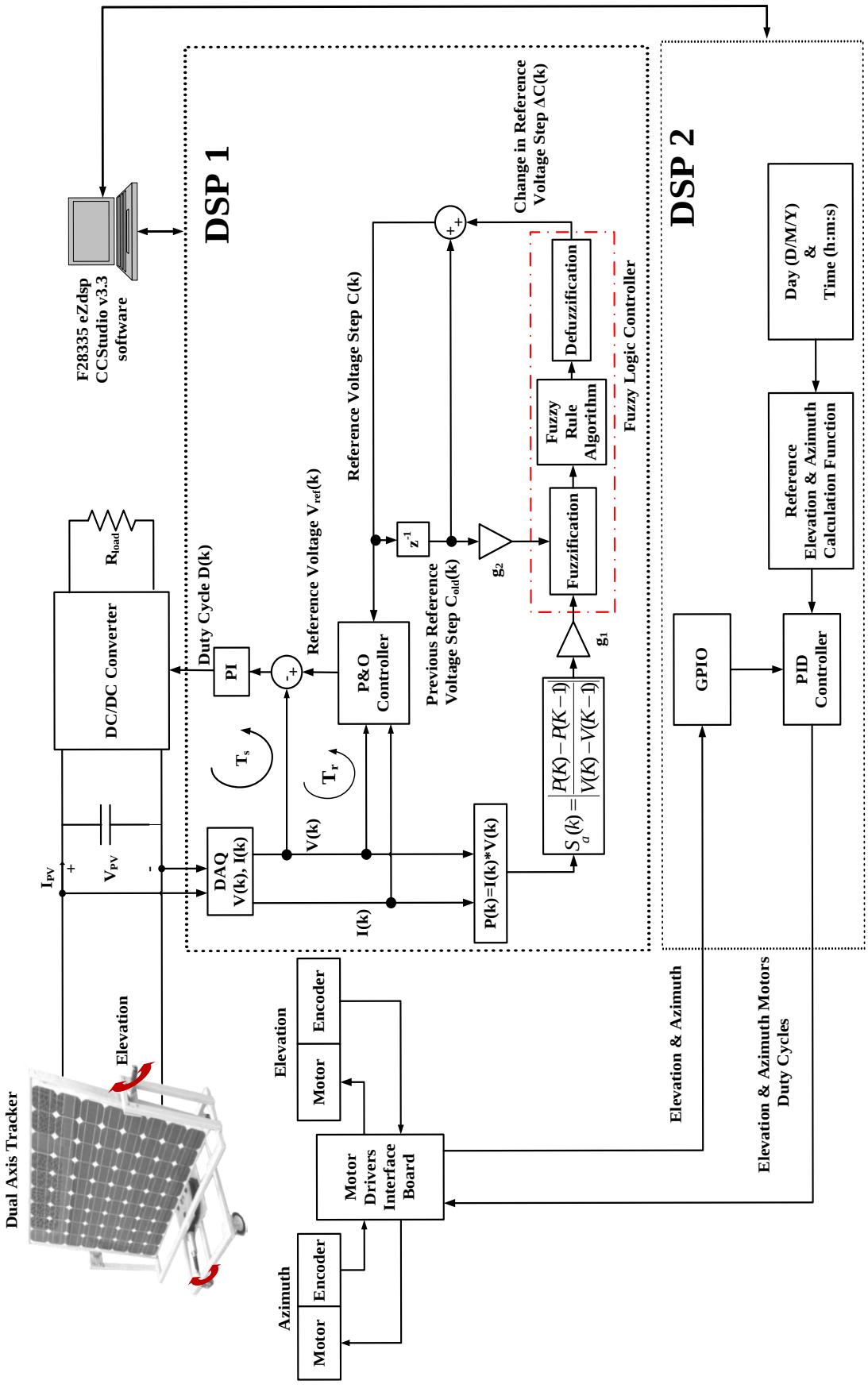


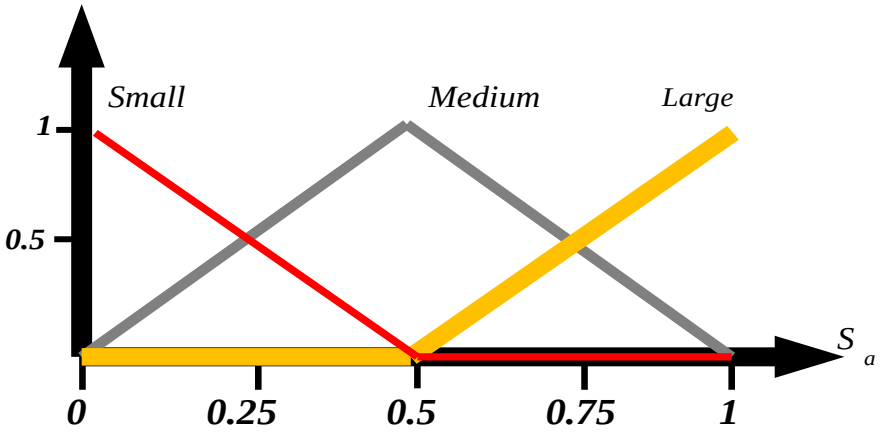
Figure 5.1: The proposed DSP based standalone solar energy system

The mathematical characterization of the triangular inputs membership function is given below.

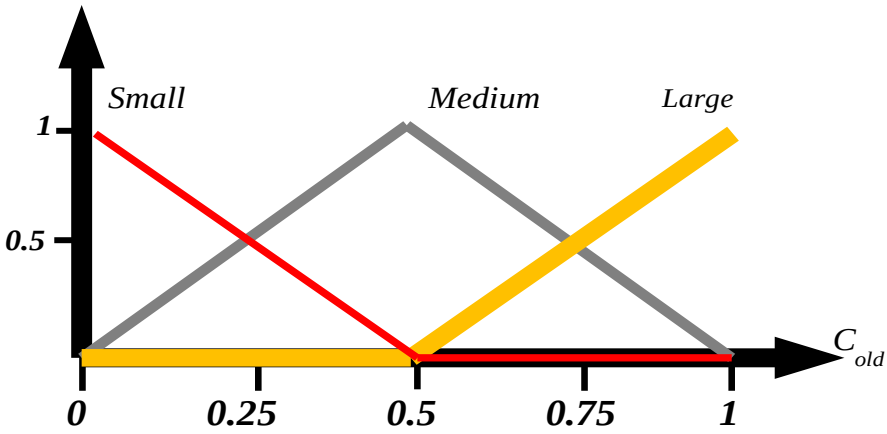
$$\mu^{Small}(u) = \begin{cases} 1 & if\ u \le 0 \\ \max\left\{0,\ 1 - \dfrac{u}{0.5}\right\} & otherwise \end{cases} \tag{5.1}$$

$$\mu^{Medium}(u) = \begin{cases} \max\left\{0, 1 + \dfrac{u - 0.5}{0.5}\right\} & if\ u \le 0.5 \\ \max\left\{0, 1 + \dfrac{0.5 - u}{0.5}\right\} & otherwise \end{cases} \tag{5.2}$$

$$\mu^{Large}(u) = \begin{cases} \max\left\{0, 1 + \dfrac{u - 1}{0.5}\right\} & if\ u \le 1 \\ 1 & otherwise \end{cases} \tag{5.3}$$



a)



b)

Figure 5.2: a) MF of the 1st input S_a b) MF of the 2nd input C_{old}

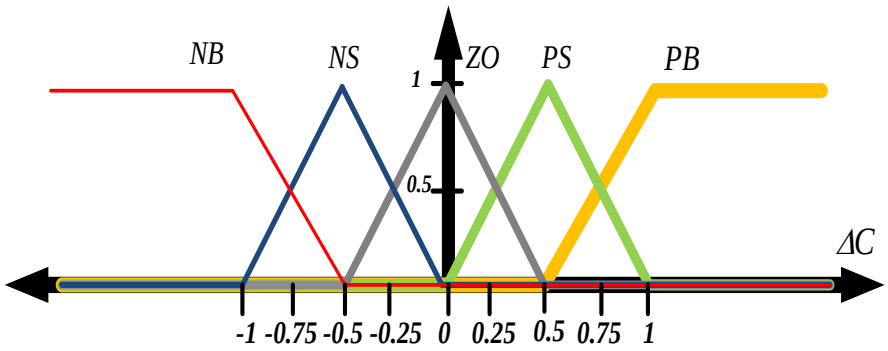


Figure 5.3: MF of the output ΔC

The output variable consists of a normalized fuzzy set of five triangular MFs: Negative Big (NB), Negative Small (NS), Zero (ZO), Positive Small (PS), and Positive Big (PB) as shown in Figure 5.3.

The mathematical characterization of the triangular output membership function is given below.

$$\mu^{NB}(\Delta C) = \begin{cases} 1 & if\ \Delta C \le -1 \\ \max\left\{0,\ 1 + \dfrac{-1-\Delta C}{0.5}\right\} & otherwise \end{cases} \tag{5.4}$$

$$\mu^{NS}(\Delta C) = \begin{cases} \max\left\{0, 1 + \dfrac{\Delta C + 0.5}{0.5}\right\} & if\ \Delta C \le -0.5 \\ \max\left\{0, 1 + \dfrac{-0.5-\Delta C}{0.5}\right\} & otherwise \end{cases} \tag{5.5}$$

$$\mu^{ZO}(\Delta C) = \begin{cases} \max\left\{0, 1 + \dfrac{\Delta C}{0.5}\right\} & if\ \Delta C \le 0 \\ \max\left\{0, 1 + \dfrac{-\Delta C}{0.5}\right\} & otherwise \end{cases} \tag{5.6}$$

$$\mu^{PS}(\Delta C) = \begin{cases} \max\left\{0, 1 + \dfrac{\Delta C - 0.5}{0.5}\right\} & if\ \Delta C \le 0.5 \\ \max\left\{0, 1 + \dfrac{0.5-\Delta C}{0.5}\right\} & otherwise \end{cases} \tag{5.7}$$

$$\mu^{PB}(\Delta C) = \begin{cases} \max\left\{0,\ 1 + \dfrac{\Delta C - 1}{0.5}\right\} & if\ \Delta C \le 1 \\ 1 & otherwise \end{cases} \tag{5.8}$$

After the fuzzification of the crisp inputs, the resulting fuzzy sets have to be compared to the rule-base. As shown in (4.9) the rule base is a set of "**If** premise **Then** consequent" rules constructed according to the designer system knowledge and experience. Depending on the value of the absolute power slope, the panel PV curve is divided into three regions as shown in Figure 5.4. Given the old perturbation step C_{old}, the controller will determine the change to the new step in order to reach the MPP.

Suppose that the operating point is at P_1, where the absolute value of the slope S_a is Large. This means that the operating point is far from the MPP. The old step C_{old} can have in this case three different values. If C_{old} is Small, then the change in step size ΔC has to be Positive Big (PB) in order to rapidly reach the MPP. Whereas if C_{old} is Medium, the change in step size ΔC has to be Positive Small (PS) in order to reach the MPP without oscillating around it. Finally if C_{old} is Large, the change in step size ΔC has to be Zero (ZO) in order to avoid exceeding the MPP in the opposite direction leading to oscillations.

As a result, Large S_a results in three rules, each with the following premise and conclusion function

- **If** S_a is Large **and** C_{old} is Small **Then** ΔC is Positive Big

$$\mu_{premise(1)} = \min(\mu_{Large}(S_a), \mu_{Small}(C_{old})) \tag{5.9}$$

$$\mu_{(1)}(\Delta C) = \min\{\mu_{PB}(\Delta C), \mu_{premise(1)}\} \tag{5.10}$$

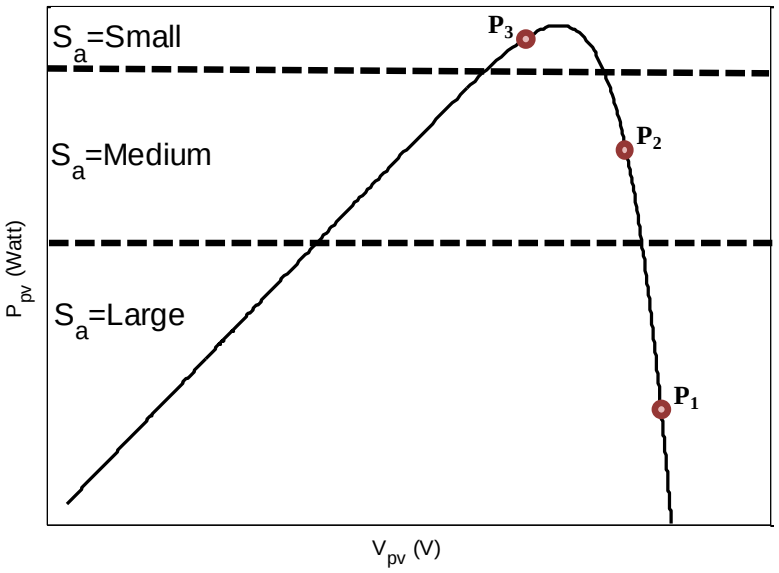


Figure 5.4: FLC based P&O PV curve illustration

- **If** S_a is Large **and** C_{old} is Medium **Then** ΔC is Positive Small

$$\mu_{premise(2)} = \min(\mu_{Large}(S_a), \mu_{Medium}(C_{old})) \tag{5.11}$$

$$\mu_{(2)}(\Delta C) = \min\{\mu_{PS}(\Delta C), \mu_{premise(2)}\} \tag{5.12}$$

- **If** S_a is Large **and** C_{old} is Large **Then** ΔC is Zero

$$\mu_{premise(3)} = \min(\mu_{Large}(S_a), \mu_{Large}(C_{old})) \tag{5.13}$$

$$\mu_{(3)}(\Delta C) = \min\{\mu_{ZO}(\Delta C), \mu_{premise(3)}\} \tag{5.14}$$

Suppose that the operating point is at P_2, where the absolute value of the slope S_a is Medium. This means that the operating point is nearer to the MPP compared to the previous case but still not quit at it. The old step C_{old} can have in this case three different values. If C_{old} is Small, then the change in step size ΔC has to be Positive Small (PB) in order to reach the MPP without oscillating around it. Whereas if C_{old} is Medium, the change in step size ΔC has to be Zero (ZO) in order to avoid exceeding the MPP in the opposite direction leading to oscillations. Finally if C_{old} is Large, the change in step size ΔC has to be Negative Small (NS) in order to not to exceed the MPP.

As a result, Medium S_a results in three rules each with the following premise and conclusion function

- **If** S_a is Medium **and** C_{old} is Small **Then** ΔC is Positive Small

$$\mu_{premise(4)} = \min(\mu_{Medium}(S_a), \mu_{Small}(C_{old})) \tag{5.15}$$

$$\mu_{(4)}(\Delta C) = \min\{\mu_{PS}(\Delta C), \mu_{premise(4)}\} \tag{5.16}$$

- **If** S_a is Medium **and** C_{old} is Medium **Then** ΔC is Positive Zero

$$\mu_{premise(5)} = \min(\mu_{Medium}(S_a), \mu_{Medium}(C_{old})) \tag{5.17}$$

$$\mu_{(5)}(\Delta C) = \min\{\mu_{ZO}(\Delta C), \mu_{premise(5)}\} \tag{5.18}$$

- **If** S_a is Medium **and** C_{old} is Large **Then** ΔC is Negative Small

$$\mu_{premise(6)} = \min(\mu_{Medium}(S_a), \mu_{Large}(C_{old})) \quad (5.19)$$

$$\mu_{(6)}(\Delta C) = \min\{\mu_{NS}(\Delta C), \mu_{premise(6)}\} \quad (5.20)$$

Suppose that the operating point is at P_3, where the absolute value of the slope S_a is Small. This means that the operating point is just close to the MPP. The old step C_{old} can have in this case three different values. If C_{old} is Small, then the change in step size ΔC has to be Zero (ZO) in order to avoid exceeding the MPP in the opposite direction leading to oscillations. Whereas if C_{old} is Medium, the change in step size ΔC has to be Negative Small (NS) in order to not to exceed the MPP. Finally if C_{old} is Large, the change in step size ΔC has to be Negative Big (NB) in order to not to exceed the MPP.

As a result, Small S_a results in three rules each with the following premise and conclusion function

- **If** S_a is Small **and** C_{old} is Small **Then** ΔC is Zero

$$\mu_{premise(7)} = \min(\mu_{Small}(S_a), \mu_{Small}(C_{old})) \quad (5.21)$$

$$\mu_{(7)}(\Delta C) = \min\{\mu_{ZO}(\Delta C), \mu_{premise(7)}\} \quad (5.22)$$

- **If** S_a is Small **and** C_{old} is Medium **Then** ΔC is Negative Small

$$\mu_{premise(8)} = \min(\mu_{Small}(S_a), \mu_{Medium}(C_{old})) \quad (5.23)$$

$$\mu_{(8)}(\Delta C) = \min\{\mu_{NS}(\Delta C), \mu_{premise(7)}\} \quad (5.24)$$

- **If** S_a is Small **and** C_{old} is Large **Then** ΔC is Negative Big

$$\mu_{premise(9)} = \min(\mu_{Small}(S_a), \mu_{Large}(C_{old})) \quad (5.25)$$

$$\mu_{(9)}(\Delta C) = \min\{\mu_{NB}(\Delta C), \mu_{premise(9)}\} \quad (5.26)$$

The FLC rules are shown in Table 5-1. The premise, which is the first part of the rule, is calculated using the inference minimum operator. The operator compares between the rules that are ON in each input MF and takes the minimum rule.

The last step in the FLC process is the defuzzification, which takes the implied fuzzy set and transforms it back to a real continuous number or a crisp output. The center of gravity (COG) defuzzification method shown in (4.10) is used in this work.

Table 5-1: FLC rules

C_{old} / S_a=dP/dV	Small	Medium	Large
Small	ZO	NS	NB
Medium	PS	ZO	NS
Large	PB	PS	ZO

$$\Delta C^{crisp} = \frac{(-1)\int \mu_{(9)}(\Delta C) + (-0.5)\int \mu_{(8)}(\Delta C) + (0)\int \mu_{(7)}(\Delta C)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{(-0.5)\int \mu_{(6)}(\Delta C) + (0)\int \mu_{(5)}(\Delta C) + (0.5)\int \mu_{(4)}(\Delta C)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{(0)\int \mu_{(3)}(\Delta C) + (0.5)\int \mu_{(2)}(\Delta C) + (1)\int \mu_{(1)}(\Delta C)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} \quad (5.27)$$

Substituting with the premises values from (5.19)-(5.26)

$$\Delta C^{crisp} = \frac{(-1)\left(\mu_{premise(9)} - \frac{\left(\mu_{premise(9)}\right)^2}{2}\right) + (-0.5)\left(\mu_{premise(8)} - \frac{\left(\mu_{premise(8)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{(-0.5)\left(\mu_{premise(6)} - \frac{\left(\mu_{premise(6)}\right)^2}{2}\right) + (0.5)\left(\mu_{premise(4)} - \frac{\left(\mu_{premise(4)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{(0.5)\left(\mu_{premise(2)} - \frac{\left(\mu_{premise(2)}\right)^2}{2}\right) + (1)\left(\mu_{premise(1)} - \frac{\left(\mu_{premise(1)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} \quad (5.28)$$

where

$$\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C) = \left(\mu_{premise(9)} - \frac{\left(\mu_{premise(9)}\right)^2}{2}\right) + \left(\mu_{premise(8)} - \frac{\left(\mu_{premise(8)}\right)^2}{2}\right) + \left(\mu_{premise(7)} - \frac{\left(\mu_{premise(7)}\right)^2}{2}\right) + \left(\mu_{premise(6)} - \frac{\left(\mu_{premise(6)}\right)^2}{2}\right) + \left(\mu_{premise(5)} - \frac{\left(\mu_{premise(5)}\right)^2}{2}\right) + \left(\mu_{premise(4)} - \frac{\left(\mu_{premise(4)}\right)^2}{2}\right) + \left(\mu_{premise(3)} - \frac{\left(\mu_{premise(3)}\right)^2}{2}\right) + \left(\mu_{premise(2)} - \frac{\left(\mu_{premise(2)}\right)^2}{2}\right) + \left(\mu_{premise(1)} - \frac{\left(\mu_{premise(1)}\right)^2}{2}\right)$$

Equations (5.27)-(5.28) can be rearranged in the following form by introducing two functions f_1 and f_2.

$$\Delta C^{crisp} = -0.5 f_1(S_a, C_{old}) + 0.5 f_2(S_a, C_{old}) \tag{5.29}$$

where

$$f_1(S_a, C_{old}) = \frac{2\left(\mu_{premise(9)} - \frac{\left(\mu_{premise(9)}\right)^2}{2}\right) + \left(\mu_{premise(8)} - \frac{\left(\mu_{premise(8)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{\left(\mu_{premise(6)} - \frac{\left(\mu_{premise(6)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)}$$

$$f_2(S_a, C_{old}) = \frac{\left(\mu_{premise(4)} - \frac{\left(\mu_{premise(4)}\right)^2}{2}\right) + \left(\mu_{premise(2)} - \frac{\left(\mu_{premise(2)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)} + \frac{2\left(\mu_{premise(1)} - \frac{\left(\mu_{premise(1)}\right)^2}{2}\right)}{\sum_{i=1}^{9}\int \mu_{(i)}(\Delta C)}$$

5.3 STABILITY ANALYSIS

The optimization of the Fuzzy-MPPT algorithm is carried out by selecting the optimum choice of three main parameters characterizing the dynamic behavior of the whole system composed by the buck converter, PV array, and MPPT controller.

These parameters are the PI controller gains (K_p, K_i) of the inner voltage control loop, the sampling interval T_r of the reference voltage, and the amplitude of the reference voltage perturbation C. The stability of the whole system can be achieved by appropriate design of the PI controller and by proper selection of the sampling interval T_r.

5.3.1 Inner Voltage-Loop Stability Analysis

The stability of the inner voltage control loop can be analyzed using the Lyapunov stability method. A positive definite function V (t) is defined as:

$$V(t) = \frac{1}{2} e^2(t) \tag{5.30}$$

where $e = V_{ref} - V_{in}$

and $\frac{de}{dt} = -\frac{dV_{in}}{dt}$

For step input reference, the derivative of $V(t)$ with respect to time is

$$\frac{dV(t)}{dt} = e(t) \frac{d(V_{ref} - V_{in})}{dt} \tag{5.31}$$

$$\frac{dV(t)}{dt} = -e(t) \frac{dV_{in}}{dt} \tag{5.32}$$

Taking the derivative of the buck converter input V_{in} with respect to time gives

$$\frac{dV_{in}}{dt} = \frac{dV_{in}}{dD} \frac{dD}{dt} \tag{5.33}$$

Defining a PI controller as $u = K_p e + K_i \int e\, dt$ and the duty cycle $D = (1 - u) = 1 - K_p e - K_i \int e\, dt$. The derivative of the control signal with respect to time is

$$\frac{dD}{dt} = -K_p \frac{de}{dt} - K_i e \tag{5.34}$$

$$\frac{dD}{dt} = K_p \frac{dV_{in}}{dt} - K_i e \tag{5.35}$$

Substitute $\frac{dD}{dt}$ in equation (5.33)

$$\frac{dV_{in}}{dt} = (K_p \frac{dV_{in}}{dt} - K_i e) \frac{dV_{in}}{dD} \tag{5.36}$$

Hence

$$\frac{dV_{in}}{dt}(1 - K_p \frac{dV_{in}}{dD}) = -K_i e \frac{dV_{in}}{dD} \tag{5.37}$$

$$\frac{dV_{in}}{dt} = \frac{-K_i e \frac{dV_{in}}{dD}}{(1 - K_p \frac{dV_{in}}{dD})} \tag{5.38}$$

As a result,

$$\frac{dV(t)}{dt} = \frac{K_i e^2 \frac{dV_{in}}{dD}}{(1 - K_p \frac{dV_{in}}{dD})} \tag{5.39}$$

Therefore, in order to prove the system stability, $\frac{dV(t)}{dt}$ can be made negative if $K_p > 0$, $K_i > 0$ and the system sensitivity function $\frac{dV_{in}}{dD} < 0$.

From the buck converter model equation (3.45), we have $V_{in} = I_{in} R_{in} = I_{in} \frac{R_o}{D^2}$. Using this equation in connection with the PV panel IV characteristics, we can observe the following two operating modes:

If the duty cycle D increases, then $\frac{R_o}{D^2}$ decreases and I_{in} will increase. As a result, V_{in} will decrease.

If the duty cycle D decreases, then $\frac{R_o}{D^2}$ increases and I_{in} will decrease. As a result, V_{in} will increase.

This shows that the rate of change in voltage is always opposite to the rate of change in duty cycle. This can be expressed in mathematical form as

$$\frac{dV_{in}}{dt} = -Ksign\left(\frac{dD}{dt}\right) \tag{5.40}$$

where K is a positive gain factor.

Multiplying both sides of equation (5.40) by $\frac{dD}{dt}$ gives

$$\frac{dV_{in}}{dt}\frac{dD}{dt} = -K\frac{dD}{dt}sign\left(\frac{dD}{dt}\right) \tag{5.41}$$

$$\left(\frac{dV_{in}}{dD}\frac{dD}{dt}\right)\frac{dD}{dt} = -K\frac{dD}{dt}sign\left(\frac{dD}{dt}\right) \tag{5.42}$$

$$\frac{dV_{in}}{dD}\left(\frac{dD}{dt}\right)^2 = -K\frac{dD}{dt}sign\left(\frac{dD}{dt}\right) \tag{5.43}$$

Since $\frac{dD}{dt}sign\left(\frac{dD}{dt}\right) > 0$ then $\frac{dV_{in}}{dD}$ is always negative. As a result $\frac{dV(t)}{dt}$ is negative and the inner voltage control loop is stable.

5.3.2 P&O Stability Analysis

Next, the stability of the P&O can be achieved by proper selection of the reference voltage-sampling interval T_r as follows. The sampling interval should be set higher than a proper threshold in order to avoid instability of the MPPT algorithm and to reduce the number of oscillations around the MPP in steady state. If the array voltage and current are sampled too fast, the algorithm may be subjected to possible mistakes caused by the transient behavior of the whole system, thus missing, even if temporarily, the current MPP of the PV panel, which is assumed to be in steady-state operation. As a result, the energy efficiency decays as the algorithm can be confused and the operating point can become unstable, entering disordered and/or chaotic behaviors [12].

To avoid this potential problem, it must be ensured that, after each reference voltage perturbation, the system reaches steady state before the next measurements of array voltage and current are acquired.

Assume that the inner closed loop is designed to behave like a second order dominant system with the desired maximum overshoot M_p and 2% settling time t_s.

$$\frac{\hat{v}_{in}(s)}{\hat{v}_{in,ref}(s)} \approx \frac{K_0\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \tag{5.44}$$

where ω_n is the natural frequency and ς is the damping factor of the dominant closed loop poles. The response of $\hat{v}_{in}$ to a small step voltage perturbation of amplitude C is

$$\hat{v}_{in}(t) = C\left[1 - \bar{e}^{\zeta\omega_n t}\cos\left(\omega_n\sqrt{1-\zeta^2}\,t\right) - \frac{\zeta\bar{e}^{\zeta\omega_n t}\sin\left(\omega_n\sqrt{1-\zeta^2}\,t\right)}{\sqrt{1-\xi^2}}\right] \tag{5.45}$$

where $\varsigma = \dfrac{-\ln M_p}{\sqrt{\pi^2 + \left(\ln M_p\right)^2}}$, $\omega_n \cong \dfrac{4}{\zeta t_s}$

Therefore $\hat{v}_{in}$ will keep within the 2% range of the reference voltage if the sampling interval T_r is higher than t_s $\left(T_r >> t_s\right)$.

5.3.3 Fuzzy-MPPT Stability Analysis

Finally, the stability of the Fuzzy-MPPT is studied by analyzing the recursive equation of the overall Fuzzy-MPPT algorithm

$$V_{ref}(k) = V_{ref}(k-1) \pm C(k) \tag{5.46}$$

this can be written as

$$V_{ref}(k) = V_{ref}(k-1) + sign(S(k))\,C(k) \tag{5.47}$$

where

$$S(k) = \frac{\Delta P(k)}{\Delta V(k)} = \frac{P(k) - P(k-1)}{V(k) - V(k-1)}, \tag{5.48}$$

$$C(k) = C(k-1) + \Delta C(k), \tag{5.49}$$

$$\Delta C(k) = f\left(S_a(k), C(k-1)\right), \tag{5.50}$$

and

$$S_a(k) = |S(k)| = \left|\frac{\Delta P(k)}{\Delta V(k)}\right| \tag{5.51}$$

Substituting in (5.47)

$$V_{ref}(k) = V_{ref}(k-1) + sign(S(k))\left(C(k-1) + f\left(S_a(k), C(k-1)\right)\right) \tag{5.52}$$

hence

$$\begin{bmatrix} V_{ref}(k) \\ C(k) \end{bmatrix} = \begin{bmatrix} 1 & sign(S(k)) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{ref}(k-1) \\ C(k-1) \end{bmatrix} + \begin{bmatrix} sign(S(k)) \\ 1 \end{bmatrix} f\left(S_a(k), C(k-1)\right) \tag{5.53}$$

In steady state

$$V_{ref}(k) = V_{ref}(k-1) = V_{ref,ss}$$

$$C(k) = C(k-1) = C_{ss}$$

as a result

$$\begin{bmatrix} V_{ref,ss} \\ C_{ss} \end{bmatrix} = \begin{bmatrix} 1 & sign(S(k)) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{ref,ss} \\ C_{ss} \end{bmatrix} + \begin{bmatrix} sign(S(k)) \\ 1 \end{bmatrix} f\left(S_{a,ss}, C_{ss}\right) \tag{5.54}$$

When the operating point reaches the MPP in steady state, $S_{a,ss}=0$ and $C_{ss}=0$ hence

$$\begin{bmatrix} V_{ref,ss} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & sign(S(k)) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{ref,ss} \\ 0 \end{bmatrix} + \begin{bmatrix} sign(S(k)) \\ 1 \end{bmatrix} f(0,0) \tag{5.55}$$

This is leads to

$$V_{ref,ss} = V_{ref,ss} + sign(S(k))\, f(0,0) \tag{5.56}$$

In order to prove our Fuzzy-MPPT convergence, we need to show that $\lim_{S_a, C_{old} \to 0} f(0,0) = 0$.

This limit can be found through the crisp output function given by (5.29):

$$\lim_{S_a, C_{old} \to 0} \Delta C^{crisp} = -0.5 \lim_{S_a, C_{old} \to 0} f_1(S_a, C_{old}) + 0.5 \lim_{S_a, C_{old} \to 0} f_2(S_a, C_{old}) \tag{5.57}$$

Finding this limit using (5.19)-(5.26) yields

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(9)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 - \frac{S_a}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 1}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ 1\}, \max\{0, -1\}] = \min[1{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(8)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 - \frac{S_a}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 0.5}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ 1\}, \max\{0, 0\}] = \min[1{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(7)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0,\ 1 - \frac{S_a}{0.5}\right\}, \max\left\{0,\ 1 - \frac{C_{old}}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ 1\}, \max\{0, 1\}] = \min[1{,}1] = 1$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(6)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 0.5}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 1}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ 0\}, \max\{0, -1\}] = \min[0{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(5)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 0.5}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 0.5}{0.5}\right\}\right] \quad (5.58)$$
$$= \min[\max\{0,\ 0\}, \max\{0, 0\}] = \min[0{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(4)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 0.5}{0.5}\right\}, \max\left\{0{,}1 - \frac{C_{old}}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ 0\}, \max\{0, 1\}] = \min[0{,}1] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(3)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 1}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 1}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ -1\}, \max\{0, -1\}] = \min[0{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(2)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 1}{0.5}\right\}, \max\left\{0{,}1 + \frac{C_{old} - 0.5}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ -1\}, \max\{0, 0\}] = \min[0{,}0] = 0$$

$$\lim_{S_a, C_{old} \to 0} \mu_{premise(1)} = \lim_{S_a, C_{old} \to 0} \min\left[\max\left\{0{,}1 + \frac{S_a - 1}{0.5}\right\}, \max\left\{0{,}1 - \frac{C_{old}}{0.5}\right\}\right]$$
$$= \min[\max\{0,\ -1\}, \max\{0, 1\}] = \min[0{,}1] = 0$$

As a result,

$$\lim_{S_a, C_{old} \to 0} \sum_{i=1}^{9} \int \mu_{(i)}(\Delta C) = \lim_{S_a, C_{old} \to 0} \left(\mu_{premise(7)} - \frac{\left(\mu_{premise(7)}\right)^2}{2} \right) = 0.5 \quad (5.59)$$

Substituting in (5.58) and (5.59) in (5.57)

$$\lim_{S_a, C_{old} \to 0} f_1(S_a, C_{old}) = \lim_{S_a, C_{old} \to 0} \left\{ \frac{2\left(\mu_{premise(9)} - \frac{\left(\mu_{premise(9)}\right)^2}{2}\right) + \left(\mu_{premise(8)} - \frac{\left(\mu_{premise(8)}\right)^2}{2}\right)}{\sum_{i=1}^{9} \int \mu_{(i)}(\Delta C)} + \frac{\left(\mu_{premise(6)} - \frac{\left(\mu_{premise(6)}\right)^2}{2}\right)}{\sum_{i=1}^{9} \int \mu_{(i)}(\Delta C)} \right\} = \lim_{S_a, C_{old} \to 0} \frac{0}{0.5} = 0 \quad (5.60)$$

$$\lim_{S_a, C_{old} \to 0} f_2(S_a, C_{old}) = \lim_{S_a, C_{old} \to 0} \left\{ \frac{\left(\mu_{premise(4)} - \frac{\left(\mu_{premise(4)}\right)^2}{2}\right) + \left(\mu_{premise(2)} - \frac{\left(\mu_{premise(2)}\right)^2}{2}\right)}{\sum_{i=1}^{9} \int \mu_{(i)}(\Delta C)} + \frac{2\left(\mu_{premise(1)} - \frac{\left(\mu_{premise(1)}\right)^2}{2}\right)}{\sum_{i=1}^{9} \int \mu_{(i)}(\Delta C)} \right\} = \lim_{S_a, C_{old} \to 0} \frac{0}{0.5} = 0$$

As a conclusion

$$\lim_{S_a, C_{old} \to 0} f(0,0) = \lim_{S_a, C_{old} \to 0} \Delta C^{crisp} = 0 \quad (5.61)$$

Equation (5.61) shows that if the system reaches MPP with $V_{ref}(k) = V_{ref,ss}$ and the FLC output $C(k) = C_{ss} = 0$, then the system will stay operating at this stable operating point unless the temperature or the insolation conditions change.

5.4 PV SYSTEM SIMULATION

In order to verify the proposed controller operation, the FLC based P&O-MPPT was integrated into the Simulink PV panel model derived in section II. Through simulation, the proposed technique is compared to the conventional P&O-

MPPT with two different duty cycle steps under varying insolation conditions. The inner-voltage control loop sampling time T_s was set to 0.3 seconds while the P&O sampling time T_r was set to be 15 seconds in order to give the PV voltage enough time to reach the reference value before the next perturbation step.

5.4.1 PV Panel Characteristic

The complete PV system simulation is constructed using the derived PV panel and power converter models. The simulated system consists of a PV panel, a DC-DC converter, and a MPPT controller as shown in Figure 5.5.

The PV panel block consists of an Embedded MATLAB Function containing the derived PV panel model shown in equation (3.18). The panel takes the temperature (T), the nominal sun insolation (Suns) and the panel voltage (V_{panel}) as inputs and outputs the panel current (I_{panel}). As shown in Figure 5.6 the buck converter block consists of a representation of equations (3.27)-(3.38). The blocks representing each equation are shown in Figure 5.7.

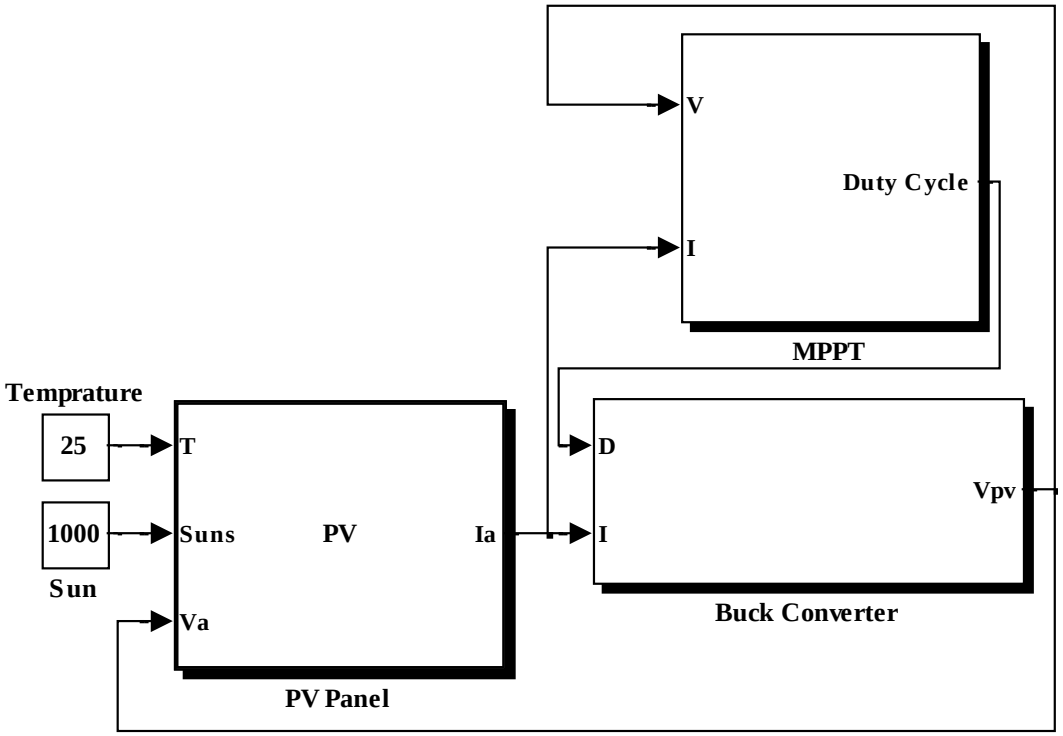


Figure 5.5: PV system simulation using MATLAB/Simulink

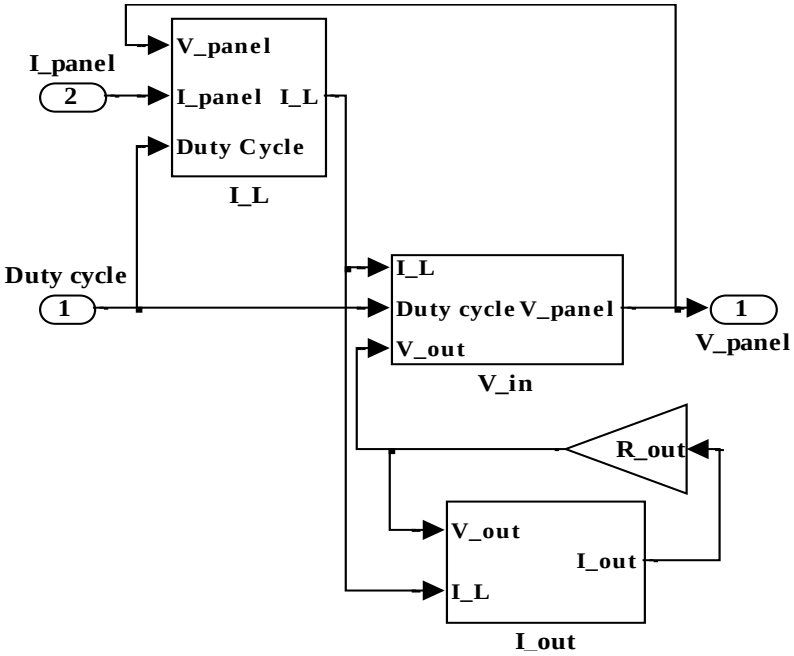
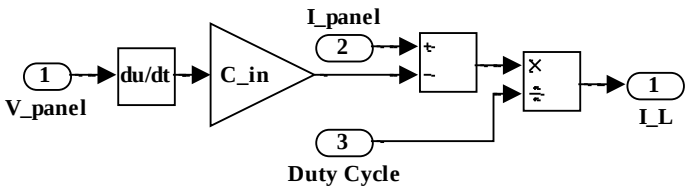
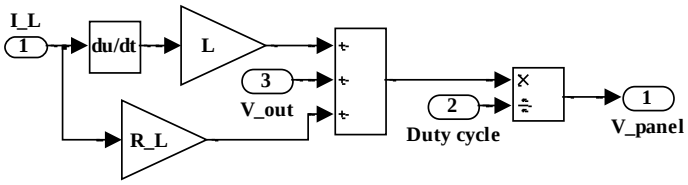


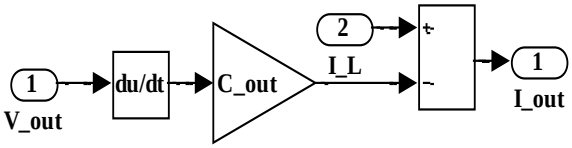
Figure 5.6: Buck converter block content



(a)



(b)



(c)

Figure 5.7: a) Buck converter inductor current representation b) Buck converter input voltage representation c) Buck converter output current representation

The complete PV system simulation is constructed using the derived PV panel and power converter model. The parameters of the Isofoton 150-Watt PV panel and the buck converter are shown in Table 5-2 and Table 5-3 respectively. Figure 5.8 shows the simulated PV panel characteristic under standard testing conditions.

Comparing Figure 5.8 with the manufacturer datasheet characteristics shown in Figure 5.9 validates the simulated results.

The MATLAB/Simulink model runs at a sampling rate of 0.05 sec. The model is used to get the PV panel characteristic through varying the power converter input duty cycle in fixed steps under STC conditions.

Figure 5.8 shows the simulated PV panel characteristic under standard testing conditions. Comparing Figure 5.8 with the manufacturer datasheet characteristics shown in Figure 5.9 validates the simulated results.

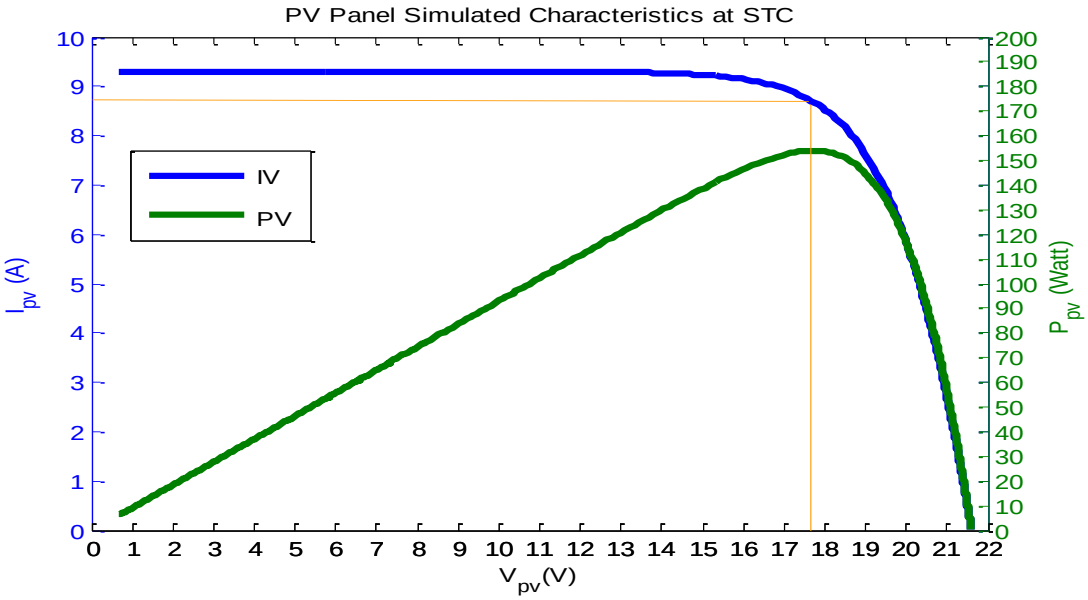


Figure 5.8: PV panel simulated characteristics at STC

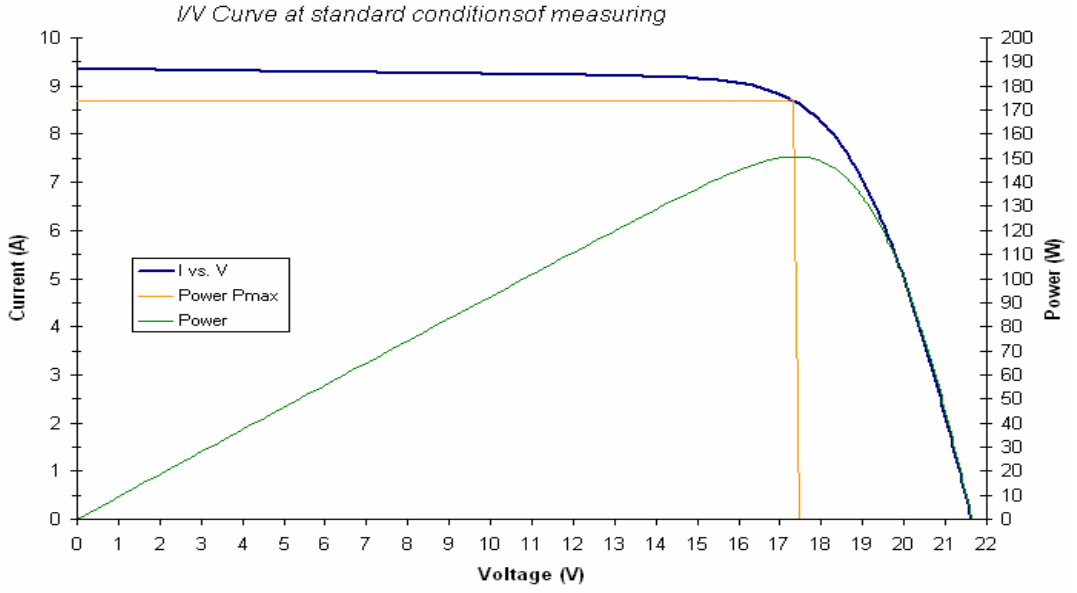


Figure 5.9: PV panel datasheet characteristics at STC

Table 5-2: PV panel parameters

P_{max}	V_{max}	I_{max}	V_{OC}	I_{SC}	N_p	N_s	FF
150 W	17.3 V	8.7 A	21.6 V	9.3 A	2	36	0.75

Table 5-3: Buck converter parameters

L	R	C_1	C_2	R_o
330 mH	0.074 Ω	470 µF	330 µF	1.6 Ω

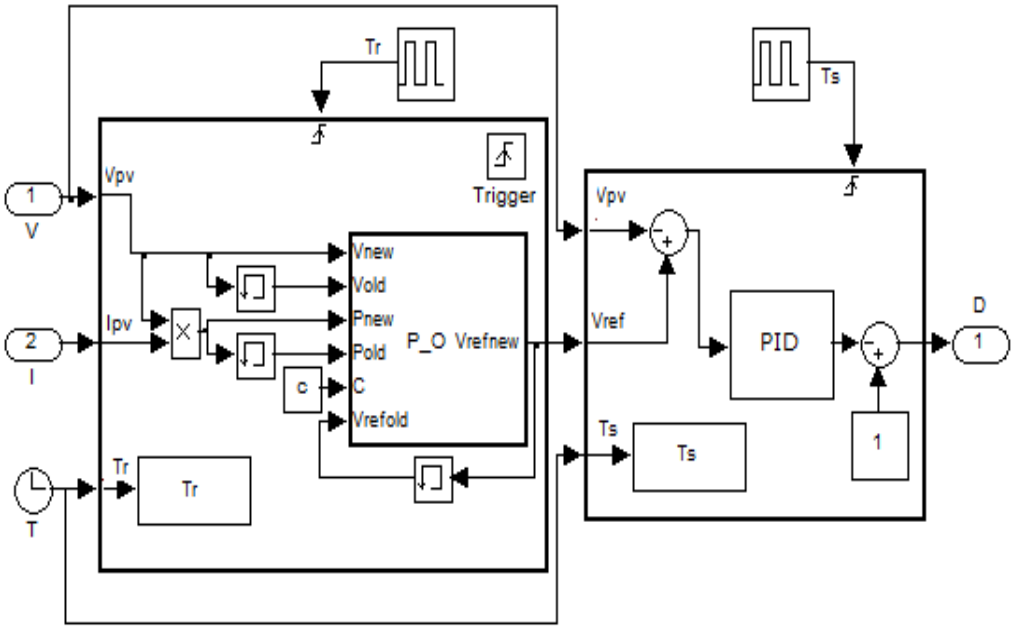


Figure 5.10: P&O-MPPT Simulink model

5.4.2 Conventional P&O Simulation

The P&O-MPPT model shown in Figure 5.10 was included in the MPPT block given in Figure 5.5. The diagram represents the P&O algorithm triggered at T_r as well as the inner voltage control loop triggered at T_s.

The P&O algorithm is simulated using two different fixed reference voltage steps ΔV_{ref} = 0.8 V and ΔV_{ref} = 0.2 V as shown in Figure 5.11 and Figure 5.12 respectively. The simulation results show that the P&O with the larger step size reaches the MPP faster than the one with the smaller step size, but on the expense of the steady state oscillations.

5.4.3 FLC Based P&O Simulation

In order to test the proposed FLC-MPPT technique, the MATLAB Fuzzy Toolbox was used to design the fuzzy controller. The first step in designing the controller is to define the FLC design parameters (inputs, outputs, Defuzzification method …etc) in the Fuzzy Inference System (FIS) Editor as shown in Figure 5.13. Then, each membership function is defined and named as shown in Figure 5.14.

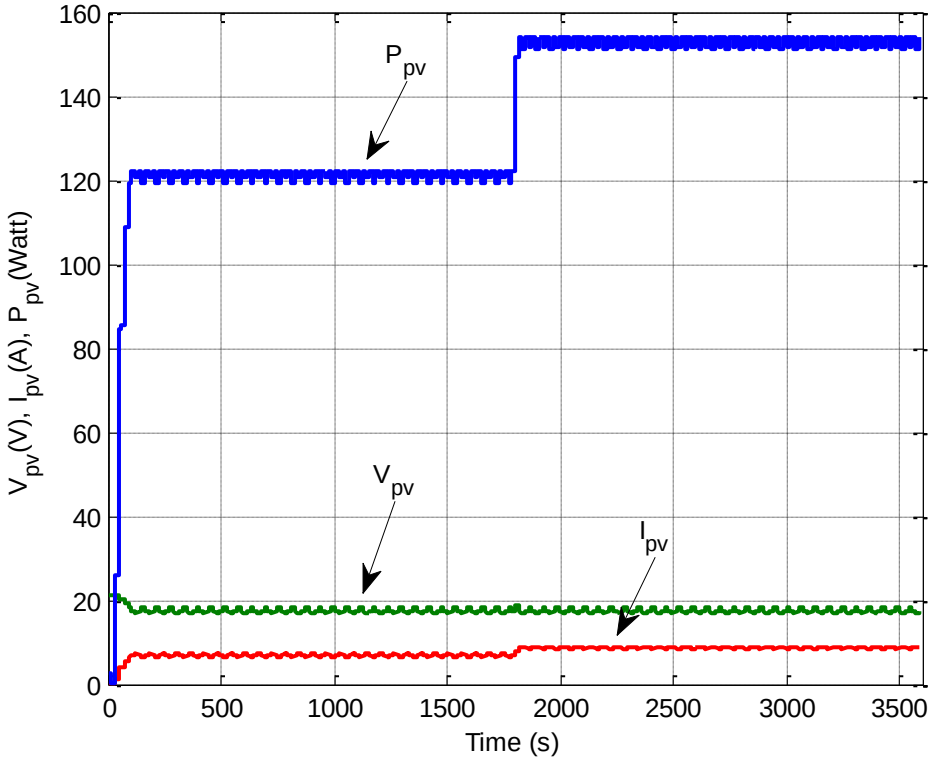


Figure 5.11: P&O algorithm simulation results with sun insolation step changes at t=0 s from 0 W/m^2 to 800 W/m^2 and at t=1800 s from 800 W/m^2 to 1000 W/m^2 and T=25^{o}C. ΔV_{ref} = 0.8 V

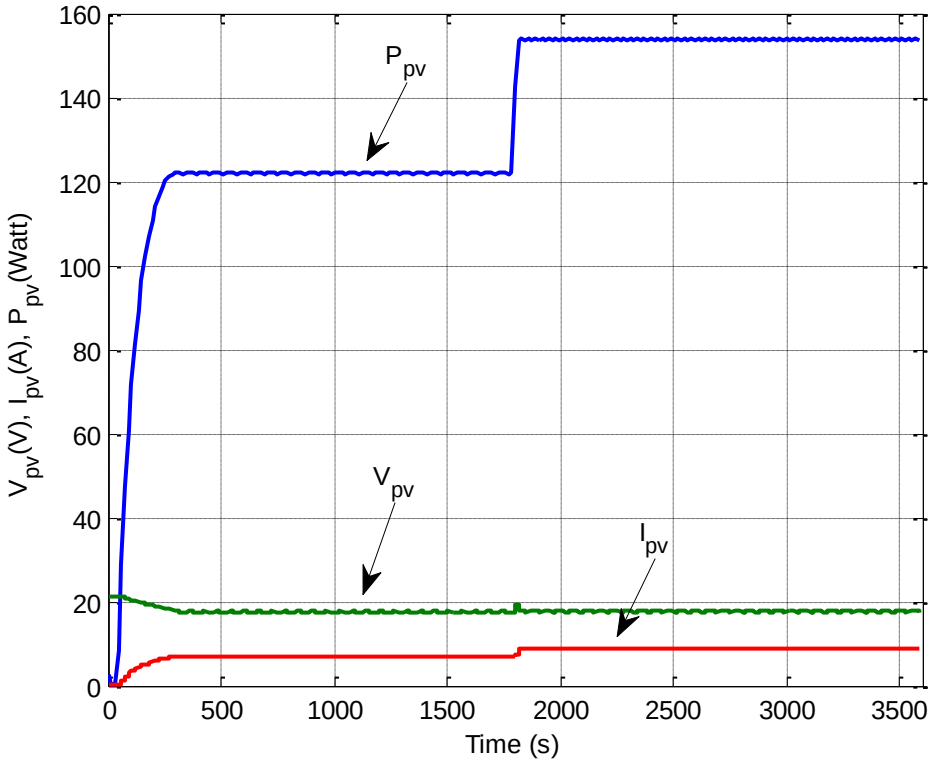


Figure 5.12: P&O algorithm simulation results with sun insolation step changes at t=0 s from 0 W/m^2 to 800 W/m^2 and at t=1800 s from 800 W/m^2 to 1000 W/m^2 and T=25^{o}C. ΔV_{ref} =0.2 V

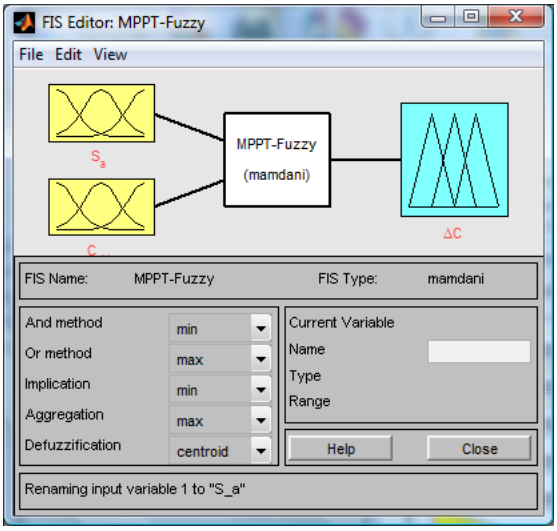
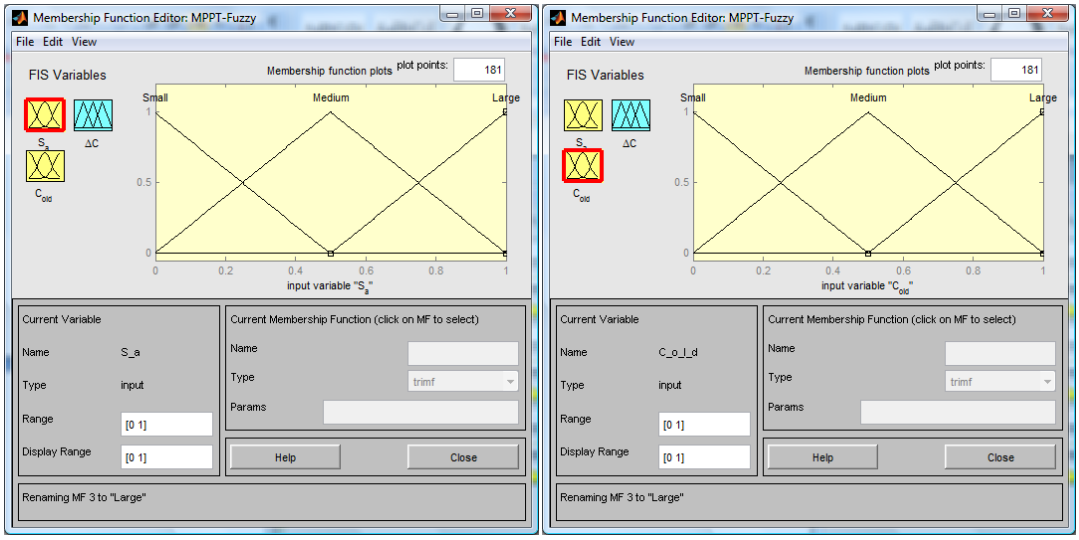
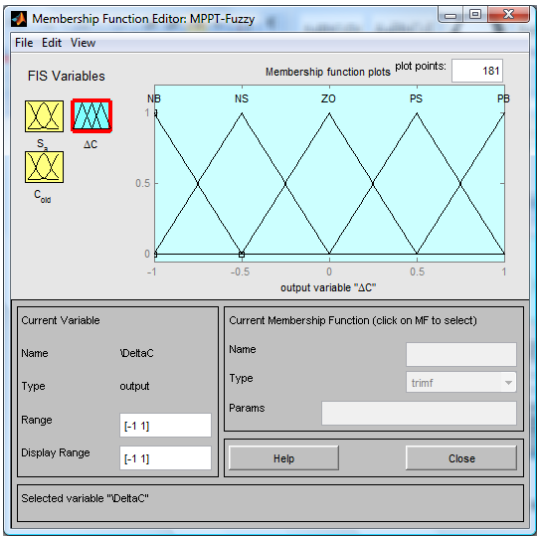


Figure 5.13: FIS editor MATLAB window



a) b)



c)

Figure 5.14: Membership function editor, MATLAB window a) S_a MF b) C_{old} MF c) ΔC MF

The last step is to define the rule base that maps the inputs to the output as shown in Figure 5.15.

The P&O block diagram shown in Figure 5.10 is altered to incorporate the FLC based step size technique as shown in Figure 5.16. The input data is normalized by tuning the gains g_{S_a} , $g_{C_{old}}$.

While running the simulation, Simulink shows the rule viewer. This viewer displays a roadmap of the whole fuzzy inference process as shown in Figure 5.17. The first two columns of plots show the inputs membership functions, the third column of plots shows the output membership functions. The last plot in the third column of plots represents the aggregate weighted decision for the given inference system. The defuzzified output is displayed as a bold vertical red line on this plot.

Moreover, the toolbox provides the designer with the Surface Viewer, which is a three-dimensional curve that represents the mapping from the FLC two inputs to the output as shown in Figure 5.18.

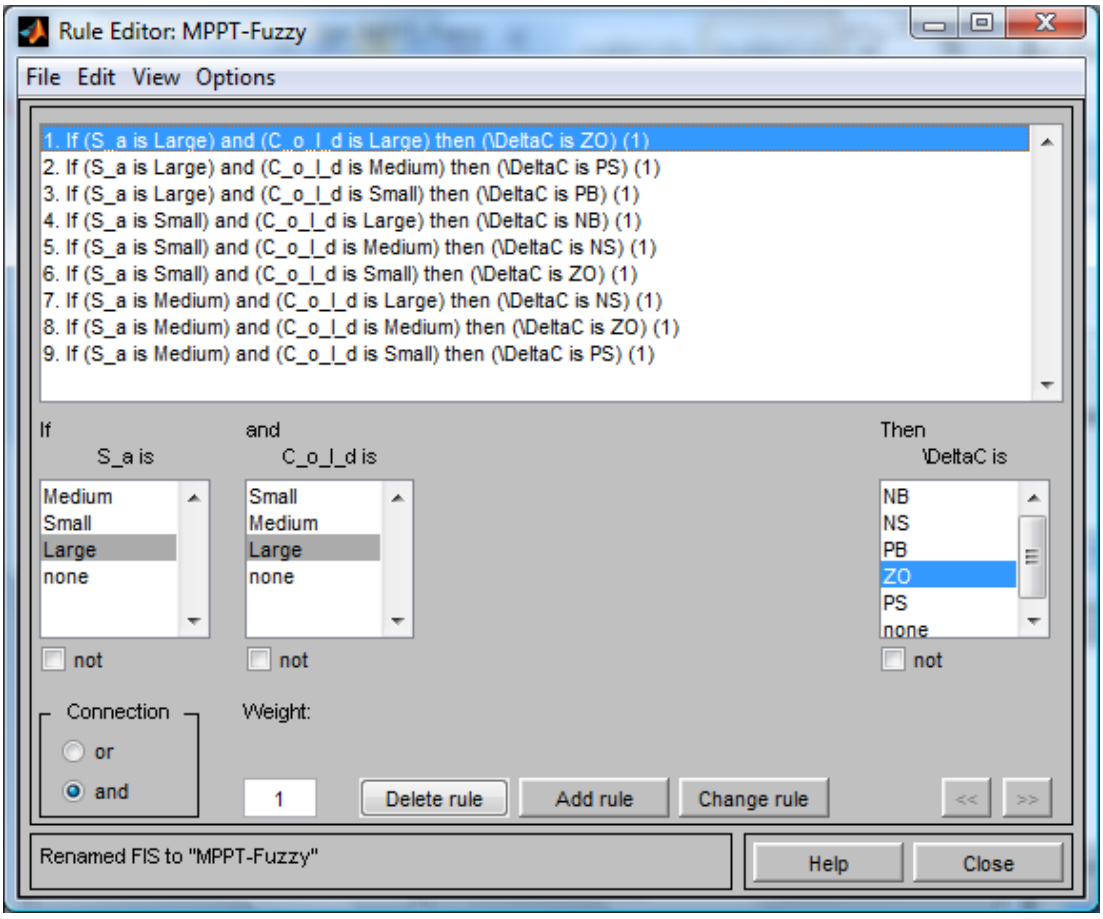
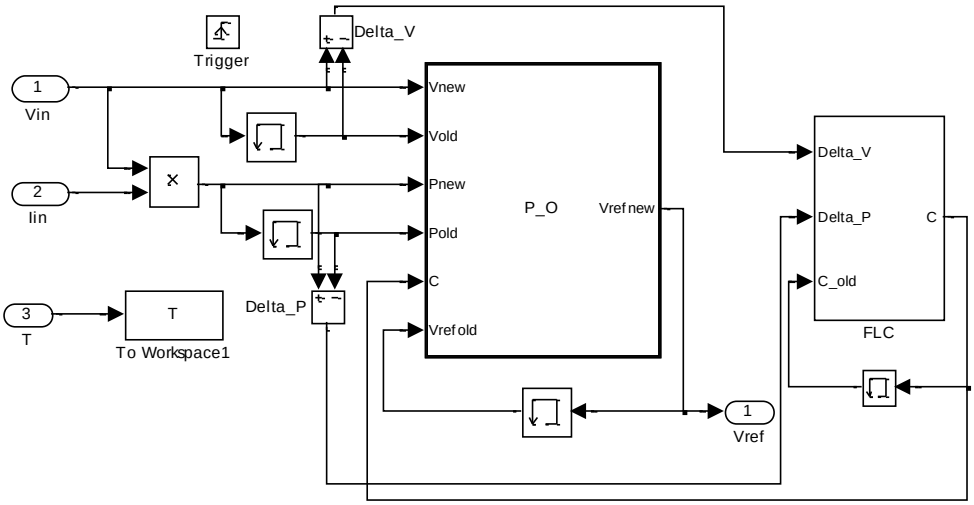
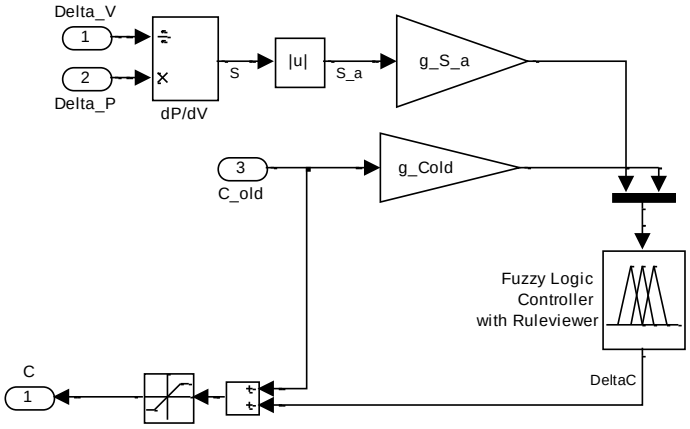


Figure 5.15: Rule editor MATLAB window



a)



b)

Figure 5.16: a) FLC based P&O-MPPT algorithm b) FLC block content

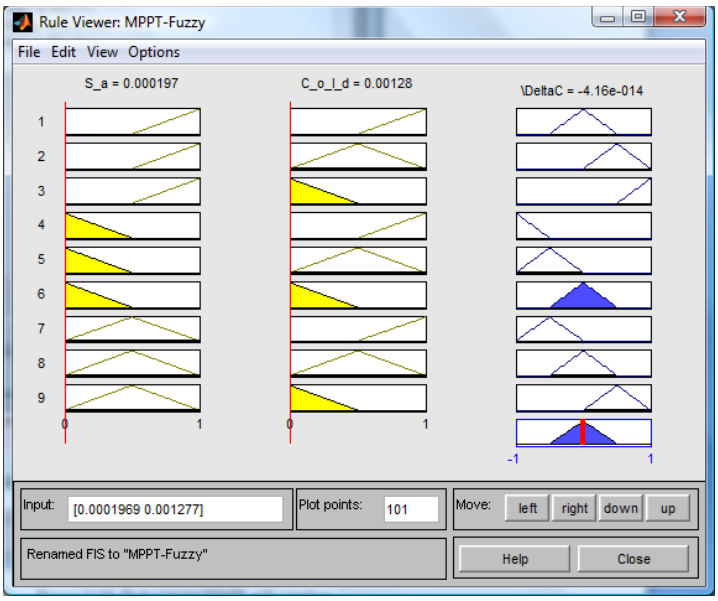


Figure 5.17: Rule viewer MATLAB window

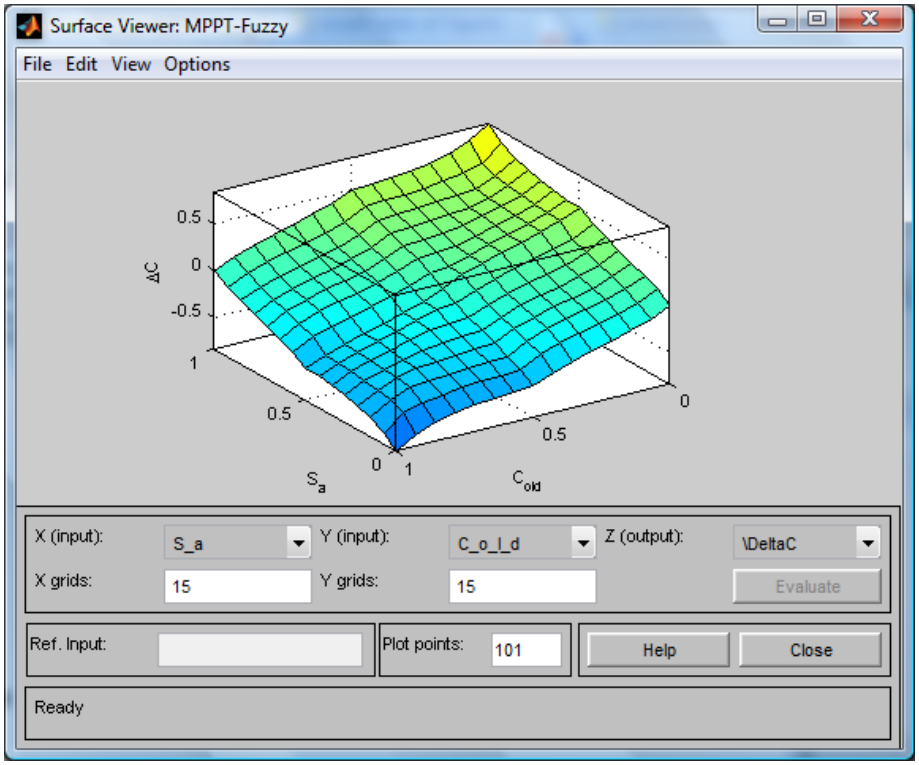


Figure 5.18: Surface viewer MATLAB window

Figure 5.19 shows the simulation results of the proposed controller. Figure 5.20 further illustrates the difference between the three controllers as it shows the MPPT simulated curves for three controllers: P&O with voltage step 0.8 V, P&O with voltage step 0.2 V, and Fuzzy-based P&O. The simulation results clearly show that the proposed Fuzzy-based P&O technique has a better performance compared to the conventional P&O (faster settling time and less steady state oscillations) resulting in higher energy produced by the PV panel.

Figure 5.21 further illustrates the higher performance of the proposed technique as it shows the overall controller output (duty cycle) produced by the three controllers. The results clearly illustrates that the FLC based P&O reaches faster to the optimal duty cycle with less oscillations compared to the conventional P&O. The FLC based P&O has a faster response to the change in insolation condition.

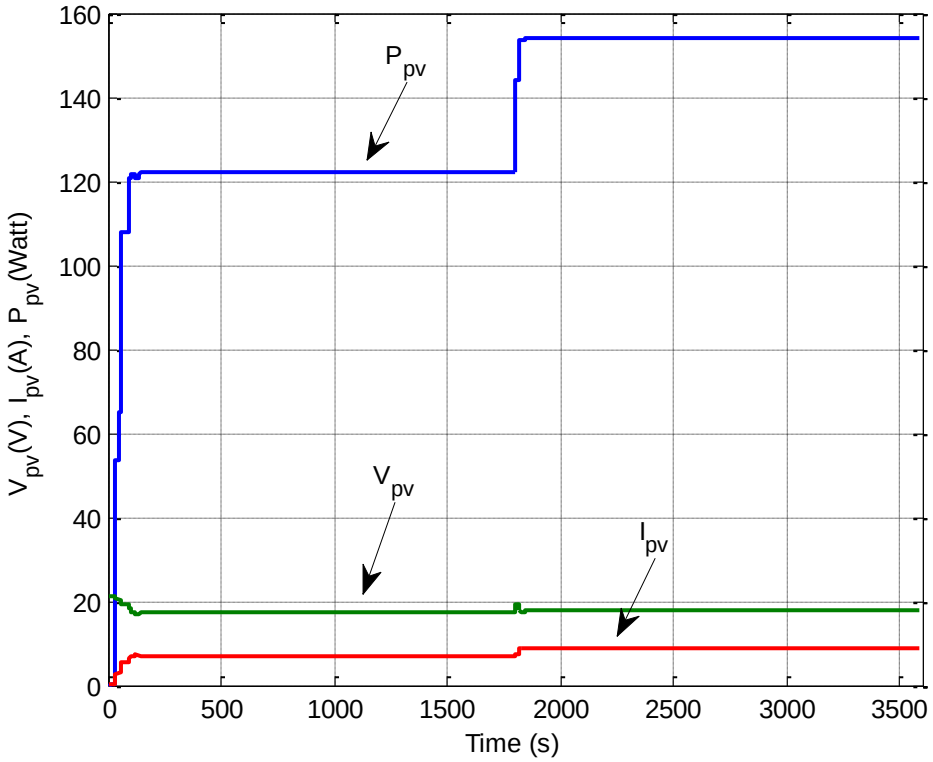


Figure 5.19: FLC based P&O simulation results with sun insolation step changes at t=0 s from 0 W/m^2 to 800 W/m^2 and at t=1800 s from 800 W/m^2 to 1000 W/m^2 and T=25^{o}C

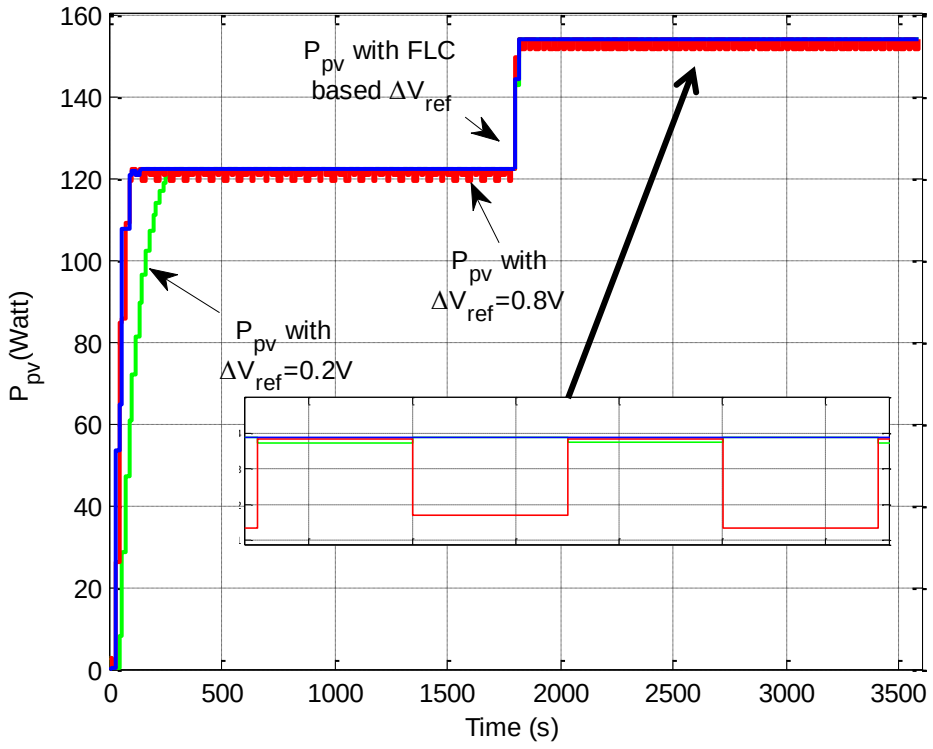


Figure 5.20: P&O algorithms power curves comparison simulation results with sun insolation step changes at t=0 s from 0 W/m^2 to 800 W/m^2 and at t=1800 s from 800 W/m2 to 1000 W/m2 and T=25^{o}C

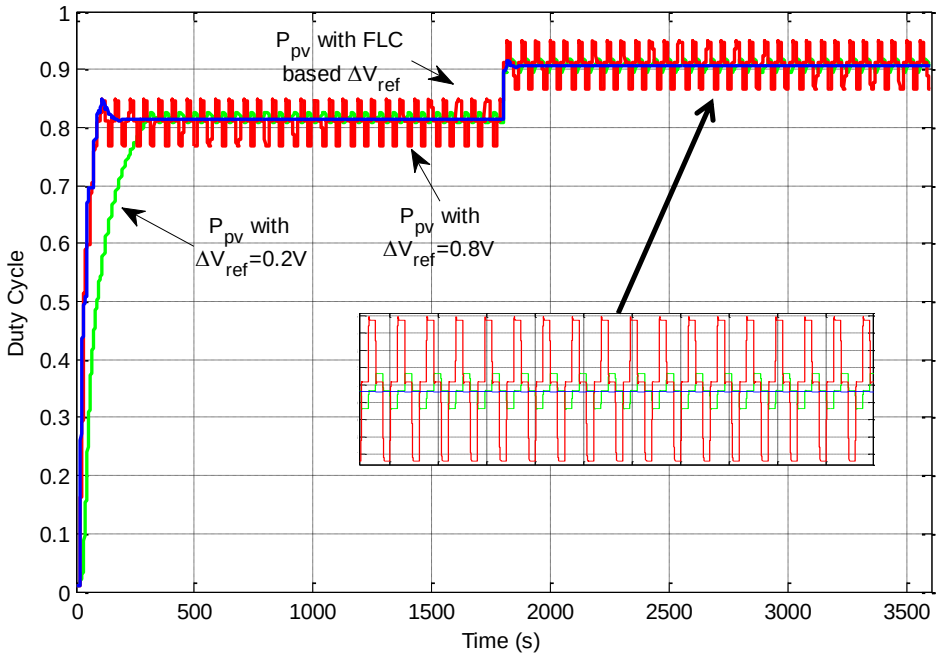


Figure 5.21: Duty cycle curves comparison simulation results with sun insolation step changes at t=0 s from 0 W/m^2 to 800 W/m^2 and at t=1800 s from 800 W/m2 to 1000 W/m2 and T=25^{o}C

CHAPTER 6 : Hardware Development & Experimental Results

6.1 PV SYSTEM HARDWARE AND STRUCTURE

The 150-Watt PV system shown in Figure 5.1 was implemented. Figure 5.8 shows the simulated PV panel characteristic under standard testing conditions. Comparing Figure 5.8 with the manufacturer datasheet characteristics shown in Figure 5.9 validates the simulated results.

Table 5-2 shows the parameters of the PV panel used. The control algorithms are developed on two separate DSPs. The first DSP is solely dedicated to the solar tracker while the second DSP is for the FLC-based MPPT.

6.1.1 Digital Signal Processor Structure

The TMS320F28335 DSP from Texas Instruments has a high performance CMOS technology with 150-MHz clock frequency, 3.3-V I/O design and 5-V DC power supply. The DSP includes also a 32-bit CPU with an IEEE-754 single-precision floating-point unit for fast interrupt response, efficient programming in C/C++ and Assembly, and a unified memory-programming model. The DSP contains six channels for direct memory access (DMA), which is used for ADC, PWM, McBSP (multichannel buffered serial port), XINTF (external interface), and SARAM (single-access RAM). In addition, the DSP supplies a great deal of operations compared to its small size. The DSP provides up to 18 ePWM (enhanced PWM) outputs, 6 HRPWM (high resolution PWM) outputs, 6 eCAP (enhanced capture) inputs, 2 quadrature encoder interfaces, three 32-bit CPU timers, 2 CAN (controller area network) modules, and one 16 channel 12-bit ADC with a conversion time of 80 ns. The F28335 DSP has also a 512-KB flash and a 68-KB RAM.

The control code is developed by the operator on a laptop using Code Composer Studio and then downloaded on the DSPs for real-time operation. Code Composer Studio is an integrated environment used to support the TMS320F28335 DSP software development. The programming environment is a practical and user-friendly tool, which includes an editor, a code generation tool, a debugger, a project manager, and a profiler. Code Composer Studio also provides probe points, real-time analysis, and visualization tools.

6.1.2 Physical Tracker Design

In order to track the sun trajectory, a two-axis tracker with a 150-Watt PV panel was manufactured as illustrated in Figure 6.2. The tracker base has a rotary table mechanism with a DC motor and a worm gear drive that rotates the panel about the vertical axis. The rotary table includes in addition a thrust bearing that is able to support the load caused by the weight of the structure. The panel is held using a frame, which is mounted on top of the rotary table. The panel is tilted around the horizontal axis using another DC motor connected to the panel through a gear with a ratio 240:1.

As explained in 3.2, the tracker is controlled using an astronomical angular trajectory generation. As shown in Figure 6.3, given the date, time, latitude, and longitude of the current location of the system, the DSP controller calculates the sun reference azimuth and elevation angles using equations (3.4) and (3.5).

Figure 6.1: Texas Instrument's TMS320F28335 DSP board

Figure 6.2: Front view of the solar tracker mechanism (panel in vertical position)

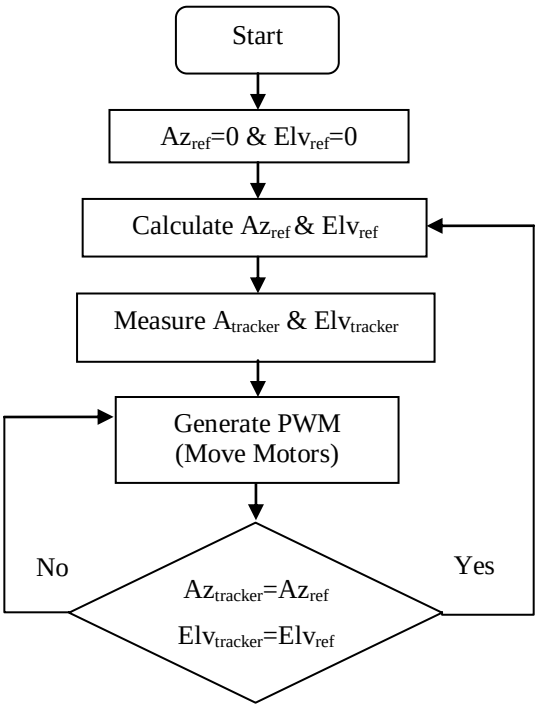


Figure 6.3: Tracker flow chart

The panel azimuth and elevation angles are next controlled to follow their reference values using a discrete PID controller for each axis as shown in Figure 6.4. Incremental optical encoders are connected to the motor shafts to provide the angular position and speed information to the DSP to implement the control algorithm.

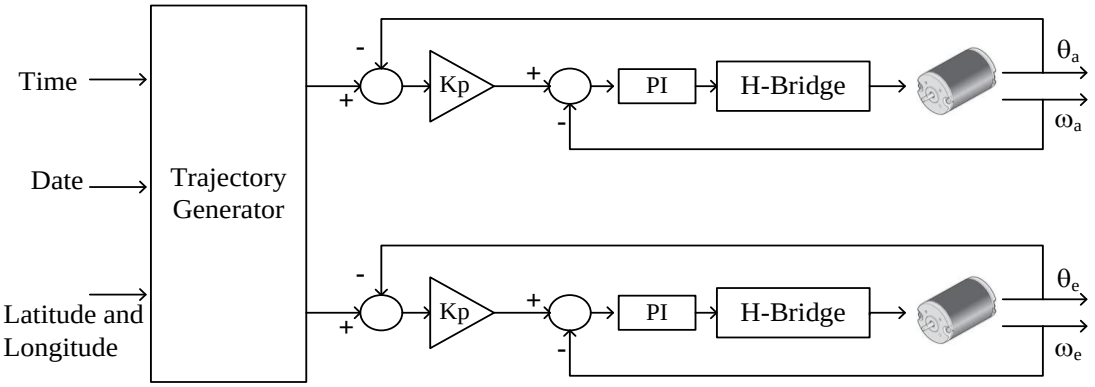


Figure 6.4: Block diagram of the tracker control system

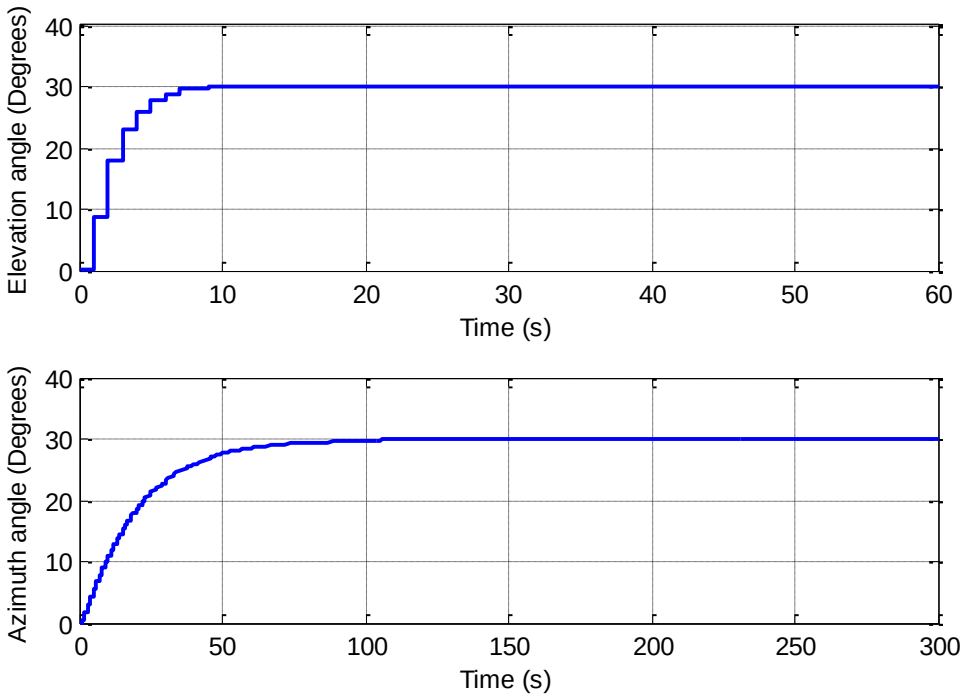


Figure 6.5: Experimental position step response (30°) for the elevation and azimuth axes

The PID gains are tuned to achieve an over-damped response with zero steady state error and a 2% settling time of around 6.75 seconds and 87 seconds for the elevation axis and azimuth axis respectively. Figure 6.5 shows the responses of the elevation and the azimuth axes to a step reference with amplitude 30°.

6.1.3 Power Converter Design

After verifying the buck converter operation using the Simulink model given in Figure 5.6 and Figure 5.7, the circuit shown in Figure 6.6 was built. The buck converter parameters are shown in Table 5-3.

6.1.4 Inner MPPT Voltage Control Loop Tuning

The performance of the inner MPPT voltage control loop is verified using step changes to the voltage reference. Figure 6.7 shows the response of the system to large and small voltage steps. For the system under consideration, the PI controller gains were optimized as $K_p = 0.01, K_i = 0.05$ to yield a settling time $t_s = 4.8\,s$. A sampling interval $T_r \approx 3 \times t_s = 15\,s$ is used in order to guaranty that the MPPT in not affected by mistakes caused by intrinsic transient oscillations of the PV system.

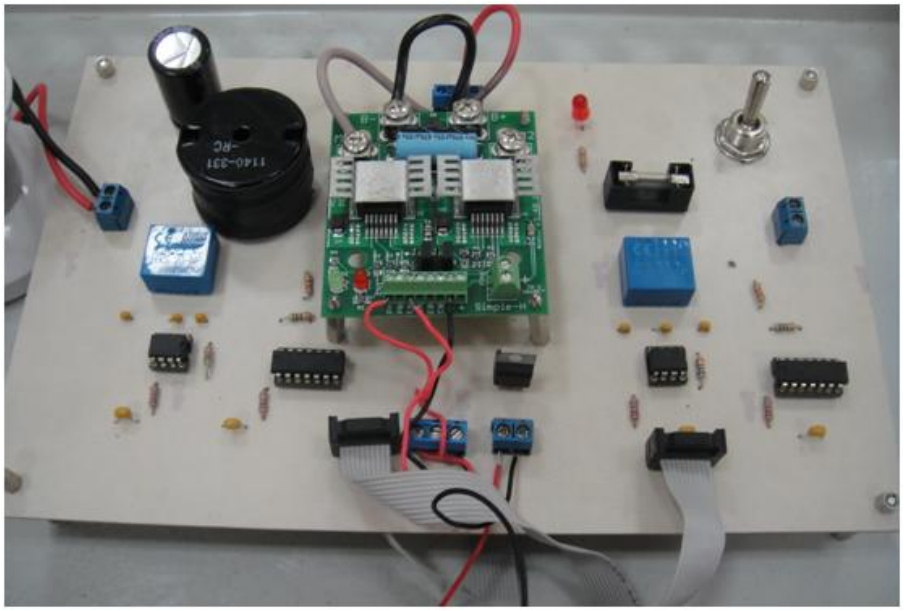


Figure 6.6: Buck converter circuit

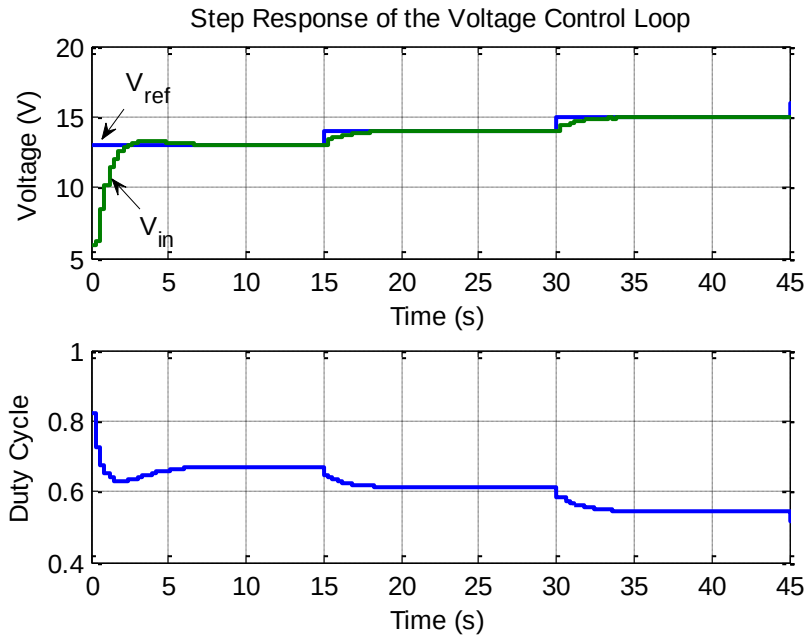


Figure 6.7: Inner voltage control loop step response

6.2 PERFORMANCE ANALYSIS OF THE MPPT CONTROLLERS

In order to analyze the dual-MPPT system operation, three tests were performed:

- Test 1: Fixed panel with conventional P&O-MPPT.
- Test 2: Fixed panel with FLC based P&O-MPPT.
- Test 3: Dual axis tracking of the panel with FLC based P&O-MPPT.

6.2.1 Conventional P&O Implementation Results

In Test 1 the panel was fixed facing true south with a tilt angle of 52 degrees which is the winter optimum angle given the location of Sharjah as shown in Table 3-1. The conventional P&O-MPPT shown in Figure 2.3 was applied to the power converter with a fixed reference step ΔV_{ref} = 0.8 V. The PV panel voltage and current were acquired and saved in the DSP. The experimental results of the conventional P&O technique are shown in Figure 6.8.

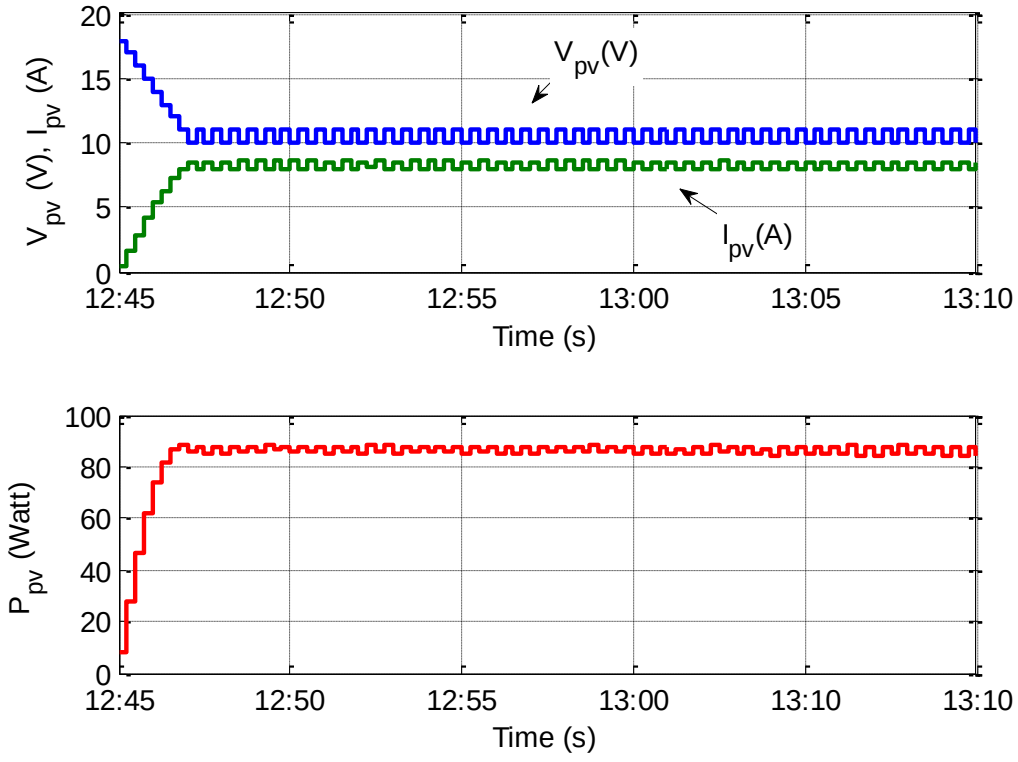


Figure 6.8: Testing results of conventional P&O with 0.8 voltage step (Nov. 21, 2011, 12:45-13:10 pm)

6.2.2 FLC Based P&O Implementation Results

In Test 2, the panel has the same panel fixture as test 1. The FLC based P&O-MPPT shown in Figure 5.1 was applied to the power converter. The experimental results of the FLC P&O technique are shown in Figure 6.9.

6.2.3 Controllers Implementation Comparison

Tests 1 and 2 compare between the proposed technique and the conventional P&O. The proposed FLC based P&O results show that the maximum power is reached within a short time (5 samples) and the oscillations around the MPP are minimal. On the other hand when the conventional P&O with ΔV_{ref} = 0.8 V was implemented, the response was slower as it reached the MPP within (six samples) and kept oscillating around it. Figure 6.10 shows the power output curve of the P&O-MPPT compared to the Fuzzy-MPPT. This figure highlights clearly the features of the proposed fuzzy based MPPT, which improves the efficiency of the system by yielding higher power while eliminating the steady state oscillations around the MPP.

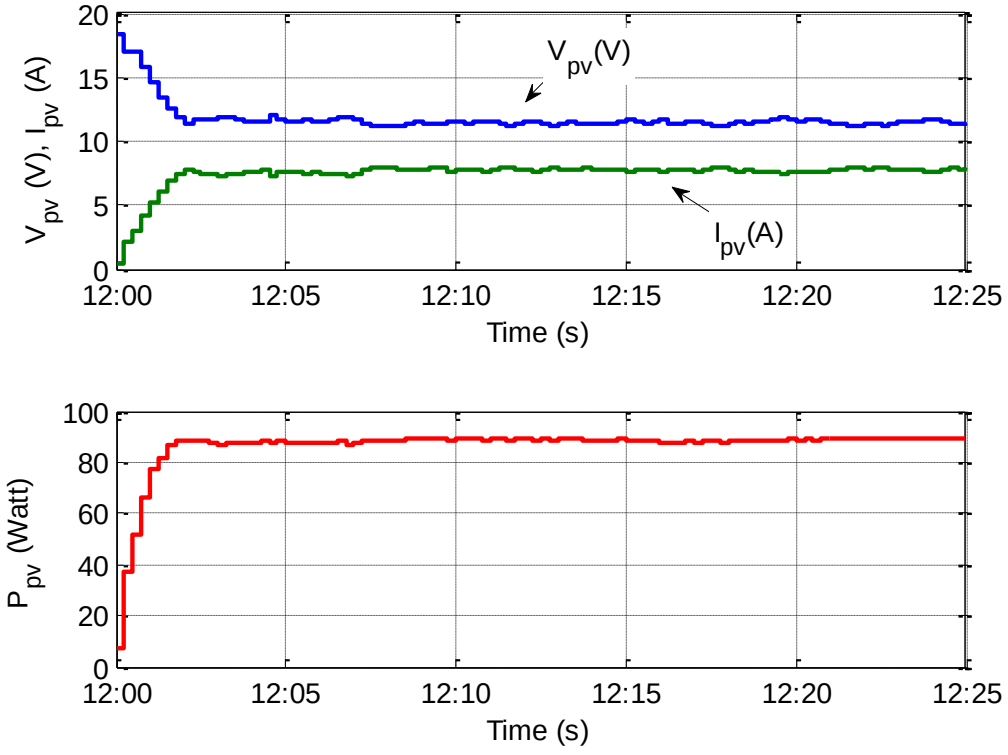


Figure 6.9: Testing results of FLC based P&O (Nov. 21, 2011, 12:00-12:25 pm)

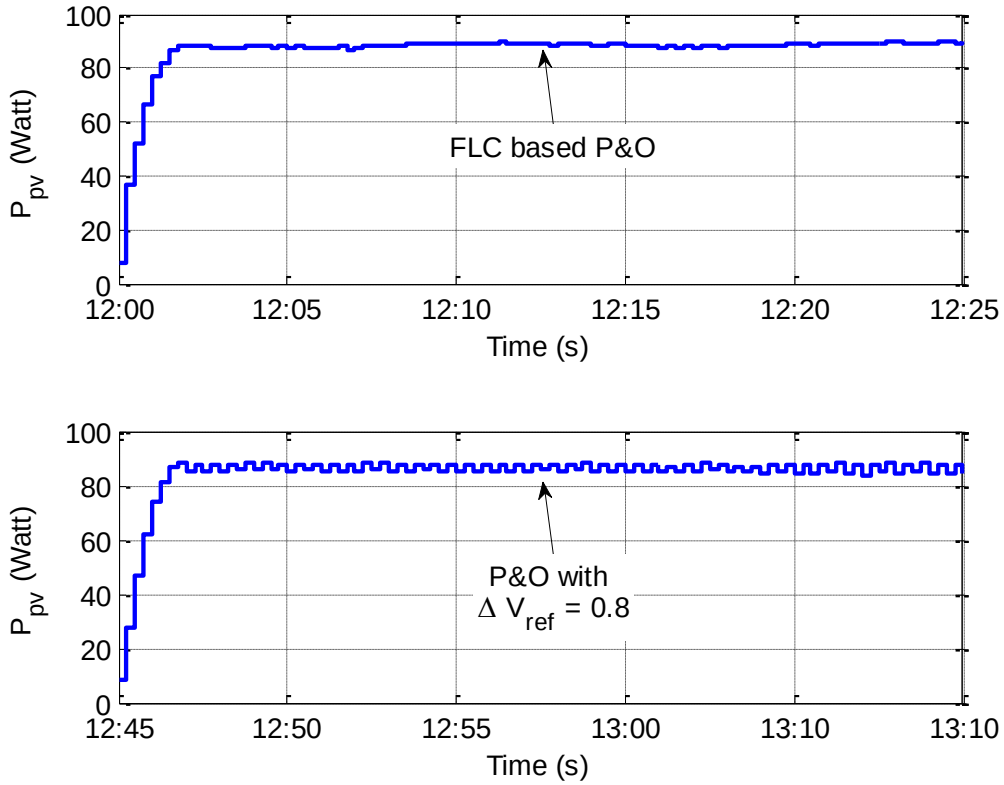


Figure 6.10: The two controllers implementation power output

Table 6-1 shows the yielded energy in kWh from the two tests. The energy is calculated using the following equation:

$$E(k) = \sum_{k=1}^{N} \left(P(k) T_r \right) = P_{Average} \times \left(N T_r \right) \tag{6.1}$$

where (NT_r) is the testing period.

Table 6-1: Energy yielded in kWh

Test Type	Average Power (kW)	E (kWh)
P&O with fixed ΔV_{ref}=0.8	0.08431	0.03513
FLC based P&O	0.08650	0.03604

The results show an enhancement of 2.59% in energy yielded by the proposed FLC based P&O controller compared to the conventional one when the panel is fixed.

6.2.4 Physical Tracker Implementation Results

Tests 1 and 3 compare the performance between the fixed panel and the dual axis tracking. Test 3 involved applying the physical tracking algorithm on the panel with the FLC based P&O. The tests were carried out in two consecutive days with

similar weather conditions. As shown in Figure 6.11, the tracking yielded higher power and the degradation of power within the one hour of testing was lower.

Moreover, Figure 6.11 shows the response of the MPPT proposed technique to varying weather conditions (clouds shading) as it was able to track the maximum power operating point as soon as the panel was un-shaded.

Using (6.1), Table 6-2 shows the yielded energy in kWh from test 3.

Table 6-2: Energy yielded in kWh

Test Type	Average Power (kW)	E (kWh)
FLC based P&O with sun tracking	0.07205	0.07205
FLC based P&O without sun tracking	0.06700	0.06700

The results show that introducing the sun tracking yielded an enhancement of 7.54% in energy.

One of the most important features of an efficient tracker is its low power consumption. Figure 6.12 shows the measured power consumption of the dual axis tracker for 1 hour period of tracking.

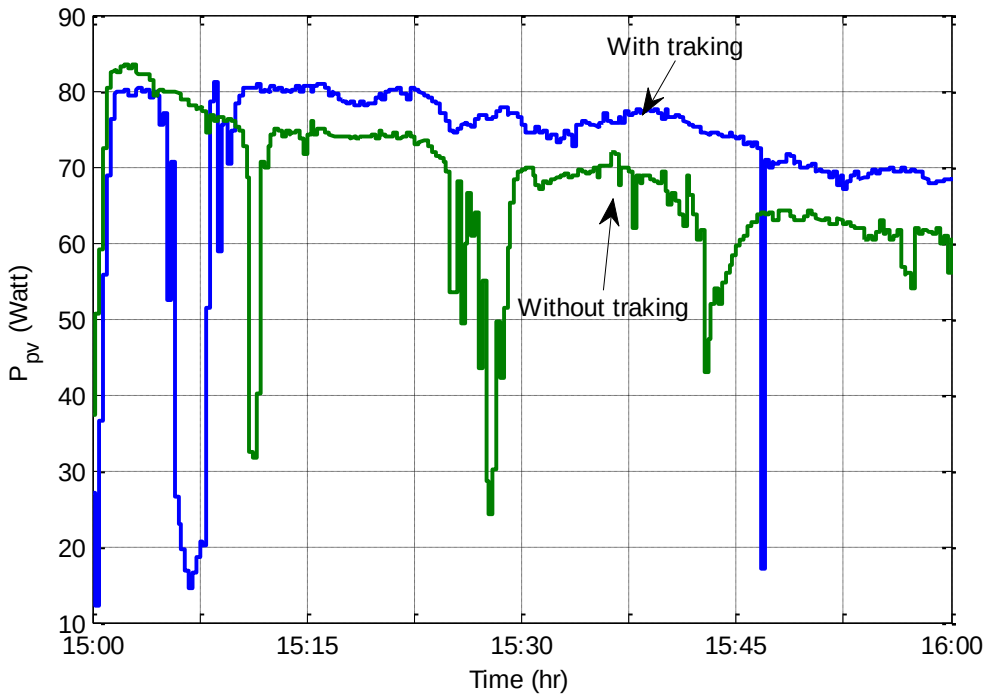


Figure 6.11: FLC based P&O implementation results with and without sun tracking (With tracking: Nov. 24, 2011, without tracking: Nov. 25, 2011, 15-16 pm)

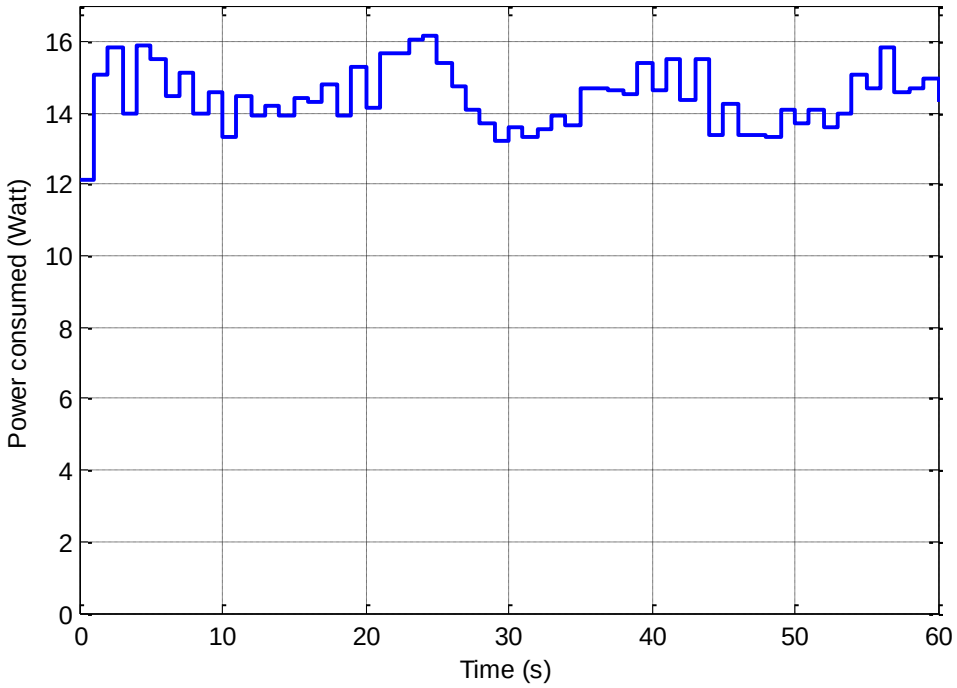


Figure 6.12: Tracker power consumption for 1 hour of testing

Using Figure 6.12 and equation (6.1) , the energy consumption of the tracker is 2.4701 Wh. The overall efficiency of the system with tracking is found to be 3.85% by subtracting the tracker consumed energy from the energy enhancement yielded by the tracker.

6.2.5 Full day testing results

In order to further verify the performance of the proposed controller, full day testing was performed.

In order to verify the FLC based system operation in steady state, the system was tested on Jan. 1 from 11:00 to 2:00 pm. The system started with the FLC based P&O MPPT and then was switched to the conventional P&O with ΔV_{ref}=0.8V after thirty minutes. The two controllers were alternated every thirty minutes for the rest of the testing period. Figure 6.13 shows the PV panel output power as well as the controller duty cycle and the insolation value throughout the testing period. In order to be able to compare between the two controllers, Figure 6.14 zooms at the period with a steady sun insolation (between 12:00 and 1:00 pm). Figure 6.14 shows that the power and duty cycle curves have more oscillations in steady state after 12:30 pm with the conventional P&O.

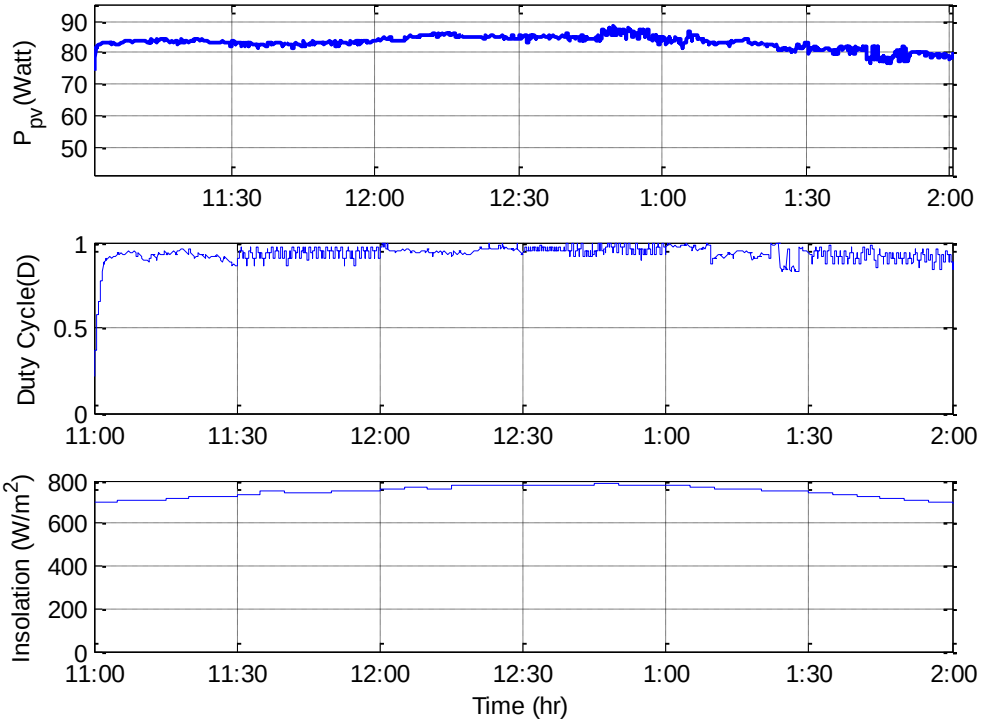


Figure 6.13: P&O full day implementation results (Jan. 1, 2012, 11 am- 2 pm)

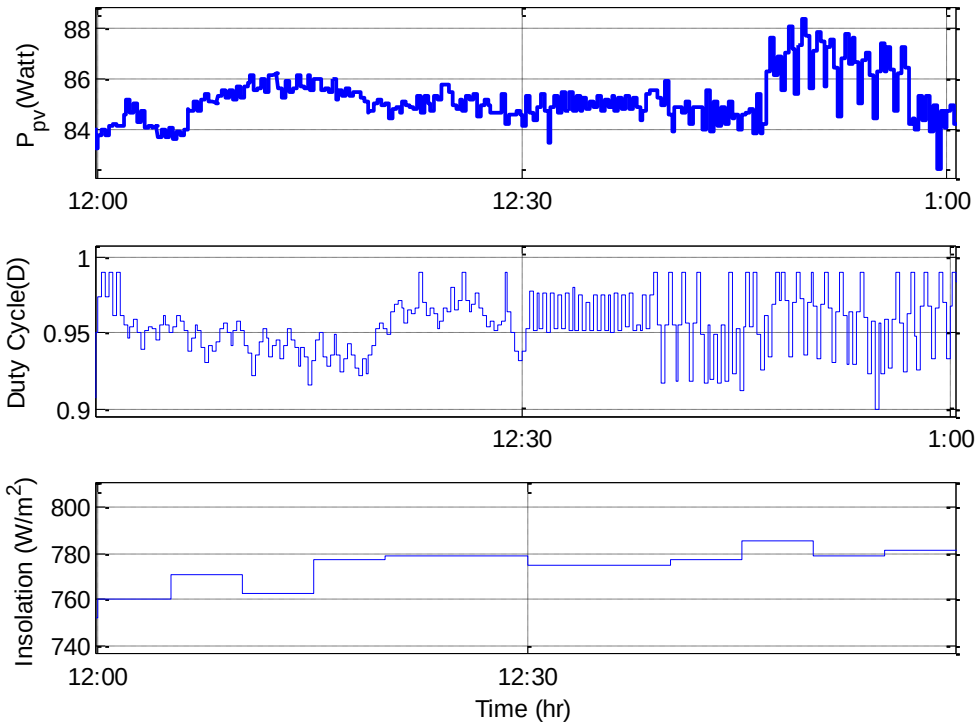


Figure 6.14: Comparison between FLC based P&O and conventional P&O

In order to verify the tracker operation, the system was tested from 10:00 am to 4:00 pm for two consecutive days. Although the testing was performed in two consecutive days, the weather conditions were different. Therefore, in order to be able to compare between the two set of results they were normalized.

As seen from Figure 6.15, with physical tracking, the system sustained the same power value the whole testing period. On the other hand, with fixed PV panel, the system power changed depending on the trajectory. The power curve starts at a low value and then starts increasing till it peaks around noon and then decreases after that. This set of results shows that sun tracking yields an enhancement of 3.83% in energy.

In order to check the system stability described in 5.3 the FLC-MPPT inputs and outputs are plotted for one hour of testing. As shown in Figure 6.16, the duty cycle reaches its maximum value at MPP and keeps operating around it while the FLC system output converges to zero. Moreover, Figure 6.16 shows the ability of the system to react to disturbances as it goes back to steady state once the disturbance is cleared.

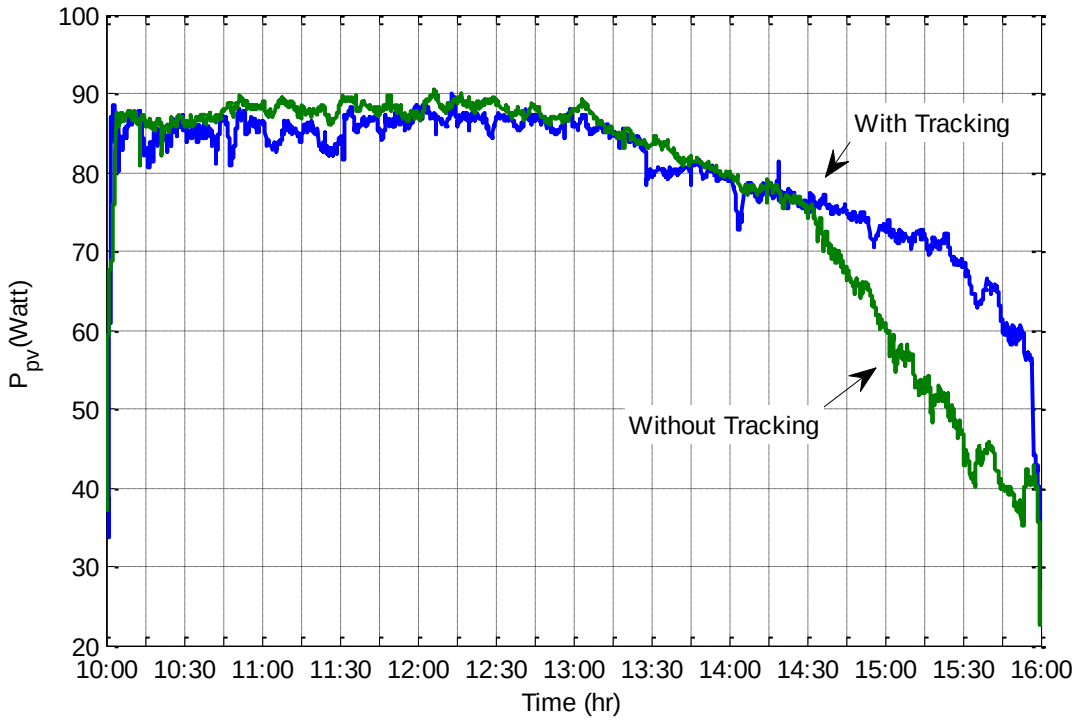


Figure 6.15: Normalized output power implementation results with and without sun tracking (With tracking: Dec. 20, 2011, 10- 4 pm, without tracking: Dec. 21, 2011, 10- 4 pm)

Plotting the absolute value of the FLC output as a function of the two FLC inputs in a 3D plot clearly shows that the system output converges to zero in steady state as shown in Figure 6.17.

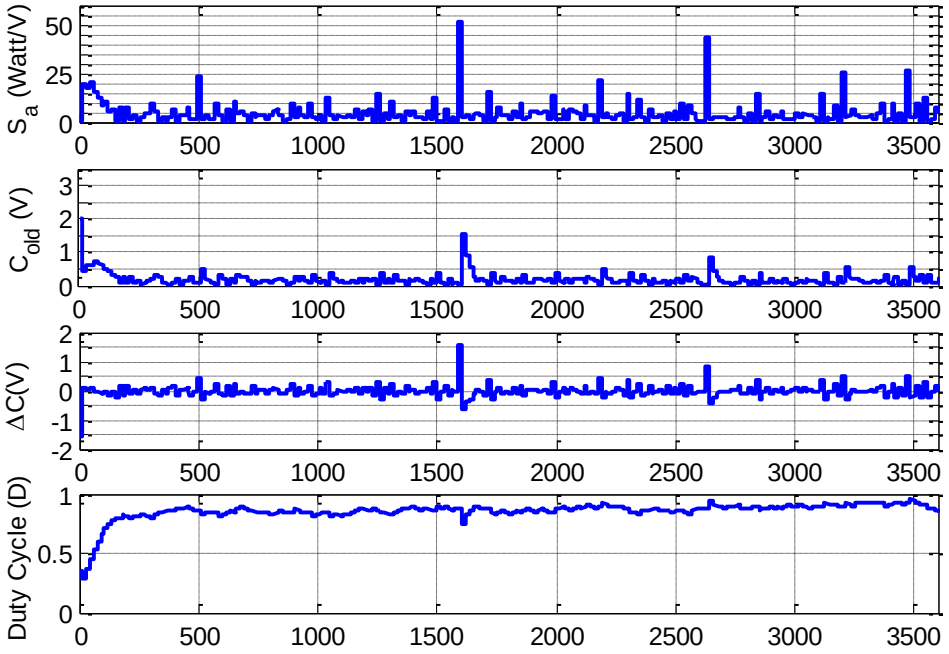


Figure 6.16: FLC inputs (S_a, C_{old}) and output (ΔC) for a clear day

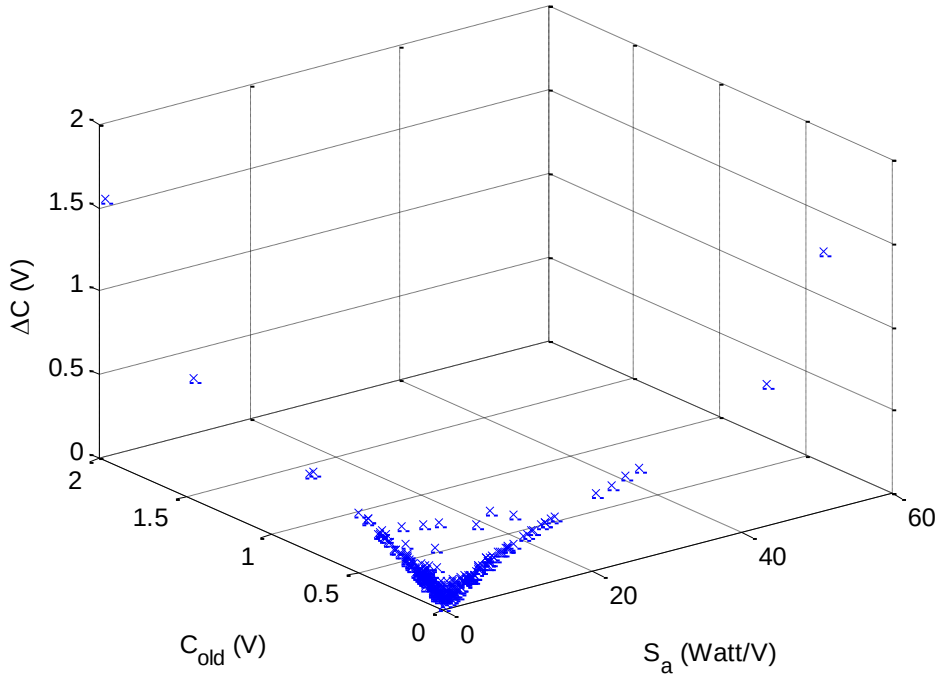


Figure 6.17: FLC absolute value of the output as a function of the two inputs for a clear day

Figure 6.18 shows the system inputs and outputs for another testing day with cloudy weather conditions. Plotting the FLC absolute value of the output as a function of its two inputs also shows that the FLC output converges to zero at steady state as shown in Figure 6.19.

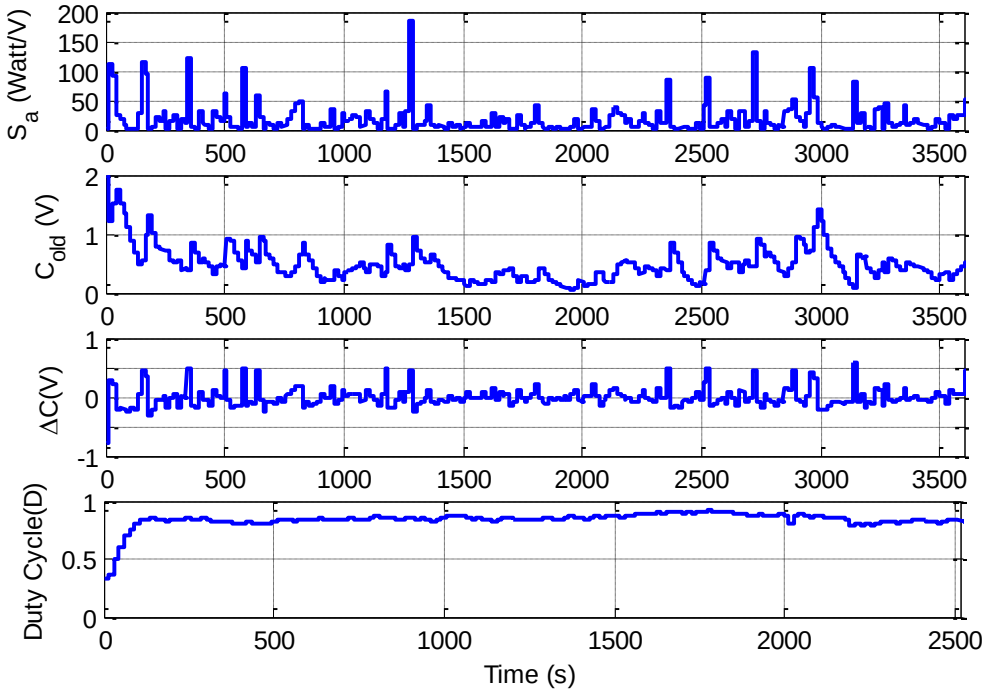


Figure 6.18: FLC inputs (S_a, C_{old}) and output (ΔC) for a clear day

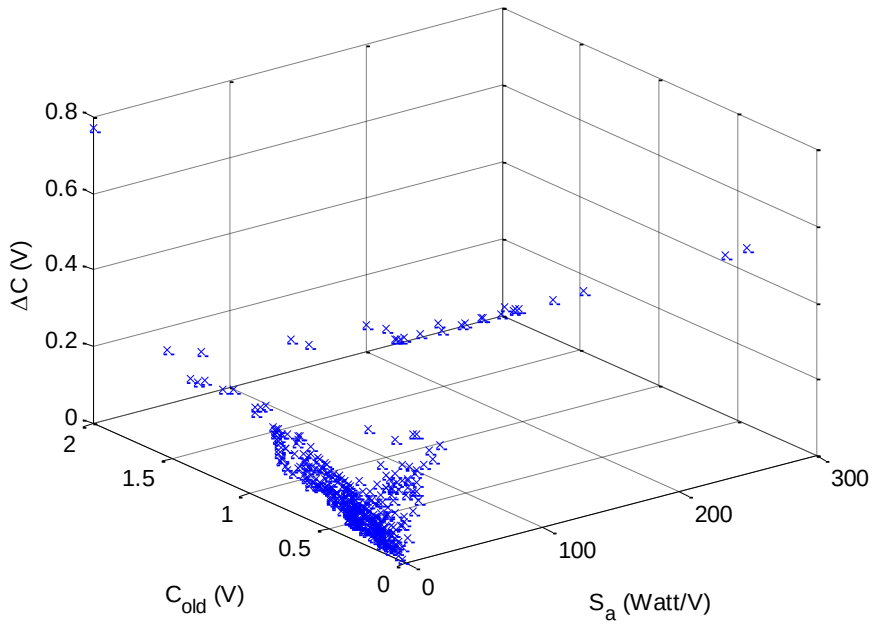


Figure 6.19: FLC absolute value of the output as a function of the two inputs for a cloudy day

CHAPTER 7 : CONCLUSION

In this research, a new dual-MPPT scheme based on fuzzy logic control (FLC) has been introduced. The first MPPT controller is an astronomical two-axis sun tracker that keeps maximum radiation on the panel throughout the day. The second controller implements a new fuzzy based MPPT technique to adaptively change the P&O perturbation step size. The proposed control scheme takes the power slope of the PV panel curve and the old voltage perturbation step as its inputs and outputs the change in the new P&O step size. The FLC output is gradually updated in order to avoid exceeding the MPP in the opposite direction leading to oscillations. The small signal analysis model of the power converter is presented along with a Lyapunov based stability analysis of the PV system. The controller was implemented on a 150-watt prototype PV system. Simulation and experimental results show clearly the advantages of the proposed technique compared to the conventional P&O. The proposed technique achieves stable operation in the entire region of the PV panel by eliminating the resulting oscillations around the maximum power operating point and enhancing the operating point convergence speed.

One of the limitations that we faced in this research work is the restriction on the outdoor testing time, as it is limited to sunshine hours. Another limitation is the choice of the testing location, as the testing requires an un-shaded fixed position that is directly exposed to sunlight throughout the day.

Further enhancement of this system can be achieved by combining the physical tracking and the MPPT controller into one DSP and making the system standalone by charging the batteries that are supplying the system.

The following is a summary of the complete work performed in this thesis:

- Conducted a thorough literature review on the different methods of MPPT tracking techniques for PV systems.
- Designed and simulated the FLC based P&O-MPPT controller using MATLAB/Simulink.
- Designed and built an astronomical dual-axes tracker.
- Verified the simulated results using a 150-watt PV system prototype.
- Implemented the FLC based P&O-MPPT controller using the TMS320F28335 DSP.
- Presented the small signal analysis model of the power converter.
- Proved the stability of the overall system using the Lyapunov stability analysis.

REFERENCES

[1] Fairley and P., "Fukushima's positive impact [Spectral Lines]," *Spectrum, IEEE*, vol. 48, no. 5, p. 8, May 2011.

[2] G. Spagnuolo et al., "Renewable energy operation and conversion schemes," *IEEE Ind. Electron. Mag.* , vol. 4, no. 1, pp. 38-51, Mar. 2010.

[3] D. Sonnenenergie, "Photovoltaic Basics," in *Planning and installing photovoltaic systems: A guide for installers, architects, and engineers*, 2nd ed., Deutsche Gesellschaft für Sonnenenergie, Ed. London: Earthscan Publications Ltd., 2008, ch. 1, pp. 1-62.

[4] B. Bose, "Global warming: energy, environmental pollution, and the impact of power electronics," *IEEE Ind. Electron. Mag.*, vol. 4, no. 1, pp. 6-17, Mar. 2010.

[5] A. Mellita and S. Kalogirou, "Artificial intelligence techniques for photovoltaic applications: A review," *Progress in Energy and Combustion Science*, vol. 34, pp. 574-632, 2008.

[6] K.-H. Chao and C.-J. Li, "An intelligent maximum power point tracking method based on extension theory for PV systems," *Expert Systems with Applications*, vol. 37, pp. 1050-1055, 2010.

[7] E. Aranda, J.Gala´ N, M. Cardona, and J.Rquez, "Measuring the I-V curve of PV generators," *IEEE Ind. Electron. Mag.*, vol. 3, no. 3, pp. 4-14, September 2009.

[8] R. Coelho, F. Concer, and D. Martins, "Analytical and experimental analysis of DC-DC converters in photovoltaic maximum power point tracking applications," in *Proc. IECON 2010*, Phoenix, 2010, pp. 2778 – 2783.

[9] W. Xiao, W. Dunford, P. Palmer, and A. Capel, "Regulation of photovoltaic voltage," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1365-1374, Jun. 2007.

[10] C. Hua, J. Lin, and C. Shen, "Implementation of a DSP-controlled photovoltaic system with peak power tracking," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 99-107, Feb. 1998.

[11] T. Patrick and P. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Trans. Energy Conv.*, vol. 22, no. 2, pp. 439-449, Jun. 2007.

[12] D. Hohm and M. Ropp, "Comparative study of maximum power point tracking algorithms," *Progress In Photovoltaics: Research And Applications*, vol. 11, pp. 47-62, 2003.

[13] V. Salas, E. Olias, A. Barrado, and A. Lazaro, "Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems," *Solar Energy Materials & Solar Cells*, vol. 90, no. 11, pp. 1555-1578, Jan. 2006.

[14] I. Houssamo, F. Locment, and M. Sechilariu, "Maximum power tracking for photovoltaic power system: Development and experimental comparison of two algorithms," *Renewable Energy*, vol. 35, pp. 2381-2387, 2010.

[15] J. Jiang, T. Huang, Y.Hsiao, and C.Chen, "Maximum power tracking for photovoltaic power systems," *Tamkang Journal of Science and Engineering*, vol. 8, no. 2, pp. 147-153, Mar. 2005.

[16] N. Femia, D. Granozio, G. Petrone, G. Spagnuolo, and M. Vitelli, "Predictive & adaptive MPPT perturb and observe method," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 934–950, Jul. 2007.

[17] C. Zhang, D. Zhao, J. Wang, and G. Chen, "A modified MPPT method with variable perturbation step for photovoltaic system," in *Proc. IPEMC '09*, Wuhan, 2009, pp. 2096-2099.

[18] C. Ahn, J. Choi, and D. Lee, "Adaptive maximum power point tracking algorithm for photovolatic power systems," *IEICE Trans. Commun.*, vol. E93-B, no. 5, pp. 1334-1337, May 2010.

[19] W. Xiao and W. Dunford, "A modified adaptive hill climbing MPPT method for photovoltaic power systems," in *Proc. IEEE PESC*, Aachen, 2004, pp. 1957–1963.

[20] R-J Wai, W-H Wang, and J-Y Lin, "Grid-connected photovoltaic generation system with adaptive step-perturbation method and active sun tracking scheme," *IEEE Trans. Ind. Electron.*, pp. 224 – 228, Nov. 2006.

[21] A Safari and S. Mekhilef, "Simulation and hardware implementation of incremental conductance MPPT with direct control method using cuk converter," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1154-1161, Apr. 2011.

[22] Y-C. Kuo, T-J. Liang, and J-F. Chen, "Novel maximum-power-point-tracking controller for photovoltaic energy conversion system," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 594-601, Jun. 2001.

[23] F. Liu, S. Duan, F. Liu, B. Liu, and Y. Kang, "A variable step size INC MPPT method for PV systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2622-2628, Jul. 2008.

[24] Q. Mai, M. Shan, L. Liu, and J. Guerrero, "A novel improved variable step-size incremental-resistance MPPT method for PV systems," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2427-2434, Jun. 2011.

[25] S. Moura and Y. Chang, "Asymptotic convergence through lyapunov-based switching in extremum seeking with application to photovoltaic systems," in *Proc. ACC*, Baltimore, 2010, pp. 3542-3548.

[26] R. Leyva et al., "MPPT of photovoltaic systems using extremum–seeking control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 249-258, Jan. 2006.

[27] Steven LS. Brunton, C. Rowley, S. Kulkarni, and C. Clarkson, "Maximum power point tracking for photovoltaic optimization using extremum seeking," in *Proc. PVSC*, Philadelphia, 2009, pp. 13-16.

[28] E. Iyasere, D. Dawson, and E. Tatlicioglu, "Backstepping PWM control for maximum power tracking in photovoltaic array Systems," in *Proc. ACC*, Baltimore, 2010, pp. 3561-3565.

[29] G. Petrone, G. Spagnuolo, R. Teodorescu, M. Veerachary, and M. Vitelli, "Reliability issues in photovoltaic power processing systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2569-2580, Jul. 2008.

[30] G. Yu, Y. Jung, J. Choi, and G. Kim, "A novel two-mode MPPT control algorithm based on comparative study of existing algorithms," *Solar Energy*, vol. 76, no. 4, pp. 455-463, 2004.

[31] N. Yadaiah and M. Veerachary, "Adaptive controller for peak power tracking of photovoltaic systems," *System Analysis Modelling Simulation*, vol. 42, no. 9, pp. 1319-1334, 2002.

[32] M. Veerachary, T. Senjyu, and K. Uezato, "Neural-network-based maximum power point tracking of coupled inductor interleaved-boost converter supplied PV system using fuzzy controller," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 749-758, Aug. 2003.

[33] C. Otieno, G. Nyakoe, and C. Wekesa, "A neural fuzzy based maximum power point tracker for a photovoltaic system," in *Proc. IEEE AFRICON 2009*, Nairobi, 2009, pp. 1-6.

[34] C. Salah and M. Ouali, "Comparison of fuzzy logic and neural network in maximum power point tracker for PV systems," *Electric Power Systems Research*, vol. 81, pp. 43-50, 2011.

[35] A. Nafeha, F. Fahmya, and E. El-Zahabb, "Evaluation of a proper controller performance for maximum-power point tracking of a stand-alone PV system," *Solar Energy Materials & Solar Cells*, vol. 75, pp. 723-728, 2003.

[36] S. Lalouni, D. Rekioua, T. Rekioua, and E. Matagne, "Fuzzy logic control of stand-alone photovoltaic system with battery storage," *Journal of Power Sources*, vol. 193, no. 5, pp. 899-907, Sep. 2009.

[37] C-S Chiu, "T-S fuzzy maximum power point tracking control of solar power generation systems," *IEEE Trans. Energy Conv.*, vol. 25, no. 4, pp. 1123-1131, Dec. 2010.

[38] N. D'Souza, L. Lopes, and X. Liu, "An intelligent maximum power point tracker using peak current control," in *Proc. PESC-05*, Recife, 2005, pp. 172 - 177.

[39] U. Yaragatti, A. Rajkiran, and B. Shreesha, "A novel method of fuzzy controlled maximum power point tracking in photovoltaic systems," in *Proc. ICIT 2005*, 2005, pp. 1421-1426.

[40] G. Zeng and Q. Liu, "An intelligent fuzzy method for MPPT of photovoltaic arrays computational intelligence and design," in *Proc. ISCID '09*, 2009, pp. 1421-1426.

[41] X. Li, K. Dong, and H. Wu, "Study on the intelligent fuzzy control method for MPPT in Photovoltaic Voltage Grid System," in *Proc. ICIEA 2008*, Singapore, 2008, pp. 708 - 711.

[42] N. Patcharaprakiti, S. Premrudeepreechacharn, and Y. Sriuthaisiriwong, "Maximum power point tracking using adaptive fuzzy logic control for grid-connected photovoltaic system," *Renewable Energy* , vol. 30, no. 11, pp. 1771-1788, 2005.

[43] C. Larbes, S. Cheikh, T. Obeidi, and A. Zerguerras, "Genetic algorithms optimized fuzzy logic control for the maximum power point tracking in photovoltaic," *Renewable Energy* , vol. 34, pp. 2093-2100, Oct. 2009.

[44] Z-J. Yang, Y-R. Chen, and J-M. Hu, "The research on application of variable universe fuzzy control to maximum power point tracking system," in *Proc. PESA2009*, 2009, pp. 1-4.

[45] C. Fengwen and W.Yiwang, "Variable universe adaptive fuzzy-PI compound control applied in maximum power point tracking for photovoltaic energy generation system," in *Proc. ICIC 2010*, 2010, pp. 514-519.

[46] N. Khaehintung and P. Sirisuk, "Application of maximum power point tracker with self-organizing fuzzy logic controller for solar-powered traffic lights," in *PEDS2007*, 2007, pp. 642-646.

[47] H. Mousazadeh, A. Keyhani, A. Javadi, and H. Mobli, "A review of principle and sun-tracking methods for maximizing solar systems output ," *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 1800-1818, 2009.

[48] L. Charles. (2002, Dec.) Optimum orientation of solar panels. [Online]. HYPERLINK "http://www.macslab.com/optsolar.html" http://www.macslab.com/optsolar.html

[49] M. Clifford and D. Eastwood, "Design of a novel passive solar tracker," *Solar Energy*, vol. 77, pp. 269–80., 2004.

[50] R-J. Wai, W-H. Wang, and C-Y. Lin, "High-performance stand-alone photovoltaic generation system," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 240-250, Jan. 2008.

[51] C. Chin, Babu A, and W. McBride, "Design, modeling and testing of a standalone single axis solar tracker using MATLAB/Simulnik," *Renewable Energy*, vol. 36, pp. 3075-3090, 2011.

[52] K. Karimov et al., "A simple photo-voltaic tracking system," *Solar Energy Materials & Solar Cells* , vol. 87, pp. 49-59, 2005.

[53] O. Bingol, A. ALTINTAS, and Y. O¨NER, "Microcontroller based solar-tracking system and its implementation," *Journal of Engineering Sciences*, vol. 12, no. 2, pp. 243-248, Jun. 2006.

[54] S. Seme, G. Štumberger, and J. Voršič, "Maximum efficiency trajectories of a two-axis sun tracking system determined considering tracking system consumption," Power Electronics," *IEEE Trans. Power Electron.*, vol. 26, no. 4, pp. 1280-1290, Apr. 2011.

[55] S. Abdallah and O. Badran, "Sun tracking system for productivity enhancement of solar still," *Desalination* , vol. 220, pp. 669-676, 2008.

[56] S. Abdallah and S. Nijmeh, "Two axes sun tracking system with PLC control," *Conversion and Management* , vol. 45, pp. 1931-1939, 2004.

[57] S. Armstrong and W. Hurley, "Investigating the effectiveness of maximum power point tracking for a solar system," in *Proc. PESC'05*, Recife, 2005, pp. 204-209.

[58] N. Nijegorodov, K. Devan, P. Jain, "Atmospheric transmittance models and an analytical method to predict the optimum slope on an absorber plate, variously orientated at any latitude." *Renewable Energy*, vol 4, pp 529-543,1997

[69] A. Jain. (2011, Nov.) Solar Concepts. [Online]. HYPERLINK "http://www.usc.edu/dept/architecture/mbs/tools/vrsolar/Help/solar_concepts.html."
http://www.usc.edu/dept/architecture/mbs/tools/vrsolar/Help/solar_concepts.html.

[60] G. Walker, "Evaluating MPPT converter topologies using a MATLAB PV model," *J. Electr. Electron. Eng.*, vol. 21, no. 1, pp. 49–56, 2001.

[61] I.-S. Kim, M.-B. Kim, and M.-J. Youn, "New maximum power point tracker using sliding-mode observer for estimation of solar array current in the grid-connected photovoltaic system," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1027-1035, Aug. 2006.

[62] W. Xiao, M. Lind, W. Dunford, and A. Capel, "Real-time identification of optimal operating points in photovoltaic power systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1017-1026, Aug. 2006.

[63] Isofoton Web site. [Online]. HYPERLINK "http://www.isofoton.com/technical/material/pdf/productos/fotovoltaica/modulos/F_T_modulo_fotovoltaico_I150S_12_ing.pdf" http://www.isofoton.com/technical/material/pdf/productos/fotovoltaica/modulos/F_T_modulo_fotovoltaico_I150S_12_ing.pdf

[64] M. Rashid, *Power Electronics: Circuits, Devices and Applications*, 3rd ed. New Jersey, USA: Pearson/Prentice Hall, 2004.

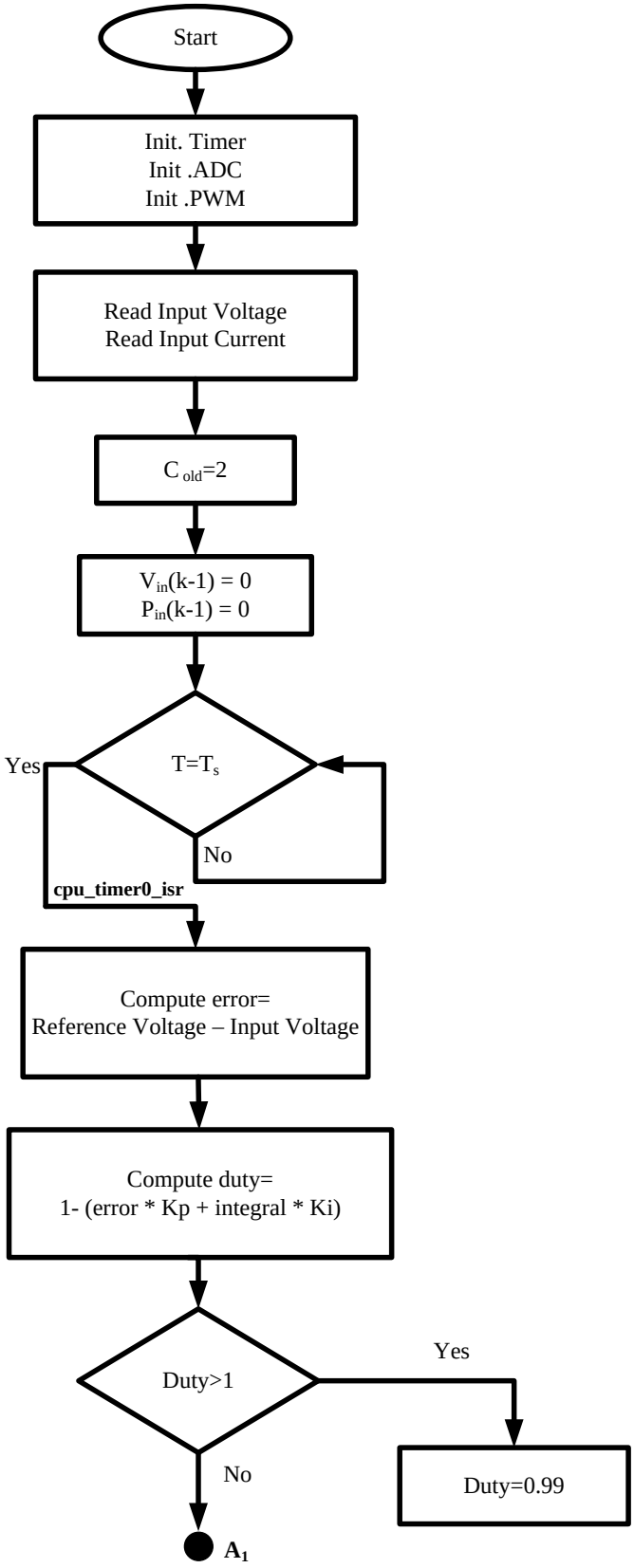
[65] R. Erickson and D. Maksimović, *Fundamental of Power Electronics*, 2nd ed. Norwell, MA: Kluwer Academic Publishers, 2001.

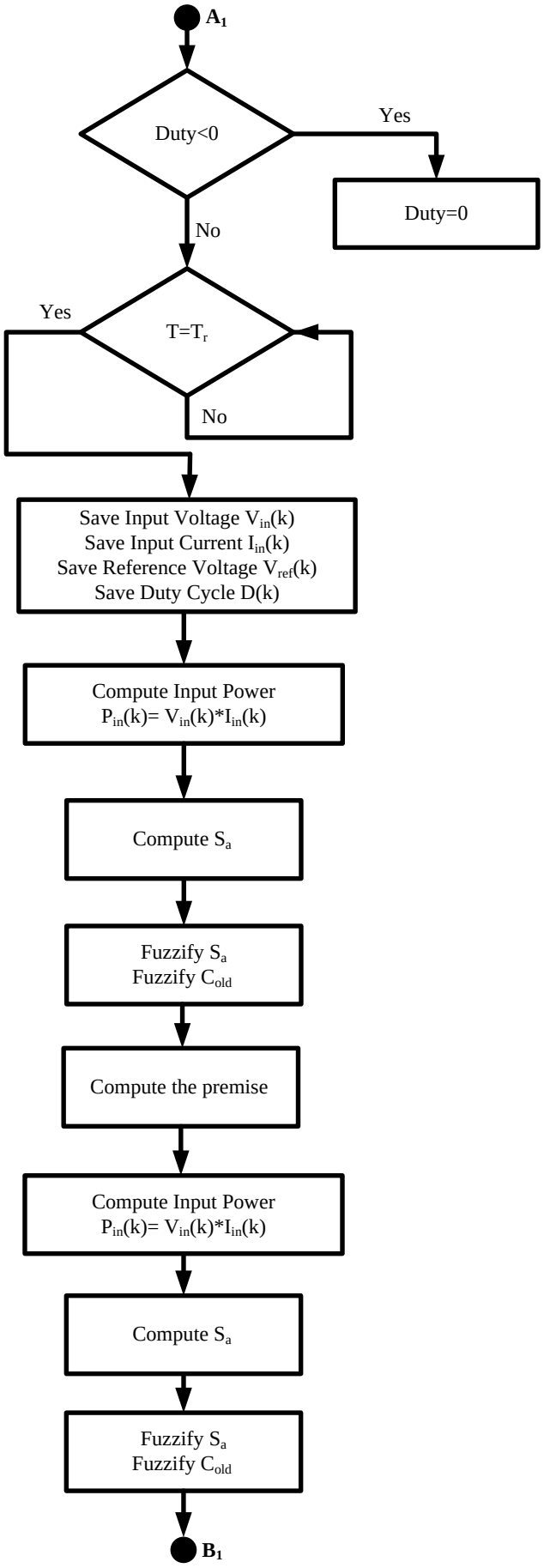
[66] L. Zadeh, "Fuzzy sets," *Inform. Control.*, vol. 8, pp. 338-353, 1965.

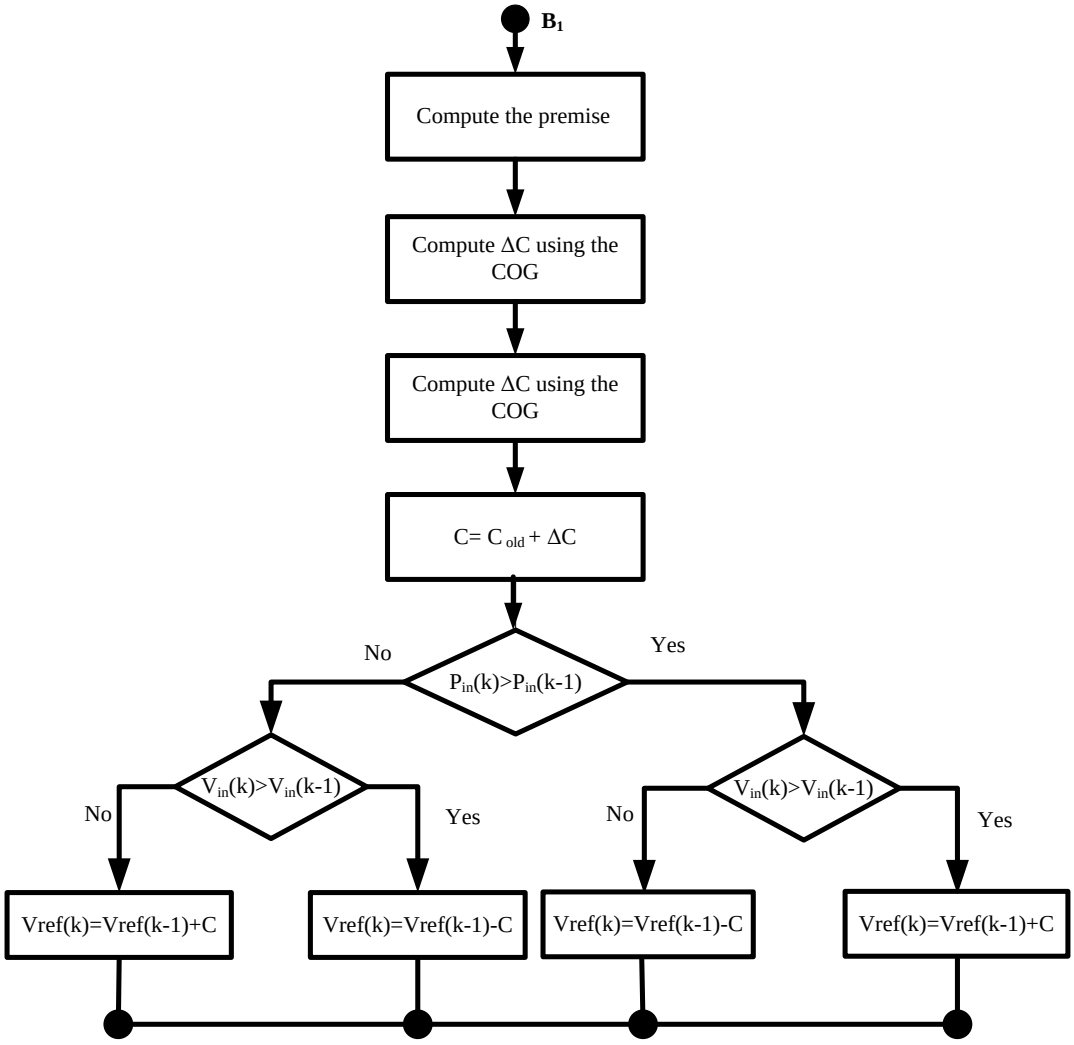
[67] K. Passino and S. Yurkovich, Fuzzy Control. CA, USA: Addison Wesley Longman, Inc., 1998.

APPENDIX A: CODES FLOW CHARTS

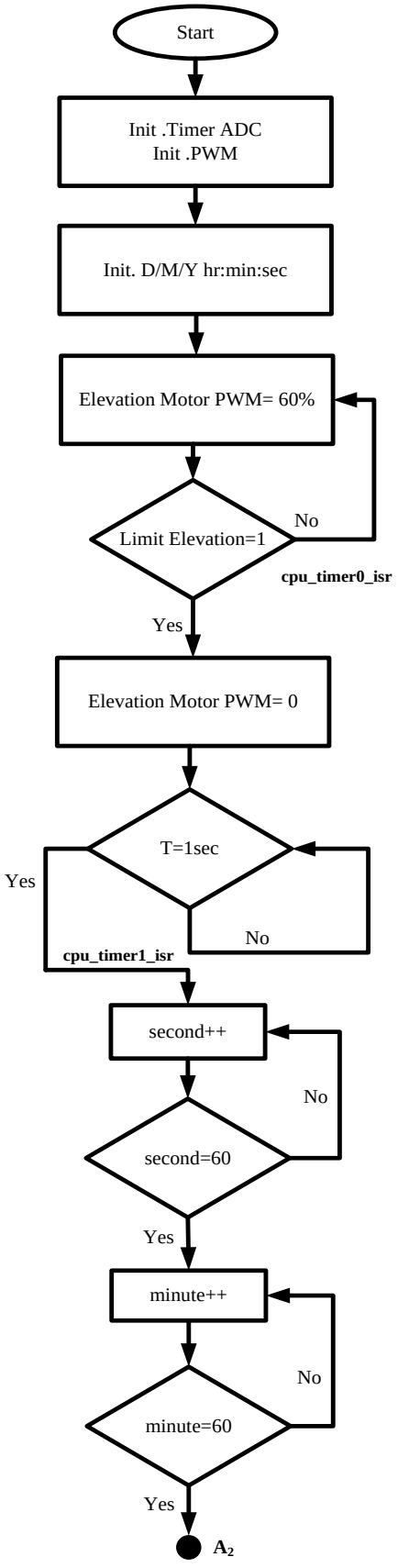
FLC Based P&O Code Flowchart:

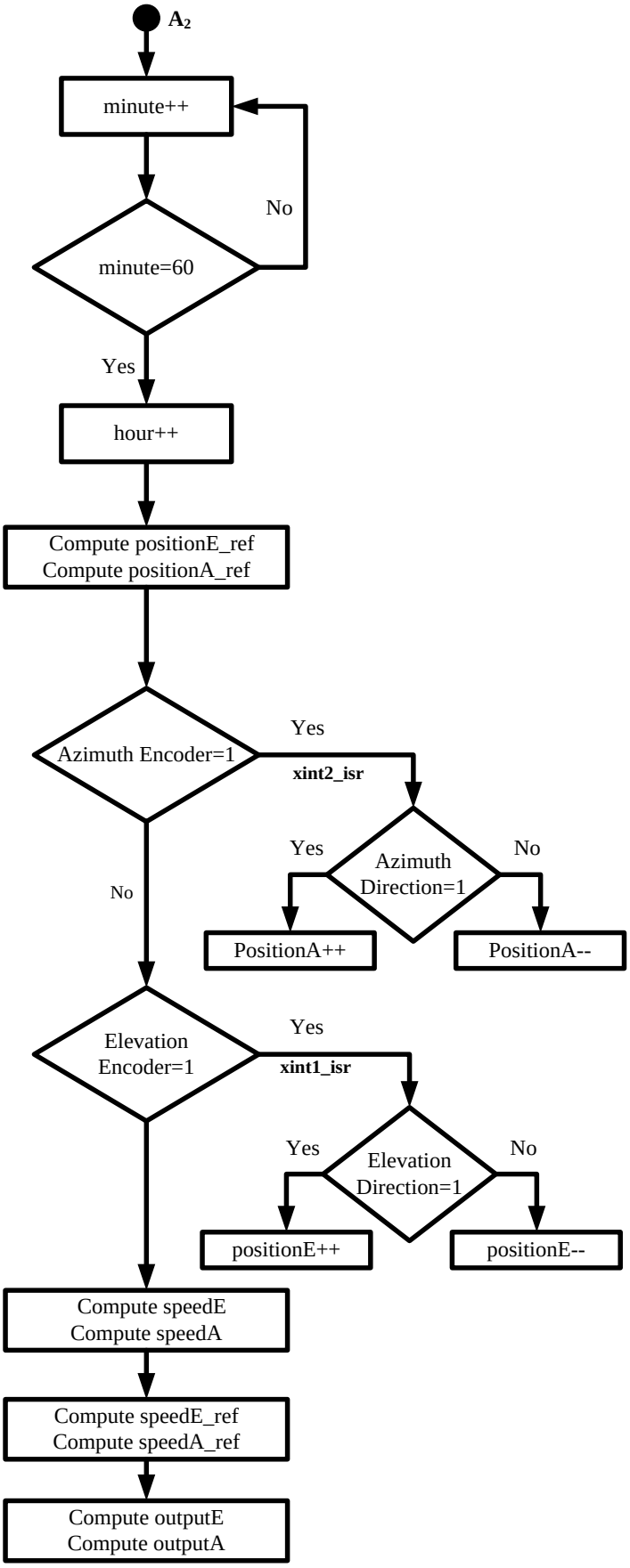



A1
Duty<0
Yes
Duty=0
No
Yes
T=Tr
No
Save Input Voltage Vin(k)
Save Input Current Iin(k)
Save Reference Voltage Vref(k)
Save Duty Cycle D(k)
Compute Input Power
Pin(k)= Vin(k)*Iin(k)
Compute Sa
Fuzzify Sa
Fuzzify Cold
Compute the premise
Compute Input Power
Pin(k)= Vin(k)*Iin(k)
Compute Sa
Fuzzify Sa
Fuzzify Cold
B1


B1
Compute the premise
Compute ΔC using the COG
Compute ΔC using the COG
C= C old + ΔC
No
Yes
Pin(k)>Pin(k-1)
Vin(k)>Vin(k-1)
No
Yes
Vin(k)>Vin(k-1)
No
Yes
Vref(k)=Vref(k-1)+C
Vref(k)=Vref(k-1)-C
Vref(k)=Vref(k-1)-C
Vref(k)=Vref(k-1)+C

Physical Tracker Code Flowchart:




A2
minute++
No
minute=60
Yes
hour++
Compute positionE_ref
Compute positionA_ref
Azimuth Encoder=1
Yes
xint2_isr
Yes
Azimuth
Direction=1
No
No
PositionA++
PositionA--
Elevation
Encoder=1
Yes
xint1_isr
Yes
Elevation
Direction=1
No
positionE++
positionE--
Compute speedE
Compute speedA
Compute speedE_ref
Compute speedA_ref
Compute outputE
Compute outputA

APPENDIX B: CODES

```
//////////////////////////////////////////////////////////////////////////////////////////////
//////
////                                                                                      ////
////                                    MPPT CODE                                         ////
////                                                                                      ////
//////////////////////////////////////////////////////////////////////////////////////////////

#include "DSP2833x_Device.h"     // DSP2833x Headerfile Include File
#include "DSP2833x_Examples.h"   // DSP2833x Examples Include File
#include <math.h>

// ADC start parameters
#if (CPU_FRQ_150MHZ)     // Default - 150 MHz SYSCLKOUT
 #define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3)   = 25.0 MHz
#endif
#if (CPU_FRQ_100MHZ)
 #define ADC_MODCLK 0x2 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 100/(2*2)   = 25.0 MHz
#endif
#define ADC_CKPS   0x1   // ADC module clock = HSPCLK/2*ADC_CKPS   = 25.0MHz/(1*2) = 12.5MHz
#define ADC_SHCLK  0x0   // S/H width in ADC module periods                 = 16 ADC clocks
#define AVG       1000  // Average sample limit
#define ZOFFSET    0x00  // Average Zero offset
#define BUF_SIZE   1000  // Sample buffer size
// Prototype statements for functions found within this file.
interrupt void cpu_timer0_isr(void);
interrupt void cpu_timer2_isr(void);
// Global variables for this example
Uint32 l,p,j,i,k;
Uint32 Duty;
Uint32 Freq;
float Voltage_in, Current_in, Current1_in;
Uint32 Voltage_ADC_in;
Uint32 Current_ADC_in;
float Presult_in=0;
float Pold=0, Duty1;
int nume=3, numc=3;
float g1=0.02,g2=0.5;
float ce[3]={0 ,(0.5/0.02),(1/0.02)};
float cc[3]={0 ,(0.5/0.5),(1/0.5)};
float rules [3][3]={{0,-(0.5),-(1)},
                    {(0.5), 0,-(0.5)},
                    {(1),(0.1),0}};
float prem,we,wc,base;
float mfe[3]={0,0,0},mfc[3]={0,0,0};
float c_count=0,e_count=0,e_int=0,c_int;
float e=0,c=2,  num=0,den=0;
float output=0;
float Vresultin[1690], Iresultin[1690], Vrefr[1690], estore [361], dstore[361], outstore[361];
float integral = 0,Kp=0.01,Ki=0.05,Vref=18,error=0,z1[2];
#define MAX(A,B)  ((A) > (B) ? (A) : (B))
#define MIN(A,B)  ((A) < (B) ? (A) : (B))
float AFF(float A)
{float B;
if (A<0)
B=-A;
else
B=A;
return (B);
}
```

```
void main(void)
{
  InitSysCtrl();
// Specific clock setting for this example:
        EALLOW;
        SysCtrlRegs.HISPCP.all = ADC_MODCLK;
        EDIS;
// Disable CPU interrupts and clear all CPU interrupt flags:
        DINT;
        InitPieCtrl();
        IER = 0x0000;
        IFR = 0x0000;
//PIE Init.
        InitPieVectTable();
        EALLOW;
        PieVectTable.TINT0 = &cpu_timer0_isr;          // interrupts that are used in this example are
// re-mapped to ISR (Interrupt Sub Routine)
// functions found within this file
        EDIS;
//Timer Init.
        InitCpuTimers();
        ConfigCpuTimer(&CpuTimer0, 150, 50000);
//ADC Init.
        InitAdc();
// Specific ADC setup for this example:
        AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;
        AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS;
        AdcRegs.ADCTRL1.bit.SEQ_CASC = 1;     // 1  Cascaded mode
        AdcRegs.ADCMAXCONV.all = 0x0008;     // Setup 2 conv's on SEQ1
        AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2;    // Use ADCINA3 Channel iin
        AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3;    // Use ADCINA4 Channel vin
        AdcRegs.ADCTRL1.bit.CONT_RUN = 1;     // Setup continuous run
// ePWM Configuration
        EALLOW;
        GpioCtrlRegs.GPAPUD.bit.GPIO6=0;  // enable pull-ups on GPIO6
        GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 1;  // configure GPIO6 as ePWM4A
        GpioCtrlRegs.GPAPUD.bit.GPIO7 = 0;  // enable pull-ups on GPIO1
        GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 1;  // configure GPIO4 as ePWM4B
        EDIS;
        Freq = 3750;                                                    //Frequency = 5KHz
        Duty1 = 0.9;
        Duty = Freq * Duty1;
        EPwm4Regs.TBPRD = Freq;
        EPwm4Regs.CMPA.half.CMPA = Duty;                          //Duty Cycle = 50%
        EPwm4Regs.CMPB = Duty;
        EPwm4Regs.CMPCTL.bit.SHDWAMODE = 1;  // immediate mode
        EPwm4Regs.TBCTL.bit.CTRMODE = 2;  // up-down-count mode
        EPwm4Regs.TBCTL.bit.PRDLD = 0;  // load TBPRD from its shadow register
        EPwm4Regs.TBCTL.bit.SYNCOSEL = 3;  // disable synchronization output select
        EPwm4Regs.TBCTL.bit.CLKDIV = 1;  // do not pre-scale SYSCLKOUT
        EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0;  // do not pre-scale SYSCLKOUT
        EPwm4Regs.AQCTLA.bit.CAU = 1;  // set ePWM3A when counter equals CMPA & counter is
increasing
        EPwm4Regs.AQCTLA.bit.CAD = 2;  // clear ePWM3A when counter equals CMPA & counter i
        EPwm4Regs.AQCTLB.bit.CAU = 1;  // set ePWM3A when counter equals CMPA & counter is
increasing
        EPwm4Regs.AQCTLB.bit.CAD = 2;  // clear ePWM3A when counter equals CMPA & counter i
//Clear Variables
        for(j=0 ; j<1690 ; j++)
        {
                Vresultin[j]=0;
                Iresultin[j]=0;
                Vrefr[j]=0;}
        i = 0;j = 0;k = 0;p=0;l=0;
```

```
EPwm4Regs.TBCTR = 0;
//Start Timer
        CpuTimer0Regs.TCR.bit.TSS = 0;                  //start timer0#
//Enable Interrupts
        IER |= M_INT1;  // enable CPU INT1 which is connected to CPU-Timer 0:
        PieCtrlRegs.PIEIER1.bit.INTx7 = 1;  // enable TINT0 in the PIE: Group 1 interrupt 7
   EINT;  // Enable Global interrupt INTM
   ERTM;  // Enable Global realtime interrupt DBGM
//  Loop Forever
   for(;;) {}
}
interrupt void cpu_timer0_isr(void)
{
p++;
if(p==6)
{
        AdcRegs.ADCTRL2.all = 0x2000;                   // Start SEQ1 in the ADC.
        while (AdcRegs.ADCST.bit.INT_SEQ1 == 0) {}  // Wait for SEQ1 Interrupt
        AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;               // Reset SEQ1
        AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;             // Clear SEQ1 Interrupt Status Flag
        //Take Voltage Sensor Reading
        Voltage_ADC_in = (AdcRegs.ADCRESULT3 >> 4);     // Shift ADCRESULT3 4-bits to the right.
        Voltage_in=(Voltage_ADC_in)/4096.0*3/0.145;
        //Take Current Sensor Reading
        Current_ADC_in = (AdcRegs.ADCRESULT2 >> 4);     // Shift ADCRESULT2 4-bits to the right.
        Current1_in=(Current_ADC_in/4096.0)*3;          //Convert Digital Value into Analog
        Current_in=(Current1_in/8.571428571428571428571428)*(20.0/0.625);          //Scale the
value
        error=Vref-Voltage_in;
        integral = integral + (error * 0.3);
        Duty1 =(1- (error* Kp + integral * Ki));
        if (Duty1>1)
                {
                        Duty1=0.99;
                }



        if (Duty1 < 0)
                {
                        Duty1 = 0;
                }
                Duty=Freq*Duty1;
                EPwm4Regs.CMPA.half.CMPA = Duty;  // Update Duty cycle
                EPwm4Regs.CMPB = Duty;
               Pold=Presult_in;
                l++;
        if (l==49)//15sec
        {
                Vresultin[j]=Voltage_in;
                Iresultin[j]=Current_in;
                Presult_in=Vresultin[j] * Iresultin[j];
                Vrefr[j]=Vref;
              we=1*(1/g1);
              wc=1*(1/g2);
              base=1;
              c_count=0;e_count=0;
e=AFF(((Presult_in-Pold)/(Vresultin[j]-Vresultin[j-1])));///(Vresultin[j]-Vresultin[j-1]))*0.3;
if ((Vresultin[j]-Vresultin[j-1])==0)
e=0;
```

```
if(e<=ce[0])
    {  mfe[0]=1;mfe[1]=0;mfe[2]=0;e_count=1;e_int=0;}
          else if (e>=ce[2])
           { mfe[0]=0;mfe[1]=0;mfe[2]=1; e_count=2;e_int=nume-1;
           }
           else
           {for(i=0;i<nume;i++)
           { if(e<=ce[i])
                    { mfe[i]=MAX(0, 1+((e-ce[i])/we));
                                                  if (mfe[i]!=0)
                                                  {e_count=e_count+1;
                                                  e_int=i;}}
           else
                    {
                     mfe[i]=MAX(0,1+((ce[i]-e)/we));
                              if (mfe[i]!=0)
                              { e_count=e_count+1;
                               e_int=i;  }
                    }
                    }
                    }
if (c<=cc[0])
{ mfc[0]=1;mfc[1]=0;mfc[2]=0;
           c_count=1;c_int=0; }
          else if (c>=cc[2])
{         mfc[0]=0;mfc[1]=0;mfc[2]=1;
           c_count=2;
           c_int=numc-1;}
          else
          {for(i=0;i<numc;i++)
          {if (c<=cc[i])
                    {mfc[i]=MAX(0,1+((c-cc[i])/wc));
                              if (mfc[i]!=0)
                              {c_count=c_count+1;
                               c_int=i;}}
                    else
                    {
                     mfc[i]=MAX(0,1+((cc[i]-c)/wc));
                              if (mfc[i]!=0)
                               {c_count=c_count+1;
                               c_int=i;}}}}
          num=0;
          den=0;
          for (k=e_int-e_count+1;k<(e_int+1);k++)
          {for (l=c_int-c_count+1;l<(c_int+1);l++)
          { prem=MIN(mfe[k], mfc[l]);
           num=num+rules[k][l]*base*(prem-(prem*prem)/2);
          den=den+base*(prem-(prem*prem)/2);
}}
 if(den!=0)
 output=num/den;
 if(num==0&&den==0)
 output=0;
c=c+output;
if (c<0)
c=0;
if(j<362)
{
estore[j]=e;
dstore[j]=Duty1;
outstore[j]=output;
}
```

```
if (Presult_in>Pold)
                        {if(Vresultin[j]>Vresultin[j-1])
                                {Vref+=c;               }
                                else
                                {Vref-=c;}}
                        else if (Presult_in<            Pold)
                        {if(Vresultin[j]>Vresultin[j-1])
                                {Vref-=c;}
                                else
                                {Vref+=c;}}
                        else
                        {Vref=Vref;}
                        if (Vref>21.6)
                        {Vref=21.6;}
                        if (Vref < 0)
                        {
                                Vref = 0;
                        }
                        j++;
                        if (j==1689)
                        {
                                j=0;
                        }
                        l=0;
                }
p=0;}
                PieCtrlRegs.PIEACK.bit.ACK1 = 1;                        //Jump to ISP again.
}
```

```
//////////////////////////////////////////////////////////////////////////////////////////////////////
////                                                                                              ////
////                        Stand Alone Sun Tracking PV Panel                                     ////
////                                                                                              ////
//////////////////////////////////////////////////////////////////////////////////////////////////////
#include "DSP2833x_Device.h"     // DSP2833x Headerfile Include File
#include "DSP2833x_Examples.h"   // DSP2833x Examples Include File
#include "math.h"
#include "time.h"
// ADC start parameters
#if (CPU_FRQ_150MHZ)     // Default - 150 MHz SYSCLKOUT
 #define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3)   = 25.0 MHz
#endif
#if (CPU_FRQ_100MHZ)
 #define ADC_MODCLK 0x2 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 100/(2*2)   = 25.0 MHz
#endif
#define ADC_CKPS   0x1   // ADC module clock = HSPCLK/2*ADC_CKPS   = 25.0MHz/(1*2) = 12.5MHz
#define ADC_SHCLK  0x0   // S/H width in ADC module periods                 = 16 ADC clocks
#define AVG       1000  // Average sample limit
#define ZOFFSET    0x00  // Average Zero offset
#define BUF_SIZE   1000  // Sample buffer size
#define DirE      GpioDataRegs.GPADAT.bit.GPIO20
#define DirA      GpioDataRegs.GPADAT.bit.GPIO6
#define Limit_elev      GpioDataRegs.GPADAT.bit.GPIO2
float pi = 3.14159265;
// Prototype statements for functions found within this file.
interrupt void cpu_timer0_isr(void);
// Global variables for this example
float ppp[1000];
Uint32 ii=0;
Uint32 jj=0;
long kk=0,pp=0;
// Initialization & Limit Variables:
int s=0, p=0, r=0,u=0, e=0, m = 0;
int i=0, j=0, k=0, v=0, w=0;
int initE=0;
int stop_east=0, stop_west=0;
Uint32 dutyA_init=0;
Uint32 dutyE_init=0;
// Encoder Measurement Variables:
int motDirA, motDirE;
int32 encA=0, encE=0;
int32 posA=0, posE=0;
int32 x1=0,x2=0, deltaE=0;
int32 y1=0,y2=0, deltaA=0;
int FLG1 =0, FLG2 =0;
//Elevation Control Variables:
float positionE_refd;                    //+ve for ccw (w.r.t elev. motor)
float positionE=0, positionE_deg=0, speedE=0, RPME=0;
float positionE_ref=0*3.14159/180, speedE_ref=0;
float integralE = 0,integralE2=0;
float KpE = 0.5, KiE = 7;       //PI gains for speed control
float KpE2 = 0.5, KiE2 =0.01;            //PI gains for position control
float errorE=0, prev_errE=0;
float errorE2=0, prev_errE2=0;
float dutyE=0, outputE;
Uint32 CMPAA_elev = 0;

//Azimuth Control Variables:
float positionA_init=0;
float positionA_refd;                    //+ve for ccw

float positionA=0, positionA_deg=0, speedA=0, RPMA=0;
float positionA_ref=-73.68*3.14159/180, speedA_ref=0;
//float positionA_ref= 0*3.14159/180, speedA_ref=0;
float integralA = 0,integralA2=0;
```

```
float KpA = 0.5, KiA =5.0;                          //PI gains for speed control
float KpA2 = 0.05, KiA2 =0;                 //PI gains for position control
float errorA=0, prev_errA=0, errorA2=0, prev_errA2=0;
float dutyA=0, outputA;
Uint32 CMPAA_az = 0;
// Trajectory Generation Variables:
Uint32 year=2011;
float month=12 , day=20, hour=12, minute=27, second=0;
struct tm *rtime;
time_t timestamp;
int n;
float t;
float d, lat, B, E, LST, h, Elev, Az, Elevation, Azimuth;
void InitEXT_INT(void);
void InitTMR0(Uint32 period);
float PWM_dutyE(void);
void PWM_updateE(float S);
float PWM_dutyA(void);
void PWM_updateA(float F);
void InitEPWM(void);
void initialize_Elev(void);
void initialize_Az(void);
interrupt void  xint1_isr(void);    // External interrupt for Elevation
interrupt void  xint2_isr(void);    // External interrupt for Azimuth
interrupt void cpu_timer0_isr(void);
interrupt void cpu_timer2_isr(void);
interrupt void cpu_timer1_isr(void);
void main(void)
{
        InitSysCtrl();
// Specific clock setting for this example:
        EALLOW;
        SysCtrlRegs.HISPCP.all = ADC_MODCLK;          // HSPCLK =
SYSCLKOUT/(HISPCP*2)=150/(3*2)=25.0 MHz
        EDIS;
 InitEPwm1Gpio();
 InitEPwm3Gpio();
// Disable CPU interrupts and clear all CPU interrupt flags:
        DINT;
        InitPieCtrl();
        IER = 0x0000;
        IFR = 0x0000;
//PIE Init.
        InitPieVectTable();
EALLOW;
        PieVectTable.TINT0 = &cpu_timer0_isr;          // interrupts that are used in this example are
        PieVectTable.TINT2 = &cpu_timer2_isr;
  PieVectTable.XINT13 = &cpu_timer1_isr;
  PieVectTable.XINT1 = &xint1_isr;
  PieVectTable.XINT2 = &xint2_isr;    // re-mapped to ISR (Interrupt Sub Routine)
// functions found within this file
        EDIS;
        //Timer Init.
        InitCpuTimers();
#if (CPU_FRQ_150MHZ)
// Configure CPU-Timer 0 to interrupt every 500 milliseconds:
// 150MHz CPU Freq, 50 millisecond Period (in uSeconds)
  ConfigCpuTimer(&CpuTimer0, 150, 10000);
  ConfigCpuTimer(&CpuTimer2, 150, 10000);
  ConfigCpuTimer(&CpuTimer1, 150, 100);
#endif

EALLOW;
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 0; //select GPIO
 GpioCtrlRegs.GPBDIR.bit.GPIO32 = 0; //input
GpioCtrlRegs.GPAMUX1.bit.GPIO1  = 0; //select GPIO
```

```
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; //output: Elevation motor direction
GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 0; //select GPIO
GpioCtrlRegs.GPADIR.bit.GPIO6 = 0; //input
GpioCtrlRegs.GPAMUX1.bit.GPIO5  = 0; //select GPIO
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1; //output
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 0; //select GPIO
 GpioCtrlRegs.GPADIR.bit.GPIO2= 0; //input
 GpioCtrlRegs.GPAMUX2.bit.GPIO20 = 0; //select GPIO
 GpioCtrlRegs.GPADIR.bit.GPIO20 = 0; //input
GpioCtrlRegs.GPAMUX2.bit.GPIO23 = 0; //select GPIO
 GpioCtrlRegs.GPADIR.bit.GPIO23 = 0; //input
EDIS;
        CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0
        CpuTimer2Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0
        CpuTimer1Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0
        InitEXT_INT();
EPwm4Regs.TBCTR = 0;
//Start Timer
        CpuTimer0Regs.TCR.bit.TSS = 0;                          //start timer0#
//Enable Interrupts
        IER |= M_INT1;  // enable CPU INT1 which is connected to CPU-Timer 0:
        // Enable CPU INT14 which is connected to CPU-Timer 2:
  IER |= M_INT14;
 // Enable CPU INT13 which is connected to CPU-Timer 1:
  IER |= M_INT13;
// Enable TINT0, XINT1, and XINT2 in the PIE: Group 1 (interrupts 7,4,5)
  PieCtrlRegs.PIEIER1.bit.INTx4 = 1;  //XINT1
  PieCtrlRegs.PIEIER1.bit.INTx5 = 1;  //XINT2
  PieCtrlRegs.PIEIER12.bit.INTx1 =1;  //XINT3
        PieCtrlRegs.PIEIER1.bit.INTx7 = 1;  // enable TINT0 in the PIE: Group 1 interrupt 7
   EINT;  // Enable Global interrupt INTM
   ERTM;  // Enable Global realtime interrupt DBGM
 EALLOW;
  SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0;
  EDIS;
  InitEPWM();
 EALLOW;
  SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;
  EDIS;
// Step 5. User specific code, enable interrupts:
// Enable CPU INT3 which is connected to EPWM1-3 INT:
  IER |= M_INT3;
// Enable EPWM INTn in the PIE: Group 3 interrupt 1-3
  PieCtrlRegs.PIEIER3.bit.INTx1 = 1;
  PieCtrlRegs.PIEIER3.bit.INTx2 = 1;
  PieCtrlRegs.PIEIER3.bit.INTx3 = 1;
// Enable global Interrupts and higher priority real-time debug events:
  EINT;  // Enable Global interrupt INTM
  ERTM;  // Enable Global realtime interrupt DBGM
//  Loop Forever
  for(;;)
  {
  if (initE == 1)
        {
                if(i==1)
                {
                        deltaE  = (x2 - x1) * 1.0;
                        speedE = (deltaE * 2.0 * pi) / (512.0 * 243 * 0.01); //rad/sec
                        RPME = (speedE * 60.0)/ (2.0 * pi);
                        positionE = (posE * 2.0 * pi) / (512.0 * 243);
      positionE_deg = positionE * (180 / pi); // degree
                        dutyE = PWM_dutyE();
                        PWM_updateE(dutyE);
                        i = 0;
                }
            if(j==1)
```

```
                    {
                        deltaA  = (y2 - y1) * 1.0;
                        speedA = (deltaA * 2.0 * pi) / (512.0 * 68.0 * 90.0 * 0.01); //rad/sec
                        RPMA = (speedA * 60.0)/ (2.0 * pi);
                        positionA =-73.68*pi/180 + (posA * 2.0 * pi) / (512.0 * 66.0 * 90.0);
                        positionA_deg = positionA * (180 / pi); // degree
                      dutyA = PWM_dutyA();
                      PWM_updateA(dutyA);
                        }                                   j = 0;
                    }

          }
 }

interrupt void cpu_timer0_isr(void)
{
          if(initE == 0)
          {
                    initialize_Elev();
          }

          if (initE == 1)
          {
          {
                        m = 0;
                    }
          }
          if (Elevation < 0 && m == 0)
          {
                    CMPAA_elev = 0;
                    errorE = 0;
                    integralE = 0;
          if(initE == 0)
                    {
                        initialize_Elev();
                    }
                    if(initE == 1)
                    {
                        m = 1;
                    }
}
if(FLG1==0)
          {
                    x1 = encE;
                    FLG1 = 1;
}
          else
          {
                    x2 = encE;
                    FLG1= 0;
                    encE=0;
                    i = 1;
          }
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
interrupt void cpu_timer2_isr(void)
{         if(FLG2==0)
          {
                    y1 = encA;
                    FLG2 = 1;           }

else
          {
                    y2 = encA;
                    FLG2 = 0;
                    encA=0;
```

```
                j = 1;
        }
}
interrupt void cpu_timer1_isr(void)
{       pp++;
        if (pp==9999)
        {
        if (year % 4 == 0)                              //Leap year
        {
                if(month==1)        {n = day;}
                else if(month==2)   {n = day+31;}
                else if(month==3)   {n = day+60;}
                else if(month==4)   {n = day+91;}
                else if(month==5)   {n = day+121;}
                else if(month==6)   {n = day+152;}
                else if(month==7)   {n = day+182;}
                else if(month==8)   {n = day+213;}
                else if(month==9)   {n = day+244;}
                else if(month==10)  {n = day+274;}
                else if(month==11)  {n = day+305;}
                else if(month==12)  {n = day+335;}
        }
        else                                                    //Common year
        {
                if(month==1)        {n = day;}
                else if(month==2)   {n = day+31;}
                else if(month==3)   {n = day+59;}
                else if(month==4)   {n = day+90;}
                else if(month==5)   {n = day+120;}
                else if(month==6)   {n = day+151;}
                else if(month==7)   {n = day+181;}
                else if(month==8)   {n = day+212;}
                else if(month==9)   {n = day+243;}
                else if(month==10)  {n = day+273;}
                else if(month==11)  {n = day+304;}
                else if(month==12)  {n = day+334;}
        }
        second++;                                               //increment seconds
        if (second == 60)
        {
                second = 0;
                minute++;

        }
        if (minute == 60)
        {
                minute = 0;
                hour++;                                         //increment hours
        }
        t = hour + minute/60 + second/3600;
        d = (pi/180)*23.45*sin((pi/180)*360*(284+n)/365);  //declination angle
        lat = (pi/180)*25.37;  //local latitude
        B = (pi/180)*360*(n-81)/364;
        E = 0.165*sin(2*B) - 0.126*cos(B) - 0.025*sin(B); //equation of time
        LST = t + (1/15.00)*(-60.00+55.21) + E;  //Solar time
        h = (pi/180)*15*(LST - 12);  //hour angle
        Elev = asin(cos(lat)*cos(h)*cos(d) + sin(lat)*sin(d));    //Elevation in rad
        if (h<0)
        {Az = -1*acos((cos(d)*sin(lat)*cos(h) - sin(d)*cos(lat))/cos(Elev));}  //Azimuth in rad (+ve for
west of south)

        else
        {Az = acos((cos(d)*sin(lat)*cos(h) - sin(d)*cos(lat))/cos(Elev));}  //Azimuth in rad (+ve for west
of south)
        Elevation = Elev*180/pi;    //Elevation in degrees
        Azimuth = Az*180/pi;  //Azimuth in degrees
```

```
            if (initE == 1 && Elevation > 0)
            {
            //Set set-points to Azimuth and Elevation
            positionE_ref = Elev;
            positionA_ref =Az;
            }
            if (Elevation < 0)
            {
            //Initialize set-points to zero after sunset
                        positionE_ref = 0;
                        positionA_ref = 0;
            }
pp=0;
}
            PieCtrlRegs.PIEACK.bit.ACK1 = 1;                        //Jump to ISP again.
}
interrupt void  xint2_isr(void)
{
            //Incrementing & decrementing for position & speed measurements (Azimuth)
                        if(DirA==1){encA--; posA--;}
                        else{encA++; posA++;}
                        PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
interrupt void  xint1_isr(void)
{
            //Incrementing & decrementing for position & speed measurements (Elevation)
                        if(DirE==1){encE--; posE--;}
                        else{encE++; posE++;}
                        PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
void InitEXT_INT(void)
{
  EALLOW;
  GpioCtrlRegs.GPAMUX1.bit.GPIO14= 0; //select GPIO
  GpioCtrlRegs.GPADIR.bit.GPIO14 = 0; //input: Elevation encoder clock
  GpioCtrlRegs.GPAQSEL1.bit.GPIO14 = 0;      // Xint1 Synch to SYSCLKOUT only
  GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 14;  // Xint1 is GPIO29
  XIntruptRegs.XINT1CR.bit.POLARITY = 1;     // Rising edge interrupt
  XIntruptRegs.XINT1CR.bit.ENABLE = 1;       // Enable Xint1
  GpioCtrlRegs.GPAMUX2.bit.GPIO27 = 0; //select GPIO
  GpioCtrlRegs.GPADIR.bit.GPIO27 = 0; //input: Azimuth encoder clock
  GpioCtrlRegs.GPAQSEL2.bit.GPIO27 = 0;      // Xint2 Synch to SYSCLKOUT only
  GpioIntRegs.GPIOXINT2SEL.bit.GPIOSEL = 27;  // Xint2 is GPIO27
  XIntruptRegs.XINT2CR.bit.POLARITY = 1;     // Rising edge interrupt
  XIntruptRegs.XINT2CR.bit.ENABLE = 1;       // Enable Xint2
 EDIS;
}
 void InitEPWM(void)
{
            EALLOW;
// PWM code for Azimuth & Elevation motors:
             EPwm1Regs.TBPRD=1000;
            EPwm1Regs.TBCTL.bit.CTRMODE=0; //Up count
            EPwm1Regs.TBCTL.bit.PRDLD=0;
            EPwm1Regs.TBCTL.bit.SYNCOSEL=3;
            EPwm1Regs.TBCTL.bit.HSPCLKDIV= 0;
            EPwm1Regs.TBCTL.bit.CLKDIV=1;  // prescale /2
            EPwm1Regs.AQCTLA.bit.ZRO=2;
            EPwm1Regs.AQCTLA.bit.CAU=1;
            EPwm3Regs.TBPRD=1000;

            EPwm3Regs.CMPA.half.CMPA=0;
            EPwm3Regs.TBCTL.bit.CTRMODE=0; //Up count
            EPwm3Regs.TBCTL.bit.PRDLD=0;
            EPwm3Regs.TBCTL.bit.SYNCOSEL=3;
            EPwm3Regs.TBCTL.bit.HSPCLKDIV= 0;
```

```
            EPwm3Regs.TBCTL.bit.CLKDIV=1;  // prescale /2
            EPwm3Regs.AQCTLA.bit.ZRO=2;
            EPwm3Regs.AQCTLA.bit.CAU=1;
            EDIS;
}
float PWM_dutyE()
{
            //Proportional Controller for position
            errorE2 = positionE_ref - positionE;
            integralE2 = integralE2 + (errorE2 * 0.01);
            speedE_ref = errorE2 * KpE2+integralE2 * KiE2;
            //PI Controller for speed
            errorE = speedE_ref - speedE;
            integralE = integralE + (errorE * 0.01);
            outputE = errorE * KpE + integralE * KiE;
                        if (Elevation > 0)
                        {
                                    if (outputE>=0)
                                    {
                                                motDirE = 1;
                                                GpioDataRegs.GPADAT.bit.GPIO1 = motDirE;// 1 = ccw w.r.t motor
                                                CMPAA_elev = (outputE/(1.769)) * 1000.00;
                                    }

                                    else
                                    {
                                                motDirE = 0;
                                                GpioDataRegs.GPADAT.bit.GPIO1 = motDirE;// 0 = cw w.r.t motor
                                                CMPAA_elev = -1 * (outputE/(1.769)) * 1000.00;
                                    }
                        }
                        else
                        {
                                    CMPAA_elev = 0;
                        }
 return CMPAA_elev;                     //return duty cycle
}
float PWM_dutyA()
{
            //Proportional Controller for position
            errorA2 = positionA_ref - positionA;
            integralA2 = integralA2 + (errorA2 * 0.01);
            speedA_ref = errorA2 * KpA2+integralA2 * KiA2;
            //PI Controller for speed
            errorA = speedA_ref - speedA;
            integralA = integralA + (errorA * 0.01);
            outputA = errorA * KpA + integralA * KiA;
                        {
                        if (outputA>=0)
                                    {
                                    motDirA =0;
                                    GpioDataRegs.GPADAT.bit.GPIO5 = motDirA;   // 0 = clockwise
                                    CMPAA_az = (outputA/(0.074)) * 1000.00;
                                    }
                                    else
                                    {
                                    motDirA =1;
                                    GpioDataRegs.GPADAT.bit.GPIO5 = motDirA;   // 1 = ccw
                                    CMPAA_az = -1 * (outputA/(0.074)) * 1000.00;
                                    }}

return CMPAA_az;                    //return duty cycle
}
void PWM_updateE(float S)
{
            EPwm1Regs.CMPA.half.CMPA = S;}    // Set compare A value (duty cycle)}
```

```
void PWM_updateA(float F)
{       EPwm3Regs.CMPA.half.CMPA = F;}    // Set compare A value (duty cycle)}
void initialize_Elev()
{
        dutyE_init = 600;               //duty cycle = 30%
        GpioDataRegs.GPADAT.bit.GPIO1 = 0;          //Direction toward limit switch
        PWM_updateE(dutyE_init);
        if(Limit_elev == 1)
        {u++;}
        if(u > 5)               //wait for 50 ms
        {
                dutyE_init = 0;
                PWM_updateE(dutyE_init);
                encE = 0;
                posE = 0;
                initE = 1;//Initialization complete
        u = 0;}}
```

VITA

Ahmad Abdel Jawad Al Nabulsi was born on June 24, 1986, in Dubai, UAE. He obtained his Bachelor of Science degree in electrical engineering with Cumlaude honor (3.55 GPA) from the American University of Sharjah on January 2009. He enrolled in the Mechatronics Masters program in the American University of Sharjah as a graduate teaching assistant (GTA), in 2009. Furthermore, he worked as a research assistant for the project Standalone Solar Energy System with MPPT funded by Emirates Foundation (EF).

Published conference papers:

1. A. Al-Nabulsi, R. Dhaouadi, and H. Rehman, "Single Input Fuzzy Controller (SFLC) based Maximum Power Point Tracking," *in Proc. ICMSAO'11*, Malaysia.

2. Ahmad Al Nabulsi, Ammar El Nosh, Abdulrahman Ahli, Mohamed Sulaiman, Rached Dhaouadi "Efficiency Optimization of a 150W PV System Using Dual Axis Tracking and MPPT" *in Proc. IEEE ENERGYCON 2010*, Bahrain.

3. A. Al-Nabulsi, M. Al-Sabbagh, R. Dhaouadi, and H. Rehman, "A 300 Watt Cascaded Boost Converter Design for Solar Energy Systems" *in Proc. EPECS'09*, Sharjah.

4. M. Al-Sabbagh, **A. Al-Nabulsi**, R. Dhaouadi, and H. Rehman, "Design and Control of A Tilt Table for Sun Tracking System," *in Proc. ICMSAO'09*, Sharjah.

Submitted transaction paper:

5. A. Al-Nabulsi, R. Dhaouadi, "Efficiency Optimization of a DSP-Based Standalone PV System using Fuzzy Logic and Dual-MPPT Control," Submitted to the *IEEE Trans. Indust. Inform*, Nov. 2011.