DESIGN AND IMPLEMENTATION OF A FUZZY LOGIC BASED MAXIMUM POWER POINT TRACKER FOR A PHOTOVOLTAIC SYSTEM

LAWRENCE KIPRONO LETTING

MASTER OF SCIENCE

(Electrical Engineering)

JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

2008

Design and Implementation of a Fuzzy Logic Based Maximum Power Point Tracker for a Photovoltaic System

Lawrence Kiprono Letting

A thesis submitted in partial fulfillment for the degree of Master of Science in Electrical Engineering in the Jomo Kenyatta University of Agriculture and Technology

2008

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signature …….. Date ……

Lawrence Kiprono Letting

This thesis has been submitted for examination with our approval as university supervisors:

Signature …….. Date ……

Dr. George N. Nyakoe

JKUAT, Kenya.

Signature …….. Date ……

Dr. Cyrus W. Wekesa

UoN, Kenya.

DEDICATION

I would like to dedicate this thesis to my mother for her constant and endless support.

ACKNOWLEDGEMENTS

First, I would like to take this opportunity to thank my supervisors Dr. G.N. Nyakoe and Dr. C.W. Wekesa for their valuable guidance and support throughout my research project. Their patience and encouragement were invaluable in the course of this research.

I would also like to thank my postgraduate colleagues including Kipyegon Edwin, Kibet Philip, Omae Oteri, Manene Franklin, and Irungu George for their support. In addition, I would like to thank the Chairman of the EEE department Dr. L.M. Ngoo for introducing us into the world of fuzzy logic. I would also like to thank D'Oketch and Kibunja at Electronics laboratory for their support. I will also not forget Barnabas Togom of Maryland, USA for his assistance in acquiring the data acquisition card. For that, I am extremely grateful.

Finally, I would like to thank my family and friends for their encouragement and support throughout my studies. Last but by no means the least, I would like to thank God for having seen me through all the challenges in my academic work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

A/D	Analogue/Digital
CCM	Continuous Conduction Mode
D/A	Digital/Analogue
DAQ	Data Acquisition Card
DCM	Discontinuous Conduction Mode
I/O	Input/Output
I-V	Current-Voltage
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracker
NB	Negative Big
NM	Negative Medium
NMM	Negative Medium Medium
NOCT	Nominal Cell Operating Temperature
NS	Negative Small
NSS	Negative Small Small
PB	Positive Big
PM	Positive Medium
PMM	Positive Medium Medium

PS Positive Small

PSS Positive Small Small

PV Photovoltaic

P-V Power-Voltage

PWM Pulse Width Modulation

TTL Transistor Transistor Logic

ZE Zero

LIST OF SYMBOLS

$\overline{I}_o$	Averaged converter output current
$\overline{I}_m$	Averaged PV module output current
$\overline{V}_m$	Averaged PV module output voltage
ΔD_k	Change in duty cycle at $k\text{-}th$ sampling instant
ΔP_k	Change in PV module power at $k\text{-}th$ sampling instant
ϑ_t	Duration in which inductor current is zero in a converter-switching period
η_{mppt}	Efficiency of maximum power point tracking
A	Curve-fitting parameter
a	Subscript indicates parameter refers to PV array
c	Subscript indicates parameter refers to solar cell
cD	Change in duty cycle
cP	Change in power
C_{SS}	Soft-start capacitor
C_T	Oscillator capacitor
D	Converter averaged duty cycle
d	Converter duty cycle
D_0	Initial value of duty cycle

D_k Duty cycle at $k\text{-}th$ sampling instant

f_s Converter switching frequency

G Solar radiation current-generator

G_a Solar radiation

i_C Capacitor current

I_c Solar cell output current

I_g Converter averaged input current

I_G Light-generated current

$I_{G,m}$ PV module light-generated current

i_L Inductor current

i_{Lmax} Maximum inductor current

i_{Lmin} Minimum inductor current

I_m PV module output current

I_{mp} PV module current at maximum power

I_o Converter averaged output current

I_o Diode or cell reverse saturation current

I_{sc} Short circuit current

I_{sc} Short-circuit current

I_{sh} Current through shunt resistance

I_T Diode current

k Boltzmann's gas constant (1.381×10^{-23} J/K)

k_d Duty cycle scale factor

k_p Power scale factor

L_{crit} Critical inductance

m Idealizing factor

m Subscript indicates parameter refers to PV module

M_p Number of solar cells in parallel in a PV array

M_s Number of solar cells in series in a PV array

N_p Number of solar cells in parallel in a PV module

N_s Number of solar cells in series in a PV module

$P_{achieved}$ Maximum power attained during maximum power point tracking

P_k PV module power at $k\text{-}th$ sampling instant

P_{mp} PV module maximum power

P_{mpp} Power at a maximum power point

q Electronic charge (1.602×10^{-19} C)

Q Transistor

R Load resistance

R' Load resistance referred to the PV module output terminals

R_C Capacitor equivalent series resistance (esr)

R_D Dead-time resistor

ref Subscript indicates the parameter at reference conditions

R_L Inductor winding resistance

R_{pvm} PV module internal resistance at maximum power

R_s Series resistance

R_{sh} Shunt resistance

R_T Oscillator resistor

S Switch

T Diode

t Time

T_a Ambient temperature

T_c Absolute cell temperature

t_{off} Transistor off-time

t_{on} Transistor on-time

T_s Converter switching period

V_C Averaged capacitor voltage

V_c Solar cell output voltage

V_{CC} Dc supply voltage

V_{CTRL} Control voltage

v_g DC-DC converter input voltage

$\overline{V_g}$ Converter averaged input voltage

V_m PV module output voltage

V_{mp} PV module voltage at maximum power

v_o DC-DC converter output voltage

$\overline{V_o}$ Converter averaged output voltage

V_{oc} PV module open-circuit voltage

V_Q	Voltage across the transistor
V_T	Diode forward voltage voltage
Z^{-1}	Unit time delay
δ	Transistor control signal
ε	Band-gap energy
μ_G	Solar cell temperature coefficient of solar radiation
$\mu_{I,sc}$	Coefficient of short-circuit current
$\mu_{V,oc}$	Coefficient of open-circuit voltage

ABSTRACT

This thesis presents a method used to optimize the energy extraction in a photovoltaic (PV) power system. The maximum power of a PV module changes with temperature, solar radiation, and load. To increase efficiency, PV systems use a Maximum Power Point Tracker (MPPT) to continuously extract the highest possible power and deliver it to the load. An MPPT consists of a dc-dc converter and a controller. The MPPT finds and maintains operation at the maximum power point using a tracking algorithm. Many such algorithms have been proposed. However, the existing methods have drawbacks in terms of efficiency, accuracy, and flexibility. Due to the nonlinear behaviour of PV module current-voltage characteristics and the nonlinearity of converters due to switching, conventional controllers are unable to give a good response in the presence of wide parameter variations and line transients.

The objective of this research was to design and implement an MPPT that uses a fuzzy logic control algorithm. Fuzzy logic, by dealing naturally with nonlinearities, offers a superior controller for this type of application. The technique also benefits from the heuristic approach to the problem that overcomes the complexity in modeling nonlinear systems. In order to achieve this goal, an MPPT model consisting of a PV module, a dc-dc converter, and a fuzzy logic controller was developed. Analysis of buck, boost, and buck-boost converter

characteristics was carried out in order to identify the most suitable topology. An integrated model of the PV module and the identified converter was simulated and the results used to derive the expert knowledge needed to formulate and tune the fuzzy logic controller. The controller was coded as a real-time control program and the MPPT implemented using a dc-dc converter controlled by a microcomputer.

The proposed method shows improved performance in terms of oscillations about the maximum power point, speed, and sensitivity to parameter variation. The results indicate that a significant amount of additional energy can be extracted from a photovoltaic module by using a fuzzy logic based maximum power point tracker. This results in improved efficiency for the operation of a photovoltaic power system since batteries can be sufficiently charged and used during periods of low solar radiation. The improved efficiency is expected to lead to significant cost savings in the long run.

CHAPTER ONE

1 INTRODUCTION

This thesis presents the design and implementation of a fuzzy logic based Maximum Power Point Tracker (MPPT) for a photovoltaic power supply. In this chapter, the background of the problem is described and the motivation for the work is presented. An overview of existing MPPT algorithms is then carried out and the need for application of fuzzy logic in maximum power point tracking is discussed. The last part gives the organization of the remainder of the thesis.

1.1 Background

There is an ever-increasing energy demand, owing to industrial development and population growth. This has led to greater interest in research and technological investments related to improved energy efficiency and use of alternative and renewable energy sources. Fossil fuels used in the production of power are also dwindling and becoming more expensive. The main challenge in replacing conventional energy sources with newer more environmentally friendly alternatives, such as solar and wind energy, is how to capture the maximum energy and deliver the maximum power at a minimum cost for a given load. A combination of two or more types of energy sources might offer the best chance of optimizing power generation by varying the contribution from each energy source depending on the

load demand. The goal is to develop and optimize maximum power tracking and control of a multi-source renewable distributed energy generation system consisting photovoltaic (PV) modules, wind generators and other sources.

At a subsystem level, photovoltaic (PV) power is a renewable energy source that is currently attracting attention and might in future replace fossil fuel dependent energy sources. However, for that to happen, PV power cost per kilowatt-hour has to be competitive in comparison to fossil fuel energy sources. The efficiency of PV modules depends on the material used in solar cells and the technology used in arranging the solar cells to form a module. Currently, PV modules have very low efficiencies with only about $12-29\%$ efficiency in their ability to convert sunlight to electrical power [1]. Gallium Arsenide solar cells have a high efficiency of 29%, while Silicon solar cells have an efficiency of about 12-14%. The efficiency can drop further due to other factors such as PV module temperature and load conditions. In order to maximize the power derived from the PV module it is important to operate the module at its optimal power point. To achieve this, a controller called a Maximum Power Point Tracker is required.

A PV module is a non-linear power source and its output power depends on the terminal operating voltage. The Maximum Power Point Tracker compensates for the varying current-voltage characteristics of the solar cell. The MPPT varies the output voltage and current from the PV module and determines the operating point that will deliver the most power. The MPPT must be able to accurately track the

constantly varying operating point where the maximum power is delivered in order to increase the efficiency of the PV module.

1.2 Statement of the Problem

Photovoltaic power is a relatively untapped source of energy due to low efficiency and relatively high cost per watt compared to fossil fuels. Thus, there still remains a lot of work to be done to make PV systems as efficient and reliable as possible. One approach to understanding and improving PV module efficiency is through modeling and simulation. After successfully modeling and simulating a PV module, it is possible to develop methods for optimizing the system operation.

Various methods of maximum power tracking in PV power applications have been reported in literature. The existing methods have drawbacks in terms of efficiency, accuracy and flexibility. This research explored ways of improving maximum power point tracking using fuzzy logic. The control algorithm uses the excellent knowledge representation and deduction capabilities of fuzzy logic to address the drawbacks of existing methods.

1.3 Objectives

The main objective of this research was to design and implement a fuzzy logic based maximum power point tracker for a photovoltaic power supply. In order to achieve this goal, an MPPT model consisting of a PV module, a dc-dc converter,

and a fuzzy logic controller was developed. Analysis of buck, boost, and buck-boost converter characteristics was then carried out in order to identify the most suitable topology. An integrated model of the PV module and the identified converter was simulated and the results used to obtain the expert knowledge needed to formulate and tune the fuzzy logic control algorithm for tracking the maximum power. The fuzzy logic controller was coded as a real-time control program and the MPPT implemented using a dc-dc converter controlled by a microcomputer.

1.4 An Overview on MPPT Algorithms

Many MPPT algorithms have been proposed and are generally categorized into the following groups: 1) perturbation and observation methods [4]–[6]; 2) incremental conductance methods [7], [8]; 3) fuzzy logic [9] and neural network based methods [10].

The "perturbation and observation" method, also known as the "hill-climbing method," is widely applied because of its ease of implementation. This method tracks the maximum power point (MPP) by repeatedly increasing or decreasing (perturbing) the module voltage and comparing the output power with that at the previous perturbing cycle. Various problems occur in this method when acquiring the maximum power. It cannot track the MPP during low solar radiation levels and when radiation changes rapidly. It also oscillates around MPP instead of directly tracking it [10], [11]. As oscillations always appear in the method, the power loss may be increased. Several improvements of the perturb and observe

algorithm have been proposed. One approach involves the use of the short-circuit current or the open-circuit voltage to determine the direction in which to perturb the module voltage. Methods based on this approach can be considered as variations of the standard perturb and observe algorithm since instead of observing the change in PV module power, change in either module short-circuit current or open-circuit voltage is used.

The "short-circuit current method" [12] performs MPPT control while a short circuit current flows for measurements in the circuit. Although this method does not have oscillations like those appearing in the standard perturb and observe method, the power loss may increase since the short circuit current flows whenever MPPT control is performed. Furthermore, it becomes difficult to perform MPPT control during periods of low solar radiation because short-circuit current decreases with solar radiation.

The "open-circuit voltage method" [13] utilizes the fact that the operating voltage is almost linearly proportional to open-circuit voltage of the PV module at MPP. It is simple, cost-effective, and avoids power loss associated with the short-circuit current method. A limitation of this method is the fact that the reference voltage does not change between samplings [10].

The "incremental conductance method", is a technique used to reduce the oscillation around the MPP. This method calculates the direction in which to perturb the module's operating point and it can determine when it has actually reached the MPP [11]. It is however, computationally intensive and the speed at

which it approaches the MPP depends on a fixed perturbation step. The perturbation step is difficult to choose when dealing with tradeoff between steady-state performance and fast dynamic response. The control circuit is also complex resulting in a higher system cost [5].

Application of fuzzy logic and artificial neural networks in MPPT control is an ongoing research field. These modern algorithms based on artificial intelligence are capable of improving the tracking performance as compared to existing conventional methods [10]. With the neural network based method, the solar radiation, temperature, module voltage and current are measured and used to identify the maximum power point of the PV module [10], [14]. Although this method can predict the maximum power point, the data acquisition and memory space requirements are very intensive and greatly affects the performance of the algorithm. This research explores ways of improving maximum power tracking using fuzzy logic.

1.5 Fundamentals of Fuzzy Logic Controllers

1.5.1 What is Fuzzy Logic?

Fuzzy Logic is a branch of Artificial Intelligence. It owes its origin to Lofti Zadeh, a professor at the University of California, Berkley, who developed fuzzy set theory in 1965 [15]. The basic concept underlying fuzzy logic is that of a linguistic variable, that is, a variable whose values are words rather than numbers (such as

small and *large*). Fuzzy logic uses fuzzy sets to relate classes of objects with unclearly defined boundaries in which membership is a matter of degree.

1.5.2 Fuzzy Sets and Membership Functions

A fuzzy set is an extension of a crisp set where an element can only belong to a set (full membership) or not belong at all (no membership). Fuzzy sets allow partial membership which means that an element may partially belong to more than one set. A fuzzy set A is characterized by a membership function μ_A that assigns to each object in a given class a grade of membership to the set. The grade of membership ranges from 0 (no membership) to 1 (full membership) written as,

$$\mu_A : U \rightarrow [0,1] \tag{1-1}$$

which means that the fuzzy set A belongs to the universal set U (called the universe of discourse) defined in a specific problem. A membership function defines how each point in the input space is mapped to a degree of membership.

For example, consider the set of membership functions for a set of tall people shown in Figure 1-1. If the set is given the crisp boundary of a classical set, it can be considered that all people taller than six feet are considered tall, while those less than six feet are short. But, such a distinction is not fully realistic. If one would however consider a smooth curve from “short” to “tall”, then the transition would make more sense. A person may be both tall and short to some degree. The output axis would be a number between 0 and 1, known as the degree of membership in a fuzzy set of height.

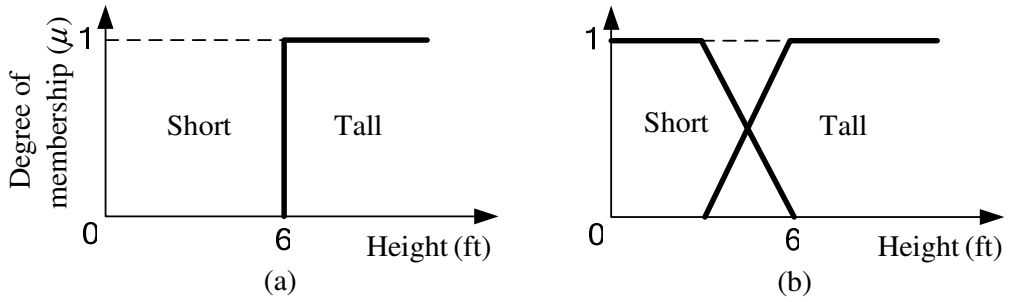


Figure 1-1 Illustration of membership functions for a set of tall people (a) crisp set (b) fuzzy set

1.5.3 Fuzzy Rules and Inferencing

The use of fuzzy sets allows the characterization of the system behaviour through fuzzy rules between linguistic variables. *A fuzzy rule* is a conditional statement R_i based on expert knowledge expressed in the form:

$$R_i : \text{IF } x \text{ is } small \text{ THEN } y \text{ is } large \tag{1-2}$$

where x and y are fuzzy variables and *small* and *large* are labels of the fuzzy sets. If there are n rules, the rule set is represented by the union of these rules i.e.,

$$R = R_1 \text{ else } R_2 \text{ else } \ldots\ldots R_n . \tag{1-3}$$

A fuzzy controller is based on a collection R, of control rules. The execution of these rules is governed by the compositional rule of inference [17] [18].

1.5.4 Fuzzy Controller Structure

The general structure of a fuzzy logic controller is presented in Figure 1-2 and comprises of four principal components:

- *Fuzzification interface*: - It converts input data into suitable linguistic values using a membership function.
- *Knowledge base*: - Consists of a database with the necessary linguistic definitions and the control rule set.
- *Inference engine*: - It simulates a human decision making process in order to infer the fuzzy control action from the knowledge of the control rules and the linguistic variable definitions.
- *Defuzzification interface*: - Converts an inferred fuzzy controller output into a non-fuzzy control action.

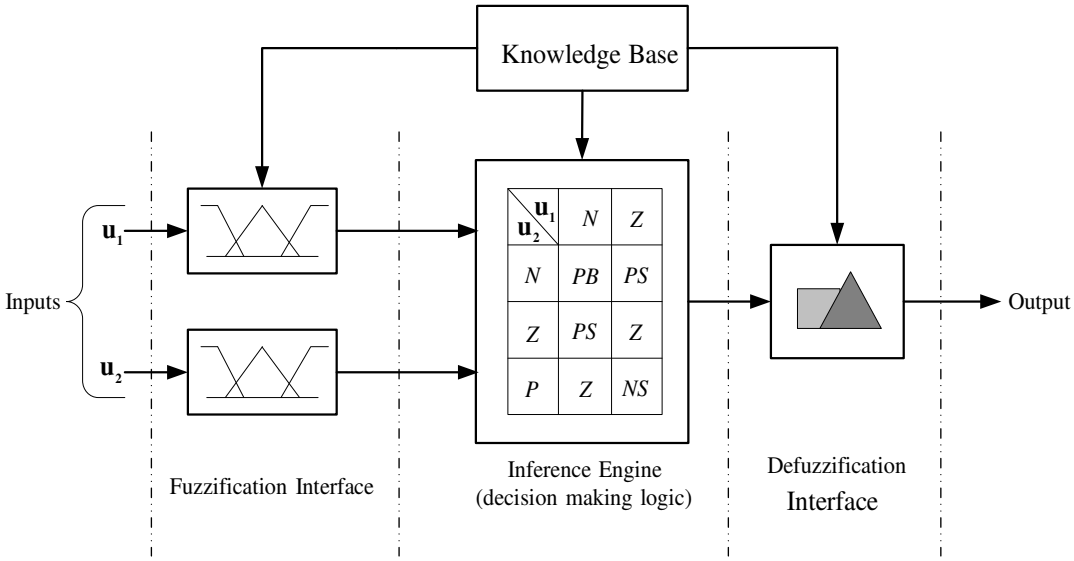


Figure 1-2 Basic configuration of a fuzzy logic controller

1.6 Application of Fuzzy Logic in MPPT Control

Dc-dc converter systems are becoming strong candidates for modern fuzzy control techniques due to their complex, nonlinear behavior, particularly for large load and line variations [16]. The highly nonlinear behavior of these power circuits is caused by the presence of a switch, which can be any electronic switch such as a transistor, a thyristor, or any other switching device. Depending on the state of the switch (ON/OFF) the plant structure exhibits very different functioning modes, resulting in a severe nonlinearity. PV modules also have nonlinear current-voltage (I-V) characteristics that are dependent on solar radiation, temperature, and degradation due to environmental effects. Therefore, their operating point that corresponds to the maximum output power varies with the environmental and load conditions.

MPPT control is therefore an intriguing subject from the control point of view, due to the intrinsic nonlinearity of dc-dc converters and PV modules. This is because an accurate model of the plant and the controller is necessary while formulating the control algorithm. There are two possible ways of overcoming this. One method is to develop more accurate nonlinear models for controllers, but the discouraging fact about taking this route is that complex mathematical derivations are involved. Even when developed, the complicated control algorithms may not be suitable for practical implementations. The other method is to employ heuristic reasoning based on human experience of the plant. Such

experience is usually collected in the form of linguistic statements and rules. In this case, no modeling is required, and the whole business of controller design reduces to the "conversion" of a set of linguistic rules into an automatic control algorithm. Here, fuzzy logic comes into play as it provides the essential machinery for performing the said conversion. Such a completely different approach is offered by fuzzy logic, which does not require a precise mathematical modeling of the system nor complex computations [17], [18]. This control technique relies on the human capability to understand the system's behavior and is based on qualitative control rules. Thus, controller design is simple, since it is only based on linguistic rules of the type: "*IF the* ***change in output power*** *is* ***positive*** *AND the* ***change in duty cycle*** *is* ***negative*** *THEN* ***reduce slightly the duty cycle***" and so on.

Fuzzy logic control relies on basic physical properties of the system, and it is potentially able to extend control capability even to those operating conditions where linear control techniques fail, i.e., large signal dynamics and large parameter variations. As fuzzy logic control is based on heuristic rules, application of nonlinear control laws to overcome the nonlinear nature of dc-dc converters is easy. Fuzzy logic offers several unique features that make it a particularly good choice for these types of control problems because:

- It is inherently robust as it does not require precise, noise-free inputs. The output control is a smooth control function despite a wide range of input variations.

- It can be easily modified to improve system performance by generating appropriate governing rules.
- Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.
- Its rule-based operation enables any reasonable number of inputs to be processed and numerous outputs generated. The control system can be broken into smaller units that use several smaller fuzzy logic controllers distributed on the system, each with more limited responsibilities.

1.7 Thesis Organization

This thesis consists of six chapters. In Chapter 1, recent opportunities and challenges to photovoltaic energy generation have been highlighted. The motivation for conducting the research is discussed as well as the expected outcome. The chapter introduces the fundamentals of fuzzy logic control, gives an overview of existing MPPT algorithms and compares possible advantages of one over the other.

Chapter 2 introduces the operation and modeling framework for PV modules. It includes a literature review on the photovoltaic modules, the solar cell characteristics and cell operation. A PV module model is developed and

simulation results are compared with those provided in the manufacturer's data sheets.

Chapter 3 covers the identification of a suitable dc-dc converter topology for maximum power tracking considering the buck, boost and buck-boost topologies. State-space averaging is used to derive a model of the identified converter. The simulation results are used to develop the control strategies and to choose the converter components in subsequent chapters.

In chapter 4, the PV module and converter models developed are combined to form a complete MPPT model. The design and implementation of the fuzzy logic-based MPPT in a microcomputer is then carried out.

The simulated and experimental results are presented in Chapter 5. Discussion of the results is also carried out.

Chapter 6 gives the thesis conclusions and the suggestions for future work.

CHAPTER TWO

2 MODELING OF A PHOTOVOLTAIC MODULE

This chapter introduces the modeling framework for a photovoltaic module. A model is developed and validated using manufacturer supplied data for a specific module. The model is used to study PV module operation characteristics with the view of formulating a suitable control strategy for extraction of maximum power. The model theory in this chapter is adapted from the *Hybrid2* theory manual [19]. *Hybrid2* is a computer simulation model for hybrid power systems developed by the University of Massachusetts. The model in *Hybrid2* can calculate the expected cost savings when maximum power point trackers are included in photovoltaic systems but has no provision for formulating or testing control strategies.

2.1 Model Theory

A PV module is composed of individual solar cells connected in series and parallel and mounted on a single panel. The goal is to calculate the power output from a PV module based on an analytical model that defines the current-voltage relationships based on the electrical characteristics of the module. As described in the following theory section, a one diode model forms the basic circuit model used to establish the current-voltage curve specific to a PV module. The theory is used to formulate a PV module model using Simulink software. This model is able to

include the effects of solar radiation level and cell temperature on the output power. The performance of PV arrays that consist of several modules connected in series or parallel is also discussed.

2.1.1 Model Inputs

The primary inputs that affect the PV module output are the parameters that define the basic module current-voltage (I-V) relationship. These parameters are determined from information supplied by the manufacturer and include the open circuit voltage and short circuit current of the module. The model inputs during simulation are the solar radiation and ambient temperature.

2.1.2 Model Outputs

The model outputs at the beginning of the simulation are 1) the light-generated current, 2) the diode (or cell) reverse saturation current, 3) the series resistance, and 4) a curve fitting parameter. The output of the model at each time step is the generated module current and voltage.

2.1.3 Modifying Parameters

These parameters affect the calculations performed in the PV module model. They relate to the module behavior under various ambient and load conditions. Modules are rated at various standard conditions. Ratings or specifications at other conditions are considered using the modifying parameters.

These parameters include the number of cells in series (N_s), solar radiation (G_a) and cell temperature (T_c) at normal operating conditions ($NOCT$).

2.1.4 Solar Cell Model

Solar cells are solid-state semiconductor devices that convert incident sunlight energy into an electrical current. Currently, Silicon and Gallium Arsenide are the most commonly used materials in the manufacture of solar cells. The equivalent circuit of a solar cell is based on the well-known single-diode representation as shown in Figure 2-1 [19]. The model contains a current source G, one diode T, a shunt resistance R_{sh}, and a series resistance R_s. R_{sh} models the surface leakage along the edges of the cell or the crystal defects along the junction depletion region, while R_s models the resistance of the diffused layer that is in series with the junction as well as the resistance of the ohmic contacts [20].

The net current I is the difference between the light-generated current I_G, the normal diode current I_T, and current through R_{sh}.

$$I = I_G - I_T - I_{sh} \tag{2-1}$$

The diode current I_T and the current through the shunt resistance I_{sh} are given by Equations (2-2) and (2-3) respectively [19].

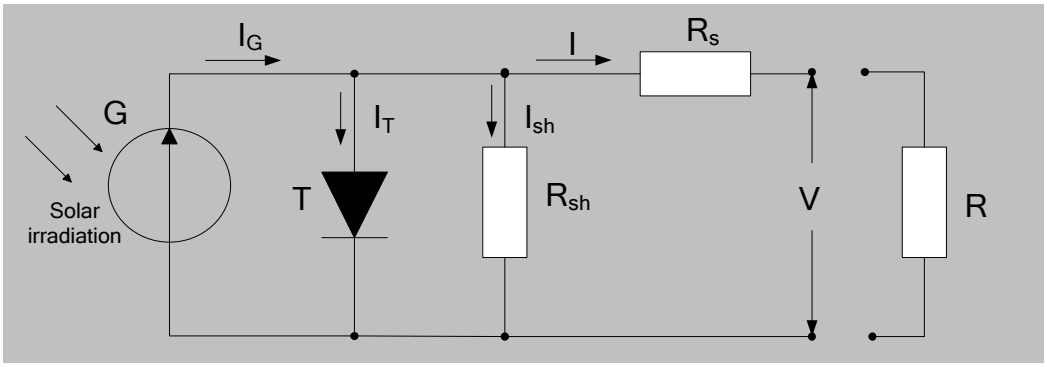


Figure 2-1 Equivalent Circuit of a Solar Cell

$$I_T = I_o \left\{ \exp\left[\frac{q}{mkT_c}(V + IR_s) \right] - 1 \right\} \tag{2-2}$$

$$I_{sh} = \frac{V + IR_s}{R_{sh}} \tag{2-3}$$

where, m is the idealizing factor, k is Boltzmann's gas constant, T_c is the absolute cell temperature, q is the electronic charge, V is the voltage imposed across the cell, and I_o is the cell reverse saturation current.

Using Equations (2-2) and (2-3), Equation (2-1) is expressed as shown in Equation (2-4).

$$I = I_G - I_o \left\{ \exp\left[\frac{q}{mkT_c}(V + IR_s) \right] - 1 \right\} - \frac{V + IR_s}{R_{sh}} \tag{2-4}$$

Normally the shunt resistance, R_{sh}, in most modern cells is very large [19]; thus $(V + IR_s)/R_{sh}$ in Equation (2-4) can be ignored. Hence,

$$I = I_G - I_o \left\{ \exp\left[\frac{(V + IR_s)}{A} \right] - 1 \right\} \tag{2-5}$$

where A is the curve fitting parameter given by,

$$A = \frac{mkT_c}{q} \tag{2-6}$$

2.2 Modeling of the PV module

2.2.1 PV Module Structure

A PV module consists of N_p parallel branches, each with N_s solar cells in series as shown in Figure 2-2 [21]. A model for the PV module is obtained by replacing each cell in Figure 2-2 by its individual solar cell model. For clarity, the following notation is used: the parameters with subscript "*m*" refer to the PV module, while the parameters with subscript "*c*" refer to the solar cell.

2.2.2 Model Parameters

Using Equation (2-5), the module current I_m, under arbitrary operating conditions is given by:

$$I_m = I_{G,m} - I_{o,m} \left\{ \exp\left[\frac{(V_m + I_m R_{s,m})}{A} \right] - 1 \right\} \tag{2-7}$$

where, $I_{G,m}$ is the module light generated current, $I_{o,m}$ is the module reverse saturation current, V_m is the module voltage, and $R_{s,m}$ is the module series resistance. The module voltage is obtained from Equation (2-7) as,

$$V_m = A\log_e\left(\frac{I_{G,m} - I_m}{I_{o,m}} + 1\right) - I_m R_{s,m} \tag{2-8}$$

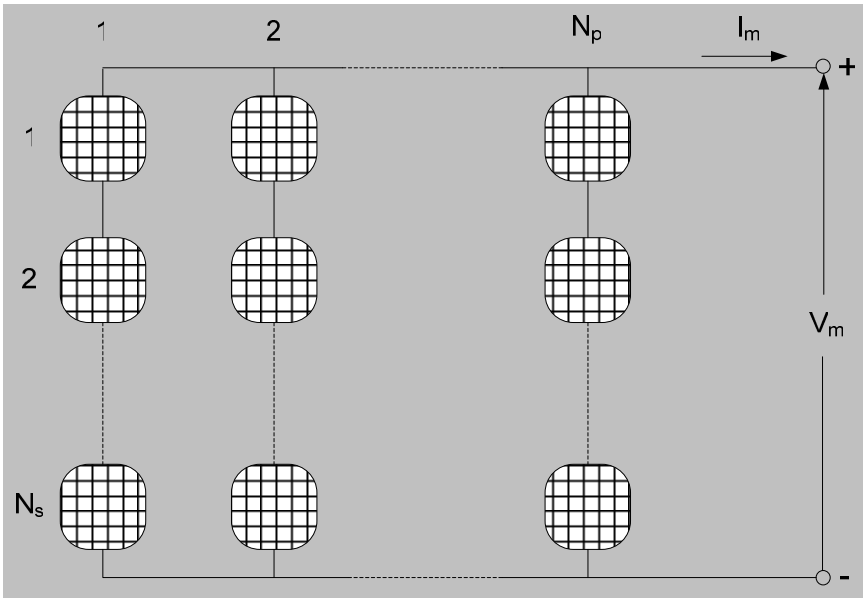


Figure 2-2 Model structure of a PV module

The relationship between PV module and solar cell parameters is given by Equations (2-9) through (2-11).

$$I_{G,m} = N_p I_{G,c} \tag{2-9}$$

$$V_m = N_s V_c \tag{2-10}$$

$$I_{o,m} = N_p I_{o,c} \tag{2-11}$$

where $I_{G,c}$ is the cell light-generated current, $I_{o,c}$ is the cell reverse saturation current, and V_c is the cell output voltage.

2.2.3 Determination of Model Parameters

Reference values for the three parameters in this model, namely, the light-generated current I_G , reverse saturation current I_o , and absolute cell temperature T_c, can be obtained indirectly using measurements of the current and voltage characteristics of a solar module at reference conditions. Measurements of current and voltage at these or other known reference conditions are often available at open circuit conditions, short circuit conditions, and maximum power conditions from manufacturer's data sheets. All quantities with the subscript '*ref*' are obtained from measurements taken at reference conditions. Traditionally, measurements of PV electrical characteristics are made at reference incident radiation of $1\ kW/m^2$ and an ambient temperature of 25^oC. These are the standard conditions used by manufacturers to test PV modules.

Light Current at Reference Conditions

At short circuit conditions, the diode current is very small and the light-generated current is equal to the short circuit current measured at the reference conditions, i.e.,

$$I_{G,ref} = I_{sc,ref} \tag{2-12}$$

Diode Reverse Saturation Current

At open circuit conditions there is no current, the exponential term in Equation (2-5) is much greater than unity.

$$I_o = I_G \exp\left(\frac{-V_{oc}}{A}\right) \tag{2-13}$$

where, V_{oc} is the open circuit voltage. At reference conditions,

$$I_{o,ref} = I_{G,ref} \exp\left(\frac{-V_{oc,ref}}{A_{ref}}\right) \tag{2-14}$$

Series Resistance

The series resistance is derived using the I-V curve at maximum power conditions and is given by [19]:

$$R_s = \frac{A_{ref}.\log_e\left[1-\frac{I_{mp,ref}}{I_{G,ref}}\right] - V_{mp,ref} + V_{oc,ref}}{I_{mp,ref}} \tag{2-15}$$

where, I_{mp} and V_{mp} are the current and voltage at maximum power point respectively. If both R_s and A_{ref} are to be positive, the maximum value of R_s occurs when A tends to zero and the maximum value of A_{ref} occurs when R_s tends zero.

Temperature Coefficients

The temperature coefficient of the short circuit current $\mu_{I,sc}$, is obtained from measurements at the reference solar radiation using,

$$\mu_{I,sc} = \frac{dI_{sc}}{dT_c} = \frac{I_{sc}(T_2) - I_{sc}(T_1)}{T_2 - T_1} \tag{2-16}$$

where T_2 and T_1 are two temperatures above and below the reference temperature. Similarly, the temperature coefficient of the open circuit voltage, $\mu_{V,oc}$ can be obtained from measurements using:

$$\mu_{V,oc} = \frac{dV_{oc}}{dT_c} = \frac{V_{oc}(T_2) - V_{oc}(T_1)}{T_2 - T_1} \tag{2-17}$$

Curve Fitting Parameter at Reference Conditions

This value can be derived by equating the experimental value of $\mu_{V,oc}$ with the value determined from an analytical expression for the derivative, dV_{oc}/dT to give:

$$A_{ref} = \frac{T_{c,ref}.\mu_{V,oc} - V_{oc,ref} + \varepsilon.N_s}{\frac{\mu_{I,sc}.T_{c,ref}}{I_{G,ref}} - 3} \tag{2-18}$$

where ε is the material bandgap energy in electron-volts (eV). The bandgap energy depends on the material used in the manufacture of solar cells. The derivation of Equation (2-18) is given in [19].

Cell Temperature

The working temperature of the cells T_c, depends exclusively on the solar radiation G_a, and on the ambient temperature T_a according to the empirical linear relation: [21]

$$T_c = T_a + \mu_G G_a \tag{2-19}$$

where the constant μ_G is the temperature coefficient of solar radiation is computed as:

$$\mu_G = \frac{T_{c,ref} - T_{a,ref}}{G_{a,ref}} \tag{2-20}$$

The open circuit voltage depends exclusively on the temperature of the solar cells and is given by,

$$V_{c,oc} = V_{c,ref} + \mu_{V,oc}(T_c - T_{c,ref}) \tag{2-21}$$

where $V_{c,oc}$ is the solar cell open circuit voltage.

2.2.4 PV Array Structure

Large PV systems consist of modules connected in arrays. Figure 2-3 shows an array of modules with M_p parallel branches each with M_s modules in series. V_a denotes the output voltage at the array's terminals, while I_a denotes the total generated current. If all the modules are identical and operating under uniform solar radiation, the PV array's current and voltage is given by Equations (2-22) and (2-23) respectively.

$$I_a = M_P I_m \tag{2-22}$$

$$V_a = M_s V_m \tag{2-23}$$

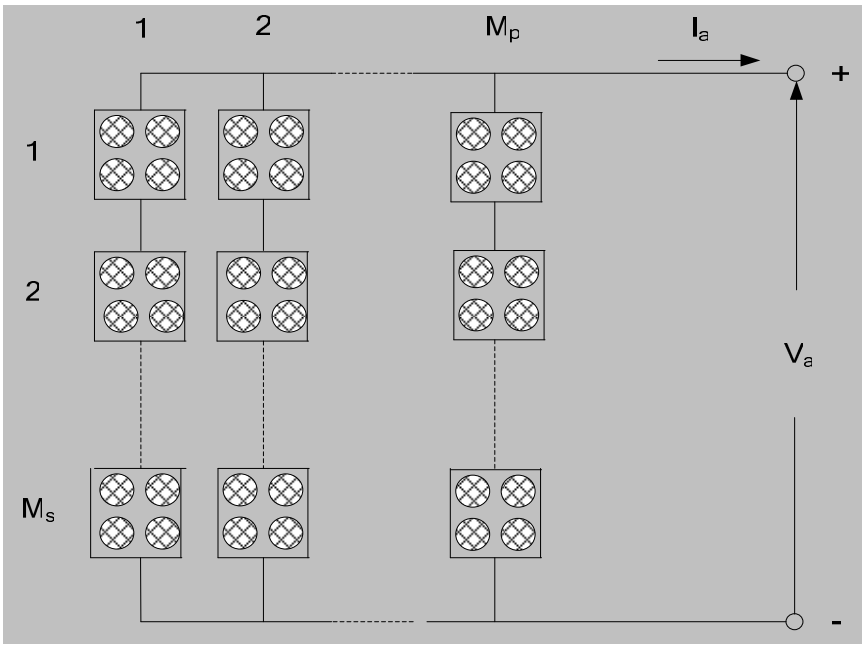


Figure 2-3 Model structure of a PV array

2.3 Simulink Models

Matlab-Simulink® is a software package that provides a graphical user interface for modeling, simulating, and analyzing dynamic systems. It offers the advantage of building hierarchical models, i.e. the system can be designed using both top-down, and bottom-up approaches. Modeling of linear and nonlinear systems in continuous time, sampled time, or a hybrid of the two is also supported [22].

2.3.1 PV Array Model

The PV array Simulink model is structured in a hierarchical manner as shown in Figure 2-4. Arrows entering the model are inputs while those leaving are the outputs. The model allows the user to define the number of modules and their connection. The array current and voltage is computed using Equations (2-22) and (2-23) respectively.

The Simulink model mask for the PV array external and internal structure is shown in Figure 2-5 and Figure 2-6 respectively. The external inputs are the solar radiation G_a (expressed as a percentage of the reference value), ambient temperature T_a, and the array current I_a from the previous time step. The model's output is the array voltage V_a, and the cell temperature T_c. The array output is solely dependent on the operation of the individual PV modules, which are assumed to be identical in this model. The parameters of the model can be set using the dialogue box shown in Figure 2-7.

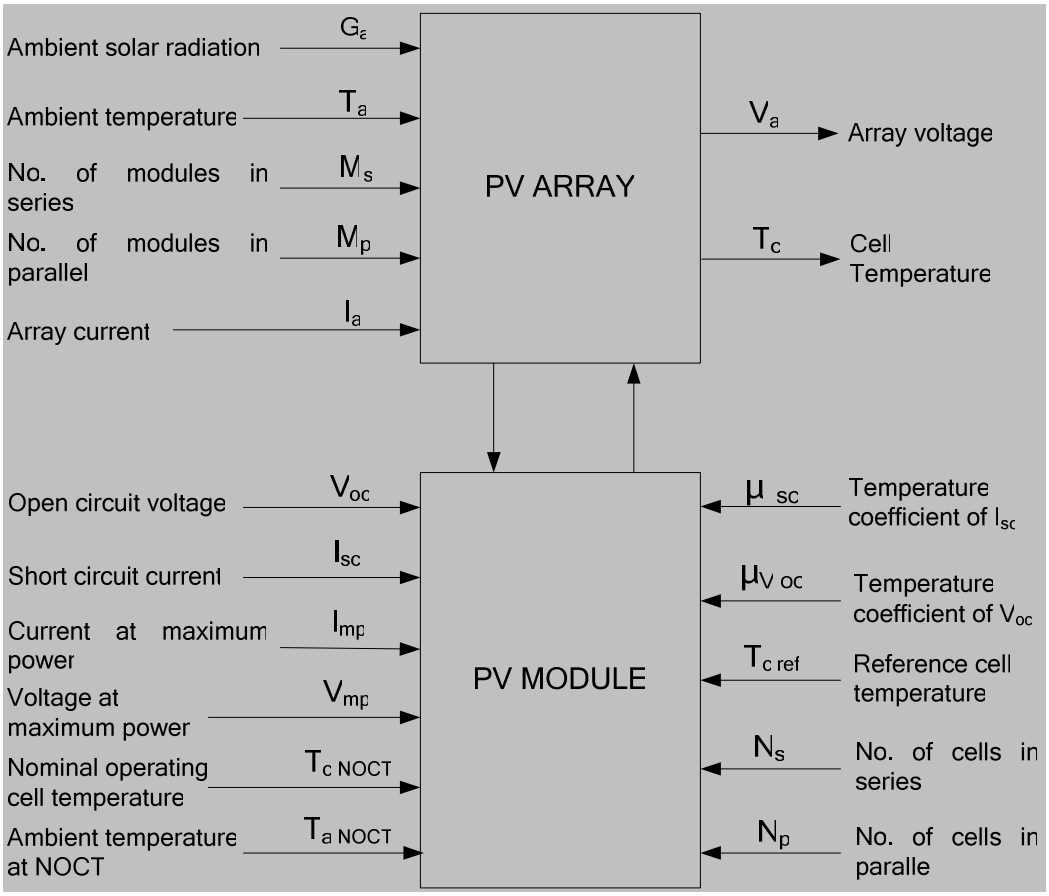


Figure 2-4 PV array Simulink model structure

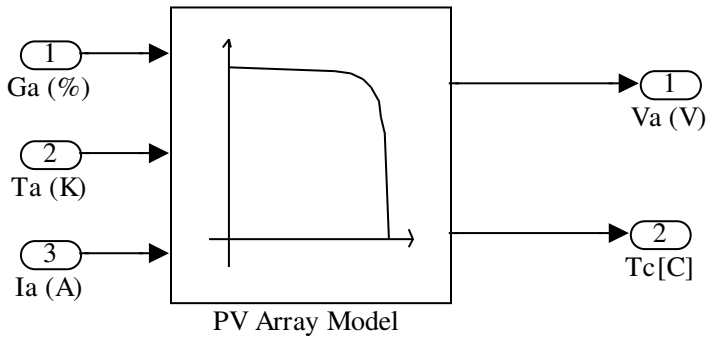


Figure 2-5 Simulink block diagram mask of a PV array - external structure

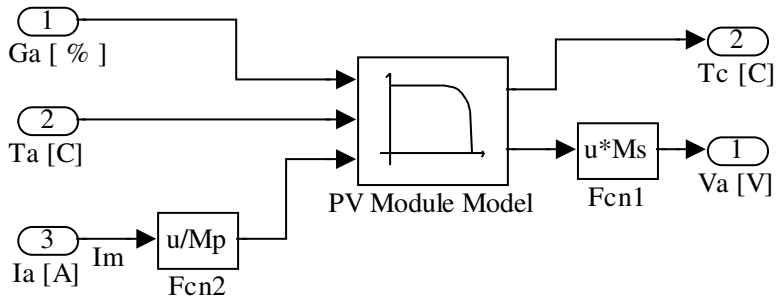


Figure 2-6 Simulink block diagram mask of a PV array - internal structure

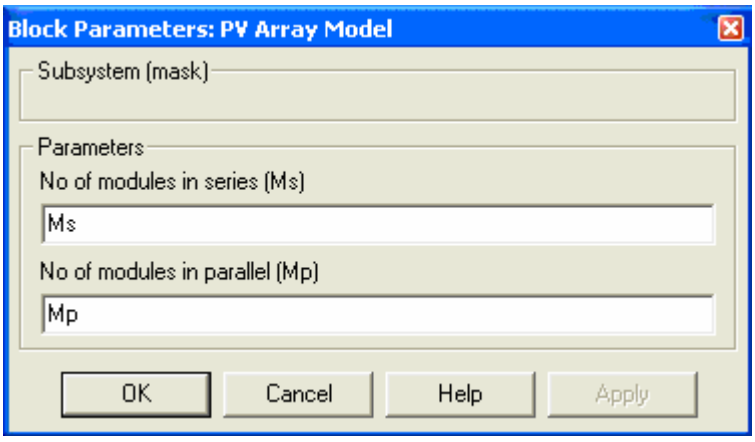


Figure 2-7 Simulink PV array model mask dialog box

2.3.2 PV Module Model

The Simulink model of a PV module is shown in the internal structure of the array model in Figure 2-6. The model is implemented using an *s-function* coded using the governing equations developed in section 2.3. An s-function is a computer language description of a Simulink block coded in MATLAB®, C++, Fortran or other supported programming languages. S-functions use a special calling syntax that enables direct interaction with Simulink equation solvers. A listing of the

module voltage s-function, which is an implementation of Equation (2-8), is given in the Appendix. The model modifying parameters are set using the dialogue box shown in Figure 2-8.

2.4 Model Validation

The model was validated using manufacturer supplied data for BP solar SX 75TU PV module. The electrical characteristics for this module are given in Table 1 [23]. The I-V characteristics of the model match the supplied data as shown in Figure 2-9. The power-voltage (P-V) curves under varying ambient temperature and solar radiation are shown in Figure 2-10. It is observed that the model provides sufficient accuracy for simulations.

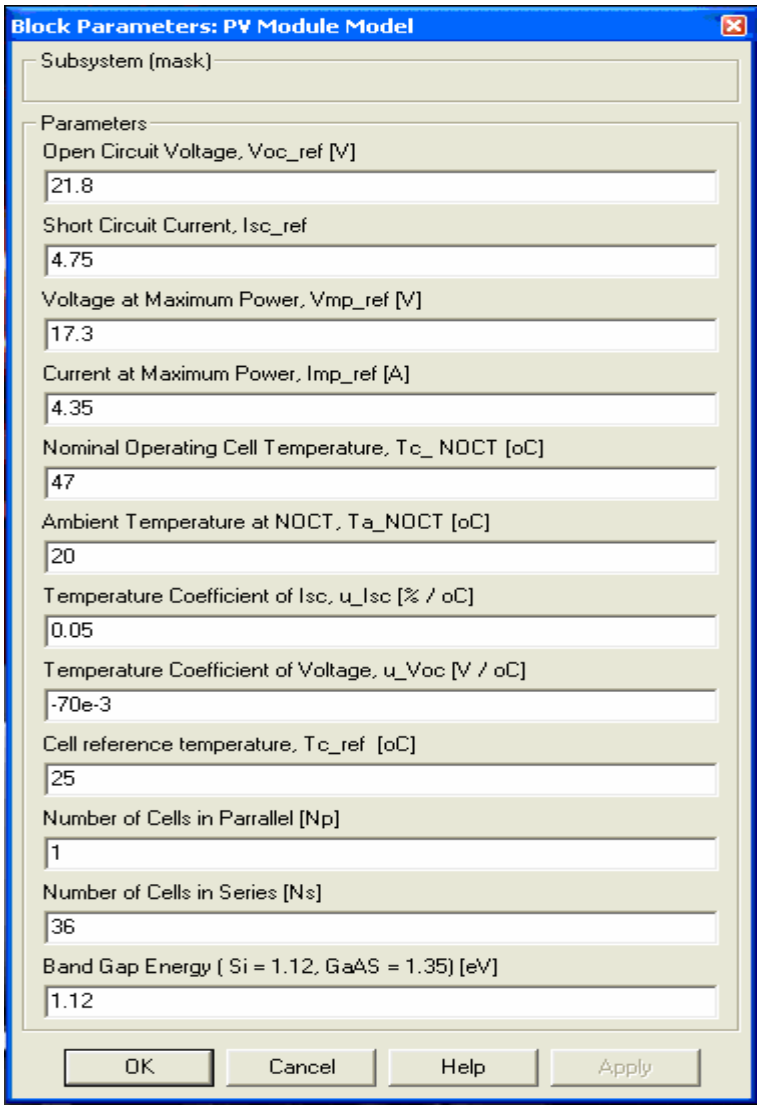
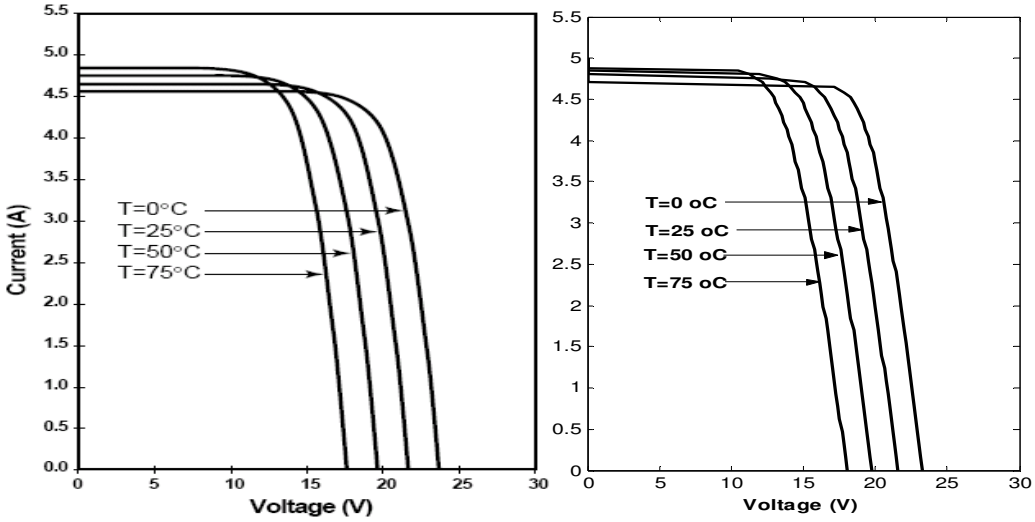


Figure 2-8 Simulink PV module model mask dialog box

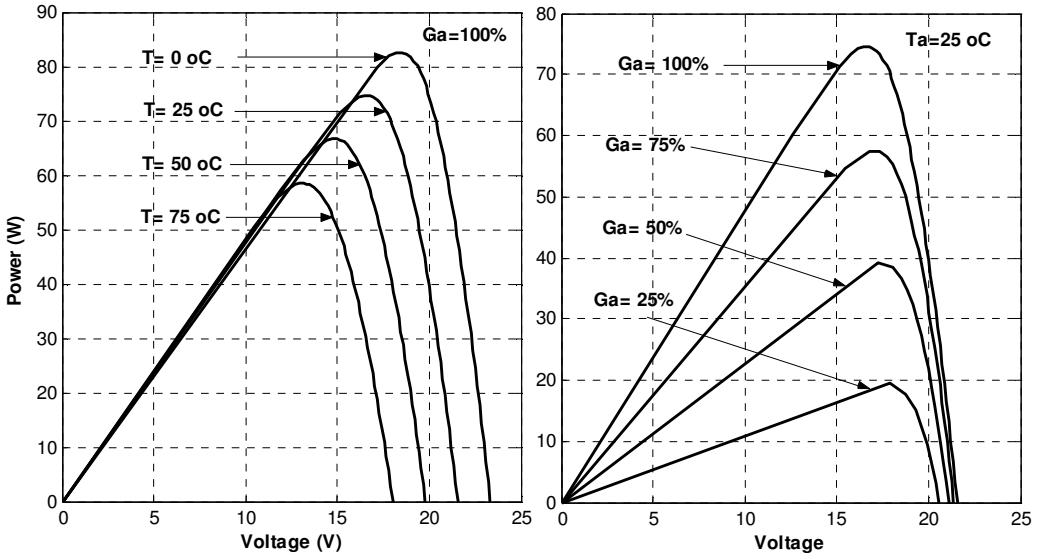
Table 2-1 Characteristics of BP SX 75TU PV module

BP SX 75TU Photovoltaic Module	
Type: Silicon Multicrystalline	
Number of Cells in series	36
Number of Cells in parallel	1
Maximum Power (P_{max})	75 W
Voltage at P_{max} (V_{mp})	17.3 V
Current at P_{max} (I_{mp})	4.35 A
Short-circuit current (I_{sc})	4.75 A
Open-circuit voltage (V_{oc})	21.8 V
Temperature co-efficient of I_{sc}	$(0.065 \pm 0.015)\% / {}^{o}C$
Temperature co-efficient of voltage	$-(80 \pm 10) mV / {}^{o}C$
Nominal Operating Cell Temperature (NOCT)	$47 \pm 2 {}^{o}C$



(a) (b)

Figure 2-9 BP Solar SX75TU PV Module I-V Curves (a) Manufacturer Supplied (b) Simulated



(a) (b)

Figure 2-10 PV module Power-Voltage curves under (a) varying temperature and (b) varying solar radiation

2.5 Conclusion

The photovoltaic module was modeled in Matlab/Simulink. The results of the simulation were compared to manufacturer supplied data and both results were found to be very close. The model is detailed enough and can be used to model any solar panel using information available in manufacturer's data sheet. The model is used in Chapter 4 to formulate a complete MPPT model.

CHAPTER THREE

3 DC-DC CONVERTERS AND MAXIMUM POWER POINT TRACKING

In this chapter the characteristics of the three basic dc-dc converter topologies; buck, boost, and buck-boost, are analyzed to determine the best topology for performing PV module maximum power point tracking. A model of the identified topology is then formulated. The model is used to carry out simulations to determine the effect of component non-idealities on converter efficiency and output voltage. The simulation results are used as the basis for developing control strategies and selecting converter components. The converter model forms the main part in the complete MPPT model used for tuning the fuzzy logic controller.

3.1 DC-DC Converters

The schematic diagram of a dc-dc converter is shown in Figure 3-1. It converts a dc input voltage $v_g(t)$, to a dc output voltage $v_o(t)$, at a different voltage level from the input [24]. It is desirable that the conversion be made with low losses in the converter. Therefore, the transistor is operated as a switch using the control signal $\delta(t)$, which is held high for a time t_{on}, and low for a time t_{off} as shown in Figure 3-2.

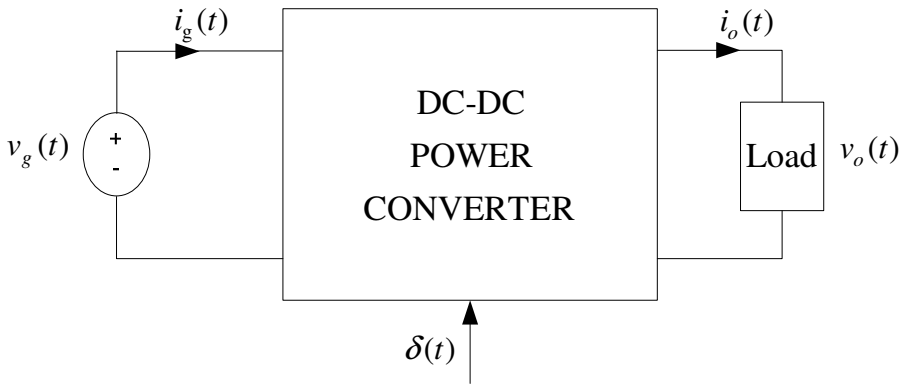


Figure 3-1 Structure of a dc-dc converter

While the transistor is on, the voltage across it is low which means that the power loss in the transistor is low. While the transistor is off, the current through it is low and the power loss is also low. The average output voltage is controlled by changing the width of the pulses while the switching period T_s is held constant. The duty cycle, $d(t)$, is a real value in the interval 0 to 1 and it is equal to the ratio of the width of a pulse to the switching period i.e. $d(t) = t_{on} / T_s$.

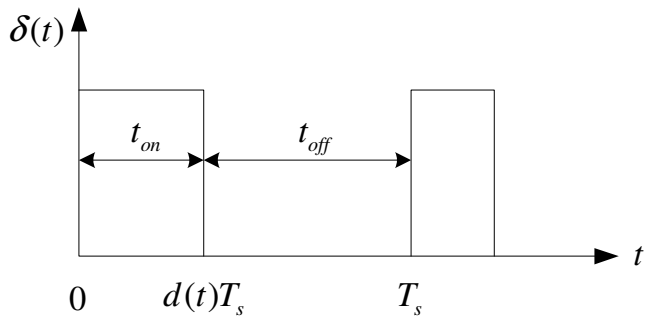


Figure 3-2 Transistor control signal, $\delta(t)$

To obtain low losses, resistors are avoided in dc-dc converters. Capacitors and inductors are used instead since ideally they have no losses. The electrical components can be combined and connected to each other in different ways, called topologies, each one having different properties. The buck, boost, and buck-boost converters are three basic converter topologies. The buck converter has an output voltage that is lower than the input voltage; the boost converter has an output voltage that is higher than the input voltage, and the buck-boost converter is able to produce an output voltage magnitude that is higher or lower than the input voltage magnitude.

3.2 Maximum Power Point Tracking

As observed in Figure 2-10, the power produced from a photovoltaic module depends strongly on the operating voltage of the load to which it is connected, as well as to the solar radiation level and cell temperature. If a variable load resistance *R*, is connected across the module's terminals, the operating point is determined by the intersection of module I-V curve and the load I-V characteristic. Figure 3-3 illustrates the operating characteristic of a PV module. It consists of two regions: Zone I is the current source region, and Zone II is the voltage source region. In Zone I, the internal impedance of the module is high, while in Zone II the internal impedance is low. The maximum power point P_{mp}, is located at the knee of the power curve. Increase in solar radiation at constant temperature causes a decrease in

internal impedance as it causes an increase in short-circuit current. An increase in temperature at constant solar radiation causes a decrease in internal impedance since it causes a decrease in open circuit voltage.

According to the maximum power transfer theory, the power delivered to the load is maximum when the source internal impedance matches the load impedance. The load characteristic is a straight line with a slope of $I/V = 1/R$. If R is small, the module operates in the region AB only and behaves like a constant current source at a value close to I_{sc}. If R is large, the module operates in the region CD behaving like a constant voltage source, at a value almost equal to V_{oc}.

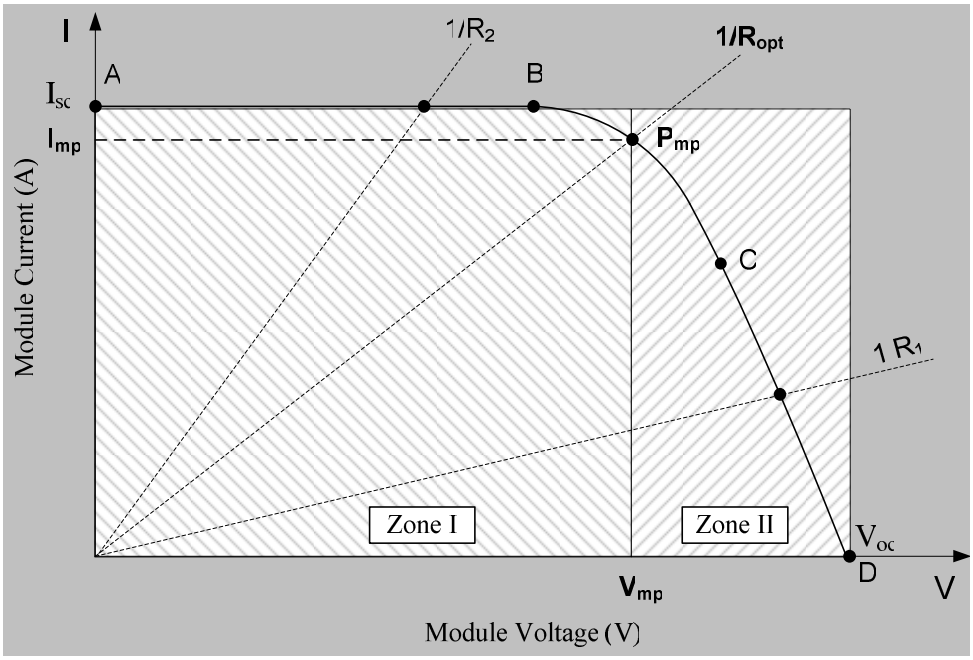


Figure 3-3 Tracking the maximum power point of a PV module

Maximum power point tracking is based on load line adjustment under varying atmospheric and load conditions by searching for an optimal equivalent output resistance R_{opt}. A dc-dc converter is used to perform load-line adjustment by varying the converter duty cycle using a controller.

3.3 Identification of a Suitable Converter Topology for Maximum Power Point Tracking

The different converter topologies are analyzed in this section in order to ascertain their performance and identify the most suitable topology for maximum power point tracking.

3.3.1 Buck Converter

For an ideal buck converter, averaged input voltage V_g, output voltage V_o, input current I_g, and output current I_o are related as follows [25]:

$$V_o = V_g D \tag{3-1}$$

$$I_o = \frac{I_g}{D} \tag{3-2}$$

where D is the equilibrium duty cycle of the converter. The dc load R connected to the converter can be expressed using Ohm's law as:

$$R = \frac{V_o}{I_o} \tag{3-3}$$

The load resistance R' referred to the input terminals of the converter can be derived from Equations (3-1) and (3-2) as:

$$R' = \frac{R_L}{D^2} \tag{3-4}$$

Since $0 < D < 1$, varying D can only increase the load seen by the source. A buck converter is therefore, only able to extract maximum power if the original load draws a higher current than the maximum power point current I_{mp}, of the PV module (Zone I of Figure 3-3).

3.3.2 Boost Converter

For an ideal boost converter, the averaged input and output values of current and voltage are related as follows [25]:

$$V_o = \frac{V_g}{1-D} \tag{3-5}$$

$$I_o = I_g (1-D) \tag{3-6}$$

The load resistance R' referred to the input side is given by:

$$R' = R(1-D)^2 \tag{3-7}$$

Since $0 < D < 1$, varying D can only decrease the load seen by the source. It is therefore noted that a tracker based on the boost converter is only able to extract maximum power if the original load draws lower current than maximum power point current I_{mp}, of the PV module (Zone II of Figure 3-3).

3.3.3 Buck-Boost Converter

For an ideal buck-boost converter, the averaged input and output values of current and voltage are related as follows [25]:

$$V_o = V_g \left(\frac{D}{1-D} \right) \tag{3-8}$$

$$I_o = I_g \left(\frac{1-D}{D} \right) \tag{3-9}$$

The load resistance R' referred to the input side is given by:

$$R' = R \left(\frac{1-D}{D} \right)^2 \tag{3-10}$$

Since $0 < D < 1$, varying D can increase or decrease the load seen by the source. The buck-boost converter is therefore able to operate both in Zones I and II.

3.3.4 Conclusion

It is noted from the analysis carried out in the preceding sub-sections that the buck-boost converter has the best performance since it is able to perform maximum power tracking in both zones I and II of Figure 3-3. A state space model of the converter is developed in the next section.

3.4 Buck-Boost Converter Model

The circuit of a buck-boost converter is shown in Figure 3-4. It consists of four basic components; transistor Q, diode T, inductor L, and capacitor C. These components are not ideal and some of the non-idealities are considered when modeling. The inductor is modeled as an ideal inductor in series with a resistance R_L. The capacitor is modeled as an ideal capacitor in series with a resistance R_C. R_L and R_C are used to model the power losses in the inductor and capacitor

respectively. The transistor has an on-state resistance, R_t while the diode has a forward voltage drop V_T. A converter can operate in two or three modes. The state space description of each mode is derived in this subsection.

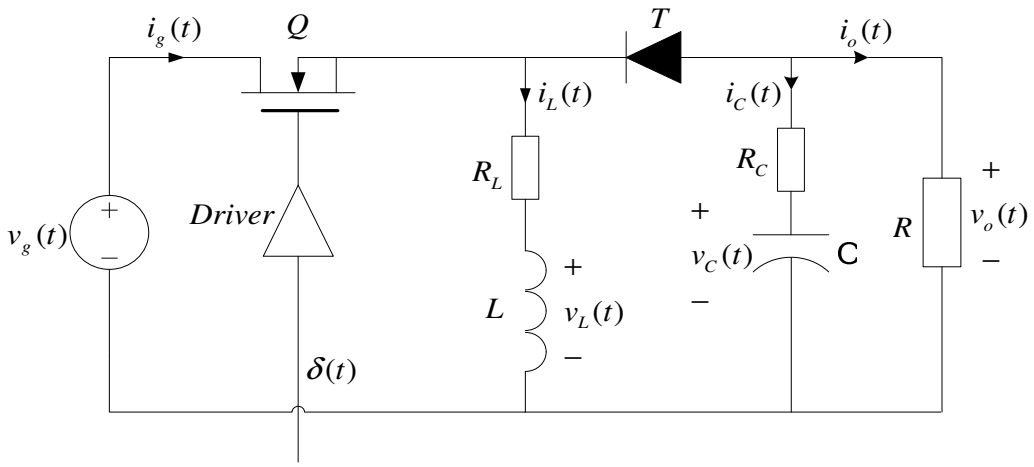


Figure 3-4 Circuit of a buck-boost converter

3.4.1 Mode 1

This mode is valid for the time interval $(0 < t < t_{on})$ when the control signal $\delta(t)$ is high. When the transistor is on, the diode is reverse biased and is not conducting. The circuit in Figure 3-5 shows the model of the buck-boost converter during t_{on}.

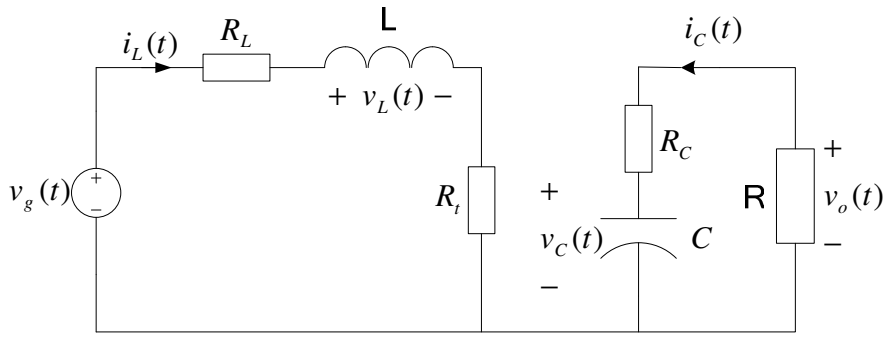


Figure 3-5 Circuit of the boost converter during t_{on}

Applying Kirchhoff's laws, the following equations are obtained from the model:

$$v_L(t) = L\frac{di_L(t)}{dt} = v_g(t) - i_L(t)(R_t + R_L) \tag{3-11}$$

$$i_C(t) = C\frac{dv_C(t)}{dt} = -\frac{v_C(t)}{R + R_C} \tag{3-12}$$

$$v_o(t) = -\frac{R}{R + R_C}v_C(t) \tag{3-13}$$

$$i_g(t) = i_L(t) \tag{3-14}$$

Re-arranging Equations (3-11) and (3-12) gives,

$$\begin{aligned}\frac{di_L(t)}{dt} &= -\frac{R_t + R_L}{L}i_L(t) + \frac{1}{L}v_g(t)\\ \frac{dv_c(t)}{dt} &= -\frac{1}{C(R + R_C)}v_C(t)\end{aligned} \tag{3-15}$$

Equations (3-13), (3-14), and (3-15) are of the form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_1\mathbf{x}(t) + \mathbf{B}_1\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_1\mathbf{x}(t) + \mathbf{E}_1\mathbf{u}(t)\end{aligned} \tag{3-16}$$

Where,

$$\mathbf{x}(t) = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix}, \tag{3-17}$$

$$\mathbf{u}(t) = \begin{bmatrix} v_g(t) \\ V_T \end{bmatrix}, \tag{3-18}$$

$$\mathbf{y}(t) = \begin{bmatrix} v_o(t) \\ i_g(t) \end{bmatrix}, \tag{3-19}$$

$$\mathbf{A}_1 = \begin{bmatrix} -\dfrac{R_t + R_L}{L} & 0 \\ 0 & \dfrac{1}{C(R + R_C)} \end{bmatrix}, \tag{3-20}$$

$$\mathbf{B}_1 = \begin{bmatrix} \dfrac{1}{L} & 0 \\ 0 & 0 \end{bmatrix}, \tag{3-21}$$

$$\mathbf{C}_1 = \begin{bmatrix} 0 & -\dfrac{R}{R + R_C} \\ 1 & 0 \end{bmatrix}, \tag{3-22}$$

$$\mathbf{E}_1 = [0]. \tag{3-23}$$

3.4.2 Mode 2

This mode is valid for the time interval $(t_{on} < t < t_{on} + \vartheta_t)$, where $t_{on} + \vartheta_t \le T_s$, and ϑ_t is the time during which the inductor current is zero. During this mode, the transistor is OFF and the diode is ON (provided $i_L(t) > 0$). The mode is valid for $(t_{on} < t < T_s)$ if $i_L(t)$ does not vanish. When the transistor is OFF, the voltage across the diode is the forward voltage drop V_T. The circuit in Figure 3-6 can therefore be

used as a model of the buck-boost converter during t_{off}, for the continuous conduction mode (CCM).

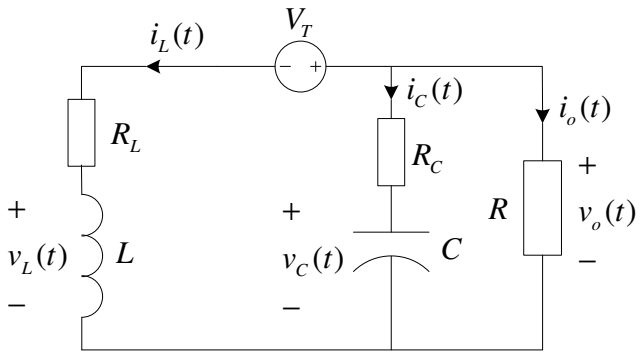


Figure 3-6 Circuit of the buck-boost converter during t_{off} (CCM)

Applying Kirchhoff's laws, the following equations are obtained from the model:

$$v_L(t) = L\frac{di_L(t)}{dt} = -R_L i_L(t) + R_C i_C(t) + v_C(t) - V_T \tag{3-24}$$

$$i_C(t) = -\frac{R}{R+R_C} i_L(t) - \frac{1}{R+R_C} v_C(t) \tag{3-25}$$

$$v_o(t) = -v_C(t) - i_C(t) R_C \tag{3-26}$$

$$i_g(t) = 0\,. \tag{3-27}$$

From Equations (3-24) to (3-27), the following state space model is obtained,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}_2\mathbf{x}(t) + \mathbf{B}_2\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_2\mathbf{x}(t) + \mathbf{E}_2\mathbf{u}(t) \end{aligned} \tag{3-28}$$

where,

$$\mathbf{A}_2 = \begin{bmatrix} -\frac{1}{L}\left(R_L + \frac{RR_C}{(R+R_C)} \right) & \frac{R}{L(R+R_C)} \\ -\frac{R}{C(R+R_C)} & -\frac{1}{C(R+R_C)} \end{bmatrix}, \tag{3-29}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 & -\frac{1}{L} \\ 0 & 0 \end{bmatrix}, \tag{3-30}$$

$$\mathbf{C}_2 = \begin{bmatrix} \frac{RR_C}{R+R_C} & -\frac{R}{R+R_C} \\ 0 & 0 \end{bmatrix}, \tag{3-31}$$

$$\mathbf{E}_2 = \mathbf{E}_1. \tag{3-32}$$

3.4.3 Mode 3

This mode is valid for the time interval $t_{on} + \vartheta_t < t \le T_s$, whenever it exists. During this mode the transistor is OFF and the diode is not in conduction (i.e. $i_L(t) = 0$). The circuit in Figure 3-7 can therefore be used as a model of the buck-boost converter during t_{off}, for the discontinuous conduction mode (DCM).

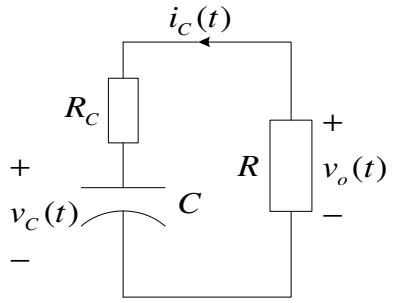


Figure 3-7 Circuit of the buck-boost converter during t_{off} (DCM)

The functioning of the circuit in this mode is given by:

$$i_L(t) = 0 \tag{3-33}$$

$$i_C(t) = C\frac{dv_C(t)}{dt} = \frac{1}{R+R_C}v_C(t) \tag{3-34}$$

$$v_o(t) = -\frac{R}{R+R_C}v_C(t) \tag{3-35}$$

$$i_g(t) = 0 \tag{3-36}$$

The corresponding standard state space model derived from Equations (3-33) to (3-36) is given by,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_3\mathbf{x}(t) + \mathbf{B}_3\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_3\mathbf{x}(t) + \mathbf{E}_3\mathbf{u}(t)\end{aligned} \tag{3-37}$$

Where,

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{C(R+R_C)} \end{bmatrix}, \tag{3-38}$$

$$\mathbf{B}_3 = [0], \tag{3-39}$$

$$\mathbf{C}_3 = \begin{bmatrix} 0 & -\dfrac{R}{R+R_C} \\ 0 & 0 \end{bmatrix}, \tag{3-40}$$

$$\mathbf{E}_3 = \mathbf{E}_1. \tag{3-41}$$

3.5 Averaged Buck-Boost Converter Model

The averaged converter model is a representation of the converter as linear time invariant system that ignores all the switching dynamics. The model is obtained by applying the method of state space averaging to the state space model.

3.5.1 Method of State Space Averaging

A converter can be considered to switch between two different linear time-invariant systems during a switching period. This is valid if the natural frequencies of the converter, and the variations of the converter inputs are much slower than the switching frequency. The converter acts as a time-invariant system while the transistor is on. While the transistor is off the converter acts as another time-invariant system and if the inductor current reaches zero, the converter acts as yet another time-invariant system. Converters therefore, switch between different time-invariant systems during each switching period. Consequently, the converter can be modeled as a time-varying system. State-space averaging [26] is one method to approximate this time-varying system with a linear continuous-time time-invariant system. This method uses the state-space description of each time-invariant system as a starting point. These state-space descriptions are then averaged with respect to their duration in the switching period. The averaged model is nonlinear and time-invariant and has the duty cycle $d(t)$, as the control signal instead of $\delta(t)$.

The state-space averaged model that describes a converter in equilibrium is given by:

$$\begin{aligned} \mathbf{0} &= \mathbf{AX} + \mathbf{BU} \\ \mathbf{Y} &= \mathbf{CX} + \mathbf{EU} \end{aligned} \tag{3-42}$$

where the averaged matrices are given by,

$$\begin{aligned} \mathbf{A} &= D_1\mathbf{A_1} + D_2\mathbf{A_2} + D_3\mathbf{A_3} \\ \mathbf{B} &= D_1\mathbf{B_1} + D_2\mathbf{B_2} + D_3\mathbf{B_3} \\ \mathbf{C} &= D_1\mathbf{C_1} + D_2\mathbf{C_2} + D_3\mathbf{C_3} \\ \mathbf{E} &= D_1\mathbf{E_1} + D_2\mathbf{E_2} + D_3\mathbf{E_3} \end{aligned} \tag{3-43}$$

D_1, D_2 and D_3 are the fraction of the switching period in which the converter operates in modes 1, 2, and 3 respectively. The equilibrium dc components are,

$$\mathbf{X} = \begin{bmatrix} I_L \\ V_C \end{bmatrix} \tag{3-44}$$

$$\mathbf{U} = \begin{bmatrix} V_g \\ V_T \end{bmatrix} \tag{3-45}$$

$$\mathbf{Y} = \begin{bmatrix} V_o \\ I_g \end{bmatrix} \tag{3-46}$$

Quantities in Equation (3-44) to (3-46) represent the equilibrium values of the averaged vectors; I_L is the inductor current, V_C is the capacitor voltage, V_g is the input voltage, V_T is the diode forward conduction voltage drop, V_o is the output voltage, and I_g is the input current. Equation (3-42) can be solved to find the equilibrium state and output vectors as:

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}\mathbf{U}$$
$$\mathbf{Y} = (-\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{E})\mathbf{U} \tag{3-47}$$

Equation (3-47) gives the converter averaged dc model.

The principle of state-space averaging is applied to the buck-boost converter model to obtain the averaged dc model. The converter is assumed to operate in the continuous conduction mode and the notations for the duty cycle are then redefined as,

$$\begin{aligned} D &= D_1, \\ D_2 &= 1 - D = D' \\ D_3 &= 0. \end{aligned} \tag{3-48}$$

3.5.2 Converter Averaged Model

Applying Equations (3-43) and (3-48) to the model equations in sections 3.4.1 and 3.4.2 gives:

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{L}\left(R_L + DRt + \frac{D'RR_C}{(R+R_C)}\right) & \frac{D'R}{L(R+R_C)} \\ -\frac{D'R}{C(R+R_C)} & -\frac{1}{C(R+R_C)} \end{bmatrix}, \tag{3-49}$$

$$\mathbf{B} = \begin{bmatrix} \frac{D}{L} & -\frac{D'}{L} \\ 0 & 0 \end{bmatrix}, \tag{3-50}$$

$$\mathbf{C} = \begin{bmatrix} \frac{D'RR_C}{R+R_C} & -\frac{R}{R+R_C} \\ D & 0 \end{bmatrix}, \tag{3-51}$$

$$\mathbf{E} = 0. \tag{3-52}$$

The dc equations are derived by expanding Equation (3-42) to obtain:

$$0 = -\frac{1}{L}\left(DR_t + R_L + \frac{D'RR_C}{R+R_C} \right) I_L + \frac{D'R}{L(R+R_C)} V_C + \frac{D}{L} V_g - \frac{D'}{L} V_D \tag{3-53}$$

$$0 = -\frac{D'R}{C(R+R_C)} I_L - \frac{1}{C(R+R_C)} V_C \tag{3-54}$$

$$V_o = \frac{D'RR_C}{R+R_C} I_L - \frac{R}{R+R_C} V_C \tag{3-55}$$

$$I_g = DI_L \tag{3-56}$$

Equation (3-54) is simplified to:

$$V_C = -D'I_L R \tag{3-57}$$

Substituting Equation (3-57) into (3-55) and simplifying, gives:

$$V_o = -V_C \tag{3-58}$$

Substituting Equations (3-57) and (3-58) into (3-53) and rearranging gives:

$$\frac{V_o}{V_g} = -\frac{D}{D'} \frac{\left(1 - \frac{D'}{D}\frac{V_D}{V_g} \right)}{\left(\frac{R_L + DR_t}{D'^2 R} \right) + \frac{1}{D'^2\left(1 + \frac{R}{R_C} \right)} + \frac{1}{\left(1 + \frac{R_C}{R} \right)}} \tag{3-59}$$

Assuming all components are ideal, i.e. $R_t = R_L = V_T = 0$, Equation (3-59) reduces to:

$$\frac{V_o}{V_g} = -\frac{D}{D'} \frac{1}{\left(1 + \frac{R_L}{D'^2 R} \right)} \tag{3-60}$$

The converter efficiency considering the average power is given by:

$$\eta = \frac{V_o I_o}{V_g I_g} = \frac{\left(1 - \frac{D'}{D}\frac{V_D}{V_g}\right)}{\left(\frac{R_L + DR_t}{D'^2 R}\right) + \frac{1}{D'^2\left(1 + \frac{R}{R_C}\right)} + \frac{1}{\left(1 + \frac{R_C}{R}\right)}} \tag{3-61}$$

Assuming $R_t = R_L = V_T = 0$, Equation (3-61) reduces to:

$$\eta = \frac{1}{1 + \frac{R_L}{D'^2 R}} \tag{3-62}$$

3.6 Buck-Boost Converter Simulink Model

The averaged converter model is implemented using the Simulink block shown in Figure 3-8. The state space averaged model given by Equations (3-49) to (3-52) is coded as an s-function whose parameters are modified using the mask dialog box shown in Figure 3-9. The model inputs are the input voltage V_g, load resistance R, and the duty cycle D, while the outputs are the load voltage V_o, inductor current I_L, input current I_g, input power P_{in}, and output power P_o.

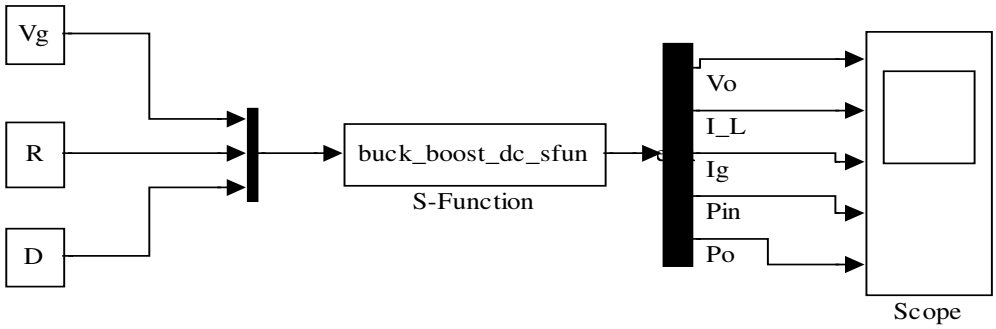


Figure 3-8 Buck-boost converter Simulink model

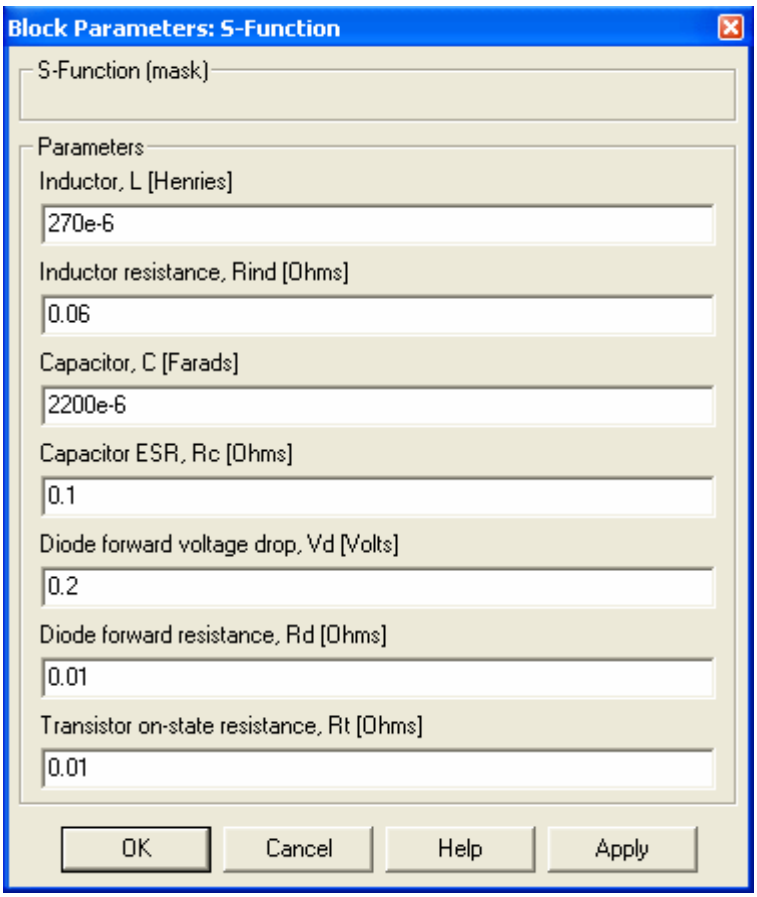


Figure 3-9 Buck-boost converter model mask dialog box

3.7 Simulation Results

Equation (3-60) is plotted in Figure 3-10 for several values of R_L / R to show the effect of inductor resistance on the output voltage. It can be seen that Equation (3-60) contains two terms. The first term, $-D / D'$, is the ideal conversion ratio with $R_L = 0$. The second term, $1/(1 + R_L / D'^2 R)$, describes the effect of the

inductor winding resistance. If R_L is much less than D'^2R, then the second term tends to unity and the conversion ratio is approximately equal to the ideal value $-D/D'$. However, increase in R_L in relation to D'^2R leads to decrease in the second term which decreases V_o/V_g as well.

As the duty cycle D approaches unity, the inductor winding resistance causes a significant effect in the V_o/V_g curve. Rather than approaching infinity at $D=1$, the curve tends to zero. This condition is expected since at $D=1$ the inductor is never connected to the output, hence no energy is transferred to the output and the output voltage tends to zero. The inductor current increases rapidly, limited only by the inductor resistance R_L. A large amount of power is lost in the inductor winding resistance while no power is delivered to the load. The converter efficiency therefore tends to zero as D approaches unity.

Equation (3-62) is plotted in Figure 3-11 for several values of R_L/R to illustrate the effect of inductor resistance on converter efficiency. It can be seen from Equation (3-62) that to obtain high efficiency, R_L should be much smaller than D'^2R. This is easier to attain at low duty cycle, where D' is close to unity, than at high duty cycle where D' approaches zero. It is observed in Figure 3-11 that the efficiency is high at low duty cycles, but decreases rapidly towards zero near $D=1$.

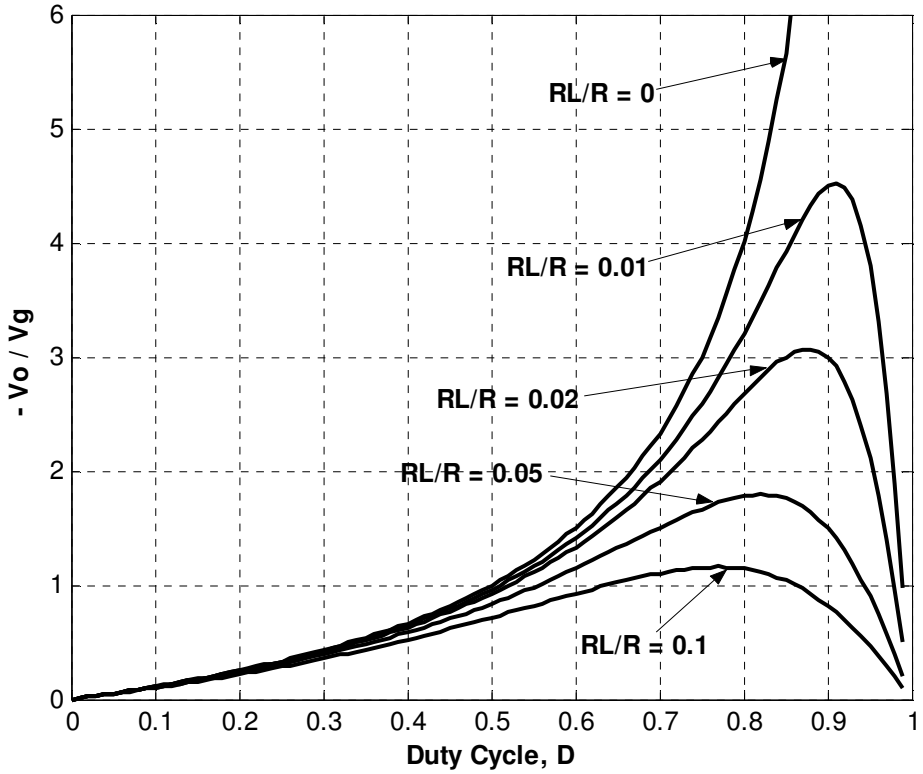


Figure 3-10 Effect of inductor resistance on buck-boost converter output voltage

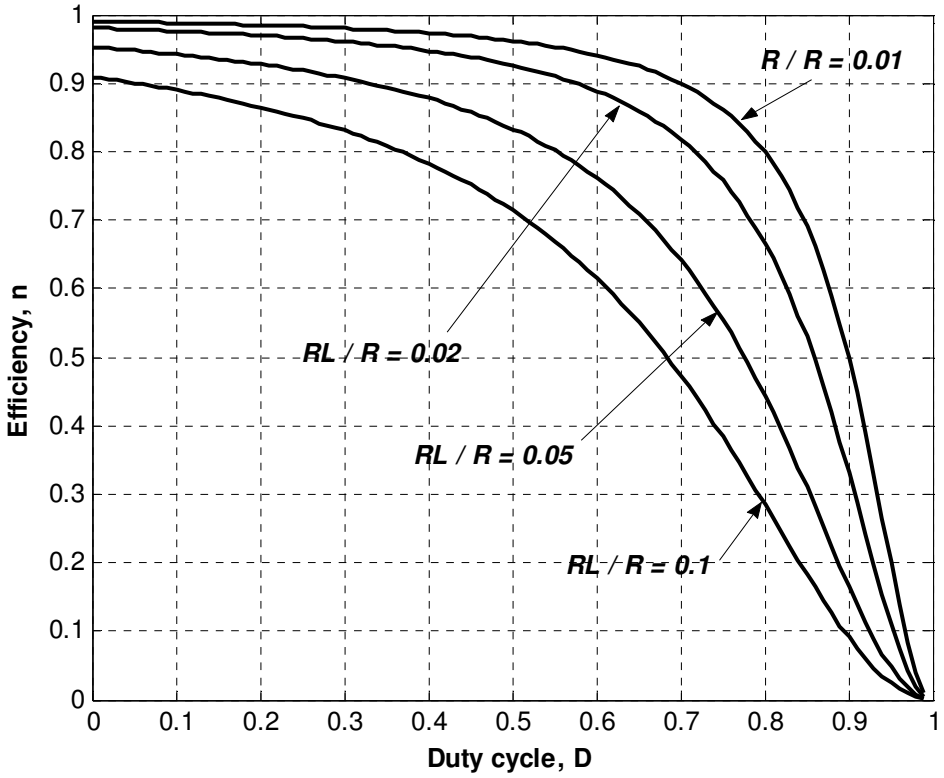


Figure 3-11 Effect of inductor resistance on buck-boost converter efficiency

3.8 Conclusion

The application of dc-dc converters in PV module maximum power point tracking has been studied in this chapter. The buck-boost converter was identified as the most suitable converter topology for maximum power tracking and a model of the converter developed. The effects of non-idealities on converter efficiency is considered in choosing converter components and setting duty cycle limits for the controller. The model is used in the next chapter to tune and test the performance of the fuzzy logic algorithm.

CHAPTER FOUR

4 FUZZY LOGIC CONTROLLER DESIGN AND HARDWARE IMPLEMENTATION

This chapter presents a study on the application of fuzzy logic to formulate a control algorithm for a maximum power point tracker. The design of the fuzzy logic controller structure and formulation of the control rules is presented. A hardware implementation of the maximum power point tracker is also carried out.

4.1 Design of Fuzzy Logic Controller Parameters

4.1.1 Controller Structure

The basic scheme of a fuzzy logic based maximum power point tracker is shown in Figure 4-1. The dc-dc converter is represented by a "black box" from which only the terminals corresponding to input voltage V_m, input current I_m from the PV module, and the controlled switch S are extracted. As observed, only two state variables are sensed; the input voltage and input current. The two values are used to calculate the input power. From these measurements, the fuzzy logic controller provides a signal proportional to the converter duty cycle which is then applied to the converter through a pulse width modulator. The modulator uses the value of D to perform Pulse Width Modulation (PWM), which generates the

control signals for the converter switch. The fuzzy logic controller scheme is a closed loop system.

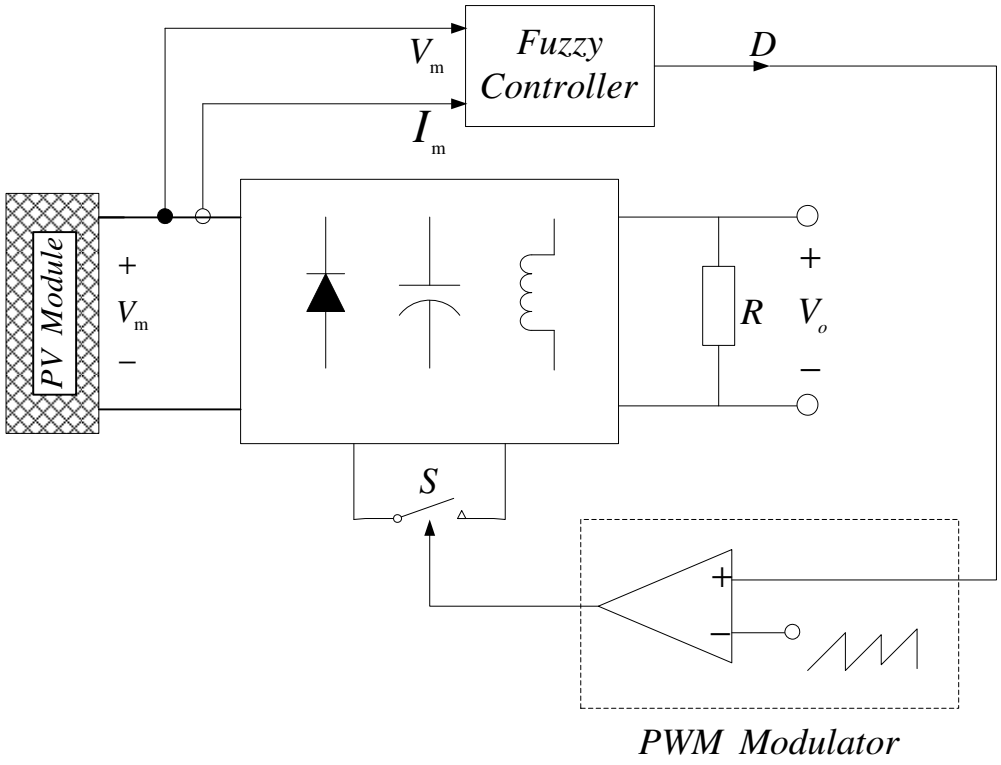


Figure 4-1 Fuzzy control scheme for a maximum power point tracker

A functional block diagram representation of the fuzzy controller is shown in Figure 4-2. The inputs to the controller are the scaled change of power, $k_p \Delta P_k$ and the previous change of duty, ΔD_{k-1}, where k_p is a power scale factor, and k is the sampling instant. The output of the fuzzy controller is the duty cycle D_k at the k-th sampling instant, and is defined as

$$D_k = D_{k-1} + k_d \Delta D_k \tag{4-1}$$

where ΔD_k is the inferred change of duty cycle by the fuzzy controller at the *k-th* sampling instant, and k_d is a duty-cycle scale factor. The block containing the term Z^{-1} in Figure 4-2 indicates a unit time delay.

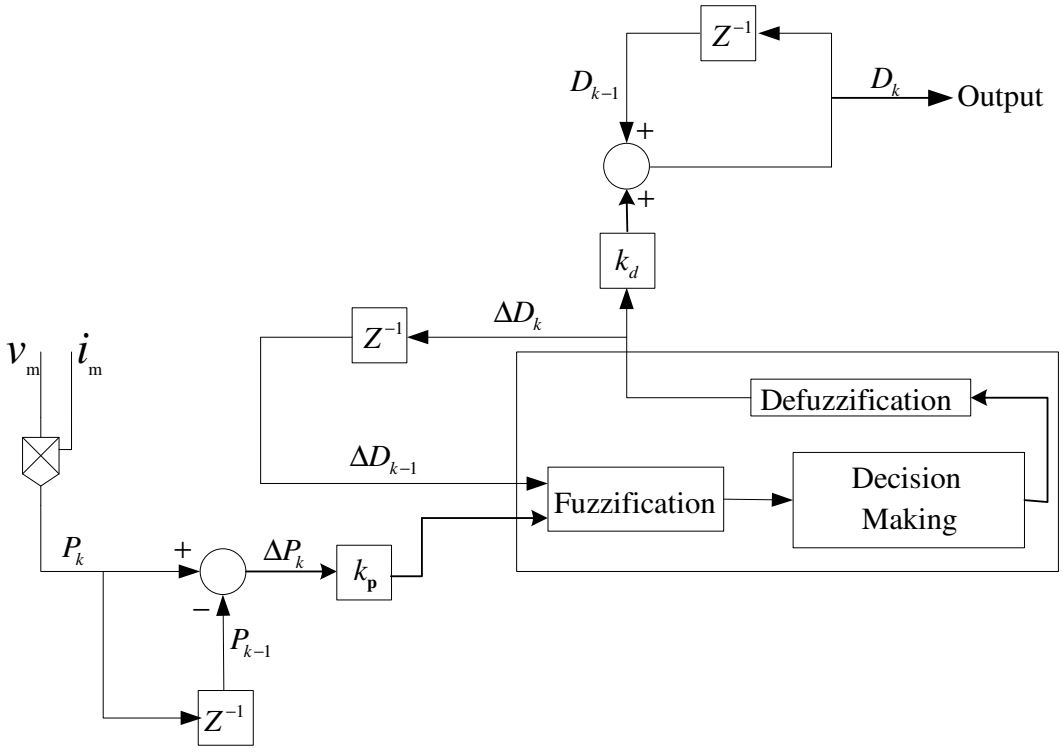


Figure 4-2 Functional block of the fuzzy controller

4.1.2 Membership Functions

Fuzzy sets for each input and output variable are defined as shown in Figure 4-3. Five fuzzy subsets *Negative Big* (NB), *Negative Small* (NS), *Zero* (ZE), *Positive Small* (PS), and *Positive Big* (PB) are chosen for the input variable ΔP_k. Eleven subsets are used for the input and output variable ΔD_{k-1}. The subsets are NB,

Negative Medium (NM), *Negative Medium Medium* (NMM), NS, *Negative Small Small* (NSS), ZE, *Positive Small Small* (PSS), PS, *Positive Medium Medium* (PMM), *Positive Medium* (PM), and PB. Eleven fuzzy subsets were chosen for ΔD_{k-1} in order to smooth the control action. As shown in Figure 4-3, triangular and trapezoidal shapes have been adopted for the membership functions; the value of each input and output variable is normalized. The same membership function is used for the output value ΔD_k and the input value ΔD_{k-1}.

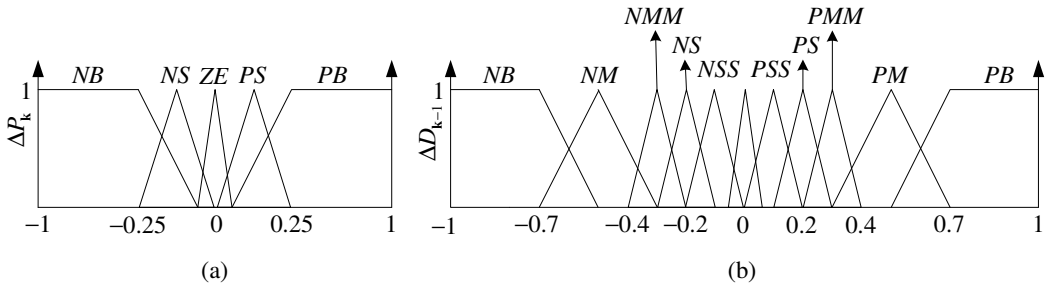


Figure 4-3 Membership functions for (a) change in power (b) change in duty cycle

The membership functions for ΔP_k and ΔD_{k-1} are made more dense at the centre in order to provide more sensitivity as the variation of power approaches zero. The duty cycle is internally limited to a maximum value of 90% to prevent operation at low efficiencies. It is also limited to a minimum value of 10% to ensure that the converter switching process does not stop as operation at $D = 0$ will indicate a false maximum power point.

4.1.3 Scaling Factors

For simplicity, the universe of discourse for each fuzzy variable was normalized to be in the range [-1 1]; this procedure involves scale mapping for the input and output data. The choice of the scale factors k_p and k_d greatly affects the bandwidth and the overall performance of the controller. The factor k_p determines the sensitivity of the controller to changes in power, and k_d determines the sensitivity to change in duty cycle. Suitable values of k_p and k_d were chosen based on simulation results.

4.1.4 Derivation of Control Rules

Fuzzy control rules are obtained from the analysis of the system behaviour. The different operating conditions are considered in order to improve tracking performance in terms of dynamic response and robustness.

The algorithm can be explained as follows: the tracking process is started with an initial duty cycle, D_0. The converter input current I_m, and voltage V_m, are then measured and used to compute the module power P_k. Then, the duty cycle is increased by the controller based on the initial changes in power and duty cycle. At stage two, I_m and V_m are measured and used to compute P_{k+1}. After gathering the past and present information of the module power, the controller makes a decision on whether to increase or decrease the duty cycle. This tracking process repeats

itself continuously until the peak power point is reached. The control rules are divided into four categories.

Category I

These rules are used to guide the controller under constant operating conditions i.e. when there are no variations in solar radiation, temperature, or load. Operation is based on the meta-rule:

*"IF the **last change in the duty cycle** has caused the **power to increase**, keep moving the **duty cycle** in the **same direction**; otherwise if it has caused **power to decrease**, move it in the **opposite direction**."*

The duty cycle is changed in adaptive steps which depend on the change in power to ensure the maximum power point is approached at a fast speed and to prevent oscillations around it. There are forty rules in this category and they are shown in Table 4-1 with a degree of support of 1. These rules are given the strongest degree of support since the task they represent describes the normal system operation. The rules are read as,

- *IF* ΔP_k *is* ***NB*** *AND* ΔD_{k-1} *is* ***NB*** *THEN* ΔD_k *is* ***PM***
- *IF* ΔP_k *is* ***NB*** *AND* ΔD_{k-1} *is* ***NM*** *THEN* ΔD_k *is* ***PMM***
- *IF* ΔP_k *is* ***NB*** *AND* ΔD_{k-1} *is* ***NS*** *THEN* ΔD_k *is* ***PSS***
- ……..
- *IF* ΔP_k *is* ***PB*** *AND* ΔD_{k-1} *is* ***PB*** *THEN* ΔD_k *is* ***PM***.

Table 4-1 Fuzzy controller rule base

ΔP_k \ ΔD_{k-1}	NB	NM	NMM	NS	NSS	ZE	PSS	PS	PMM	PM	PB
NB	PM	PMM	PS	PSS	PSS	NB	NSS	NSS	NS	NMM	NM
NS	PM	PMM	PS	PSS	PSS	NS	NSS	NSS	NS	NMM	NM
ZE	NB	NM	NMM	NS	NSS	ZE	PSS	PM	PMM	PM	PB
PS	NM	NMM	NS	NSS	NSS	PS	PSS	PSS	PS	PMM	PM
PB	NM	NMM	NS	NSS	ZE	PS	PSS	PSS	PS	PMM	PM

Key: degree of support = 1.00
= 0.50
= 0.25

Category II

The rules in this category guide the controller when there are sudden changes in solar radiation, temperature, or load leading to an overall shift in the optimum point. There are four rules in this category and are found in the vertical column of Table 4-1 corresponding to $\Delta D_{k-1} = 0$, and a degree of support of 0.5. This condition is rare and the rules are used to return the system to normal operation where category I rules are activated to search for the new optimum operating point.

Category III

The rules in this group are used to ensure that the maximum power transfer search only stops when the true maxima has been reached. Several false maxima are

introduced due to the quantization effect shown in Figure 4-4. Since the input signals are digitized, the continuous curve is broken into a series of plateaus (points with constant power). It is observed from Figure 4-4 that the steeper the curve, the shorter the plateau. Since the optimum point satisfies the condition $\partial P / \partial D = 0$, the controller might recognize any large plateau as a maximum power point and stop. There are ten rules in this category and are found in the horizontal column of Table 4-1 corresponding to $\Delta P_k = 0$. The rules are given a degree of support of 0.5 since the condition is rare.

Category IV

There is only one rule in this category. The rule is activated when the system reaches the optimum point and it is used to stabilize operation at the maximum power point. It is given a weight of 0.25 as shown in Table 4-1. The rule is read as,

- *IF* ΔP_k *is* ***ZE*** *AND* ΔD_{k-1} *is* ***ZE*** *THEN* ΔD_k *is* ***ZE***

Figure 4-5 shows the fuzzy controller rule surface. It is a graphical representation of the rule base. In this figure, $cD(k-1)$ is the previous change in duty cycle, $cP(k)$ is the current change in module power, and $cD(k)$ is the corresponding change in duty cycle.

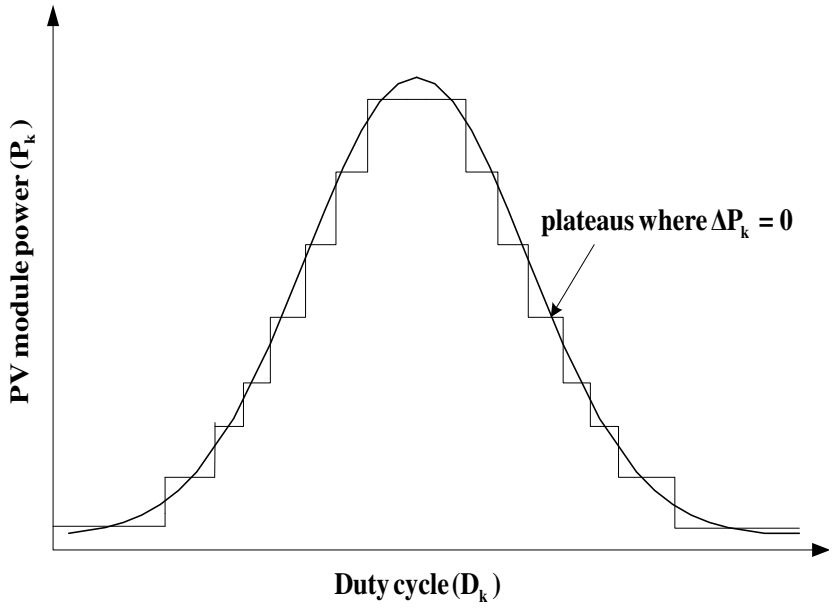


Figure 4-4 Quantization effect during maximum power search

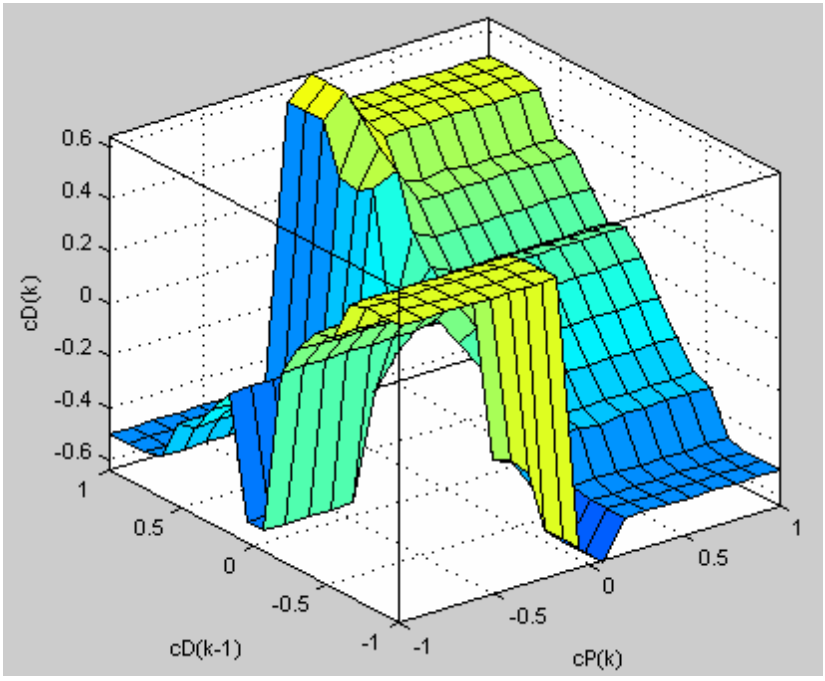


Figure 4-5 Graphical representation of Table 4-1

4.1.5 Tuning of Control Rules

A system model of the block diagram shown in Figure 4-1 was formulated in the Simulink/Matlab environment using the models of the PV module and the buck-boost converter formulated in Chapters 2 and 3 respectively. The fuzzy logic algorithm was formulated using the fuzzy logic toolbox and the rules were fine tuned by simulations. This involved the choice of the number of rules, setting membership function boundaries, and determining suitable values for the scale factors k_p and k_d.

4.1.6 Decision Making

From the membership functions in Figure 4-3, it is observed that every input and output belongs to at most two fuzzy sets. A maximum of four rules are therefore activated at any sampling instant during which the input signals ΔP_k and $\Delta D_{k=1}$ are processed. For instance, let $\Delta P_k = 0.2$ and $\Delta D_{k-1} = -0.35$. From Figure 4-5, change in power ΔP_k belongs to the fuzzy set PS and PB with a degree of membership $\mu_{PS}(\Delta P_k) = 0.33$, and $\mu_{PB}(\Delta P_k) = 0.71$. Change in duty cycle ΔD_{k-1} belongs to the fuzzy set NMM and NM with $\mu_{NMM}(\Delta D_{k-1}) = 0.5$, and $\mu_{NM}(\Delta D_{k-1}) = 0.25$. The degree of membership for the other membership functions is zero. Therefore, the following four rules are activated,

- *Rule 1: IF* ΔP_k *is* ***PS*** *AND* ΔD_{k-1} *is* ***NM*** *THEN* ΔD_k *is* ***NMM***

- *Rule 2: IF* ΔP_k *is* ***PS*** *AND* ΔD_{k-1} *is* ***NMM*** *THEN* ΔD_k *is* ***NS***
- *Rule 3: IF* ΔP_k *is* ***PB*** *AND* ΔD_{k-1} *is* ***NM*** *THEN* ΔD_k *is* ***NMM***
- *Rule 4: IF* ΔP_k *is* ***PB*** *AND* ΔD_{k-1} *is* ***NMM*** *THEN* ΔD_k *is* ***NS***.

The inference result of each rule consists of two parts, the degree of fulfillment, DOF_i of the individual rule, and the weighting factor w_i, according to the rule. The degree of fulfillment (DOF) is obtained by means of Mamdani's min-fuzzy implication [15] and w_i is retrieved from the control rule table. The degree of fulfillment of each rule using Mamdani's min fuzzy implication is given by,

$$DOF_i = \min\{\mu_{\Delta P}(\Delta P_k), \mu_{\Delta D_{k-1}}(\Delta D_{k-1})\} \tag{4-2}$$

and the output of each rule is given by,

$$z_i = (DOF_i)w_i \tag{4-3}$$

where z_i denotes the fuzzy representation of change in duty cycle inferred from the *i-th* rule. Since the inferred output is fuzzy, the defuzzification operation is performed to obtain a crisp output.

The fuzzy inference system with the Mamdani's min fuzzy implication method for inputs $\Delta P_k = 0.2$ and $\Delta D_{k-1} = -0.35$, is illustrated in Figure 4-6. The degree of fulfillment of *Rule* 1 is given by:

$$DOF_1 = \mu_{PS}(\Delta P_k) \wedge \mu_{NM}(\Delta D_{k-1}) = 0.33 \wedge 0.25 = 0.25 \tag{4-4}$$

where $\wedge$ = minimum (AND) operator. The rule output z_1, is given by the truncated membership function $\overline{NMM}$. Similarly, degrees of fulfillment for *Rules* 2, 3 and 4 are evaluated using Equation 4-5 to give:

$$DOF_2 = 0.33 \tag{4-5}$$

$$DOF_3 = 0.25 \tag{4-6}$$

$$DOF_4 = 0.5 \tag{4-7}$$

The corresponding outputs for *Rules* 2, 3 and 4 are the truncated membership functions $\overline{NS}$, $\overline{NMM}$ and $\overline{NS}$ respectively, as shown in Figure 4-6. The total fuzzy output is the union of all the component membership functions and is given by,

$$\mu_{out}(Z) = \mu_{\overline{NMM}}(Z) \vee \mu_{\overline{NS}}(Z) \vee \mu_{\overline{NMM}}(Z) \vee \mu_{\overline{NS}}(Z) \tag{4-8}$$

where $\vee$ = maximum (OR) operator. The fuzzy output $\mu_{out}(Z)$ is shown in Figure 4-6.

4.1.7 Defuzzification

The output of an inference process is a fuzzy set specifying a distribution space of fuzzy control actions defined over an output universe of discourse. Defuzzification is the conversion of this fuzzy output to crisp output suitable for a control action. A defuzzification strategy is aimed at producing a non-fuzzy control action that best represents the possibility distribution of an inferred fuzzy control action. Unfortunately, there is no systematic procedure for choosing a defuzzification strategy [18]. This process involves the operation:

$$Z_0 = defuzzifier(Z) \tag{4-9}$$

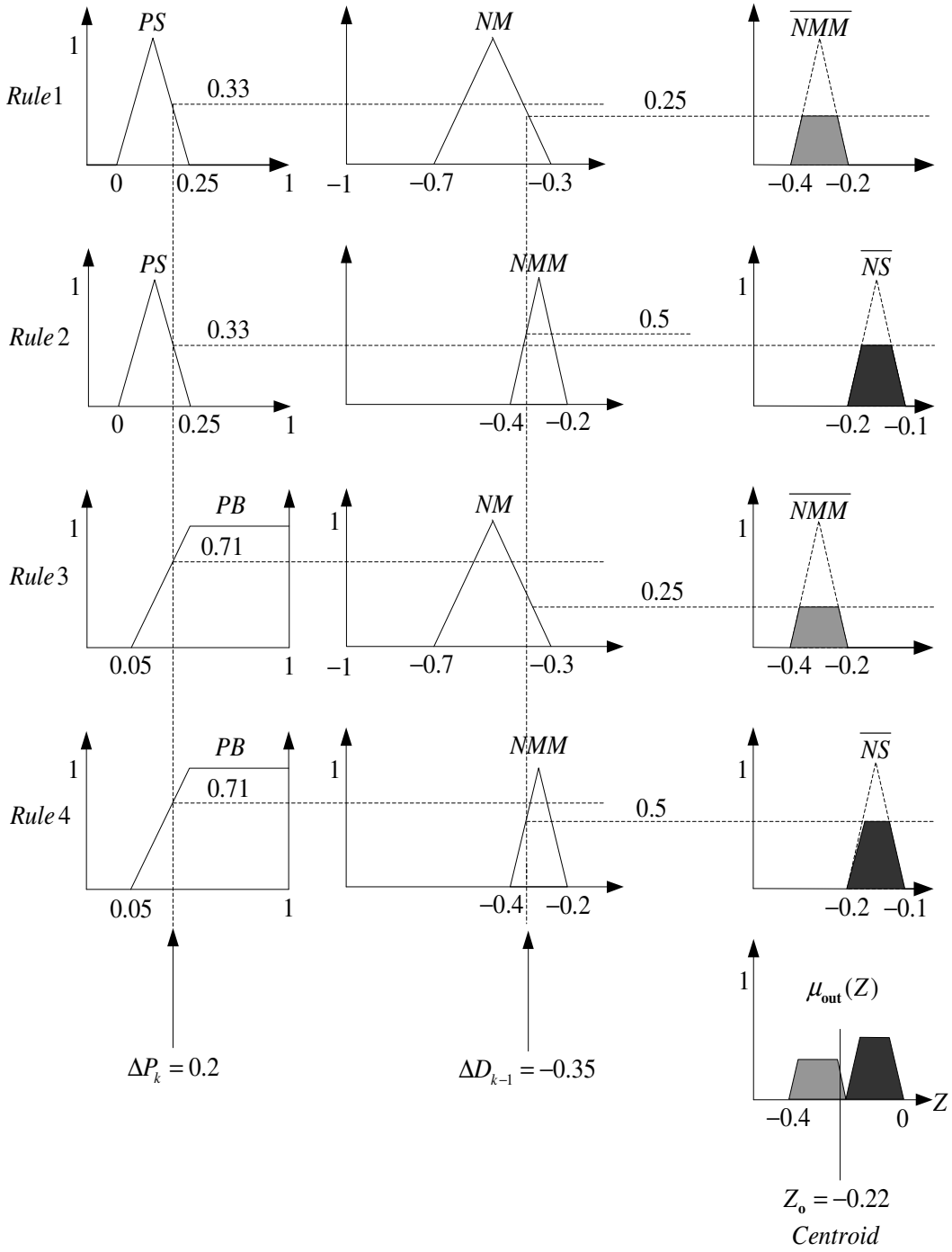


Figure 4-6 Fuzzy inferencing and defuzzification using Mamdani method

The term Z_0 in Equation (4-9) is the non-fuzzy control output and *defuzzifier* is the defuzzification operator. There are various defuzzification methods which include, centre of area (COA), bisector, mean of maxima, sum of maxima, etc. The COA method commonly known as centre of gravity (or centroid) is used in this work. The method is computationally intensive but more accurate than other methods. In this method, the crisp output Z_0 of the Z variable is taken to be the geometric centre of the output fuzzy value $\mu_{out}(Z)$, where $\mu_{out}(Z)$ is formed by taking the union of all the contributions of rules whose $DOF > 0$ [15] as shown in Figure 4-6. The general expression for the COA method in the case of a discretized universe of discourse is given by,

$$Z_0 = \frac{\sum_{i=1}^{n} Z_i \mu_{out}(Z_i)}{\sum_{i=1}^{n} \mu_{out}(Z_i)}. \tag{4-10}$$

A crisp value for the change in duty cycle is calculated by applying Equation (4-10) to the output fuzzy value $\mu_{out}(Z)$ in Figure 4-6.

4.2 Simulation Model

The fuzzy MPPT algorithm was simulated by implementing a complete system model in Matlab/Simulink as shown in Figure 4-7. The system is composed of a PV source, buck-boost converter, and a fuzzy logic controller block.

4.2.1 PV Source

This block simulates the nonlinear I-V characteristic of a PV module using the model developed in Chapter 2. BP solar SX 75TU PV module was used for the simulation. This solar panel has a rated power of $75W$. The inputs for this block are the ambient solar radiation G_a , ambient temperature T_a , and the module current I_m . The outputs are the cell temperature T_c and module voltage V_m which are recorded using *scope 2*.

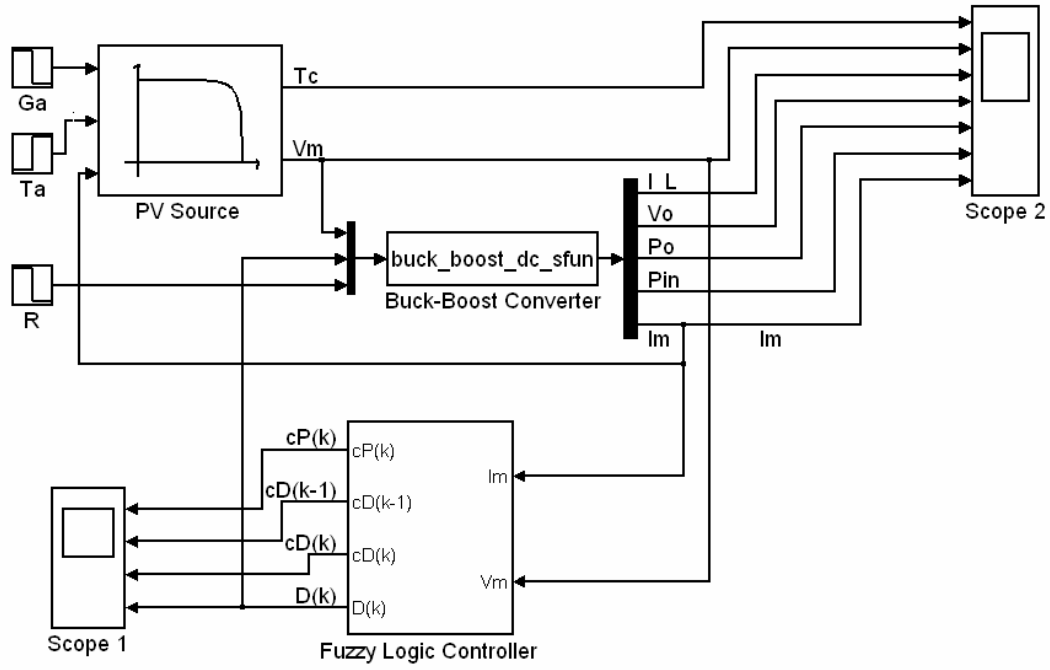


Figure 4-7 Fuzzy MPPT Simulink Simulation Model

4.2.2 Fuzzy Logic Controller

This model simulates the fuzzy MPPT algorithm. It computes the desired duty cycle of the buck-boost converter using the procedure shown in Figure 4-2. This block performs the fuzzification, decision making and defuzzification processes. The inputs are the PV module voltage and current, while the output is the converter duty cycle. All controller parameters i.e. the current change in power $cP(k)$, previous change in duty cycle $cD(k-1)$, current change in duty cycle $cD(k)$, and the current value of duty cycle $D(k)$ are recorded by *scope 1*. $cP(k)$ and $cD(k-1)$ are used by the controller to determine $cD(k)$. The duty cycle $D(k)$ used to drive the converter is determined by updating the previous value using $cD(k)$.

4.2.3 Buck-Boost Converter

This block simulates the buck-boost converter characteristics using a Simulink s-function, *buck_ boost_dc_ sfun*. The s-function was coded using the model equations of the buck-boost converter developed in Chapter 3. The Matlab code for the *buck_ boost_dc_ sfun* is listed in the Appendix. The use of an s-function greatly improves simulation speed compared to a converter model developed using discrete components available in the Simulink *SimPowerSystems* toolbox. The s-function model also allows simulations of step changes in load which cannot be easily achieved in a discrete component model. The inputs for

this block are; the module voltage V_m, duty cycle D_k, and load resistance R. The outputs are the input current I_m, input power P_{in}, inductor current I_L, output voltage V_o, and output power P_o. All the output parameters are recorded using *scope 2*.

4.3 Hardware Design Overview

Maximum Power Point Tracker design involves the selection of dc-dc converter components and implementation of the control algorithm. In this design, the buck-boost converter topology is chosen in order to maximize the efficiency of the PV array. The selection of the converter topology was carried out in Chapter 3. A schematic diagram of the maximum power point tracker is shown in Figure 4-8. Implementation of the control algorithm involves the design of current and voltage sensing circuits and formulation of the control software. The controller senses analogue voltages and currents proportional to measured quantities. The inputs are digitized and processed in a micro-controller in order to determine, and output a suitable duty cycle that ensures the system operates at its maximum power point. USB-1208FS data acquisition (DAQ) card manufactured by Measurement Computing Corporation is used.

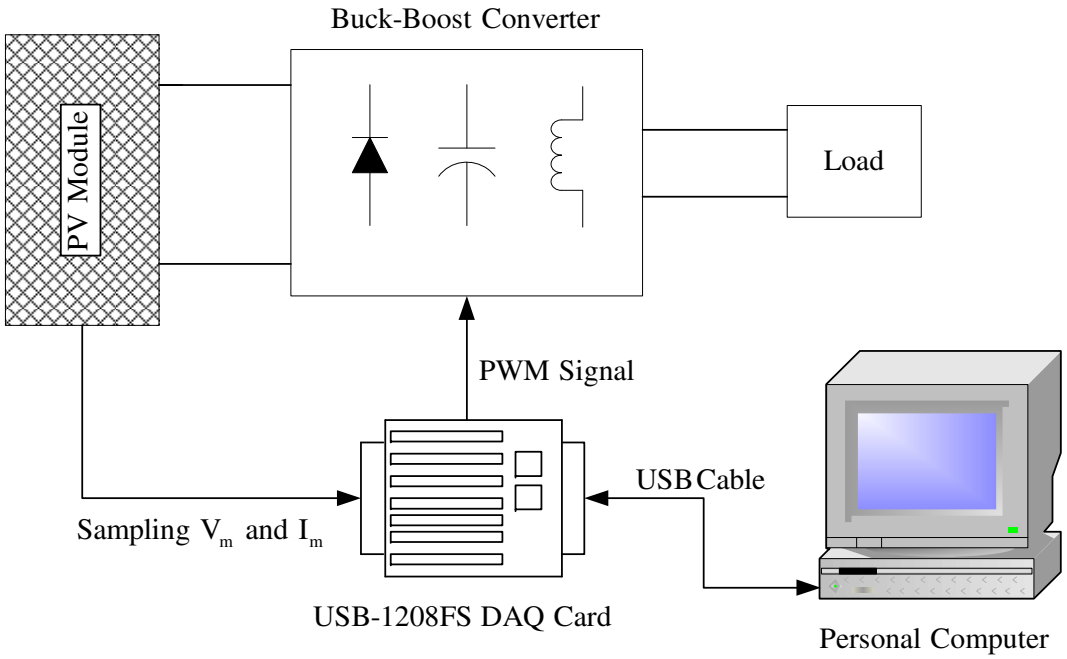


Figure 4-8 Schematic diagram of a PC-based MPPT using fuzzy logic

4.4 Buck-Boost Converter Operation

The buck-boost converter circuit is shown in Figure 4-9. When the transistor Q, is on, energy is transferred to the inductor *L*. When the transistor turns off, inductor current flows through the diode T. Energy stored in *L* is transferred to the capacitor *C* and the load *R*. This transfer action results in an output voltage of opposite polarity to that of the input. Neither the input current nor the output current is continuous, although the inductor current may be continuous or discontinuous.

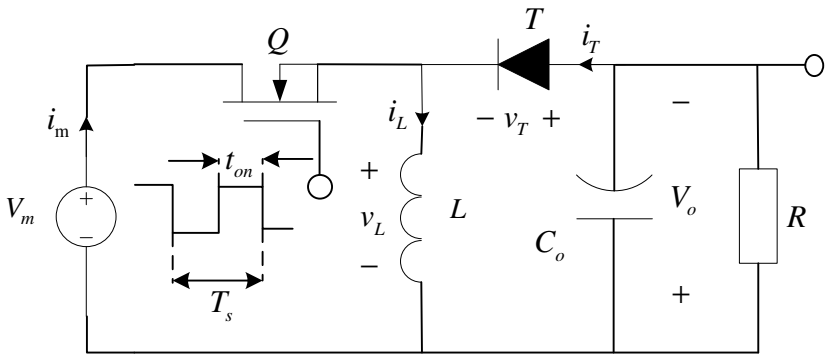


Figure 4-9 Buck-boost converter circuit

4.4.1 Continuous Inductor Current

The circuit voltage and current waveforms for the buck-boost converter operating in continuous conduction mode are shown in Figure 4-10. The voltage drop across the inductor v_{L} is given by:

$$v_{\mathrm{L}} = L\frac{di}{dt} \tag{4-11}$$

Assuming a constant input and output voltage, the change in inductor current Δi_L is derived from Equation (4-11) with $v_{\mathrm{L}} = V_{\mathrm{m}}$, $di = \Delta i_{\mathrm{L}}$, $dt = t_{on}$. V_m is the input voltage, and t_{on} is the transistor on-time. Δi_L is therefore given by,

$$\Delta i_L = \frac{V_m}{L} t_{on} = -\frac{V_o}{L}(T_s - t_{on}) \tag{4-12}$$

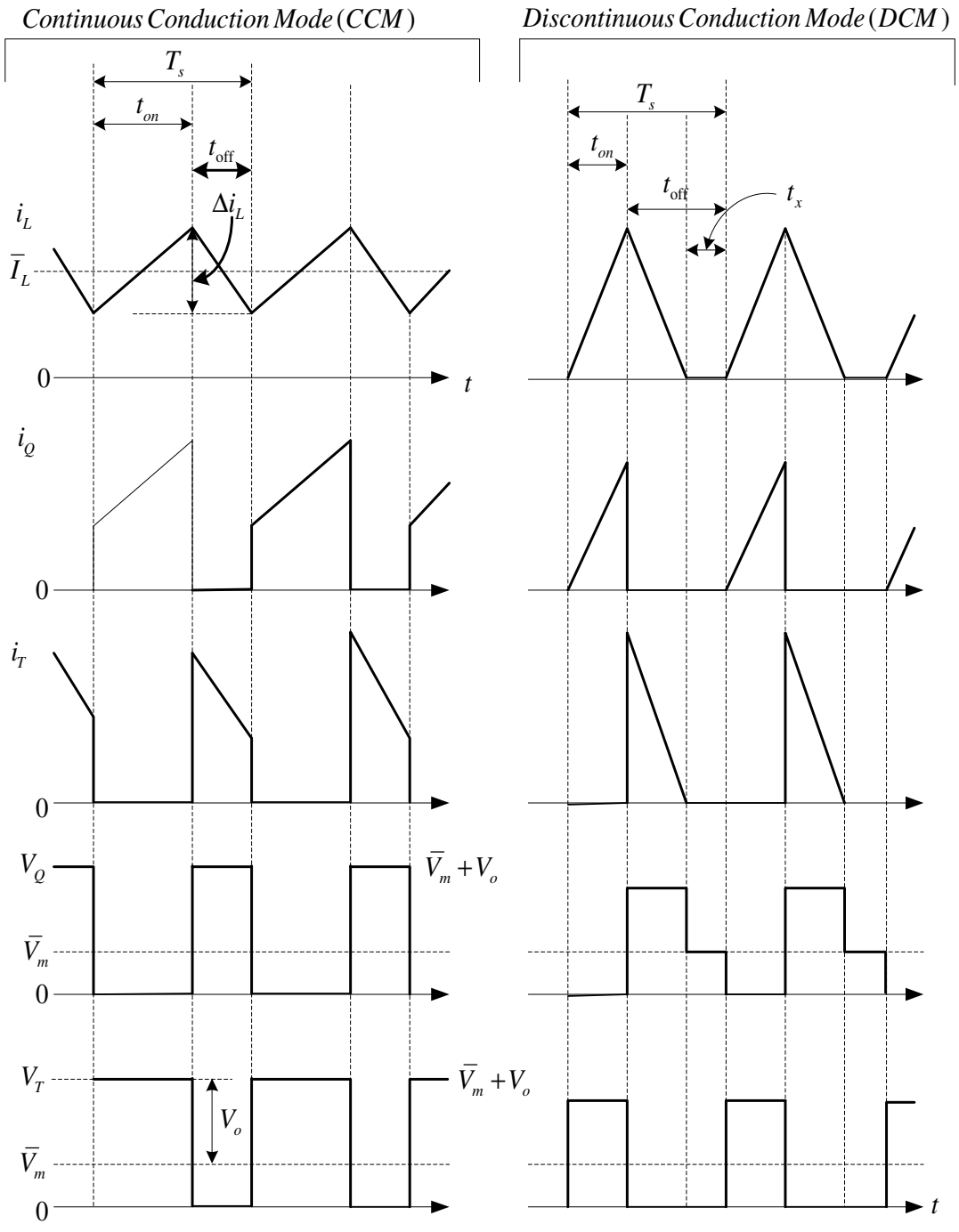


Figure 4-10 Buck-boost converter waveforms for continuous inductor current and discontinuous inductor current waveforms

thus,

$$\frac{V_o}{V_m} = \frac{\bar{I}_m}{\bar{I}_o} = -\frac{D}{1-D} \tag{4-13}$$

where T_s is the transistor switching period, $D = t_{on} / T_s$, V_o is the output voltage, $\bar{I}_m$ is the average input current, and $\bar{I}_o$ is the average output current. For $D < 0.5$, the output voltage magnitude is less than the input voltage while $D > 0.5$,the output is greater in magnitude than the input. Using Equations (4-12) and (4-13) the maximum change in inductor current $\Delta i_{L\max}$, is given by,

$$\Delta i_{L\max} = \frac{D V_m T_s}{2L} \tag{4-14}$$

The maximum inductor current $\Delta i_{L\max}$,and the minimum inductor current $\Delta i_{L\min}$, are given by,

$$i_{L\max} = \bar{I}_L + \frac{\Delta i_L}{2} \tag{4-15}$$

$$i_{L\min} = \bar{I}_L - \frac{\Delta i_L}{2} \tag{4-16}$$

where $\bar{I}_L$ is the average inductor current.

4.4.2 Discontinuous Inductor Current

The change from continuous to discontinuous inductor current conduction occurs when the average inductor current $\bar{I}_L$, is given by,

$$\bar{I}_L = \frac{i_{L\max}}{2} = \Delta i_L \tag{4-17}$$

where from Equation (4-12),

$$i_{L\max} = \frac{V_o}{L}(T_s - t_{on}) \tag{4-18}$$

The output voltage for discontinuous conduction is evaluated from

$$i_{L\max} = \frac{V_m}{L}t = -\frac{V_o}{L}(T_s - t_{on} - t_x) \tag{4-19}$$

which gives,

$$\frac{V_o}{V_m} = -\frac{D}{1 - D - \frac{t_x}{T_s}} \tag{4-20}$$

Using Equation (4-19) and

$$\overline{I}_L = \frac{1}{2}Di_{L\max} \tag{4-21}$$

yields

$$\overline{I}_L = \frac{V_m T_s D}{2L} \tag{4-22}$$

It is observed from Figure 4-10 that the input and output currents are proportional to the inductor current. The inductor current depends on the duty cycle and at the boundary between continuous and discontinuous conduction:

$$\begin{aligned} \overline{I}_m &= \frac{Di_{L\max}}{2} \\ \overline{I}_o &= \frac{(1-D)i_{L\max}}{2} \end{aligned} \tag{4-23}$$

where,

$$i_{L\max} = \Delta i_L = \frac{\overline{I}_L}{2} \tag{4-24}$$

Using $\overline{I}_m = D\overline{I}_L$, Equation (4-24) becomes

$$\overline{I}_m = \frac{V_m T_s D^2}{2L} \tag{4-25}$$

Assuming the converter is lossless, that is $V_m \overline{I}_m = V_o \overline{I}_o$

$$\frac{V_o}{V_m} = \frac{\overline{I}_m}{\overline{I}_o} = \frac{V_m T_s D^2}{2L\overline{I}_o} = \frac{V_o T_s D^2}{2L\overline{I}_m} = D\sqrt{\frac{T_s R}{2L}} \tag{4-26}$$

At the boundary between continuous and discontinuous conduction, Equation (4-25) can be modified using Equation (4-13) to give

$$\overline{I}_o = \frac{V_m}{2L} T_s D(1-D) \tag{4-27}$$

The critical (or minimum) inductance required to ensure continuous conduction is derived from Equation (4-27) and is given by,

$$L_{crit} = 0.5RT_s(1-D)^2 \tag{4-28}$$

At a low output current or low input voltage there is a likelihood of discontinuous conduction. To avoid this condition, a larger inductance value is needed, which worsen transient response.

4.5 Buck-Boost Converter Design

Converter design involves the choice of suitable component values that meet the desired operating conditions. The component values that need to be determined are the input capacitor, inductor, and the output capacitor. The input capacitor is required because the input current of the buck-boost converter is discontinuous. The

main goal of this design is to ensure that the converter operates in the continuous conduction mode since it lowers the peak current flowing through the inductor, and consequently reduces the losses in the converter.

The buck-boost converter conduction mode changes when it is used in PV applications, since the input voltage and current change continuously with the atmospheric conditions. In addition, the converter duty cycle D, is changed continuously in order to track the maximum power point of the PV module. The choice of the converter switching frequency and the inductor value is a compromise between converter efficiency, cost, power capability and weight. For example, the higher the switching frequency, the lower the inductor core size, but the power switch losses are higher. Also, by using a large inductor value, the peak to peak current ripple Δi_L, is smaller, requiring lower current rating power switches, but the converter size is increased substantially because a larger inductor core is required [25][27].

4.5.1 Inductor Selection

The inductor value L, required to operate the converter in the continuous conduction mode is calculated using Equation (4-28). In order to determine the values of R and D to be used in calculating L_{crit}, the worst case condition has to be considered. This occurs when the PV array operates at its maximum capacity at minimum load. It can be assumed that the load is constant at $1\ \Omega$, the converter is

lossless and operates in CCM. The PV module internal resistance at maximum power R_{pvm} is given by,

$$R_{pvm} = \frac{V_{mp}}{I_{mp}} \tag{4-29}$$

where V_{mp} and I_{mp} are the voltage and current at maximum power respectively. A $14W$ solar panel with $V_{mp} = 14.5\ V$, and $I_{mp} = 0.97\ A$ was selected for use in the experiment. R_{pvm} for this panel is obtained using Equation (4-29) as $17.01\ \Omega$. For operation at the maximum power point the load resistance R, must be equal to R_{pvm}. The value of D required for operation at maximum power is obtained from Equation (3-10) by setting $R' = R_{pvm}$ and rearranging to give,

$$D = \frac{1}{1 + \sqrt{\frac{R_{pvm}}{R}}} \tag{4-30}$$

Using Equation (4-29) with $R_{pvm} = 17.01\Omega$ and $R' = 1\Omega$, gives the required value of D as 0.195. Choosing a switching frequency $f_s = 40\,kHz$ and using Equation (5-18), gives $L_{crit} = 81.0\mu H$. It is however noted that R_{pvm} increases with decrease in solar radiation. A more suitable value of L_{crit} is therefore determined with the limiting case of $D = 0$. This gives $L_{crit} = 125\mu H$. However, a larger inductance than L_{crit} is chosen to ensure continuous inductor current during periods of low solar radiation and in the presence of large loads. A $270\,\mu H$ PCV-2-274-10L series

Bobbin inductor from Coilcraft Incorporation was chosen for the converter. It has a $10\,A$ r.m.s. current rating and a maximum resistance of $0.060\,\Omega$ [28].

4.5.2 Input Capacitor Selection

An input capacitor is needed in order to limit the input voltage ripple due the switching action. Since the short circuit current of the PV module is $1.2\,A$, the maximum current ripple to be handled by the input capacitor is approximately $1\,A$. Figure 4-11 shows the capacitor current ripple due to variations in input current. The charge stored in the capacitor ΔQ is given by,

$$\Delta Q = \int i \cdot dt = \frac{\Delta i_m}{2} \cdot \frac{T_s}{2} \tag{4-31}$$

Using $\Delta i_m = 1A$ and $T_s = 1/f_s = 25\times10^{-6}s$, Equation (4-31) gives $\Delta Q = 6.25\times10^{-6}C$.

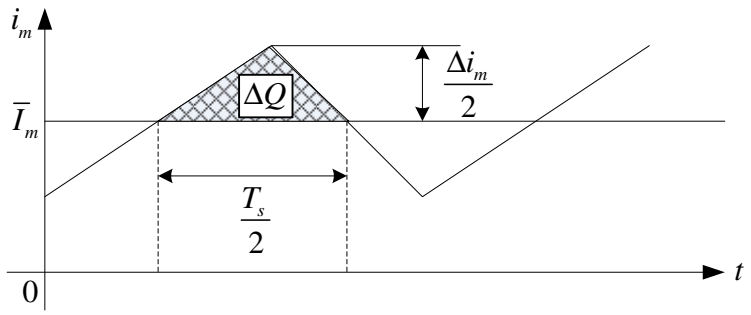


Figure 4-11 Input capacitor ripple current

With an input voltage of $14.5\,V$ at maximum power, the minimum capacitor size required to regulate the input voltage ripple to 1% is given by,

$$C = \frac{\Delta Q}{\Delta V} = \frac{(6.25\times10^{-6}C)}{(0.01)(14.5V)} = 43.10\mu F \tag{4-32}$$

A $50\mu F, 30V$ electrolytic capacitor was selected for the input.

4.5.3 Output Capacitor Selection

A suitable value of capacitance is required in order to attain the desired output voltage ripple. The output voltage ripple is the capacitor ripple voltage. The output current $\overline{I}_o$ is provided solely from the capacitor during the period t_{on} when the switch is conducting, thus

$$\Delta V_o = \frac{1}{C}\int i \cdot dt = \frac{1}{C} t_{on}\overline{I}_o \tag{4-33}$$

Substituting for $\overline{I}_o = V_o / R$ gives

$$\Delta V_o = \frac{t_{on}V_o}{CR} \tag{4-34}$$

Rearranging gives the fractional peak-to-peak voltage ripple in the output voltage as,

$$\frac{\Delta V_o}{V_o} = \frac{1}{CR} t_{on} = \frac{DT_s}{CR} \tag{4-35}$$

Considering the extreme case where, $D = 0.9$ (D is limited to maximum value of 0.9 to reduce power loss) and $R = 1\,\Omega$, the minimum capacitor size required to regulate the output voltage ripple to 1% is obtained by rearranging Equation (4-35) as,

$$C = \frac{DT_s}{(\Delta V_o / V_o)R} \tag{4-36}$$

Using $D = 0.9$, $T_s = 25 \times 10^{-6} s$, $R = 1\ \Omega$, and $\Delta V_o / V_o = 0.01$A Equation (4-36) gives $C = 2250 \mu F$. A $2700 \mu F, 50V$ electrolytic capacitor was selected for the output.

4.5.4 Diode Selection

Diode choice is a trade off between breakdown voltage, speed, and forward voltage. A higher forward voltage drop increases the converter power loss. However, a fast switching diode is needed to act as a switch for the energy in the inductor. If the diode is slow to react, the efficiency of the converter is low and damaging high voltage transients will develop. In case of these transients and the possibility of large output voltages if the load is suddenly disconnected, the diode must have a high breakdown voltage. The best combination of these features could be found in SB530 Schottky diode from RS Components, which has a reverse breakdown voltage of 30 V and a forward voltage drop of $0.57V$ at a rated current of 5A [29].

4.5.5 Switching Transistor Selection

An n-channel enhancement mode power MOSFET IRFZ24N from Fairchild Semiconductor Corporation was selected as a high speed-switching device for the buck-boost converter. The rating of this device is 55V and 17A, well above maximum operating voltage and current which are 21 V and 1.2 A of the converter.

It has a maximum gate leakage current of only $100nA$ and a low on state resistance of 0.07Ω [30]. The low on-state resistance leads to small conduction loss hence higher converter efficiency.

4.6 Pulse Width Modulation

Pulse Width Modulation (PWM), is a method of controlling the amount of power to a load by switching the load on and off. The amount of power delivered to the load is proportional to the percentage of time that the load is switched on. PWM signals can be generated in a number of ways namely: analogue circuitry, PWM chips, digital counters, and using timers on microcontrollers. The SG3525A PWM controller from ST Microelectronics was chosen for this purpose because it can be operated to give a duty cycle of $0-100\%$.

4.6.1 SG3525A PWM IC Operation

The pin configuration of the IC is shown in Figure 4-12 [31]. The PWM controller has two complementary outputs called Output A (pin 11) and Output B (pin 14). Both outputs have a duty cycle limited to less than 50% and they are never high at the same time. If the frequency of each output is F, then the frequency of the sum of the outputs is $2F$. Example waveforms are shown in Figure 4-13 where the frequency of each output is $20kHz$. When both outputs are used in parallel, the switching waveform in the circuit will have a frequency of $40kHz$ as shown in Figure 4-13.

The SG3525A PWM IC allows control of a parameter called the dead time which is a fixed length of time where both outputs are off. This feature is useful when two MOSFETs are used, one switched by each output, in order to attain 100% duty cycle. The negative feedback can easily force the PWM portion of the controller to go to 100% duty cycle, which means that the switches will never turn off. If the switches never turn off, the inductor will never discharge and the inductor current becomes very large. The dead time is used to ensure that both MOSFET switches have a guaranteed amount of off time, preventing this problem.

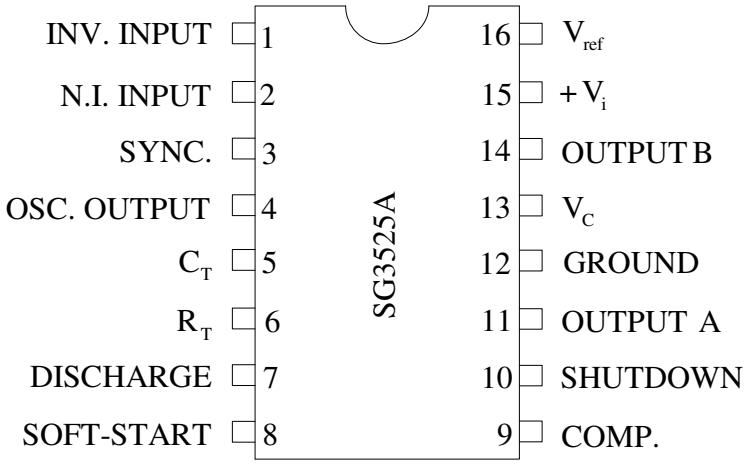


Figure 4-12 SG3525A pin configuration

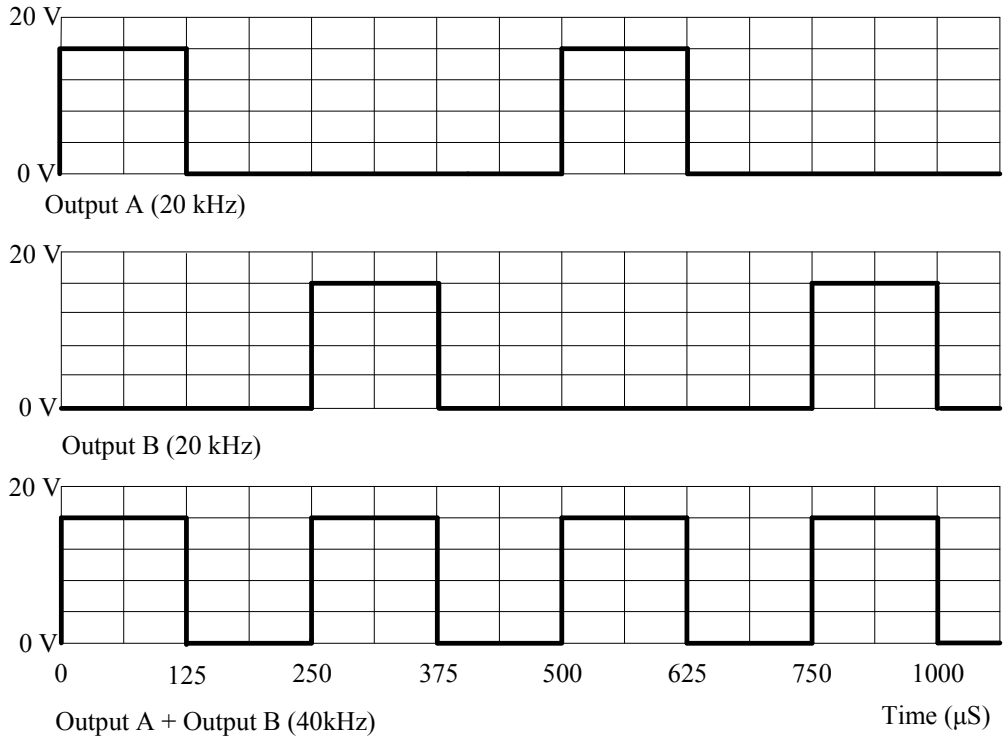


Figure 4-13 SG3525A output waveforms

4.6.2 Design Procedure

The biasing circuit for the PWM IC is shown in Figure 4-14 [31]. The design process involves choosing the values for the oscillator resistor R_T, dead-time resistor R_D, and oscillator capacitor C_T, in order to produce the desired switching frequency and a dead time that limits the maximum duty cycle. The switching frequency is given by [31],

$$f = \frac{1}{C_T(0.7R_T + 3R_D)} \tag{4-37}$$

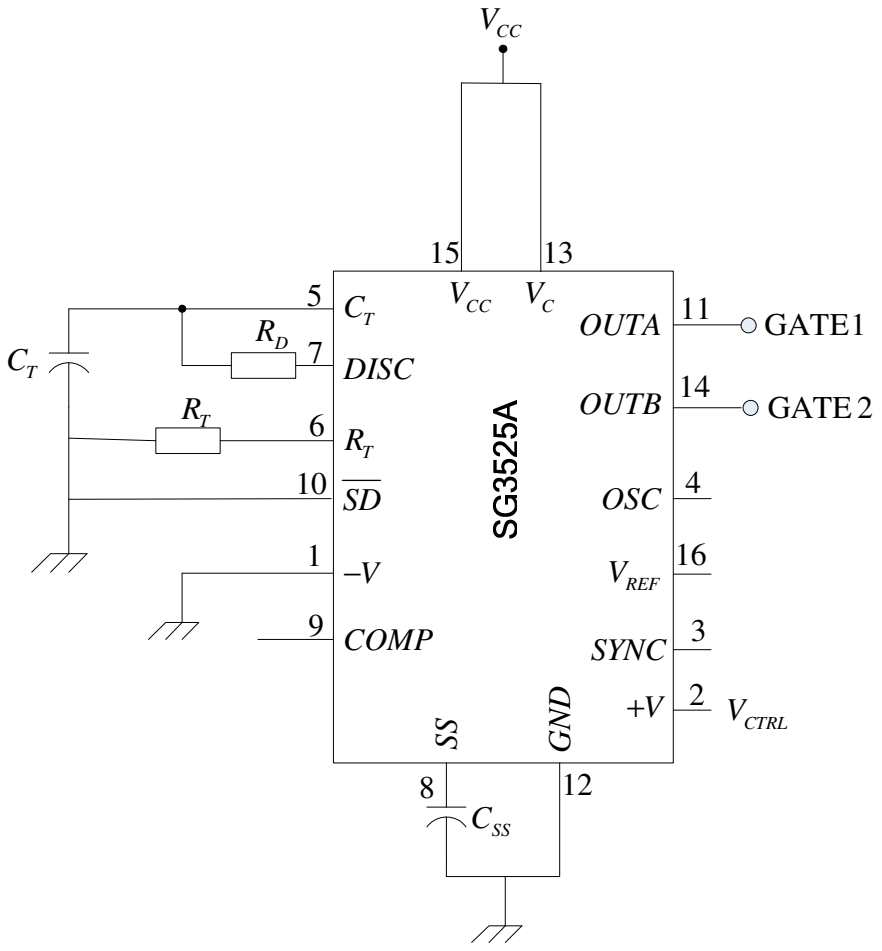


Figure 4-14 SG3525A biasing circuit

A switching frequency of $40\,kHz$ was selected for the circuit. Choosing $C_T = 0.01\mu F$ and $R_D = 0$, R_T is calculated using Equation (4-37) to give $R_T = 3.6\,k\Omega$. The dead time resistor R_D was set to zero so that the maximum duty cycle could be software controlled using the control voltage V_{CTRL}, supplied to pin 2. The minimum and maximum duty cycle is attained with an input voltage of $0.7V$ and $3.6V$ respectively [31]. The output drivers, Pins 11 and 14, can each sink or

source a maximum current of $100\,mA$. The output voltage waveform has a maximum value of $+V_{CC}$ and a minimum of zero. With $V_{CC} = 12V$, the output drivers meet the IRFZ24N MOSFET requirements which has a rated a gate current of $100\,nA$ and a minimum gate threshold voltage of $4V$.

The rate at which the pulse-width is allowed to increase during start-up is determined by the soft-start capacitor C_{SS}, connected to pin 8. This capacitor is charged by a $50\,\mu A$ current source in the IC. When the capacitor reaches approximately 2.5 volts, the soft-start is complete. This feature prevents high duty cycles at start up which cause large inductor currents. Choosing a soft-start that takes 1 second, C_{SS} is calculated using the equation:

$$C_{SS} = \frac{\Delta Q}{\Delta V} = \frac{It}{\Delta V} \tag{4-38}$$

Using $I = 50\mu A$, $t = 1s$, and $\Delta V = 5V$, Equation (4-38) gives $C_{SS} = 20\mu F$.

The SG3525A PWM IC outputs with $C_T = 0.02\mu F$, $R_T = 37\,k\Omega$ and $V_{CTRL} = 2.5V$ are shown in Figure 4-15.

Figure 4-15 SG3525A PWM IC test outputs

4.7 Data Acquisition System

The data acquisition system consists of voltage and current sensing circuitry and USB-1208FS data acquisition (DAQ) card. The DAQ card acquires the PV module current and voltage, which is used by the maximum power point tracking software to determine the converter duty cycle. A control voltage (V_{CTRL}) corresponding to the desired duty cycle is passed through the DAQ card, which feeds it to PWM IC.

4.7.1 USB-1208FS Data Acquisition Card

The USB-1208FS is a USB 2.0 full-speed device supported under Microsoft® Windows®. It features eight analog inputs, two 12-bit analog outputs, 16 digital I/O connections, and one 32-bit external event counter [32]. The device appears as shown in Figure 4-16. The analog inputs are software configurable for either eight 11-bit single-ended inputs, or four 12-bit differential inputs. The input/output (I/O) ports are Transistor Transistor Logic (TTL) compatible with a programmable sampling rate of up to $50\,kHz$. I/O connections are made to the screw terminals located along each side of the DAQ card.

Figure 4-16 USB-1208FS data acquisition card

4.7.2 Sensing Circuitry

The PV module voltage is sensed directly by the microcontroller through a voltage divider network while a 0.1Ω current sensing resistor is used to measure the current. The sensing circuitry is shown in Figure 4-17.

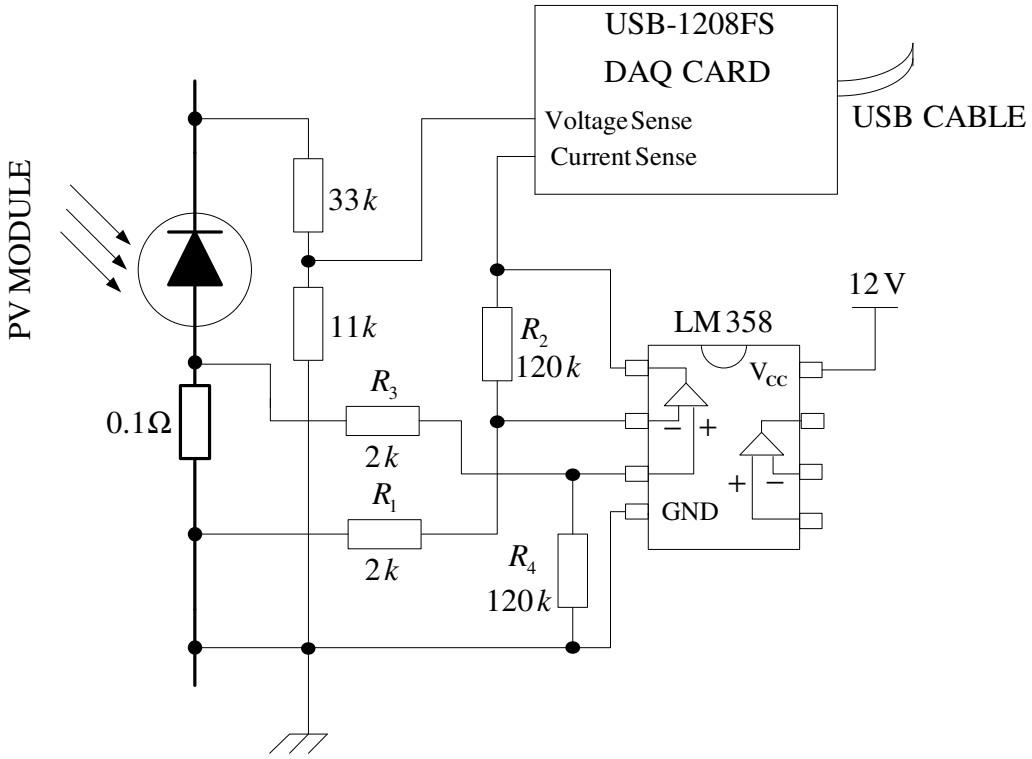


Figure 4-17 Sensing circuitry

For a maximum current of $1.2A$ the current sensing resistor produces a voltage drop of $0.12V$. A non-inverting differential amplifier was used to increase this voltage to a level where the microcontroller was easily able to read. The low cost LM358 IC was chosen for the differential amplifier. The amplification of the

amplifier is directly proportional to R_1 / R_2. With $R_2 = 120 k\Omega$ and $R_1 = 2 k\Omega$ it gives an amplification of 60. This amplifies the $0.12V$ at $1.2A$ over the current sensing resistor to $7.2V$. The features of LM358 IC are given in reference [33].

4.8 Features of the Software

The control software was coded in C++ using Microsoft Visual Studio 6.0. A flowchart of the software is shown in Figure 4-18. The software acquires the PV module voltage and current periodically from the DAQ card and uses the fuzzy algorithm to determine the required change in duty cycle in the search for maximum power. The duty cycle is changed by modifying the control voltage signal supplied to pin 2 of SG3525A PWM controller using the DAQ card. A control voltage of $0.9V$ corresponds to zero duty cycle while $3.3V$ corresponds to 100% duty cycle. The output ports of the DAQ card are latched which ensures that the system operates at constant duty cycle until the next update. An initial duty cycle of 10% is used with initial values of change in power ΔP and change in duty ΔD set to zero. All the input values and controller parameters are logged to a data file and displayed on the screen using the user interface shown in Figure 4-19. The inputs are sampled at $10 Hz$ and the duty cycle is also updated at the same rate. A listing of the software is given in the Appendix.

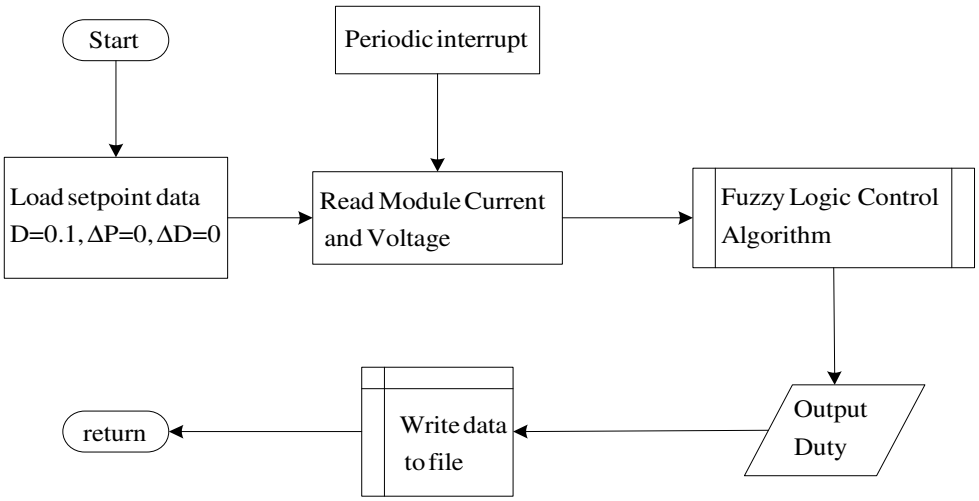


Figure 4-18 Flowchart describing operation of the software

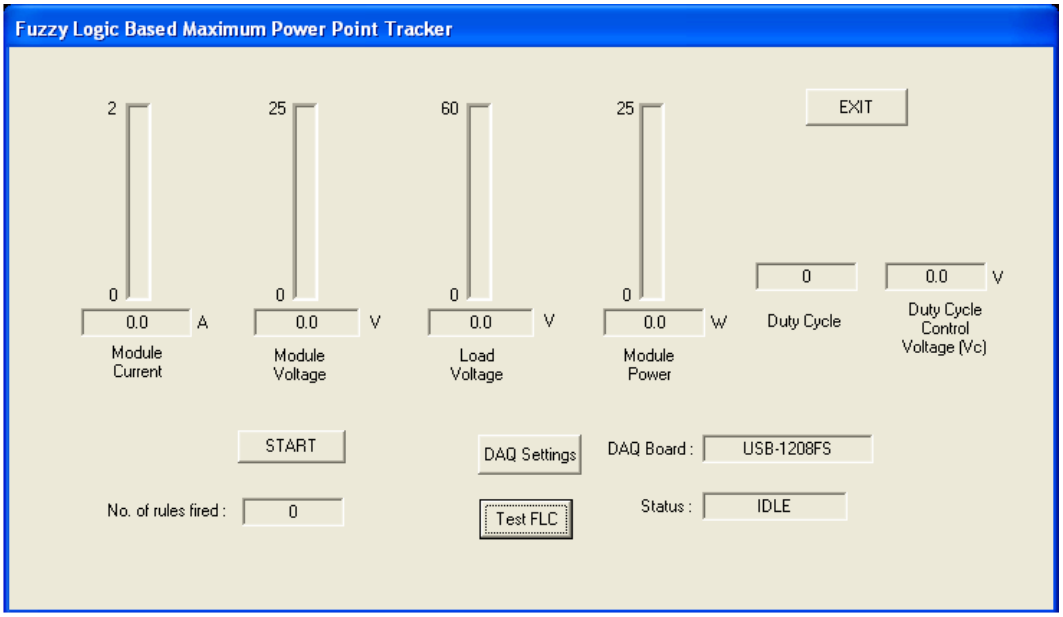


Figure 4-19 Control software user interface

4.9 Conclusion

A complete fuzzy logic based maximum power point tracker for a PV system was successfully designed and implemented. The design of the fuzzy logic controller structure and formulation of the algorithm has been presented. The power circuit is based on a buck-boost converter, while the controller used a microcomputer with a fuzzy logic control algorithm that searches for the optimum duty cycle and transfers the peak power from the solar panel to a resistive load. The system is ready to be implemented in a dedicated microcontroller and fitted to a larger installation.

CHAPTER FIVE

5 RESULTS AND DISCUSSION

This chapter presents a simulation and experimental results of the fuzzy logic based maximum power point tracker. It consists of a PV module, buck-boost dc-dc converter, data acquisition card, and a resistive load. The fuzzy logic control algorithm developed in Chapter 4 is used to search for the optimal operating point under different operating conditions. A computer was used for data acquisition, running the control algorithm, data storage, as well as data display and analysis.

5.1 Simulation Results

In order to investigate the performance and accuracy of the proposed fuzzy MPPT technique simulations were performed under time-invariant and time varying conditions. The model in Figure 4-7 was first simulated in open loop mode without the fuzzy controller in order to determine a data set consisting of the expected maximum power under different values of solar radiation. The data was generated by varying the module input current from zero to short-circuit current for each value of solar radiation and recording the output power. The maximum power point for each value of solar radiation was then obtained from the data. The achieved maximum power during the simulation at each time step t_k was recorded for comparison with the expected maximum power. At the end of the simulation,

the data was used to determine the efficiency of the maximum power point tracking algorithm using the formula,

$$\eta_{mppt}(t_k) = \frac{P_{achieved}(D_k, t_k)}{P_{mpp}(t_k)} \tag{5-1}$$

where, $P_{achieved}(D_k, t_k)$ is the power achieved by the algorithm at duty cycle D_k, and $P_{mpp}(t_k)$ is the actual maximum power possible. The average of η_{mppt} over all t gives a good measure of the effectiveness of the algorithm. In general, the fuzzy logic algorithm, operating at sampling frequencies between $1kHz$ and $10kHz$, achieved an average tracking effectiveness of over 95%. This means that over 95% of the available power was captured by the tracking algorithm.

For comparison purposes, the maximum power achieved by the algorithm was plotted with the expected actual maximum power under different operating conditions. Figure 5-1 shows the power track when the solar radiation varies at $0.3\ Hz$ between 50% and 100%. This fast and wide variation of solar radiation was used to test the effectiveness of the algorithm under rapidly changing conditions. The tracking effectiveness of the simulation of Figure 5-1 was 98.6%. Without the MPPT less than 20% of the available power could be delivered to the load of 100Ω used in the simulation.

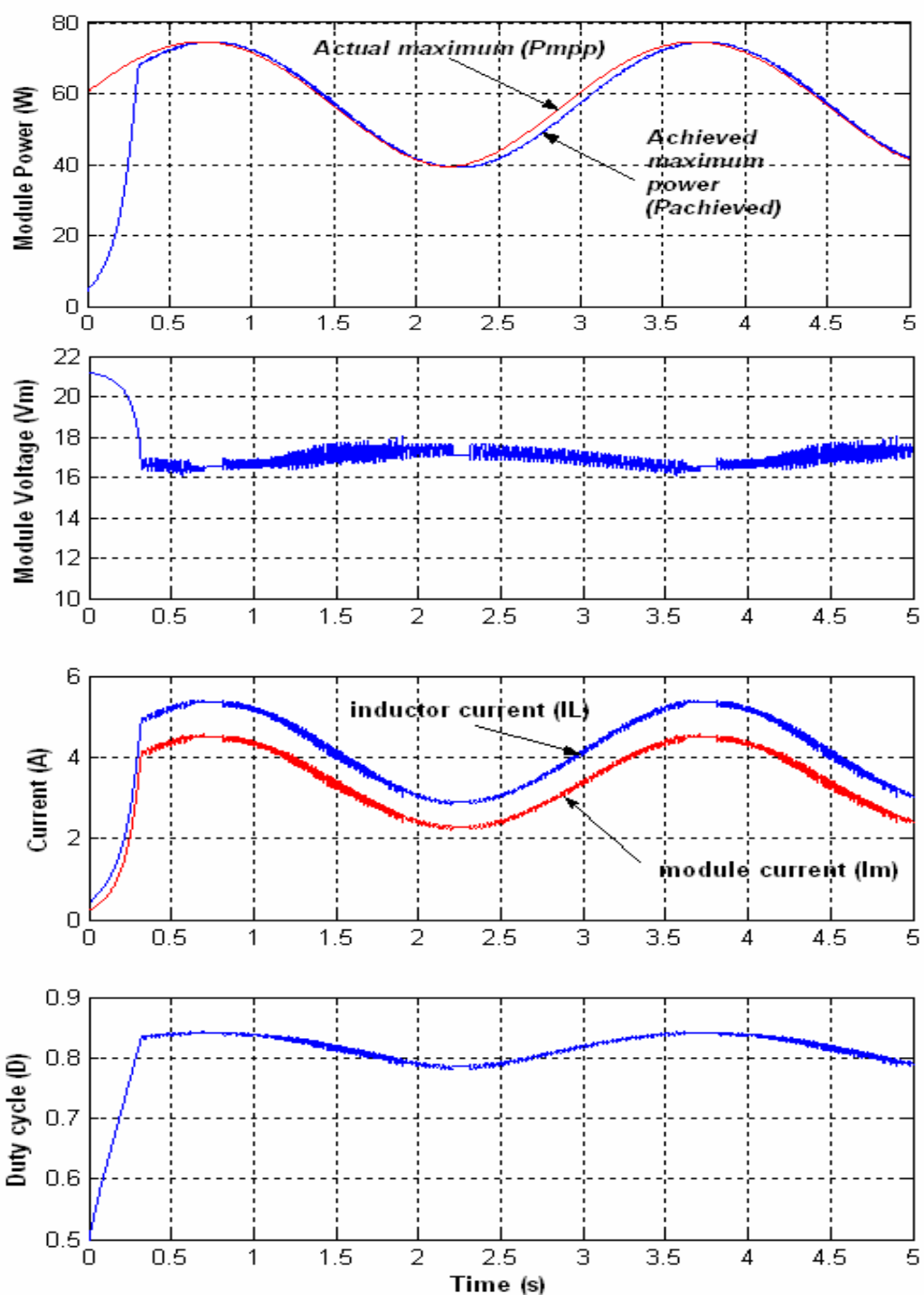


Figure 5-1 Maximum power point tracking under 50% sinusoidal variation of solar radiation

However, it was observed that the maximum power delivered without the tracker depends on the size of the load. The variation of module voltage, module current, inductor current, and the duty cycle are also shown in Figure 5-1. The inductor current and the module current have no overshoots due to the smooth variation of the duty cycle.

As a further test on the performance of the algorithm, step changes of load by 50% were introduced in addition to the time variation of solar radiation. The simulation results are shown in Figure 5-2. The algorithm exhibits a recovery time of less than $0.1s$. This is the time taken to return the system back to operation at maximum power after the step change in load. There is no appreciable change in the efficiency of the algorithm under step changes in load. Figure 5-2 also shows how the algorithm moves the duty cycle while tracking the maximum power. The variation in duty cycle is fast but smooth which avoids oscillations around the maximum power.

Having proven the effectiveness of the algorithm under time-varying conditions, it was also applied to the time-invariant case. This is a situation where the solar radiation and temperature of the PV module are held constant in time. In this static case, the algorithm attained an efficiency of over 99%. The track of this simulation is shown in Figure 5-3. Under this static case, the algorithm is also able to reach the maximum power point in under $0.1s$ when there are step changes in load.

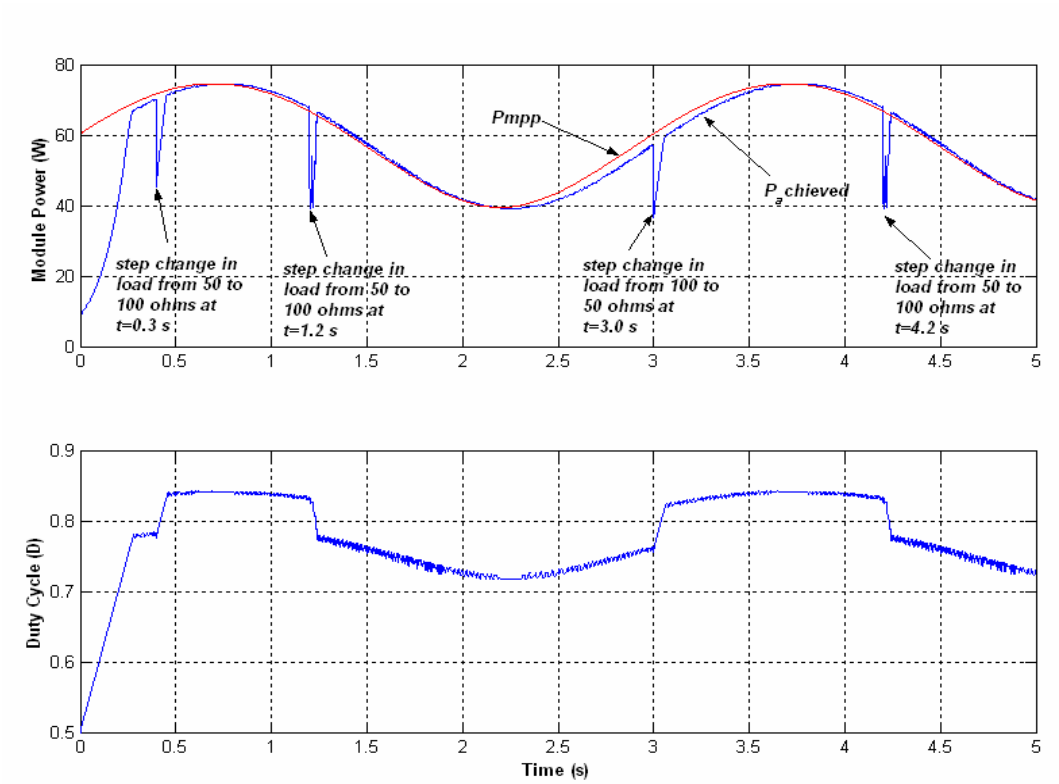


Figure 5-2 Maximum power point tracking under 50% sinusoidal variation of solar radiation and step changes in load

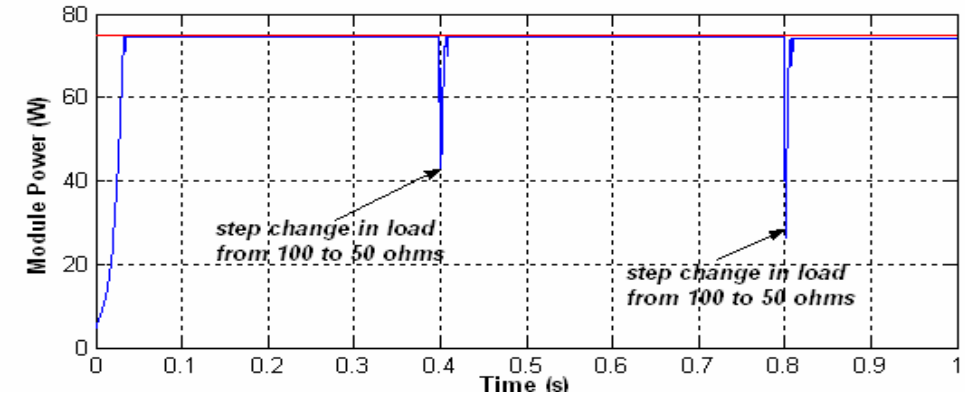


Figure 5-3 Maximum power point tracking under time-invariant solar radiation and step changes in load

The performance of the algorithm considered so far assume that the system has perfect knowledge of the power that results from a particular duty cycle $D(k)$. In reality, however, the algorithm will not have such perfect measurements. Due to the limitations of the analog-to-digital converter and the current and voltage sensing circuits, the measurements of the PV module's voltage and current, and hence the module power is subjected to noise. Therefore, to test the robustness of the algorithm in the presence of measurement noise, the simulation was extended. Gaussian noise was added to the power measurements before the algorithm used them. The Matlab *awgn()* function was used to add white Gaussian noise to the measured power signal. The signal-to-noise ratio is specified in decibels. A noise power value of $7\,dB$, which represents a signal-to-noise ratio of 5 was used. This represents an absolute worst case noise level of $15W$ when the PV module is operating at its maximum power rating of $75W$.

Although the algorithm did not perform effectively in the presence of measurement noise as in the noise-less case, the results of the simulations were satisfactory. For the case where the solar radiation varied in time, the tracking efficiency of the algorithm (averaged over a single run) was generally in the range of $90-95\%$. It was observed that the efficiency improved by decreasing the change in power scale factor k_p and increasing the change in duty scale factor k_d. The simulation was run with $k_p = 0.1$ and $k_d = 0.01$. The control parameters were entered using the controller dialog box shown in Figure 5-4. Due to the stochastic

nature of the noise in the simulation, the performance varied from one simulation run to the next. The track of one such run is given in Figure 5-5.

The simulation in Matlab proved essential for developing and fine tuning the control rules and membership functions of the maximum power point tracking algorithm. The simulation results show that the algorithm is robust and is able to handle variations in the operating conditions of a PV module. However, for the hardware implementation, the fuzzy logic algorithm was coded in C++ using Microsoft Visual Studio 6.0. This became necessary because the drivers for the data acquisition card were not available in Matlab. The C++ control software was tested, and debugged using the Matlab simulation as a benchmark. A set of test points was selected randomly from the design space, representing several of the possible input conditions that would be encountered by the maximum power point tracker. Each test point was input into the Matlab model as well as the C++ controller code. Utilizing the output of the Matlab model, the controller code was debugged until the controller code output matched the output of the Matlab model. The program dialog box for testing the fuzzy controller outputs is shown in Figure 5-6. It displays the inferred output for the given inputs as well as identifying the activated rules and their firing strengths.

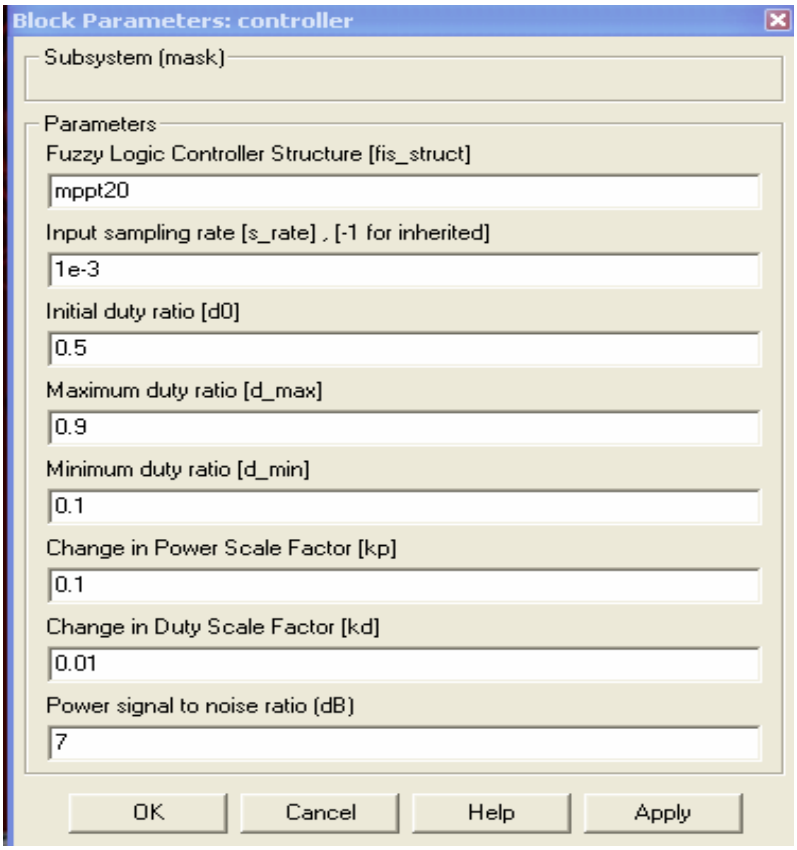


Figure 5-4 Controller mask dialog box

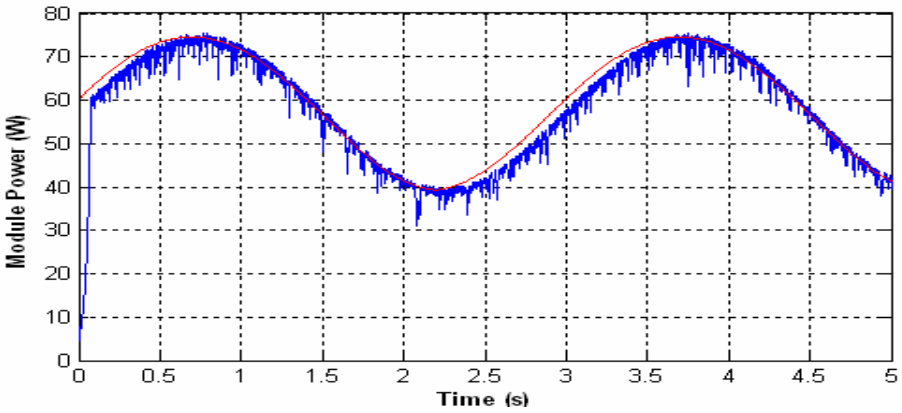


Figure 5-5 Maximum power point tracking under time-varying solar radiation in the presence of noise

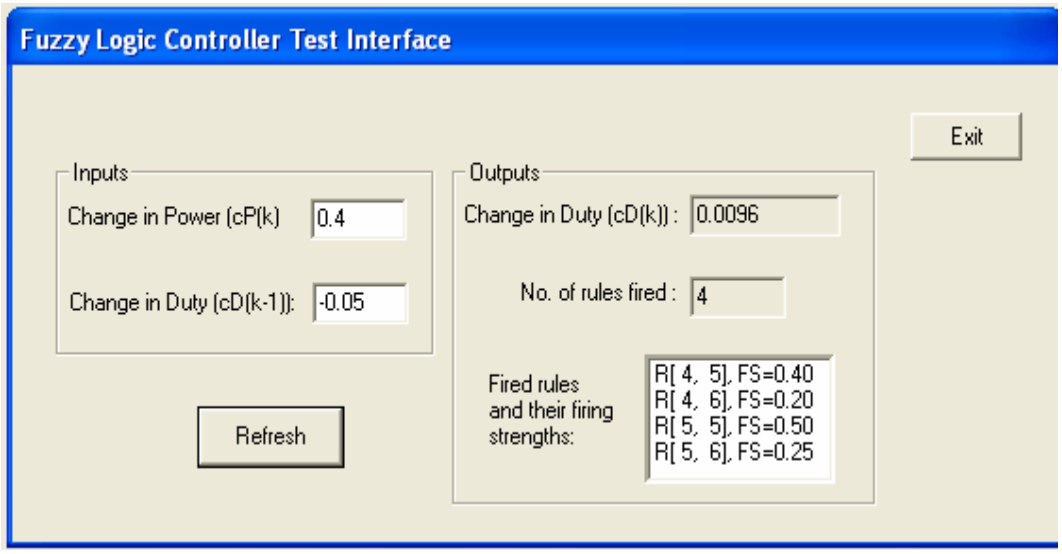


Figure 5-6 Fuzzy controller test interface dialog box

5.2 Experimental Results

The system shown in Figure 4-8 was implemented as explained in the preceding sections and used to supply power to a resistive load. The main goal was to transfer as much power as possible to a $100\,\Omega$ resistor. A PV module used for the collection of the experimental data is *Kenital* 14 W solar panel. The module has a maximum power output of $14W$ and a $22.5V$ open-circuit voltage at an irradiation of $1000W/m^2$ at 25^oC. The PV module specifications provided by the manufacturer are shown in Table 5-1 [34].

The performance of the MPPT system was first tested using a dc power supply with different current settings. The MPPT was able to vary the output voltage until the maximum current set was drawn from the power supply which corresponds

to transfer of maximum available power to the load. The MPPT was then connected to a PV module and tested during a clear sunny day. A clear sunny day was chosen to ensure a constant solar radiation was available since the circuit does not include ways of measuring the solar radiation level. Figure 5-7 shows the maximum power attained with the MPPT and the normal operating power (without MPPT) in one of the test runs. The system with MPPT was found to attain an average maximum power of $12.5W$. It is clear that the module power is maintained at an optimum value with slight variations due to noise in current and voltage measurements. Figure 5-7 also shows that the attained power for the same load without MPPT is only $4\ W$. These results indicate an increase of about 200% in PV output power in the presence of the fuzzy MPPT.

Table 5-1 Electrical specifications for *Kenital* 14W PV module

Maximum power, P_m (W)	14
Short circuit current, I_{sc} (A)	1.20
Open circuit voltage, V_{oc} (V)	22.5
Voltage at maximum power point, V_{mp} (V)	14.5
Current at maximum power point, I_{mp} (A)	0.97
Size ($mm \times mm \times mm$)	430x406x22

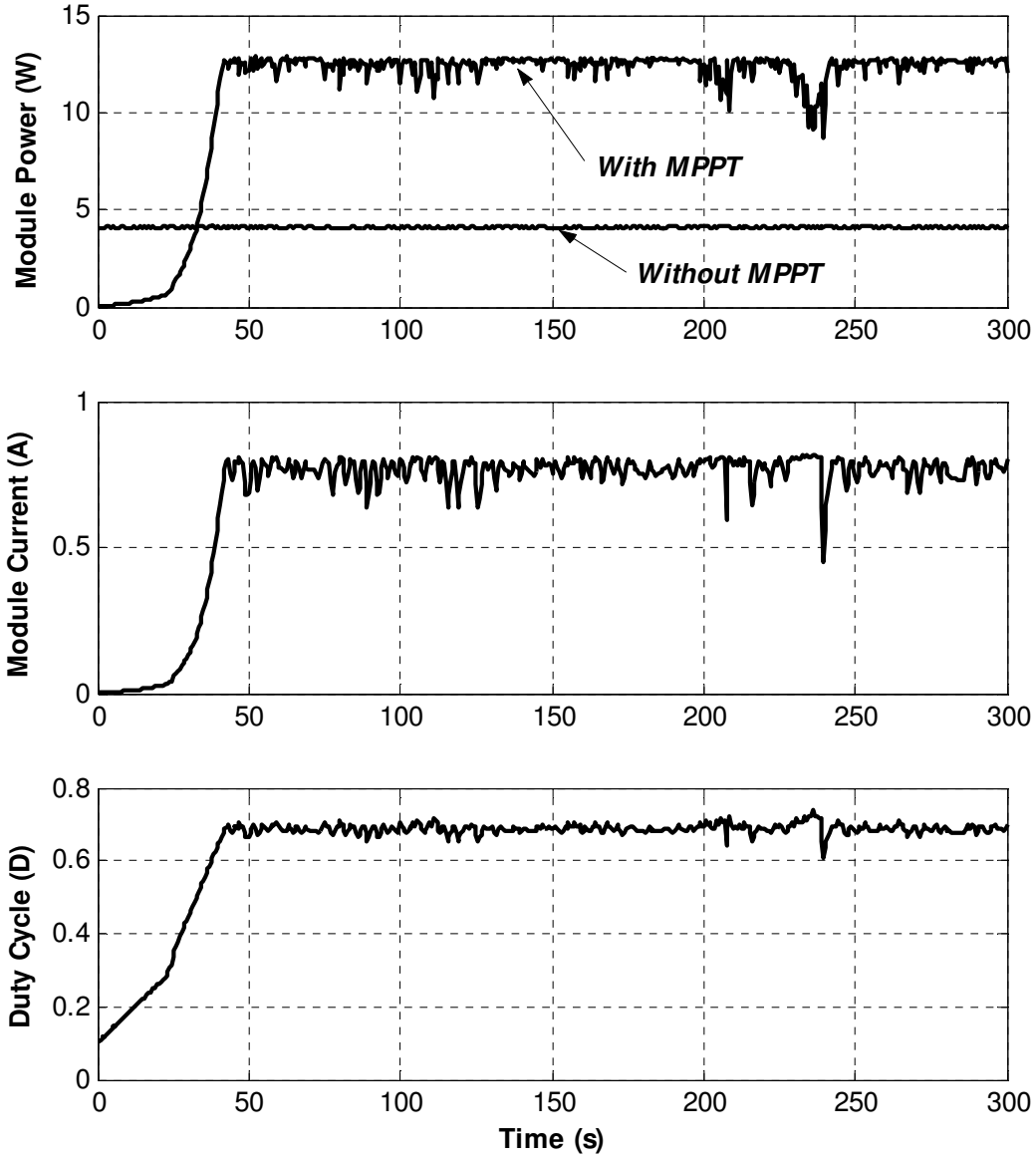


Figure 5-7 Measured results under high solar radiation on a clear sunny day at 12.30pm

Extensive measurements indicate that the amount of increase in PV module output power highly depends on the environmental conditions and the load level. This shows the importance of using MPPT to get a good matching between the load

and the PV module. It was however noted that the system could not track the maximum power under very cloudy conditions. Under low solar radiations the solar panel generates very low currents due to its low power rating and the system could not be able to sense the low currents and track the power adequately. It is however expected that the system will give good performance when fitted in PV systems with high power ratings.

5.3 Conclusion

The simulation results have shown that the controller is inherently robust with an efficiency of over 95% under wide input variations. The experimental results also show that system reliably stabilizes the maximum power transfer under time invariant input conditions. Further experimental tests are required in order to determine the stability of the algorithm under rapidly changing atmospheric and load conditions.

CHAPTER SIX

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

The aim of this thesis was to develop a method to optimize the energy extraction in a photovoltaic power system. The concept of PV module maximum power point tracking has been presented and various methods of addressing existing challenges are explored. A fuzzy logic based algorithm for tracking the maximum power is proposed in this work. In order to formulate and implement the algorithm, a system model is needed. The various components and subsystems are analyzed, modeled, validated, and combined together to produce a complete maximum power point tracker model. A hardware implementation was then carried out to determine the performance of the algorithm in a practical setup.

Analysis of different dc-dc converter topologies showed that the buck-boost topology is the most suitable for a maximum power tracker. The PV module and the buck-boost converter were modeled and validated in Simulink while the fuzzy logic algorithm was formulated using the Fuzzy Logic Toolbox in Matlab. The complete Maximum Power Point Tracker model was formed by combining the PV module and the converter model with the fuzzy logic controller. The MPPT model was used to tune the fuzzy logic controller rules and membership functions. The PV module

model was found to be sufficiently accurate and can model any solar panel using information supplied in manufacturer data sheets. Simulation results show that the proposed fuzzy logic algorithm has an average efficiency of 95% under rapidly varying conditions and in the presence of measurement noise. The results show that compared to other MPPT techniques, it provides improved performance in terms of oscillations about the maximum power point, speed and sensitivity to parameter variation. This is possible since fuzzy logic controller rules can be assigned separately for the various regions of operation resulting in effective small-signal and large-signal operation.

A hardware design and implementation of the MPPT was then carried out in order to test the performance of the proposed algorithm. The design was broken down into several smaller components, which are described in turn. For each sub-circuit, a thorough description of the relevant design issues and decisions was provided. Experimental results show that the proposed algorithm is able to transfer peak power from a PV module to the load. The results indicate that a significant amount of additional energy can be extracted from a photovoltaic module by using a fuzzy logic based maximum power point tracker. This results in improved efficiency for the operation of a photovoltaic power system since batteries can be sufficiently charged and used during periods of low solar radiation. The improved efficiency is expected to lead to significant cost savings in the long run.

6.2 Future Work

The overall control goal in a photovoltaic power system is not to deliver the maximum amount of possible power to the load or batteries, but only as much as is needed at any given time; this can be referred to as power matching. Usable power that is not extracted from a particular solar panel (because it is not operating at its current maximum power point) is dissipated as heat through its surface. In a practical application like a water pump in a remote village, the MPPT circuit must include a way to determine how much power is needed at any given time and perform maximum power tracking only to a level where power requirements are met. The MPPT should also include a second voltage regulation stage that maintains a steady output regardless of variations in the load demand. These improvements will greatly boost the functionality of the system in a practical setup.

As part of future work, ways of implementing a fuzzy logic algorithm in a dedicated single-chip microcontroller needs to be addressed. The control algorithm is fairly complex and modifications need to be made in order to meet the limited memory space and speed of microcontrollers. The microcontroller must also incorporate timers, PWM input and output, A/D and D/A interfaces, and interrupts for timing control and communications. The overall circuit also needs to be modified to include ways of supplying power to the control circuit using batteries charged by the MPPT. The redesigned circuit should be implemented in a printed circuit board.

On the whole, it is concluded that the overall objective of formulating and implementing a fuzzy logic based maximum power point tracker for a photovoltaic power system has been met. Although there is a large amount of work that can and should still be done, the work in this thesis has created a solid foundation to allow that work to continue.

.

REFERENCES

[1] Ocran, T.A. et al, *Artificial Neural Network Maximum Power Point Tracker for Solar Electric Vehicle*, Tsinghua Science & Technology, Vol. 10, No. 2, pp. 204-208, 2005.

[2] Hua C., Lin J., *A modified tracking algorithm for maximum power tracking of solar array,* Energy Conversion and Management 45 (2004) 911-925, Elsevier Ltd. 2003.

[3] Salas V., et al, *New algorithm using only one variable measurement applied to a maximum power point tracker,* Solar Energy Materials & Solar Cells 87 (2005) 675-684, Elsevier B.V. 2004.

[4] C. Hua and C. Shen, *Comparative study of peak power tracking techniques for solar storage systems*, in *Proc. IEEE Appl. Power Electron. Conf. and Expo.*, Feb. 1998, vol. 2, pp. 676–683.

[5] Koutroulis E., Kalaitzakis K., Voulgaris N.C., *Development of a Microcontroller Based Photovoltaic Maximum Power Point Tracking Control System*, IEEE Transactions on Power Electronics, Vol. 16, No. 1, 2001.

[6] J. H. R. Enslin, D. B. Snyman, *Simplified feed-forward control of the maximum power point tracker for photovoltaic applications*, Proc. Int. Conf. IEEE Power Electron. Motion Control, 1992, vol. 1, pp. 548–553.

[7] M. Bodur and M. Ermis, *Maximum power point tracking for low power photovoltaic solar panels*, in Proc. IEEE Electro Tech. Conf., 1992, vol. 2, pp. 758–761.

[8] C. R. Sullivan and M. J. Powers, *A high-efficiency maximum power point trackers for photovoltaic array in a solar-powered race vehicle*, in Proc. IEEE PESC, 1993, pp. 574–580.

[9] M. Veerachary, T. Senjyu, and K. Uezato, *Feed-forwardmaximum power point tracking of PV systems using fuzzy controller*, IEEE Trans. Aerosp. Electron. Syst., vol. 38, no. 3, pp. 969–981, Jul. 2002.

[10] Ocran, T.A. et al, *Artificial Neural Network Maximum Power Point Tracker for Solar Electric Vehicle*, Tsinghua Science & Technology, Vol. 10, No. 2, pp. 204-208, 2005.

[11] Hohm D.P., Ropp M.E., *Comparative Study of Maximum Power Point Tracking Algorithms*, Progress in Photovoltaics: Research and Applications 2003; 11:47-62 (DOI: 10.1002/pip.459).

[12] Noguchi T., Togashi S., and Nakamoto R., *Short-current pulse-based maximum power point tracking method for multiple photovoltaic-and converter module system*, IEEE Trans. Ind. Electron., vol. 49, no. 1, pp. 217–223, Feb. 2002.

[13] Enslin J.R., Wolf M.S., Snyman D.B., Sweigers W., *Integrated photovoltaic maximum power point tracking converter,* IEEE Trans. Ind. Electron., 1997, 44(6): 769-773.

[14] Baghat A.B.G, et al, *Maximum power point tracking controller for PV systems using neural networks,* Renewable Energy 30 (2005) 1257-1268, Elsevier Ltd. 2004.

[15] Bose B. K, *Modern Power Electronics and AC Drives*, University of Tennessee, Prentice Hall, 2002, Chapter 11.

[16] B. Tomescu, H.F. VanLandingham, *Improved Large Signal Performance of Paralleled DC/DC Converters Current Sharing Using Fuzzy Logic Control*, IEEE Trans. PE, vol. 14 No. 3, May 1999, pp. 573-577.

[17] Lee C. C, *Fuzzy logic in control systems: Fuzzy Logic Controller- Part I*, IEEE Transactions on Systems Man and Cybernetics Vol. 20, NO.2, March/April 1990, pages 404-417

[18] Lee C.C, *Fuzzy logic in control systems: Fuzzy Logic Controller- Part II*, IEEE Transactions on Systems, Man and Cybernetics, Volume: 20, NO.2, March/April 1990, pages 418-435.

[19] Manwell J.F, Rogers A. Hayman G., Avelar C.T., McGowan J.G., *Hybrid2 – A Hybrid System Simulation Model - Theory Manual*, Renewable Energy Research Laboratory, Department of Mechanical Engineering, University of Massachusetts,1998, pp 105-117.

URL:http://www.ecs.umass.edu/mie/labs/rerl/hy2/theory/pdf/complete.pdf

(date: 12th Aug. 2005)

[20] Tyagi, M.S., *Introduction to Semiconductor Materials and Devices*, John Wiley & Sons, 1991.

[21] Anca D. Hansen et al, *Models for a Stand-Alone PV system,* Risø National Laboratory, Roskilde, Norway, December 2000.

URL: http://www.risoe.dk/solenergi/rapporter/pdf/sec-r-12.pdf (date: 15th Nov. 2005)

[22] Mathworks Corporation, *Simulink Reference Manual,* Matlab Release 13 documentation. URL:

http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/slref.pdf (date: 23rd Sep. 2005)

[23] BP Solar Global Marketing , *BP Solar SX 75TU PV module data sheet*, 2002.

URL: http://ww.solardepot.com/pdf/BPSX75TU.pdf (date: 20th Oct. 2005)

[24] R.W. Erickson, D. Makasimovic, *Fundamentals of Power Electronics*, 2nd edition, Kluwer Academic Publishers, 2001, page 311.

[25] Mohan N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

[26] Middlebrook R. D., Cuk, S. (1976), *A general unified approach to modeling switching converter power stages*, IEEE Power Electronics Specialists Conference Record, pp. 18-34, 1976.

[27] Rashid Muhammad H., *Power Electronics, Circuits, Devices, and Applications*, Prentice-Hall India, 3rd Edition, pp. 186-204, 2004.

[28] Coilcraft Incorporation, *PCV-2-274-10L inductor datasheet*,

URL: http://www.coilcraft.com/pdfs/pcv.pdf (Date: 20th Jan. 2006)

[29] Vishay Semiconductors, *SB530 diode datasheet*,

URL: http://www.ortodoxism.ro/datasheets/vishay/sb520.pdf (Date: 20th Jan. 2006)

[30] Fairchild Semiconductor Inc., IRFZ24N datasheet,

URL: http://www.irf.com/product-info/datasheets/data/irfz24n.pdf (Date: 20th Jan. 2006)

[31] ST Microelectronics, *SG3525A datasheet*,

URL: http://www.st.com/stonline/books/pdf/docs/4286.pdf. (Date: 20th Jan. 2006)

[32] Measurement Computing Corp., *USB-1208FS user's guide*,

URL: http://www.measurementcomputing.com/PDFmanuals/usb-1208fs.pdf (Date: 20th Jan. 2006)

[33] National Semiconductors, LM358 datasheet,

URL: http://www.national.com/ds/LM/LM158.pdf (Date: 20th Jan. 2006)

[34] Kenital ,14W solar panel datasheet,

URL: http://www.kenital.com/category.asp?ID=8 (Date: 20th Jan. 2006)

APPENDICES

A-1 S-Functions

1 Buck-Boost Converter S-function

```
function [sys,x0,str,ts] =buck_boost_dc_sfun(t,x,u,flag,Rind,Rc,L,C,Vd,Rd,Rt)
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage of the S-function.
%
switch flag,
  % Initialization %
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  % Outputs %
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u,Rind,Rc,L,C,Vd,Rd,Rt);
  % Unhandled flags %
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,4, and 9 are not used, so return an emptyu
  % matrix
  case { 1,2,4, 9 }
    sys=[];
  % Unexpected flags (error handling)%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
%
%=============================================================
====================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.

%%=============================================================
function [sys,x0,str,ts] = mdlInitializeSizes()
```

```
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 6;
sizes.NumInputs      = 3;
sizes.DirFeedthrough = 1;   % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  =[];
ts  =[-1 0];   % sample time

% end mdlInitializeSizes
%=============================================================
====================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================

function sys = mdlOutputs(t,x,u,Rind,Rc,L,C,Vd,Rd,Rt)
D=u(2); %duty cycle
D1=1-D;

R=u(3);
U=[u(1);Vd]; %inputs

% update state space matrices
A=[-1/L*(D*Rt+Rind+D1*R*Rc/(R+Rc)) D1*R/(L*(R+Rc));-D1*R/(C*(R+Rc))
-1/(C*(R+Rc))];
B=[D/L -D1/L;0 0];
Cm=[D1*R*Rc/(R+Rc) -R/(R+Rc);D 0];

X=-inv(A)*B*U; % state vector
Y=Cm*X; % output vector

Pin=U(1)*Y(2); %input power
Po=(Y(1))^2/R; %output power
sys=[X',Y',Pin,Po];
%end
% end mdlOutputs
```

2 PV Module Voltage S-function

```
function [sys,x0,str,ts] = module_voltage(t,x,u,flag)
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage of the S-function.
%
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end


%
```

```
%=============================================================
====================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================
====================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;  % dynamically sized
sizes.NumInputs      = -1;  % dynamically sized
sizes.DirFeedthrough = 1;   % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [0];   % inherited sample time

% end mdlInitializeSizes
%=============================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================
%u(1)=Rs, u(2)=A, u(3)=IG(light-generated current), u(4)=Io (reverse saturation
%current), u(5)=Im (module current)
%====================
function sys = mdlOutputs(t,x,u)
if u(3)<=(u(5)-u(4)) sys=[0];
else sys=u(2)*log((u(3)-u(5))/u(4)+1)-u(5)*u(1);
end

% end mdlOutputs
```

A-2 Fuzzy Logic Controller C++ Files

1. fuzzy.h

```
// fuzzy.h: interface for the fuzzy class.
//
//////////////////////////////////////////////////////////////////////

#if
!defined(AFX_FUZZY_H__BDEC3FB0_C8BC_486F_A31A_EE98F050CED0__
INCLUDED_)
#define
AFX_FUZZY_H__BDEC3FB0_C8BC_486F_A31A_EE98F050CED0__INCLU
DED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "mf.h"
#include "ruleIndex.h"

const unsigned int
MAXLEN=20,NUM_INPUTS=2,NUM_OUTPUTS=1,NUM_ROW=5,NUM_CO
L=11,NUM_RULES=NUM_ROW*NUM_COL;
const float MF_RANGE_MIN=-1.0,MF_RANGE_MAX=1.0;

//char NAME[]="MPPT",TYPE[]="MAMDANI";

enum {NBr=0,NSr,ZEr,PSr,PBr};//row id for rule base
enum
{NBc=0,NMc,NMMc,NSc,NSSc,ZEc,PSSc,PSc,PMMc,PMc,PBc};//collumn id
for rule base
enum  {CP,CD};

class fuzzy
{
private:

                char name,type;
                unsigned int numInputs,numOutputs,numRules;
                int fired;
```

```
                //float ccd;//current change in duty cycle;

                mf cD[NUM_COL]; // MFs for change in duty
                mf cP[NUM_ROW];// MFs for change in power
                mf rbMf[NUM_ROW][NUM_COL];// rule base


public:
                fuzzy();
                virtual ~fuzzy();
   void fuzzify(float,float);
                float defuzzify();
                unsigned int get_numInputs();
                unsigned int get_numOutputs();
                unsigned int get_numRules();
                int get_fired();
                float get_ruleFs(int,int);
                float get_ruleWt(int,int);
                ruleIndex firedRule[NUM_RULES];
};

#endif //
!defined(AFX_FUZZY_H__BDEC3FB0_C8BC_486F_A31A_EE98F050CED0__
INCLUDED_)
```

2. mf.h

```
// mf.h: interface for the mf class.
//
//////////////////////////////////////////////////////////////////////

#if
!defined(AFX_MF_H__792C6BA9_4F58_4026_A780_6A22F5E9728C__INCLU
DED_)
#define
AFX_MF_H__792C6BA9_4F58_4026_A780_6A22F5E9728C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

const float STEP=0.01;
```

```
class mf
{
		float a,b,c,d;//boundaries
		bool id;// id=1 when fired else 0
		float fs; //firing strength
		float wt; //rule weight if mf is used for output
public:
		mf();
		float get_a();
		float get_b();
		float get_c();
		float get_d();
		float get_fs();
		void set_fs(float,int=0);//set firing strength
		void set_wt(float);//set weight
		float get_centroid();
		float get_uz(float,float,float,float);
		float get_area();
		void set_param(float,float,float,float);
		void set_param(float,float,float);
		void set_id();
		void reset_id();
		bool get_id();
		float get_wt();// return weight
		void image(mf);
		virtual ~mf();

};

#endif //
!defined(AFX_MF_H__792C6BA9_4F58_4026_A780_6A22F5E9728C__INCLU
DED_)
```

3. ruleIndex.h

```
// ruleIndex.h: interface for the ruleIndex class.
//
//////////////////////////////////////////////////////////////////////
```

```
#if
!defined(AFX_RULEINDEX_H__72487254_97C7_47AE_AAE5_284132207673
__INCLUDED_)
#define
AFX_RULEINDEX_H__72487254_97C7_47AE_AAE5_284132207673__INCL
UDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class ruleIndex
{
                int i,j;
public:
                ruleIndex();
                void set_i(int);
                void set_j(int);
                int get_i(){return i;};
                int get_j(){return j;};
                virtual ~ruleIndex();

};

#endif //
!defined(AFX_RULEINDEX_H__72487254_97C7_47AE_AAE5_284132207673
__INCLUDED_)
```

4. fuzzy.cpp

```
// fuzzy.cpp: implementation of the fuzzy class.
//
//////////////////////////////////////////////////////////////////////

#include <float.h>
#include <MSVCRT.LIB>
#include "stdafx.h"
#include "FuzzyController.h"
#include "fuzzy.h"

#ifdef _DEBUG
#undef THIS_FILE
```

```
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

fuzzy::fuzzy()
{

numInputs=NUM_INPUTS;numOutputs=NUM_OUTPUTS;numRules=NUM_R
ULES;

//initialize MFs for change in duty (cD)
cD[PBc].set_param(0.5,0.7,1,1); // parameters for trap MF PB
cD[PMc].set_param(0.3,0.5,0.7); // parameters for tri MF PM
cD[PMMc].set_param(0.2,0.3,0.4); // parameters for tri MF PMM
cD[PSc].set_param(0.1,0.2,0.3);// parameters for tri MF PS
cD[PSSc].set_param(0.0,0.1,0.2);// parameters for tri MF PSS
cD[ZEc].set_param(-0.1,0,0.1);// parameters for tri MF ZE
cD[NBc].image(cD[PBc]);
cD[NMc].image(cD[PMc]);
cD[NMMc].image(cD[PMMc]);
cD[NSc].image(cD[PSc]);
cD[NSSc].image(cD[PSSc]);

//initialize MFs for change in power (cP)
cP[PBr].set_param(0.25,0.5,1,1); // parameters for trap MF PB
cP[PSr].set_param(0.0,0.25,0.5); // parameters for tri MF PS
cP[ZEr].set_param(-0.1,0,0.1);// parameters for tri MF ZE
cP[NSr].image(cP[PSr]);
cP[NBr].image(cP[PBr]);


//initialize rule base
rbMf[NBr][NBc]=cD[PMc];//rule no.1
rbMf[NBr][NMc]=cD[PMMc];//rule no.2
rbMf[NBr][NMMc]=cD[PSc];//rule no.3
rbMf[NBr][NSc]=cD[PSSc];//rule no.4
rbMf[NBr][NSSc]=cD[PSSc];//rule no.5
rbMf[NBr][ZEc]=cD[NBc];//rule no.6
rbMf[NBr][PSSc]=cD[NSSc];//rule no.7
```

rbMf[NBr][PSc]=cD[NSSc];//rule no.8
rbMf[NBr][PMMc]=cD[NSc];//rule no.9
rbMf[NBr][PMc]=cD[NMMc];//rule no.10
rbMf[NBr][PBc]=cD[NMc];//rule no.11

rbMf[NSr][NBc]=cD[PMc];//rule no.12
rbMf[NSr][NMc]=cD[PMMc];//rule no.13
rbMf[NSr][NMMc]=cD[PSc];//rule no.14
rbMf[NSr][NSc]=cD[PSSc];//rule no.15
rbMf[NSr][NSSc]=cD[PSSc];//rule no.16
rbMf[NSr][ZEc]=cD[NSc];//rule no.17
rbMf[NSr][PSSc]=cD[NSSc];//rule no.18
rbMf[NSr][PSc]=cD[NSSc];//rule no.19
rbMf[NSr][PMMc]=cD[NSc];//rule no.20
rbMf[NSr][PMc]=cD[NMMc];//rule no.21
rbMf[NSr][PBc]=cD[NMc];//rule no.22

rbMf[ZEr][NBc]=cD[NBc];//rule no.23
rbMf[ZEr][NMc]=cD[NMc];//rule no.24
rbMf[ZEr][NMMc]=cD[NMMc];//rule no.25
rbMf[ZEr][NSc]=cD[NSc];//rule no.26
rbMf[ZEr][NSSc]=cD[NSSc];//rule no.27
rbMf[ZEr][ZEc]=cD[ZEc];//rule no.28
rbMf[ZEr][PSSc]=cD[PSSc];//rule no.29
rbMf[ZEr][PSc]=cD[PMc];//rule no.30
rbMf[ZEr][PMMc]=cD[PMMc];//rule no.31
rbMf[ZEr][PMc]=cD[PMc];//rule no.32
rbMf[ZEr][PBc]=cD[PBc];//rule no.33

rbMf[PSr][NBc]=cD[NMc];//rule no.34
rbMf[PSr][NMc]=cD[NMMc];//rule no.35
rbMf[PSr][NMMc]=cD[NSc];//rule no.36
rbMf[PSr][NSc]=cD[NSSc];//rule no.37
rbMf[PSr][NSSc]=cD[NSSc];//rule no.38
rbMf[PSr][ZEc]=cD[PSc];//rule no.39
rbMf[PSr][PSSc]=cD[PSSc];//rule no.40
rbMf[PSr][PSc]=cD[PSSc];//rule no.41
rbMf[PSr][PMMc]=cD[PSc];//rule no.42
rbMf[PSr][PMc]=cD[PMMc];//rule no.43
rbMf[PSr][PBc]=cD[PMc];//rule no.44

rbMf[PBr][NBc]=cD[NMc];//rule no.45
rbMf[PBr][NMc]=cD[NMMc];//rule no.46

```
rbMf[PBr][NMMc]=cD[NSc];//rule no.47
rbMf[PBr][NSc]=cD[NSSc];//rule no.48
rbMf[PBr][NSSc]=cD[NSSc];//rule no.49
rbMf[PBr][ZEc]=cD[PSc];//rule no.50
rbMf[PBr][PSSc]=cD[PSSc];//rule no.51
rbMf[PBr][PSc]=cD[PSSc];//rule no.52
rbMf[PBr][PMMc]=cD[PSc];//rule no.53
rbMf[PBr][PMc]=cD[PMMc];//rule no.54
rbMf[PBr][PBc]=cD[PMc];//rule no.55

// set rule weights for rules whose weights are not equal to 1

rbMf[NBr][ZEc].set_wt(0.5);//rule no.6
rbMf[NSr][ZEc].set_wt(0.5);//rule no.17

rbMf[ZEr][NBc].set_wt(0.5);//rule no.23
rbMf[ZEr][NMc].set_wt(0.5);//rule no.24
rbMf[ZEr][NMMc].set_wt(0.5);//rule no.25
rbMf[ZEr][NSc].set_wt(0.5);//rule no.26

rbMf[ZEr][NSSc].set_wt(0.25);//rule no.27
rbMf[ZEr][ZEc].set_wt(0.25);//rule no.28
rbMf[ZEr][PSSc].set_wt(0.25);//rule no.29

rbMf[ZEr][PSc].set_wt(0.5);//rule no.30
rbMf[ZEr][PMMc].set_wt(0.5);//rule no.31
rbMf[ZEr][PMc].set_wt(0.5);//rule no.32
rbMf[ZEr][PBc].set_wt(0.5);//rule no.33

rbMf[PSr][ZEc].set_wt(0.5);//rule no.39
rbMf[PBr][ZEc].set_wt(0.5);//rule no.50

}

unsigned int fuzzy::get_numInputs() { return numInputs;} // return number of
inputs

unsigned int fuzzy::get_numOutputs() { return numOutputs;} // return number of
outputs

unsigned int fuzzy::get_numRules() { return numRules;} // return number of rules
```

```
void fuzzy::fuzzify(float ccP, float ccD) {

		int j;

		//reset all MF ids
		for (j=NBr;j<=PBr;j++)
		{
				cP[j].reset_id();
		}

		for (j=NBc;j<=PBc;j++)
		{
		  cD[j].reset_id();
		}


//scale inputs to MF allowed range
		if (ccP<MF_RANGE_MIN) ccP=MF_RANGE_MIN;
   if (ccP>MF_RANGE_MAX) ccP=MF_RANGE_MAX;
   if (ccD<MF_RANGE_MIN) ccD=MF_RANGE_MIN;
   if (ccD>MF_RANGE_MAX) ccD=MF_RANGE_MAX;

//test set membership for current change in power (cP) parameter
		for (j=NBr; j<=PBr;j++)
				if ((ccP >= cP[j].get_a())&&(ccP <= cP[j].get_d()))
				{
				cP[j].set_fs(ccP);
				if (cP[j].get_fs()>0) cP[j].set_id();
				}

//test set membership for change in duty (ccD) parameter
		for (j=NBc; j<=PBc;j++)
		if ((ccD >= cD[j].get_a())&&(ccD <= cD[j].get_d()))
		{
				cD[j].set_fs(ccD);
				if (cD[j].get_fs()>0) cD[j].set_id();
		}

}

float fuzzy::defuzzify() {

		int i,j;
```

```
		fired=0;
		float ccD,area,areasum=0,moment,momentsum=0,fs,fs1,fs2;

		for (i=NBr;i<=PBr;i++)
				for (j=NBc;j<=PBc;j++)
				if ((cP[i].get_id())&&(cD[j].get_id()))
				{
						fired+=1;
						firedRule[fired-1].set_i(i);
						firedRule[fired-1].set_j(j);
						fs1=cP[i].get_fs();
						fs2=cD[j].get_fs();
						if (fs1>fs2) fs=fs2;
						else fs=fs1;
						rbMf[i][j].set_fs(fs,1);

		moment=rbMf[i][j].get_centroid();

		area=rbMf[i][j].get_area()*rbMf[i][j].get_wt();
						areasum+=area;

		momentsum+=moment*area;

						}

if ((areasum ==0)||(momentsum ==0))
		ccD=0.0;
else ccD=momentsum/areasum;

return ccD;

}

int fuzzy::get_fired() {
		return fired;
}

float fuzzy::get_ruleFs(int i,int j) {
		return rbMf[i][j].get_fs()*rbMf[i][j].get_wt();
}

float fuzzy::get_ruleWt(int i,int j) {
		return rbMf[i][j].get_wt();
```

```
}

fuzzy::~fuzzy(){}
```

5. mf.cpp

```
// mf.cpp: implementation of the mf class.
//
//////////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "FuzzyController.h"
#include "mf.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

mf::mf(){
   a=0;
                b=0;
                c=0;
                d=0;
                fs=0;
                id=0;
                wt=1;
}

mf::~mf(){}

float mf::get_a() {
                return a;
}

float mf::get_b() {
                return b;
```

```
}

float mf::get_c() {
                return c;
}

float mf::get_d() {
                return d;
}

bool mf::get_id() {
                return id;
}

float mf::get_wt() {
                return wt;
}

void mf::set_fs(float val,int id ) {
                if (id==0) {
                if ((val<a)||(val>d)) fs=0;
                else if ((val < b)&&(b != a)) fs=(val-a)/(b-a);
                else if ((val > c)&&(c != d)) fs=(val-d)/(c-d);
   else fs=1;
                }
                else fs=val;
}


float mf::get_fs() {
                return fs;
}


float mf::get_uz(float z,float b1,float c1,float y) {
                float x;
                if ((z<=a)||(z>d)) x=0;
                else if ((z < b1)&&(b1 != a)) x=(z-a)/(b1-a);
                else if ((z > c)&&(c != d)) x=(z-d)/(c1-d);
   else x=y;
                return x;
}
```

```
float mf::get_centroid()
{
   if (fs==0)
                    {
                                    AfxMessageBox("The firing strength of MF is zero
when calculating CENTROID ");
                                    return 0;
                    }

                   float b1,c1,num=0,den=0,z=a+STEP,uz=0;
                    b1=a+fs*(b-a);
                    c1=d+fs*(c-d);

                    while (z<=d)
          {

                                    uz=get_uz(z,b1,c1,fs);
                                    num+=z*uz;
                                    den+=uz;
                                    z+=STEP;
                    }

                    if (den==0)
                    { AfxMessageBox("Division by zero when calculating
CENTROID");
                      return 0;
                    }
                                    return num/den;
}

float mf::get_area()

{
float b1,c1,area;
                    b1=a+fs*(b-a);
                    c1=d+fs*(c-d);
                    area=0.5*fs*(d-a+c1-b1);
                    return area;
}

void mf::set_param(float a1,float b1,float c1,float d1) {
                    a=a1; // trap mf
```

```
                b=b1;
                c=c1;
                d=d1;
}

void mf::set_param(float a1,float b1,float c1) {
                a=a1; // tri mf
                b=b1;
                c=b1;
                d=c1;
}

void mf::set_id() {
                id=1;
}

void mf::set_wt(float w) {
                wt=w;
}

void mf::reset_id() {
                id=0;
}

void mf::image(mf val) {
                set_param(-val.d,-val.c,-val.b,-val.a);
}
```

6. ruleIndex.cpp

```
// ruleIndex.cpp: implementation of the ruleIndex class.
//
//////////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "FuzzyController.h"
#include "ruleIndex.h"
#include "fuzzy.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
```

```
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

ruleIndex::ruleIndex()
{
		i=-1;
		j=-1;

}

void ruleIndex::set_i(int x)
{
		i=x;
}

void ruleIndex::set_j(int x)
{
		j=x;
}

ruleIndex::~ruleIndex()
{

}
```

A-3 MPPT Circuit Diagram