4EE33: ELECTRICAL PROJECT

SOLAR MAXIMUM POWER POINT TRACKING AND FORECASTING

A

Report

Submitted in partial fulfillment of the

4EE33: Electrical Project

course for the

Award of B. Tech (Electrical) Degree

By

Het Patel (18EE022)

Galav Bhatt (18EE027)

Jash Patadia (18EE030)

Project Group No: 4

Under the guidance of

Dr. Akshay Pandya

Associate Professor

Electrical Engineering Department

Electrical Engineering Department

Birla Vishvakarma Mahavidyalaya (Engineering College)

(An Autonomous Institute)

Vallabh Vidyanagar: 388120

(May 2022)

Birla Vishvakarma Mahavidyalaya (Engineering College)

An Autonomous Institution

Vallabh Vidyanagar 388120

Certificate

Certified that, work presented by **Mr. Het Patel** bearing ID. No:**18EE022** & Enrollment No:180070109036, **Mr. Galav Bhatt** bearing ID. No: **18EE027** & Enrollment No: 180070109002, **Mr. Jash Patadia** bearing ID. No: **18EE030** & Enrollment No: 180070109040 in the Project entitled **SOLAR MAXIMUM POWER POINT TRACKING AND FORECASTING** for 4EE33: Electrical Project of B. Tech (Electrical) Final year is satisfactory to warrant its acceptance for the purpose of evaluation. It is to be understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusions drawn but approve the study for the purpose for which it is submitted.

Dr. Akshay Pandya (Project Faculty Guide)
Associate Professor,
Electrical Engineering Department

Dr. Rashesh Mehta
Head of Department,
Electrical Engineering Department

Date:

Acknowledgement

We would like to take this opportunity to express our sincere gratitude towards our project guide **Dr. Akshay Pandya**, for his unwavering support, exemplary guidance and monitoring throughout the course of this project. His guidance and blessings made it easier to implement the project in a most efficient way, which resulted in the smooth completion of the project. Additionally, we are very grateful to **Dr. Rashesh Mehta** for valuable support and **Prof. Swapnil Arya**, whose invaluable suggestions and ideas expanded our technical potential. Also, **Prof. Dharita Patel** for her guidance and valuable support throughout this project as our project coordinator.

We are also thankful to the Electrical Engineering Department, BVM Engineering College, for allowing us to use their infrastructure. We express our gratitude towards the lab assistants and technicians for providing all the necessary equipment needed for the project. We are glad to have moral support and technical expertise from our colleagues.

Finally, we would like to thank our institute for providing us with the liberty and platform to conduct research at their infrastructure without any constraints; through the institute's support, the impactful work would not have been executed.

Het Patel (18EE022)
Galav Bhatt (18EE027)
Jash Patadia (18EE030)

Abstract

For many years, fossil fuels have filled the void of most of the energy demands. Fossil fuels available in nature are also limited and since to fulfil this large demand, they will work out in only few decades. Hence, an alternative to fossil fuels is needed which is available in the form of sun, wind, tides, energy from earth, etc. Of all this, solar energy is available in huge amount daily. Large amount of progress has been made to convert electrical energy from solar energy efficiently using different techniques. One such method is developing a converter which can extract Maximum Power from solar panel. Thus, the aim of the project is to design MPPT using Buck Converter which will lead us to extracting maximum power that a solar panel can produce at given conditions of temperature and irradiance. To harvest maximum power from a PV module, MPPT based on P&O algorithm is implemented for the converter. Additionally, a machine learning based system is built on the data collected from sensor, which forecasts the solar MPPT system output based on temperature and irradiance. The entire project is simulated in MATLAB and likewise the hardware model is prepared.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Definition
MPPT	Maximum Power Point Tracking
P & O	Perturb & Observe algorithm
V_{mp}	Maximum panel voltage
I_{mp}	Maximum panel current
V_{oc}	Open circuit voltage
I_{sc}	Short circuit current
R_s	Series resistance
R_{sh}	Shunt resistance
I_l	Photovoltaic current
I_d	Diode current
FF	Fill factor
H	Efficiency
P_{in}	Input solar radiation in w/m^2
A	Area of solar cell in m^2
Θ	Angle of light falling on solar panel
T	Temperature
Irr	Irradiance
W_p	Maximum power
STC	Standard Test Conditions
ΔI_l	Estimated inductor ripple current
f_s	Switching frequency of converter
ΔV_{out}	Desired output ripple voltage
V_{in}	Typical input voltage
V_{out}	Desired output voltage
V_{in}	Typical input voltage
C	Capacitance
L	Inductance
Y	Target variable
$B_0, B_1, B_2..$	Slope/Weights of variables
$X_1, X_2, X_3..$	Variables
E	Error
MAE	Mean absolute error
MSE	Mean squared error
RMSE	Root mean squared error
R^2	R Squared
Yi	Actual value
$\hat{y}_i$	Predicted value
N	Total number of observations

Chapter 1: Introduction

Solar maximum power point tracking is a concept initiated in order to derive maximum power from a solar PV system with respect to weather conditions. There are certain techniques used in order extract maximum power from solar PV system. Before that, it becomes necessary to understand the basic functionalities and terms associated with solar cell.

1.1Solar Cell Construction and its Equivalent Circuit

A solar cell is defined as an electrical device which converts light energy into electricity. It consists of a front contact at the top, emitter – base junction or p – n junction diode in middle and back contact at the bottom. Numerous solar cells are connected to make solar PV modules. Several, solar PV modules are connected to form solar PV array which can be connected in series or parallel as per the power requirement. It is essential to study the characteristics of solar cell in order to understand the working of solar PV system. Sunlight falling on Earth is essentially photon bundles, with each photon in a bundle having a finite amount of energy. For, generation of electricity photons must be absorbed by solar cell and it depends upon the energy of photon and the band gap energy Solar cell equivalent circuit consists of diode, resistances and the connected load.

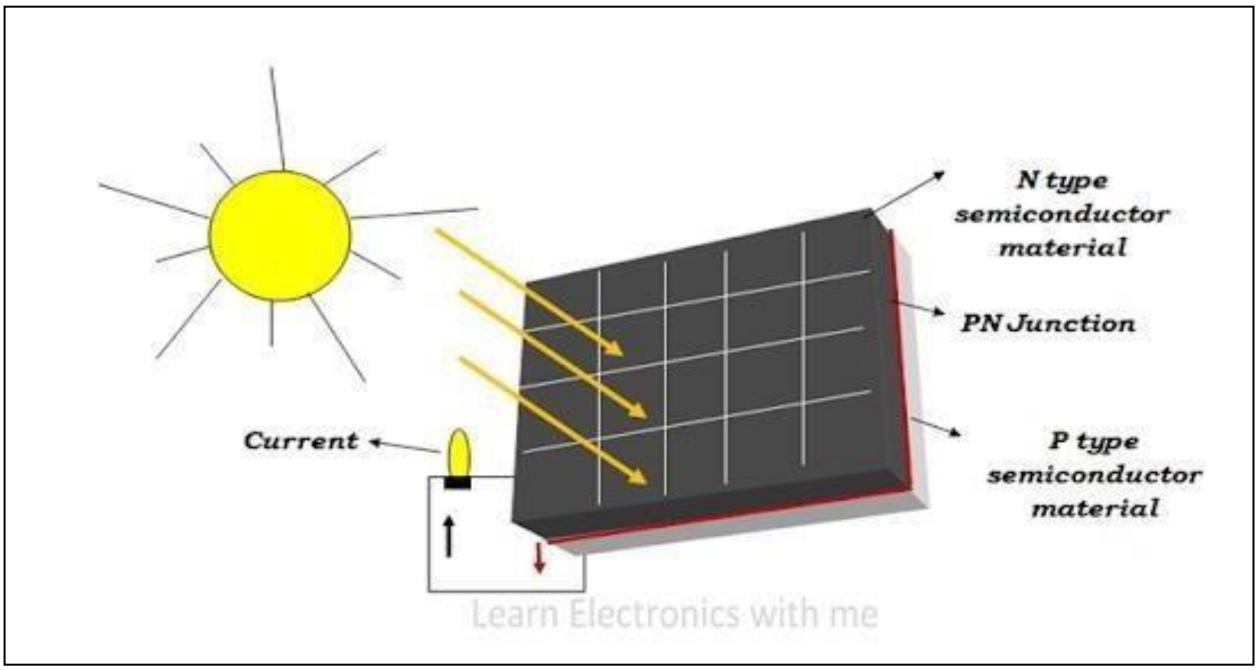

Figure 1.1 Solar cell construction

I_L → Photo current generated due to illumination of solar cell.

I_D → Diode Characteristics in dark (This is sink for the current), some current pass-through diode and doesn't reach the load.

Diode should be of good quality to reduce the leakage current, Resistance Rsh, Rs are there because the diode is not an ideal diode.

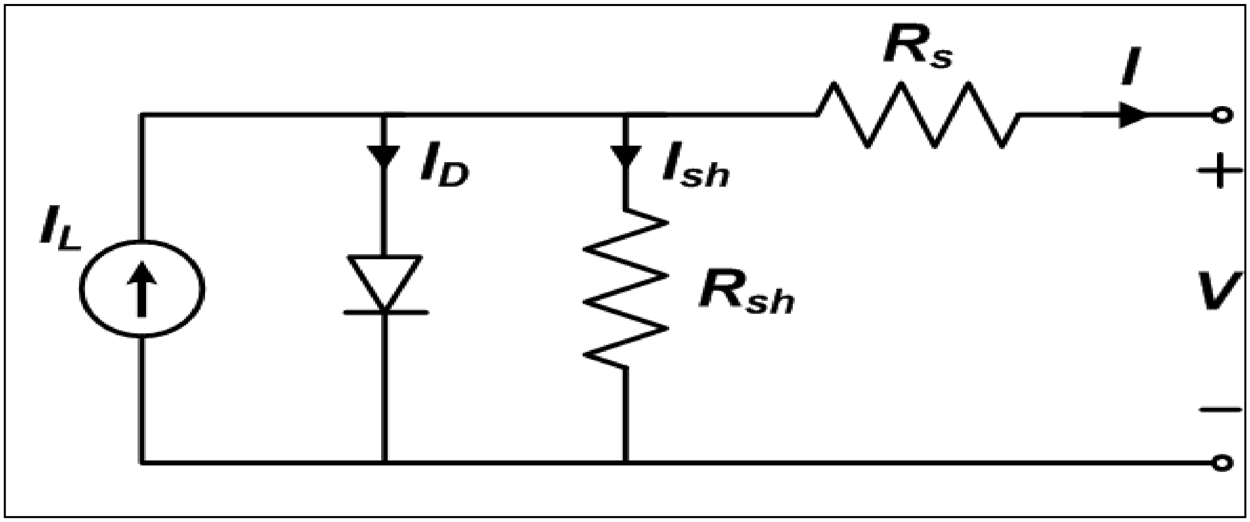

Figure 1.2 Solar cell equivalent circuit

Rs (Series Resistance) is the result of bulk resistance of semi-conductor, resistance of metal electrodes, contact resistance between semi-conductor material and metal electrode. It drops the output voltage as well as maximum output power.

Rsh (Shunt Resistance) is the result of leakage across the p-n junction around the edge, crystal defects, pin holes, impurity precipitates. It should be as high as possible otherwise it acts as a sink for the current and that will reuse power to the load. The diode acts as bypass diode and its useful in carrying the extra current when there is shadow at a certain part of the panel which act as a load to other part of the panel.

1.2 Working of a Solar Cell: -

Photons in the sunlight are absorbed by front face of semiconductor material used in solar cell.

As the photon gets absorbed into band gap, electron-hole pairs are generated. Electrons have a negative charge, whereas holes have a positive charge. When a solar PV system is connected to a load, electron-hole pairs near the junction become separated. Holes accumulate at the positive terminal (anode), while electrons accumulate at the negative terminal (cathode).

Electric potential is built at the terminals due to separation of positive and negative charges. Voltage developed at the terminals of a solar cell is used to drive the DC current in the circuit.

The amount of electricity generated by our solar PV system depends on various parameters. Hence, the electricity generated throughout the day is not constant.

There are several parameters of solar cell that determine the **effectiveness of sunlight to electricity conversion**. The list of solar cell parameters is given below: -

- **Short Circuit Current (Isc)**
  Short circuit current is the current through solar cell when the voltage across the solar cell is zero. Here the positive and negative terminals of the solar cell is short circuited. It depends upon the area of solar cells, number of photons, spectrum of incident light etc.

- **Open Circuit Voltage (Voc)**
  The maximum voltage that a solar panel can produce. Here both the terminals of solar cell are kept open. It mainly depends upon the cell technology and operating temperature.

- **Maximum Power Point**
  The maximum power that a solar panel can produce at STC (Standard Test Conditions). Higher the Pm (Maximum Power) better is the cell, A solar cell panel can operate at many current and voltage combinations. But a solar panel can produce maximum power only while operating at a certain voltage and current combination.
  Pm = Vm x Im
  Vm – Voltage at Maximum Power Point,
  Im – Current at Maximum Power Point.

- **Current at Maximum Power Point (Im)**
  This is the current produced by a solar cell when it is operating at full power. Im (Current at Maximum Power Point) will always be less than Isc (Short Circuit Current).

- **Voltage at Maximum Power Point (Vm)**
  This is the voltage produced by a solar cell when it is operating at full power. Vm (Voltage at Maximum Power Point) is always less than Voc (Open Circuit Voltage).

- **Fill Factor (FF)**
  Fill Factor is the ratio of area covered by Im – Vm rectangle and Isc – Voc rectangle. It is given in terms of percentage and higher its value, better will be the cell.
  Fill Factor (FF) = **Vm x Im / Voc x Isc**
  Vm → Voltage at Maximum Power Point in Volts.
  Im → Current at Maximum Power Point in Ampere.
  Voc → Open Circuit Voltage.
  Isc → Short Circuit Current.

- **Efficiency (η)**
  The efficiency of solar cell is defined as the maximum output power (Pm) divided by the input power (Pin). It is given in terms of percentage and it indicates that how much the radiation power gets converted into electricity.
  Efficiency (η) = **(Vm x Im / Pin x A) x 100.**
  Pin→ Input solar radiation at standard test condition is taken as 1000 watt per meter square.

Im→ Current at Maximum Power Point in Ampere.
Vm→ Voltage at Maximum Power Point Volt.
A→ Area of solar cell in Meter Square.

Solar cell parameters solely act upon the solar cell technology and weather conditions like Temperature, Irradiance, Wind speed, Humidity etc. The manipulations caused in solar cell parameters is the result of weather conditions and cell technology.

The solar panel manufacturers usually give all the solar panel parameters measured on the Standard Test Condition (STC). Hence, usually we obtain the power less than the given due to fluctuating weather conditions

1.3 Solar Cell V - I & P - V Graph

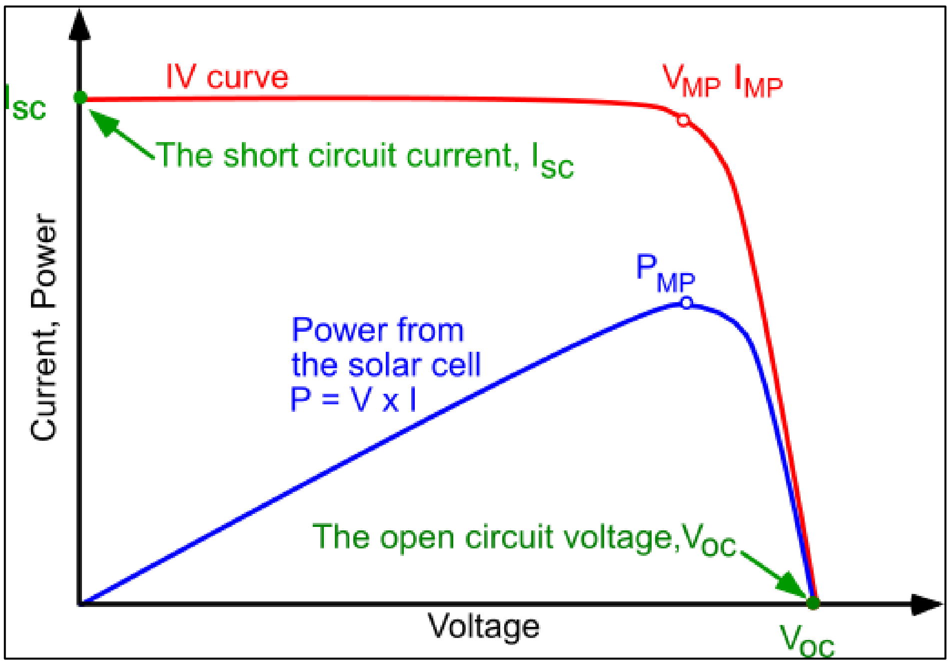

Figure 1. 3 V - I Characteristics of solar cell

The above graph represents the V – I Characteristics of solar cell, we can see that X – axis have voltage and Y – axis have current & power. Isc represents the short circuit current, Voc represents open circuit voltage, initially as the open circuit voltage increase the short circuit current remains constant and then after saturation level the current drops severely.

The output power graph is represented by blue line we can see that a particular point of voltage and current we get the maximum power which is denoted by P_{MP}. The graph moves up and down depending upon the weather conditions like temperature and Irradiance. As the temperature increases; voltage decreases, current increases up to 25°C and above those current decreases. With the increase in irradiance the output current increases and vice versa.

In order to extract maximum power from solar PV system we need to operate the system nearby the maximum power point voltage and maximum power point current. The main purpose of

Solar MPPT, is to operate the solar PV system around the voltage and current which offers the maximum power point. Different techniques can be used in order to extract maximum power from solar PV system.

There are certain environmental factors on which the solar p-v graph depends, like temperature, irradiance, humidity, wind speed, inclination of solar PV system etc.

1.4 Factors Affecting Electricity Generation of Solar Cell

- ✓ **The Conversion Efficiency (η)**

  Only some fraction of light gets converted into electricity, the ratio of electrical energy generated to the input light energy is referred as conversion efficiency. It is always fixed depending upon the material and the manufacturing process. Efficiency of a solar cell is given in terms of maximum power that solar cell can generate for a given input solar radiation.

  Efficiency (η) **= (Vm x Im / Pin x A) x 100.**

  Pin→ Input solar radiation at standard test condition is taken as 1000 watt per meter square.

  Im→ Current at Maximum Power Point in Ampere.

  Vm→ Voltage at Maximum Power Point Volt.

  A→ Area of solar cell in Meter Square.

- ✓ **The amount of light (Pin)**

  The amount of light falling on solar cell keeps on changing throughout the day, the electric current generated by solar cell directly depends upon the light falling on it. From, morning to evening as the sunlight on the solar cell increases, current increases as a result output power also increases. On the other hand, from afternoon to evening as the sunlight decreases, current decreases as a result the output power drops down. The voltage is relatively not affected by solar radiation intensity.

- ✓ **Solar Cell Area (A)**

  The current output is directly proportional to solar cell area, when area is large the amount of electric current generated by it will be large. When we divide the current generated by solar cell and its area we get current density (J), the current density is fixed for a solar cell, depending upon its manufacturing technology and material.

  $J_{sc} = I_{SC}/A$ (mA/cm^2)

  J_{SC} → Current Density.

  I_{SC} → Short Circuit Current.

  A → Area of solar cell.

  We can clearly see from the above equation, that as area of solar cell increases the I_{SC} increase, because the current density is fixed only area of solar cell is variable. As I_{SC} increases the I_m also moves up.

- **Angle of Light Falling on Solar Cell (θ)**
  The angle of sunlight with respect to the solar cell has a significant impact on output power; solar cells produce the most power (for a given light intensity) when sunlight falls perpendicular to their surfaces.. When the light does not fall perpendicular to the surface, It always gives less power as some part of light gets reflected and only a small part of light gets utilize to generate electricity.

- **Solar Cell Operating Temperature(T)**
  The solar cell output voltage, efficiency and output power depends upon the cell temperature. Solar cells are encapsulated within the glass which results in heating of solar cells, As the temperature increases the output voltage decreases as a result, the output power and efficiency of solar cell gets degraded. The output current increases up to 25 °C, but as we move above that ambient temperature the current also decreases and usually in countries like India the average temperature lies between 30°C - 35°C.

1.5 Types of Solar Panel

- Polycrystalline - Polycrystalline are solar panels that consist of several crystals of silicon in a single PV cell.
- Monocrystalline - A monocrystalline solar panel is a solar panel comprising monocrystalline solar cells. These cells, like semiconductors, are made from a cylindrical silicon ingot grown from a single crystal of high purity silicon..
- Amorphous silicon (a-Si) - The most developed thin-film solar cell is amorphous silicon solar cells.1.6 Hardware and Software Requirements

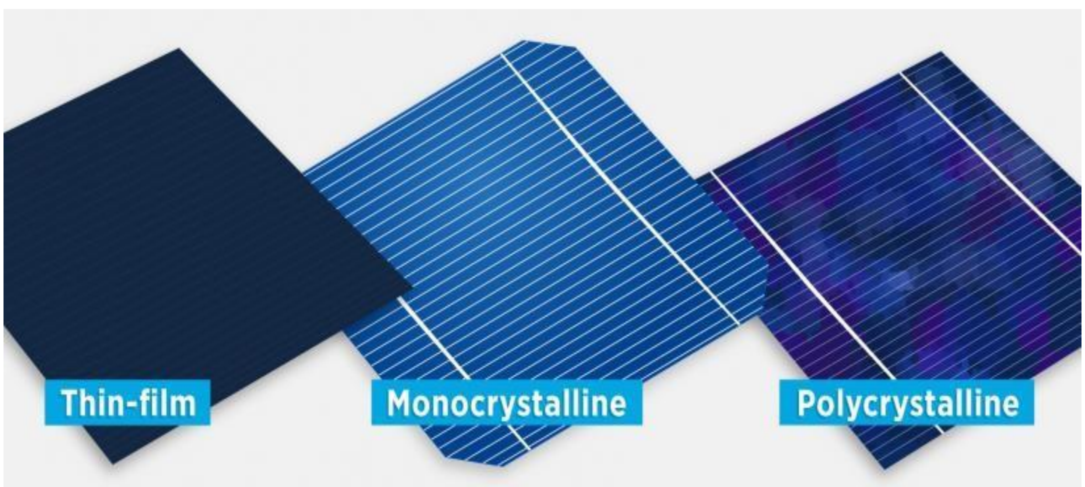

Figure 1.4 Solar panel types

Hardware Requirements: -

Sr. No.	NAME	Quantity
POWER SUPPLY		
1	Solar panel(12v 40W)	1
2	Variable dc supply	1
SENSORS		
3	Current sensor(ACS712)	2
4	22K ohms resistor	2
5	80K ohms resistor	2
6	DHT 11 sensor	1
7	LDR module	1
MONITORING & DATA COLLECTION		
8	SD card module	1
9	16x2 LCD display	1
10	SD card	1
PROCESSING UNIT		
11	Arduino UNO	1
BUCK CONVERTER		
12	220uF 25V capacitor	2
13	1100uF 25V capacitor	2
14	IRFZ44n MOSFET	4
15	High optocoupler	2
16	ferite core	1
17	Inductor coil(4m)	1
18	Heat sink	1
MISCELLANEOUS		
19	Buck converter	1
20	1K ohms potentiometer	1
21	DC bulb(12V 35W)	1
22	Solar panel stand	1

Figure 1.5 Project components list

Software Requirements: -

- MATLAB Simulink.
- Arduino IDE.
- Python Jupyter Notebook.
- Streamlit library.
- Scikit Learn, NumPy, Pandas and Matplotlib.

Chapter 2: Project Definition, Identification and Objective

2.1 Definition

Recently, renewable energy technology has been swiftly developed where it has an important role in clean energy application. Solar energy is an important type of renewable energy because it directly produces electrical energy using PV modules and an MPPT algorithm to maximize output power. The objective of obtaining MPP in PV systems is to regulate the actual operating voltage of PV panels to the voltage at maximum power point. Majorly, solar MPPT is used as a charge controller for battery charging at maximum voltage.

2.2 Identification

The conversion of solar radiation into electricity by PV cells depends on cell type and is typically between 5% and 20%. Hence, most of the absorbed energy is wasted. When the solar energy is converted to electrical energy the basic parameter that comes into picture is the operating voltage of load and considering the p-v graph of the panel we cannot always achieve the maximum power by directly loading the solar panel thus we need an intermediate device known as MPPT (Maximum Power Point Tracking). So, the purpose of creating an intermediate stage called MPPT charge controller is to convert most of this energy into useful energy and provide maximum power. The main drawback of directly loading the solar panel is that the solar panel operates at the voltage which is required by the load and hence we are not able to extract the maximum power which is generated by the solar panel. Thus, to solve this problem we need a intermediate device or stage which is MPPT. Here, MPPT helps in maintaining the solar voltage close to its maximum power voltage. Further, the forecasting of maximum power will be done using the data collected from the MPPT with the help of machine learning techniques.

2.3 Objective

An MPPT (Maximum Power Point Tracking Device) is an electronic DC to DC converter that acts as intermediate stage for extracting maximum power out of solar panel and using it to charge a battery at the required voltage which may not be equal to maximum voltage thus, solar MPPT device increases the overall efficiency of solar PV system by extracting maximum power that a solar panel will be able to generate depending upon the respective weather parameters.

MPPT solar charge controller is required for any solar power system that needs to extract the most power from the PV module; it forces the PV module to operate at a voltage close to the maximum power point to draw the most available power.

Here, the project involves the use of DC – DC buck converter in solar maximum power point device. Initially the design of entire system is done on MATLAB Simulink and further the hardware implementation for the same is done. Additionally, a machine learning based user interface system will be built where the power operator can forecast corresponding solar MPPT output depending upon the temperature and irradiance at that instance.

Along with DC – DC buck converter, Arduino Uno is used for taking decisions of operating duty cycle of buck converter which will be achieved by the Perturb & Observe Algorithm which is loaded in Arduino Uno.

The Perturb & Observe Algorithm is used in order to achieve the respective voltage and current point where the system achieves maximum power point. The output of Perturb & Observe will be the duty cycle which will be as input to switching device (MOSFET) of DC – DC buck converter.

2.4 Perturb & Observe (P&O) Algorithm

Typically, P&O method is used for tracking the MPP. A minor perturbation is introduced in this technique to cause the PV module's power to vary. The PV output power is measured on a regular basis and compared to the previous power.

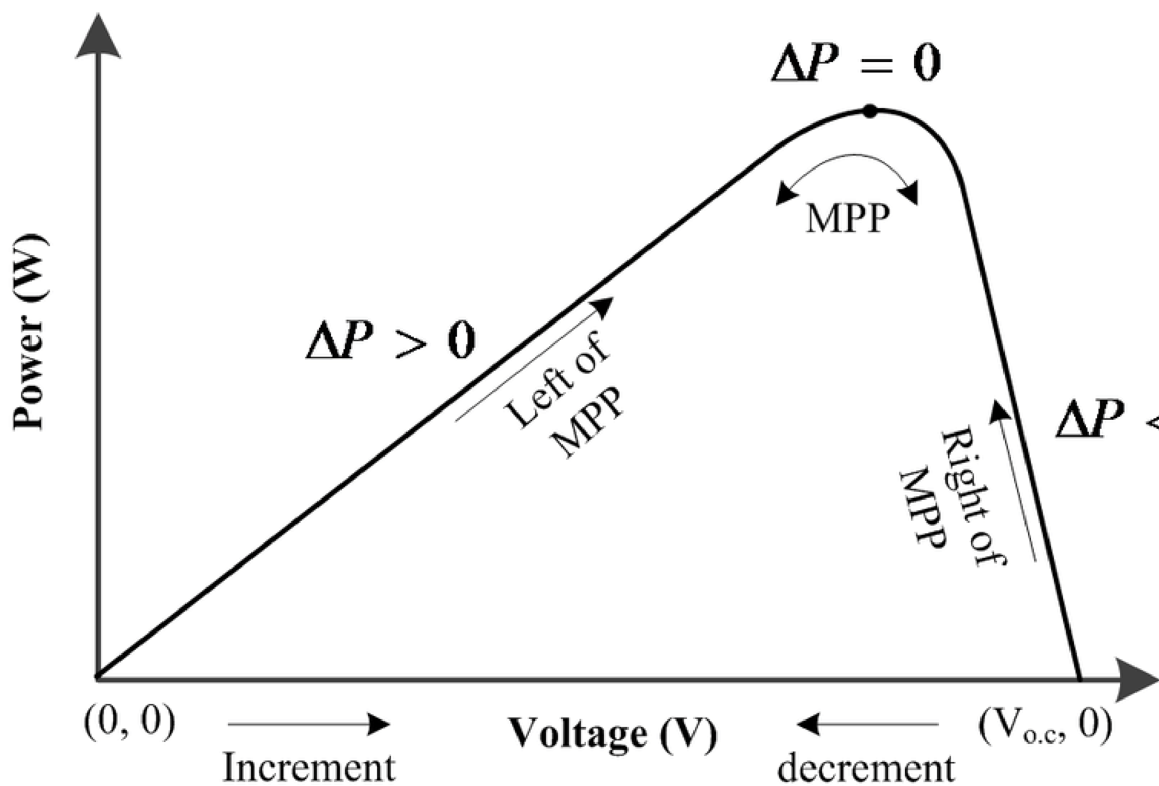

Figure 2.1 Solar P - V graph

The solar P – V graph depicts the relation between solar panel output power and its corresponding voltage. As the voltage increases, the solar output power increases accordingly. But at a certain point of voltage and current we get the maximum power and our primary goal behind solar MPPT technique is to move around the maximum power point. At certain point as the saturation in the solar cell takes place, current moves down with the increase in voltage as a result we can see that power moves down after maximum power point.

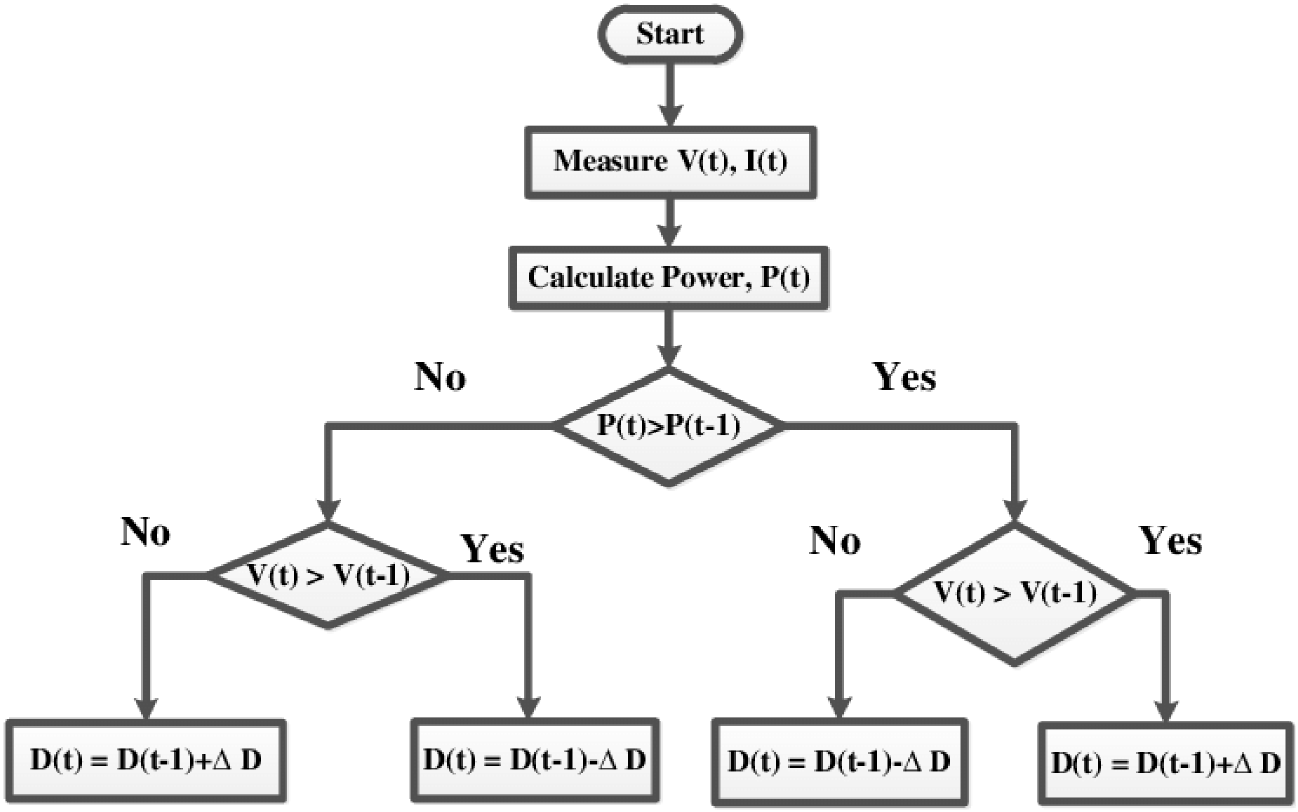

Figure 2.2 Flowchart of P & O Algorithm

In this algorithm perturbation is provided to the PV module or the array voltage. The PV module voltage is increased or decreased to check whether the power is increased or decreased. When an increase in voltage leads to an increase in power, this means the operating point of the PV module is on the left of the MPP. Here we will increase the duty cycle in order to achieve the maximum power point, similarly if the decrease in voltage leads to decrease in power, then also duty cycle increases. Hence, here the perturbation is towards the right. Further, if the increase in voltage leads to decrease in power the operating point is towards the right of maximum power point. Hence, here we will decrease the duty cycle in order to achieve the maximum power point. Similarly, if the decrease in voltage leads to increase in power, then also, we will decrease the duty cycle and reach at the maximum power point. Hence, here the perturbation is towards the left as the operating point is at the right side of the maximum power point. The algorithm is loaded in Arduino Uno microcontroller and its further utilized in order to obtain the duty cycle which act as an input to buck converter switching part.

Chapter 3: Project Plan

Project Planning & Task Distribution				
Sr. No	Task	Duration (Days)	Start Date	End Date
1	Project Identification & Definition	10	16/12/2021	25/12/2021
2	Extensive Study of Research Paper	15	26/12/2021	9/1/2022
3	Analyzing the requirement and market opportunity for solar MPPT	5	10/1/2022	14/01/2022
4	Comparative Study of MPPT Algorithms	5	15/01/2022	19/01/2022
5	Project Design on MATLAB Simulink and ML Implementation on simulation data	20	20/01/2022	8/2/2022
6	LCD Interfacing with Arduino Uno	2	9/2/2022	11/2/2022
7	Current Sensor and Voltage Sensing Calibration along with Interfacing of Arduino Uno	10	12/2/2022	21/2/2022
8	Buck Converter Design on Hardware	20	22/2/2022	14/3/2022
9	Perturb and Observe Algorithm Coding in Arduino Uno IDE	5	15/3/2022	19/3/2022
10	Temperature and Irradiance Sensor Calibration along with interfacing of Arduino Uno	10	20/3/2022	29/3/2022
11	Interfacing of SD Card Module	5	30/3/2022	2/4/2022
12	Data Collection in SD Card Module	5	3/4/2022	7/4/2022
13	Machine Learning Implementation and User Interface System	5	8/4/2022	12/4/2022
14	Testing of Entire System	3	13/4/2022	16/4/2022

Table 3.1 Project planning & task distribution

Project Gantt Chart: -

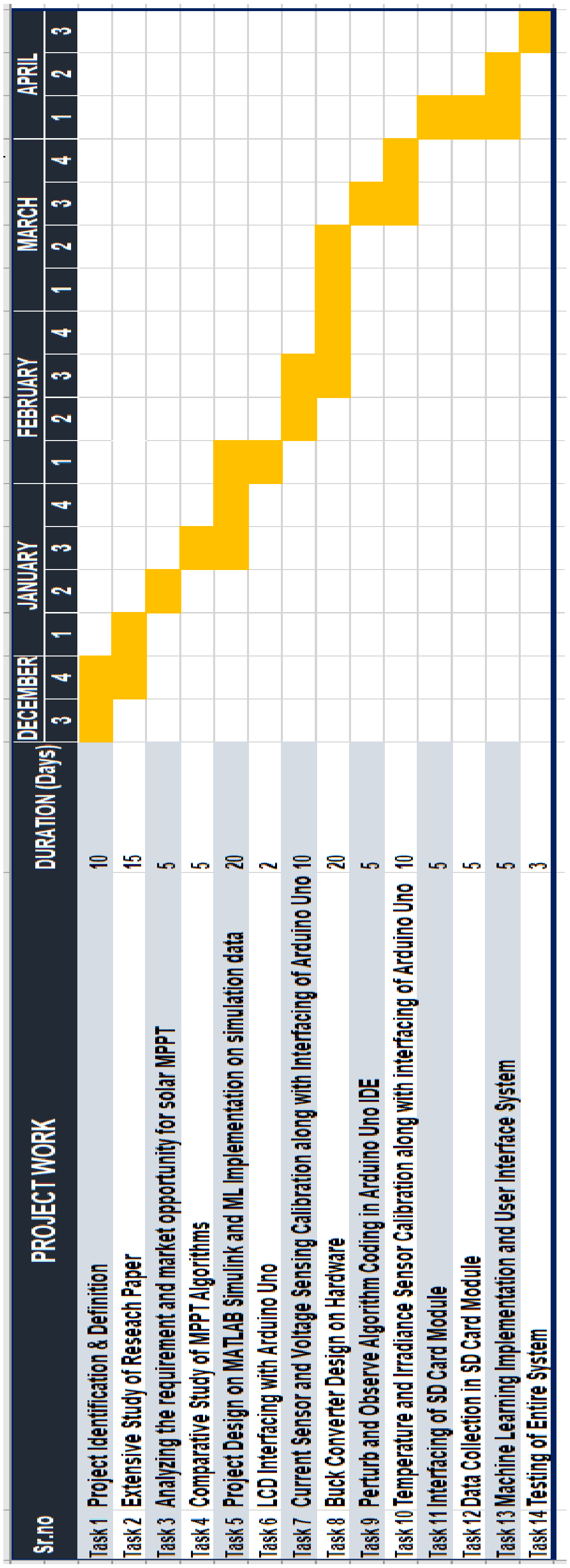

Figure 3.1 Project gantt chart

The project involves multiple stages where initial stages include Study of research paper, simulation of data and then in the next stage we have implemented hardware design and machine learning model on the data collected through the sensors and associated MPPT Output.

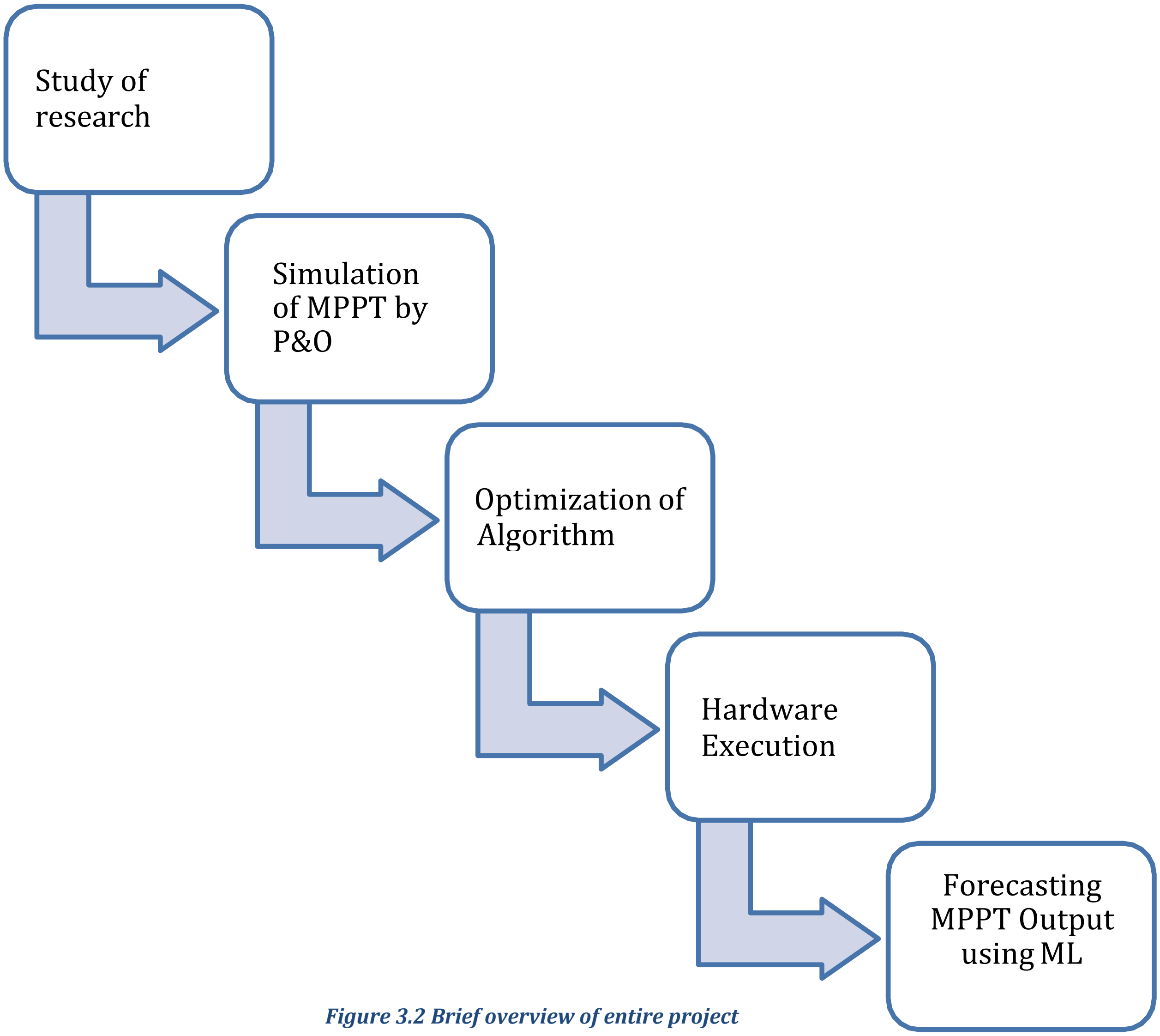

Figure 3.2 Brief overview of entire project

Initially, the machine learning model will be built on simulation of MPPT of 40W Solar PV Panel. After achieving satisfactorily results we will move towards hardware and implement machine learning technique on the data collected through hardware sensors and MPPT output.

Chapter 4: Literature Survey

4.1 Design of a P-&-O algorithm based MPPT charge controllerfor a stand-alone 200W PV system

Salman, S., AI, X. & WU, Z. Design of a P-&-O algorithm based MPPT charge controller for a stand-alone 200W PV system. Springer Conference on *Prot Control Mod Power Syst* **3,** 25 (2018). https://doi.org/10.1186/s41601-018-0099-8.

Takeaways

Solar cells convert sunlight into electricity, but they have significant drawbacks such as high initial cost, low photo-conversion efficiency, and intermittent operation. The current-voltage characteristics of solar cells are affected by solar insolation and temperature, resulting in a variation in the maximum power point (MPP). A microcontroller-based battery charge controller with maximum power point tracker (MPPT) is designed in this paper to harvest the maximum power available from the PV system under given insolation and temperature conditions, thereby improving PV system efficiency and battery lifetime. Among the various MPPT techniques, the perturb and observe (P&O) technique produces excellent results and is thus used. This project entails creating an MPPT charge controller with a DC/DC buck converter and a microcontroller.

This paper implements MPPT based on a modified P&O algorithm to harvest the most power from a PV module. The hardware, which consists of a high efficiency DC/DC buck converter and a microcontroller based MPPT controller, has been tested for battery charging. A conventional charge controller and a designed charge controller are compared when charging a 12 V battery from a 200 W solar PV module. It demonstrates that the modified P&O algorithm provides efficient and reliable maximum power tracking performance under rapidly changing irradiance and temperature conditions.

4.2 MPPT Techniques for PV Systems

D. Beriber and A. Talha, "MPPT techniques for PV systems," 4th International Conference on Power Engineering, Energy and Electrical Drives, 2013, pp. 1437- 1442, doi: 10.1109/PowerEng.2013.6635826.

Takeaways

This paper proposes a thorough comparison of four maximum power tracking techniques: Perturb and Observe (P&O), Incremental Conductance (InC), a fuzzy logic-based tracking technique, and a less well-known method based solely on photovoltaic current measurement. The disadvantage of the three investigated methods, P&O, InC, and one sensor algorithm, is that the operating point oscillates around the maximum power point at steady state, resulting in waste of the output panel's available energy. The simulation results show that the proposed fuzzy logic controller (FLC) can track maximum power faster and more consistently than the other methods investigated.

The perturb and observe controller is very simple and easy to implement. A drawback of P&O algorithm is that, at steady state, the system's operating point oscillates around the MPP giving rise to the waste of the available power. The perturbation step-size must be carefully chosen; the step size determines how quickly the MPP is reached; larger step-sizes allow for faster tracking but increase oscillations around the MPP. There is a tradeoff between dynamic and consistent performance. The InC method, which is more complex than the P&O method, allows for a slight reduction in the amplitude of the oscillation, but the system may not operate at the MPP. InC method suffers from the same problems associated to P&O algorithm such as requirement of ad-hoc tuning parameters, tradeoff between dynamics and steady state performance. The major advantage of the single current sensor technique is the fact that it uses the measurement of only one variable: the photovoltaic current. The proposed FLC provides faster and stable tracking of maximum power as compared to the other MPPT methods studied in this paper.

4.3 Analysis and simulation of the P&O MPPT algorithm using a linearized PV array model

Villalva, Marcelo & Gazoli, Jonas & Filho, Ernesto. (2009). Analysis and simulation of the P&O MPPT algorithm using alinearized PV array model. 189 - 195. 10.1109/COBEP.2009.5347755.

Takeaways

This paper shows that the performance of the perturb and observe (P&O) maximum power point tracking (MPPT) algorithm is improved when the electronic converter is correctly controlled. To obtain the transfer function of the electronic converter and to design a voltage compensator for the converter input voltage, a linearized photovoltaic (PV) array model is used. In this project, a buck converter is used.

The results have demonstrated that the direct duty-cycle control is not recommended in this kind of application. First, one should not employ the MPPT algorithm as the controller of the converter because a simple PI compensator can achieve this task with low stead-state error and improved dynamic response. Second, the MPPT algorithm's behavior is heavily influenced by the converter dynamics. At each sampling interval, the algorithm senses the voltage and current of the PV array and perturbs the reference (in this case, the voltage reference) based on algorithm decisions. If the reference voltage is not properly followed by the converter the MPPT algorithm senses wrong current and voltage values and gets confused. In this paper the P&O algorithm has been employed but these conclusions are valid for other algorithms. Using the simplest MPPT algorithm, as well as other algorithms already presented in the literature, one can conclude that even the simplest MPPT method can produce good results if the converter is controlled by a simple feedback controller with PI compensator.

4.4 Predicting solar power output using machine learning techniques

Solar energy is one of the world's leading renewable energy sources, and it is growing. However, it is dependent on sunlight, which is a finite natural resource. This makes power output predictability critical for the integration of solar photovoltaics into traditional electrical grid systems.

Standard data science techniques have been applied to predicting the solar power output in 12 different locations. The results show that LGBM is the best base model while the stacked model gave the overallbest performance.

Finally, the approach followed in this work can be used as a standard procedure to solve predictive analytics problems in the renewableenergy space.

4.5 Maximum Power Point Tracking of PV System Based on Machine Learning

Takruri, Maen & Farhat, Maissa & Barambones, Oscar & Ramos-Hernanz, & Turkieh, & Badawi, & AlZoubi, & Sakur, Abdus. (2020). Maximum Power Point Tracking of PV System Based on Machine Learning. Energies. 13. 692. 10.3390/en13030692.

Takeaways

This project investigates the conditions that lead to the maximum power point of a photovoltaic (PV) panel. It demonstrates that the maximum power point is extremely sensitive to external disturbances such as temperature and light. It proposes a novel method for increasing the PV panel output power when connected to a DC/DC boost converter with variable load conditions.

The author compares the performance of the system when using each of two machine learning algorithms, generalized regression neural network (GRNN) and SVR. Both algorithms are well known for their regression, function modelling, and prediction capabilities.

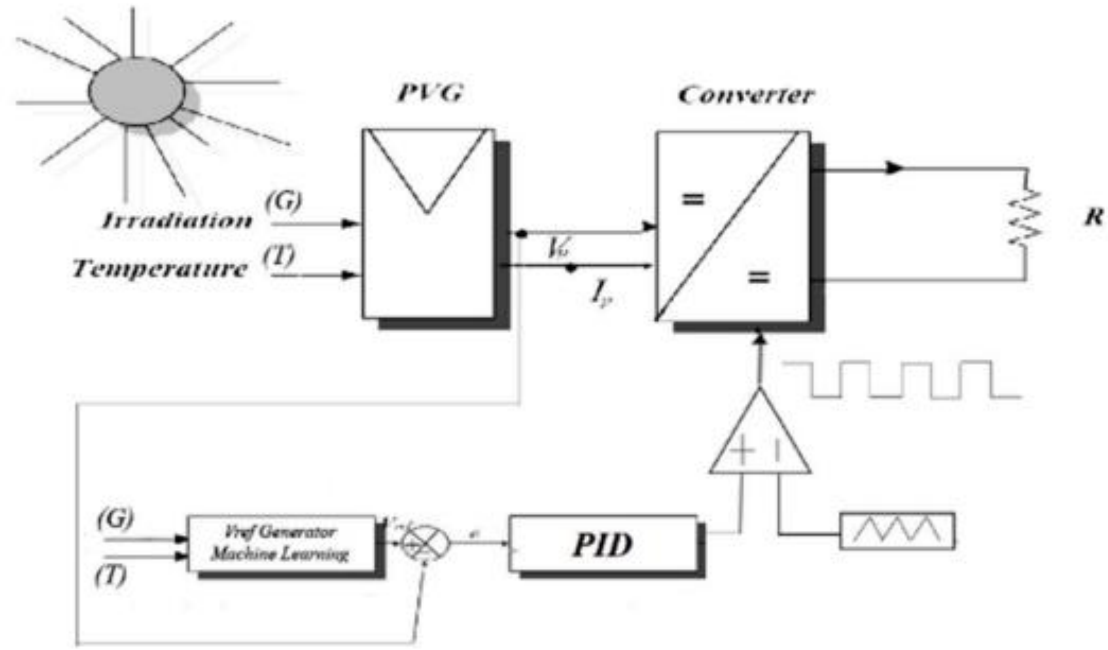

Figure 4.1 Block diagram of author's proposed Solar MPPT System

It predicts V for a given temperature, irradiation pair, which corresponds to the maximum output power point of the panel. To be fully functional, the machine learning block must go through two stages: the learning stage (also known as the training stage) and the running stage (also referred to as the testing stage). The learning stage's goal is to construct and validate a function that models the relationship of temperature and irradiation with the reference voltage V using historical (training) data.

The predicted voltage will be used as a reference by the PID controller to force the system to operate at its best point on the characteristic curve.

Chapter 5: Block Diagram, Simulation and Results

5.1 Block Diagram

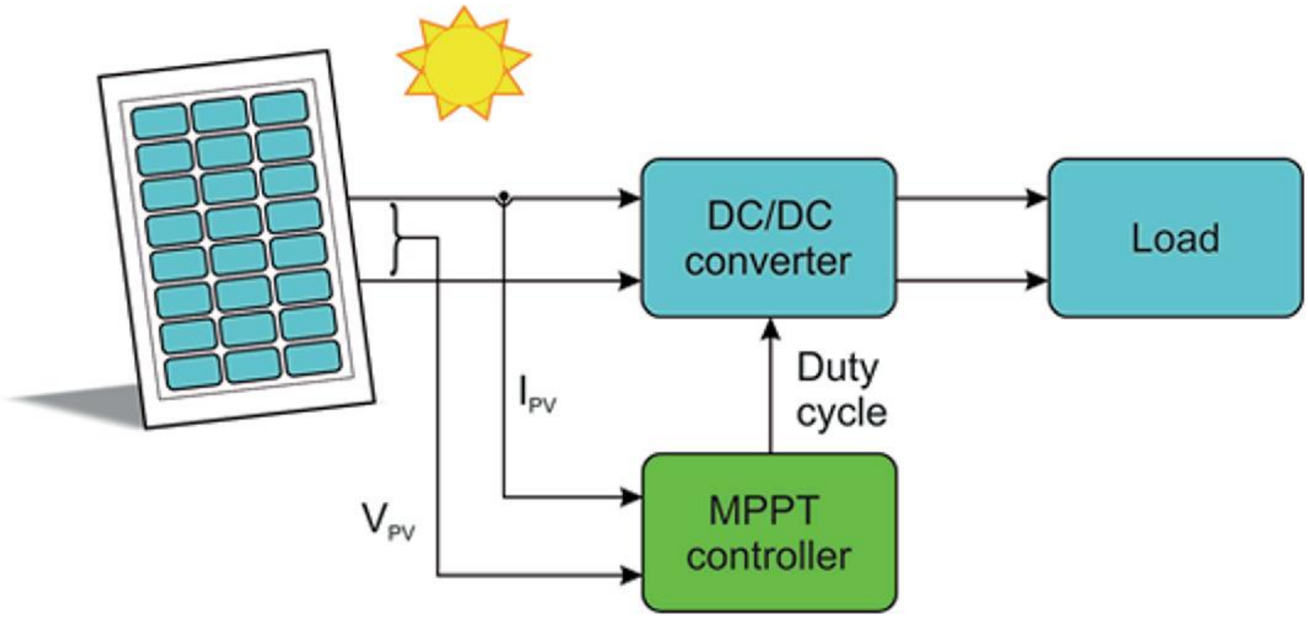

Figure 5.1 Block diagram of Solar MPPT system

5.2 Description of the Block Units:

- ✓ **Solar PV Panel**

  For our system, we have used 40W solar panel with maximum power voltage (Vmp) 19.25V, maximum power current (Imp) 2.08A, short circuit current (Isc) 2.22A and open circuit voltage (Voc) 22.5V.

- ✓ **MPPT Controller**

  We have implemented Perturb & Observe (P&O) algorithm for the adjustment of duty cycle, Arduino Uno is used as a MPPT controller where P&O algorithm is uploaded, and its input will be the solar panel voltage and current with respect to the respective irradiance and temperature.

- ✓ **DC/DC Converter**

  Buck converter is used in our system where the voltage steps down and current steps up in accordance with the duty cycle received from the MPPT controller.

- ✓ **Load**

  A DC bulb of 12V and 35W is used as our load.

5.3 Circuit Diagram/Simulation

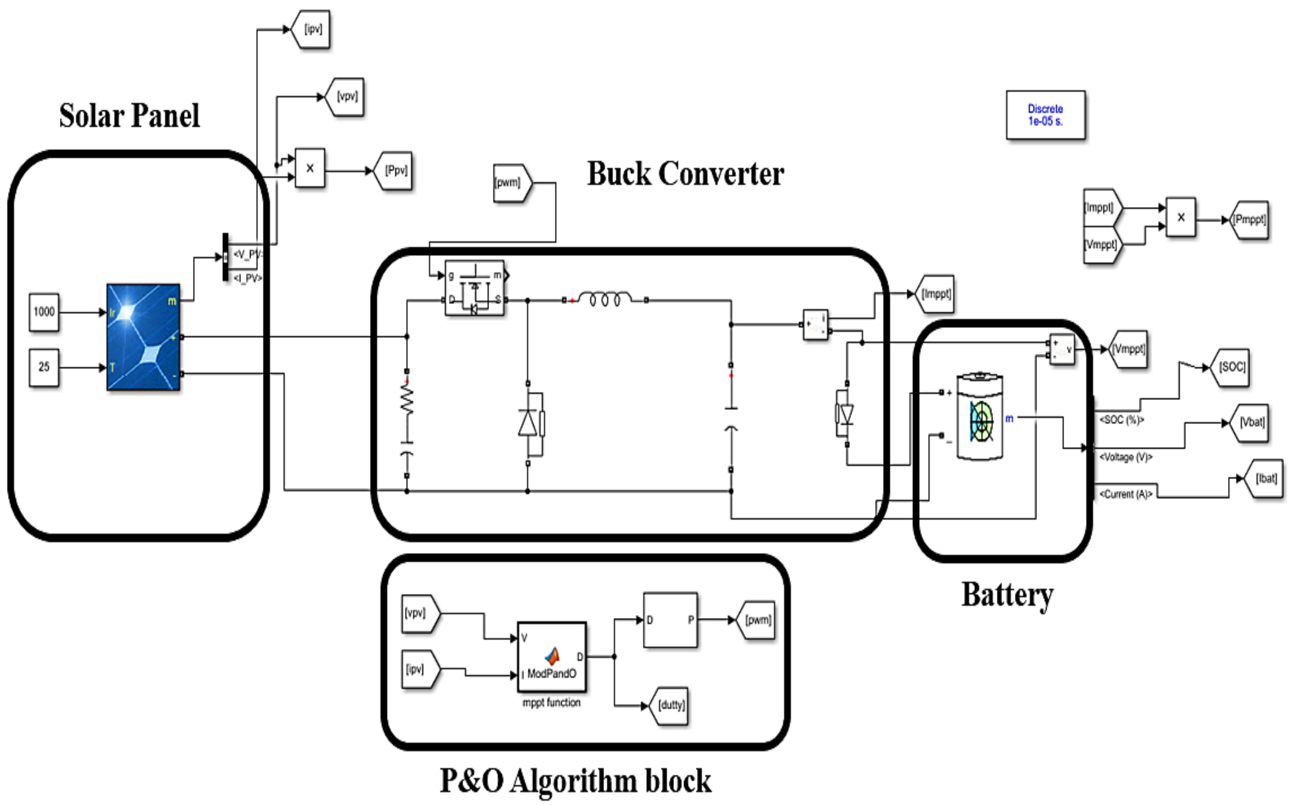

Figure 5.2 Circuit Diagram/Simulation of Solar MPPT system

Before building the solar MPPT PV system on the hardware it becomes essential to analyze and verify entire design as well as algorithm on Simulink. Hence, we have designed the entire solar MPPT PV system on the MATLAB Simulink platform.

The steps taken to design the entire solar MPPT PV system on MATLAB Simulink are as follow: -

Step 1: -

We selected a solar panel model and modified the parameters i.e. W_P maximum power (40W), V_{oc} open circuit voltage (22.5V) and I_{SC} short circuit current (2.22A).

Step 2: -

We designed a buck converter by following the conventional method of designing a buck converter. The steps to design buck converter are given below: -

- ✓ Fix the output power associated with load.
- ✓ Fix the input and output voltage.

- ✓ Find the current requirement of the load.
- ✓ Finalizing the ripple current and voltage of buck converter.
- ✓ Based on ripple voltage and ripple current calculating the value of capacitor and inductor.

Step 3: -

Afterwards, we implemented Perturb & Observe Algorithm (P&O) in MATLAB Simulink through MATLAB coding block where solar panel voltage and current are sensed, and power is calculated and by comparing values previous power and present power duty cycle is generated through P&O algorithm.

Step 4: -

For analyzing the results of solar MPPT we have used various scopes. Which helped us in optimizing our P&O algorithm so that we achieve the maximum power point as early as possible and with great accuracy.

Step 5: -

This system is tested for both standard condition i.e. 1000 irradiance and 25˚c and in varying condition also where we created a function block to generate varying temperature and irradiance.

Step 6: -

To collect the data for machine learning we used data logging feature of MATLAB where we can store data in excel by the use of scope. Through data logging we collected the variation in MPPT output caused by variation in temperature and irradiance.

Step 7: -

Here we have used a battery of 50mAH as a load which is charged with maximum power that is extracted using MPPT device. To monitor the charging of battery we used a scope where we are monitoring state of charge, voltage and charging current.

5.4 Simulation Results

Solar Panel Voltage, Current and Power with Standard Test Conditions (STC) Temperature and Irradiance for direct loading

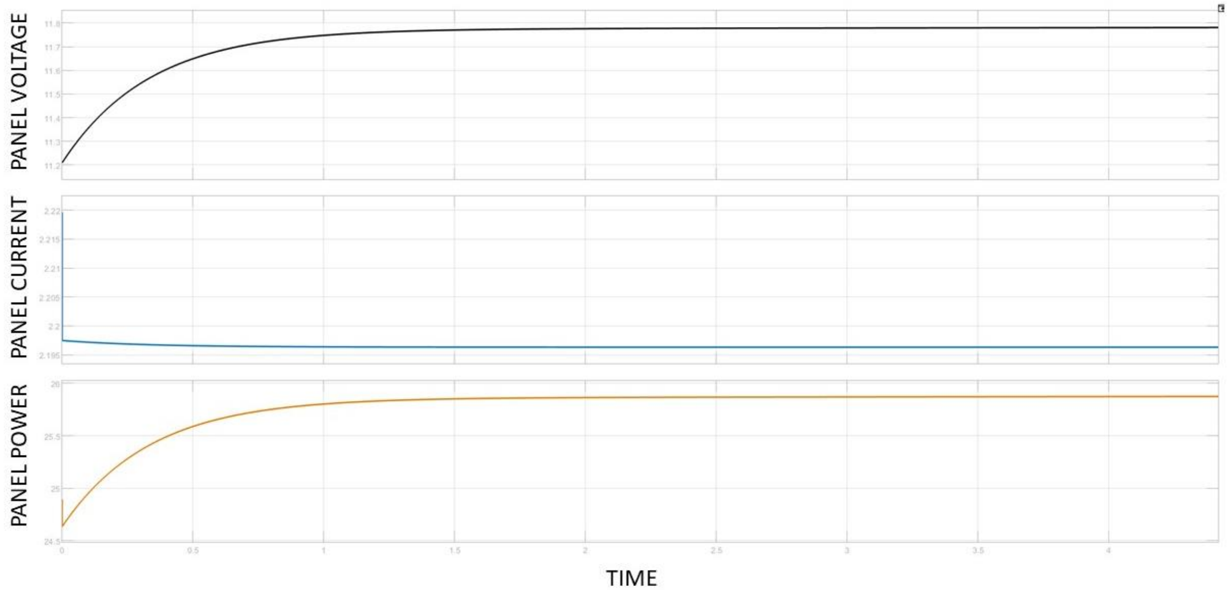

Figure 5.3 Solar panel voltage, current and power with STC temperature and irradiance for direct loading

Here, the first curve represents the panel voltage with respect to time. The second curve shows panel current with respect to time and last curve shows panel power with respect to time, for constant temperature (25°C) and irradiance (1000W/M^2). Here we can observe that the panel will operate at battery voltage which is not the maximum power voltage hence we are unable to utilize to maximum power capacity of the solar panel.

Solar Panel Output Current, Voltage and Power with Standard Temperature (STC) and Irradiance for MPPT Loading

Here, the first curve represents the panel voltage, next curve represents the panel current, and the last curve represents the panel power when a MPPT device is used to load the panel. We can observe that the panel is operating at the maximum power voltage which is around 19.25V, hence we are able to achieve maximum power out of solar panel**.**

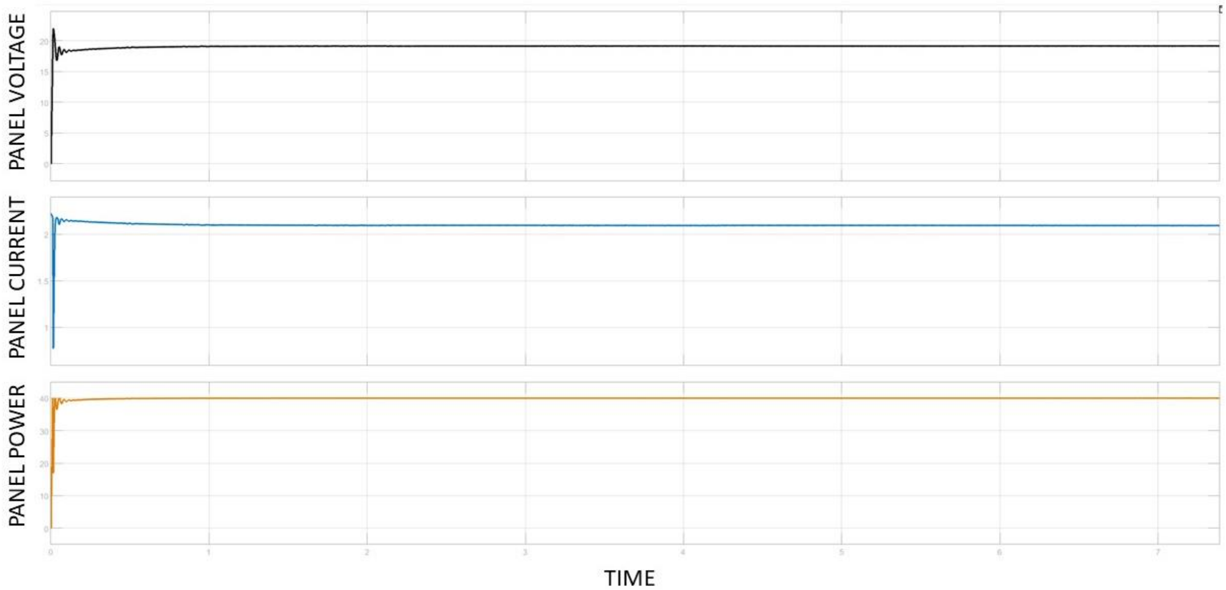

Figure 5.4 Solar panel voltage, current, and power with STC temperature and irradiance for MPPT loading

Comparative Statement for Solar Panel with & without MPPT device at Standard Test Conditions (STC) Temperature and Irradiance

- At standard test condition temperature (25) and irradiance (1000) we found that without using MPPT solar panel was generating maximum output power in the range of 24 to 26 Watt. On the other hand, after using MPPT along with solar panel we were able to achieve the maximum output power in the range of 38 to 40 Watt.

Solar Panel Voltage, Current and Power with Variable Temperature and Irradiance for direct loading

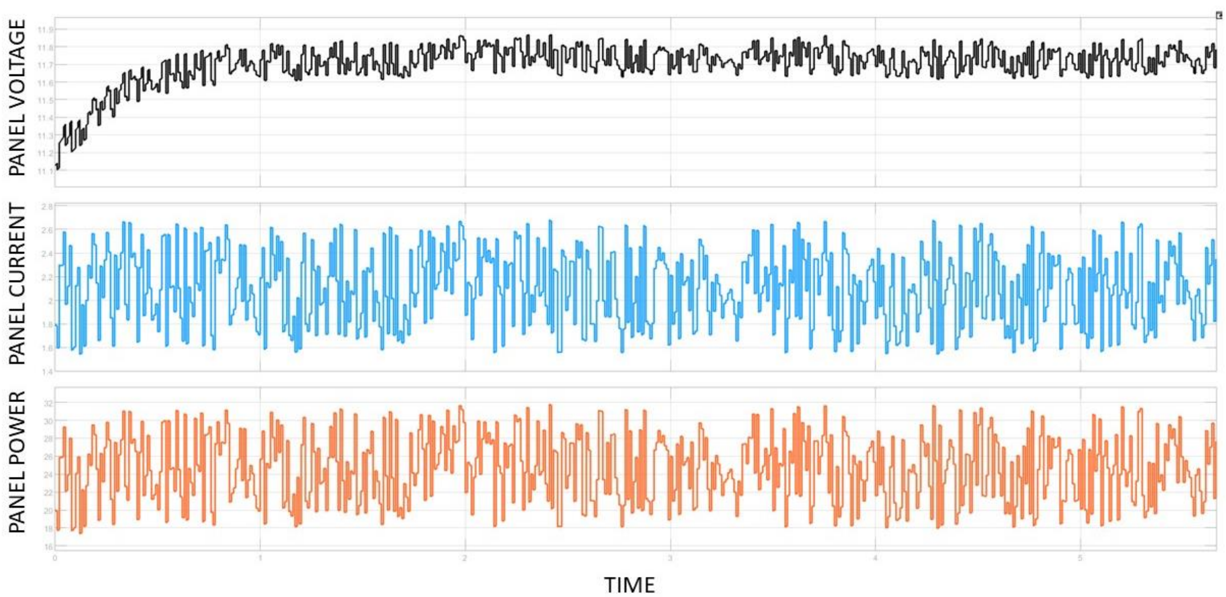

Figure 5.5 Solar panel output power, voltage, current with variable temperature and irradiance for direct loading

Here we have directly loaded the solar panel with battery and as a result the solar panel is operating at the battery voltage which may or may not be the maximum power voltage and due to that our 40W solar panel may not produce maximum power for a given condition and as a result the overall utilization efficiency of the panel reduces because we are not making use of the maximum capacity of the panel. Below graph depicts the relation of solar panel voltage, current and power with variable temperature, Irradiance for direct loading. First curve represents panel voltage, second curve represents panel current and last curve represents panel power.

Solar Panel Output Current, Voltage and Power with variable Temperature and Irradiance for MPPT Loading

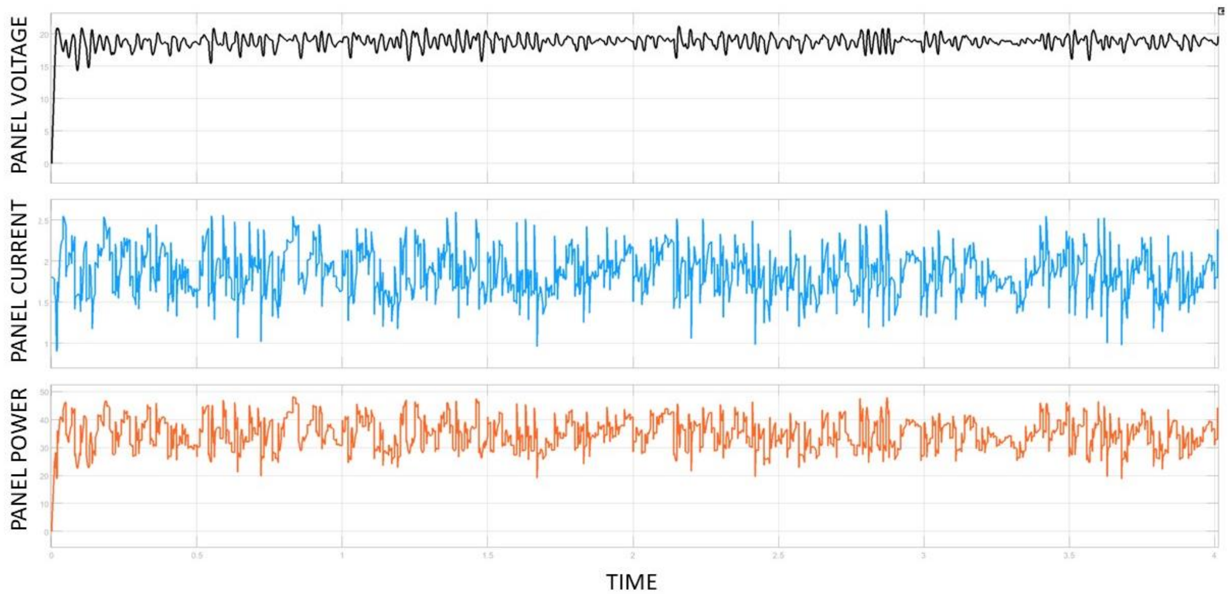

Figure 5.6 Solar panel output power, voltage and current with variable temperature and irradiance for MPPT loading

In the above graph, first curve shows solar panel voltage change with respect to time for variable conditions. Next curve shows the change in solar panel current with respect to time and the last curve shows the change in solar panel power as the condition's changes. Here we can see that the solar panel voltage remains almost constant as the maximum power voltage at a given instance. Thus, as compared to direct loading we can increase the power generation by the panel by significant amount.

Comparative Statement for Solar Panel with & without MPPT device at Variable Temperature and Irradiance

- At variable temperature and irradiance, we found that without using MPPT solar panel was operating at battery voltage instead of maximum power voltage for a particular instance. Thus, we were extracting maximum power of solar panel but after using MPPT as an intermediate stage the solar panel operating voltage was shifted to the maximum power voltage and hence, we were able to achieve maximum power at a particular instance.

Effect of Variable Temperature and Irradiance on solar MPPT output power.

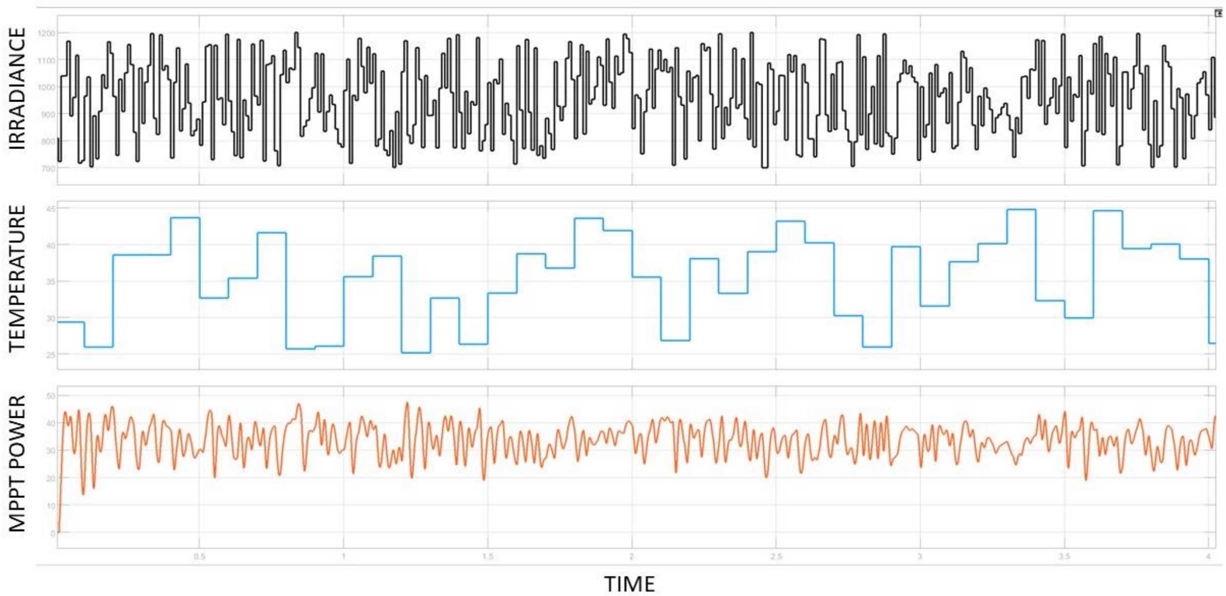

Figure 5.7 Relation between Solar MPPT output power vs irradiance, temperature

The first curve represents the change in irradiance with respect to time. Next curve depicts the change of temperature with respect to time. Last curve represents the change in maximum power point which is caused due to change in irradiance and temperature. Here as temperature remains constant and irradiance increases there is a significant amount of increase in the maximum power and if the temperature increases with irradiance constant there is a decrease in the maximum power.

Chapter 6: Hardware Implementation and Results

After completion of solar maximum power point tracking system on MATLAB Simulink, we further moved towards developing the entire system on hardware. Initially we planned to design the entire circuit on a breadboard and after testing it on the breadboard we will have a transition towards of entire system on brown board.

6.1 The steps taken to design the entire solar MPPT PV system on Hardware

Step 1: -

Interfacing of 16 X 2 LCD display with Arduino Uno for displaying the present values of solar panel operating voltage, current, power and duty cycle to verify the proper working of Perturb & Observe algorithm and to monitor panel parameters.

Step 2: -

Interfacing of current sensor and sensing voltage through voltage divider circuit in order to calculate output power of solar panel and buck converter. The output power of solar panel is calculated to predict duty cycle through P & O algorithm.

For voltage divider we have used two resistors of 22k ohm and 80k ohm to step down 24 volt to 5 volt which is the maximum limit of voltage that can be given to Arduino at analog pin.

$V_{in} = V_{SOLAR} \times (R_2/R_1+R_2)$
$R_1 \rightarrow$ 80k ohm.
$R_2 \rightarrow$ 22k ohm.
$V_{in} \rightarrow$ Voltage input to analog pin A2 of Arduino Uno.
$V_{SOLAR} \rightarrow$ Solar panel voltage supplied to buck converter.

The current sensor used is ACS712 hall effect sensor which converts the value of current passing through it to the equivalent value of voltage in the range of 0 – 5 Volt.

Step 3: -
Calibration of Current and Voltage Sensor

After successful interfacing of current sensor and voltage sensor with Arduino Uno. It was necessary that calibration of above-mentioned sensor was performed to ensure accurate value of current and voltage which was done by initially making current and voltage zero and subtracting the offset value. Further, a known value of current and voltage was supplied using variable DC supply and to get the accurate value a gain was multiplied and to ensure less fluctuating in reading average of 50 values of each was taken.

Step 4: -

We designed a buck converter by following the conventional method of designing a buck converter. The steps to design buck converter are given below: -

- ✓ **Fix the output power associated with load.**
  Here, we must fix our output power of buck converter because there is no conventional method to design a buck converter with varying output power. In our case, we fixed our buck converter output around 25W, and we will consider a DC bulb of 25 W and 12 – 15 V as our respective load.

- ✓ **Fix the input and output voltage.**
  Similarly, as mentioned above we also must fix input and output voltage because to calculate inductance and capacitance we require fix output and input voltage. Here, we have taken our output voltage as 12V and input voltage as 19.25V, the reason for taking input voltage as 19.25V is that in our P-V curve of solar panel the maximum power for ideal conditions lies around voltage 19.25 – 19.5 V.

- ✓ **Find the current requirement of the load.**
  To find the current requirement of the load we supplied the rated voltage to the load and then calculated the current drawn. We achieved this by using variable dc supply. Here, the main reason to find the current requirement is that we need to find current ripple which is required to calculate capacitance and inductance.

- ✓ **Finalizing the ripple current and voltage of buck converter.**
  As we have fixed input/output voltage and current which had led us to fixed ripple current and ripple voltage. If, we want less ripple in our output current and voltage the values of inductor and capacitor will be higher than compared to more ripple in output current and output voltage.

- ✓ **Based on ripple voltage and ripple current calculating the value of capacitor and Inductor.**

$$C = \frac{\Delta I_L}{8 \times f_s \times \Delta V_{out}}$$

Where
ΔI_L = estimated inductor ripple current
f_s = minimum switching frequency of the converter
ΔV_{out} = desired output voltage ripple

Figure 6.1 Buck converter capacitance calculation

$$L = \frac{V_{out} \times (V_{in} - V_{out})}{\Delta I_L \times f_s \times V_{in}}$$

Where

V_{in} = typical input voltage

V_{out}= desired output voltage

f_s= minimum switching frequency of the converter

ΔI_L = estimated inductor ripple current

Figure 6.2 Buck converter inductance calculation

For the calculation of inductor and capacitor values the above given two formulas were used respectively as described the required ripple voltage and ripple current were fixed along with the switching frequency of MOSFET.

- ✓ **Selecting MOSFET, considering the above parameters**
  Considering the maximum current which will flow through the MOSFET and the maximum reverse voltage which will be applied on MOSFET (OFF condition). Hence, we selected IRFZ44n as our MOSFET. Further, the switching frequency at which the buck converter will be operating must also be considered while selecting the MOSFET.

Step 5: -

Perturb & Observe algorithm implementation on hardware with the use of Arduino Uno. Followed by modification of the algorithm for proper functioning and results in which we set the delta in such a way that we can track the maximum power point as early as possible without the loss of accuracy. It was achieved by coding on Arduino IDE platform.

Step 6: -

Interfacing of temperature & humidity sensor and light intensity sensor with Arduino Uno. The use of this sensors is to collect temperature, humidity and light intensity which will be useful while training machine learning model where we will predict the MPPT output with the help of this data.

Step 7: -

Interfacing of SD card Module with Arduino Uno. The data from above mentioned sensors i.e. voltage sensor, current sensor, temperature & humidity sensor and light intensity sensor are collected in SD card in the form of text file which can be imported to excel sheet for further exploratory data analysis and training machine learning model which will help user/solar operator to predict the generation of final MPPT output without worrying about any technical terms.

Step 8: -

Final Circuit after testing on breadboard was implemented on brown board to eliminate the problems caused by lose connections of breadboard.

6.2 Problems & Solutions while Designing Hardware

Problem: Ineffective triggering of MOSFET

Arduino Uno was not able to properly supply sufficient current and square wave at high frequencies.

Solution

We used a triggering circuit to trigger the MOSFET and design a simple circuit to turn 'ON' and 'OFF' the LED on 1HZ frequency then we proceeded towards 1khz switching frequency and lamp as load. Afterwards, to deal with low current supply we used external 9V battery and optocoupler to generate pulses.

Problem: Damage to Arduino due to high gate current drawn by MOSFET

Without isolation MOSFET may draw high gate current which can damage Arduino pin

Solution

To solve this problem, we used optocoupler to isolate MOSFET gate from Arduino pin and, hence protecting it. Also due to optocoupler there was a limitation to the switching frequency so we replaced normal optocoupler with high speed optocoupler which can switch at higher frequency.

Problem: Less voltage and current on the output side then predicted due to improper selection of capacitor and inductor

This problem was caused because we took in consideration the values of capacitor and inductor same as simulation.

Solution

To deal with this problem we designed a buck converter from basics using conventional method i.e. fixing switching frequency, fixing ripple current and voltage etc.

Problem: slip of point from maximum power point.

In this problem after reaching maximum power point our algorithm was not able to hold that point and was slipping away from it.

Solution

Here we changed the switching frequency and also modified delay in the program also the delta was adjusted, the main solution was to modify delay as it seemed the main problem causer as it was disturbing the flow of algorithm

6.3 Hardware Glimpse and Results

Glimpse of Solar MPPT device on breadboard: -

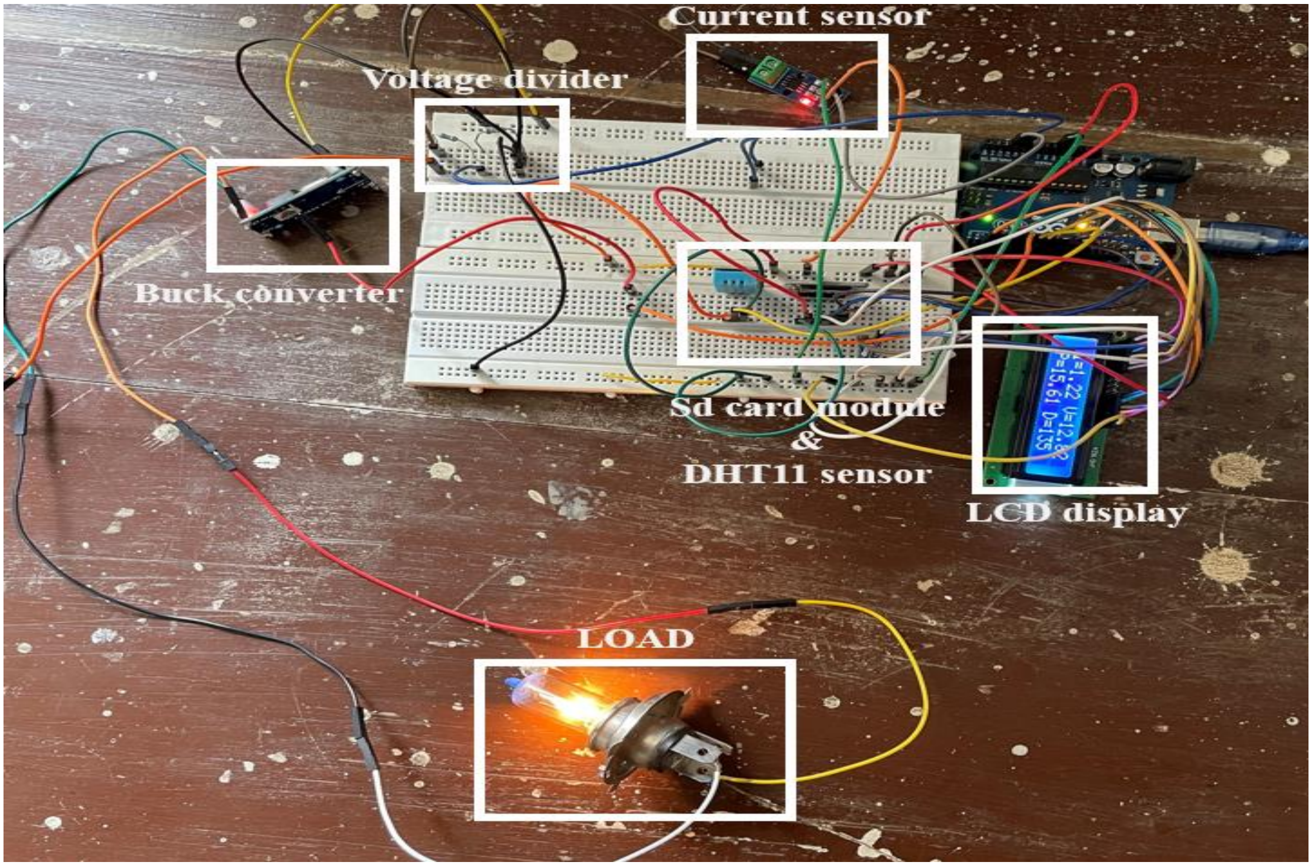

Figure 6.3 Glimpse of solar MPPT device on breadboard

The above figure depicts the initial testing stage of solar MPPT device and a DC bulb 12V 35W. Each part of the circuit is highlighted by the white color box and its interfacing is done on the breadboard.

The main purpose behind designing and implementing the entire circuit on the breadboard is to test it and analyze the proper working of each component associated with the circuit.

Glimpse of Solar MPPT device on brown board: -

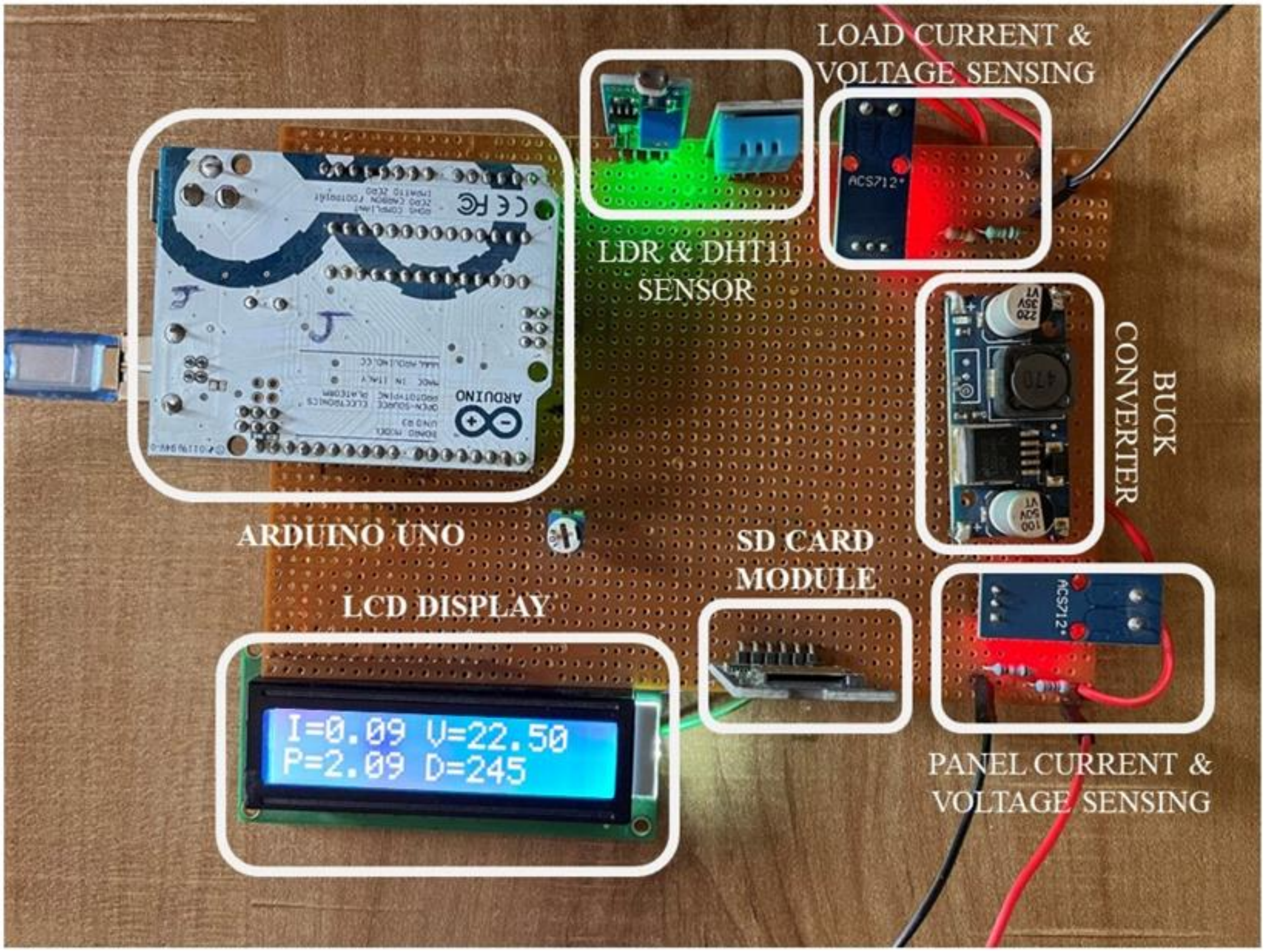

Figure 6.4 Glimpse of solar MPPT device on brown board

Here the final circuit is executed and assembled on the brown board. The connections with respect to the circuit are in the backside of the brown board.

6.4 Code for Hardware Implementation

The entire code is divided into multiple steps and those are given below:-

Step1:-

Declaration of various variables used , declaration of input/output pins based on requirements including library for various functions carried out throughout the code.

```
#include "DHT.h"
#define DHTPIN 7     // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
#include <SPI.h> //Library for SPI communication (Pre-Loaded into Arduino)
#include <SD.h> //Library for SD card (Pre-Loaded into Arduino)
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,5,4,3,2);


//ip parameters

float cur_now;
float volt_now = 0.00;
float power_now = 0.00;
float cur_prev;
float volt_prev = 0.00;
float power_prev = 0.00;
float avg_c, avg_v ;

//op parameters

float cur_op;
float volt_op = 0.00;
float power_op = 0.00;
float avg_cop, avg_vop ;
float t,h,irr;

//duty variables

int duty = 220;
int duty_max = 245;
int duty_min = 10;
int delta = 5;
```

Figure 6.5 Declaring variables required for entire code

Step 2:-

Here various pins are set for input/output purpose for example we are using pins A1 and A2 for sensing current and voltage of solar panel respectively, here in this section of code we have also changed the PWM switching frequency of pin number 6 as we are going to supply duty cycle to the buck converter using this pin.

```
void setup()
{
     TCCR0B = TCCR0B & B11111000|B00000001;
    pinMode(A5,INPUT); //light intensity pin
    //ip pins
    pinMode(A1,INPUT); //current pin
    pinMode(A2,INPUT); //voltage pin
    //op pins
    pinMode(A4,INPUT); //op current pin
    pinMode(A3,INPUT); //op voltage pin
    pinMode(6,OUTPUT); // dutty output
    dht.begin();
    Serial.begin(9600);
    lcd.begin(16,2);

    Initialize_SDcard();
}
```

Figure 6.6 Pin setup for Input/Output operation

Step 3: -

Now after setup we move towards the infinite loop part of the code the task that are to be repeated are performed inside this loop, here we are calculating the values of panel voltage , current, power and MPPT output voltage , current and power this calculation are done inside a separate function which is the called here.

```
void loop()
{
    lcd.clear();
    //ip parameters calculation

    current();  //reading current value and taking average to reduce error
    Serial.println("current=");
    Serial.println(cur_now);


    voltage();  // reading voltage and converting to original value
    Serial.println("voltage=");
    Serial.println(volt_now);

    // calculating power

    power_now = cur_now*volt_now;
    Serial.println("power=");
    Serial.println(power_now);

    //op parameters calculation

    current_op();  //reading current value and taking average to reduce error
    Serial.println("current_op=");
    Serial.println(cur_op);


    voltage_op();  // reading voltage and converting to original value
    Serial.println("voltage_op=");
```

Figure 6.7 Panel voltage, current, power and MPPT output power calculation

Voltage, current and power calculation are followed by MPPT function. It is a function where our perturb and observe algorithm is stored which gives us the duty cycle, which will be the input for buck converter. After that we are storing the present values of voltage current and power which will be used to compare next values by the MPPT function. The duty cycle provided by the

```
power_op = cur_op*volt_op;
Serial.println("power_op=");
Serial.println(power_op);

// mppt algorithm

mppt();


Serial.println("duty=");
Serial.println(duty);
volt_prev = volt_now;
cur_prev = cur_now;
power_prev = power_now;
lcdprint();
avg_c=0;
avg_v=0;

avg_cop=0;
avg_vop=0;
```

Figure 6.8 MPPT function (P & O Algorithm) and duty

MPPT function is then given as output on pin 6 with the help of analog Write, which produces a PWM signal with max duty cycle of 255 and minimum of 0.

Step 4:-

Now the duty cycle is checked if it is within the permissible limit or not by using if else for our case the maximum duty cycle is 245 and minimum is 10. Now for sensing temperature and humidity we are using DHT11 sensor and a function is created which will give us temperature and humidity , similarly for measuring light intensity we are using LDR module and a function named intensity is created, now to store this data for machine learning purpose we are using sd card module and to write data to sd card we have created a function name Write_SDcard(); , which stores values of temperature, humidity, light intensity and power to sd card.

```
if(duty>245)
{
duty= duty_max;
}
else if(duty<10)
{
duty=duty_min;
}


TandH();

intensity();

Write_SDcard();
```

Figure 6.9 Duty cycle condition, temperature, humidity, and Irradiance calculation function

Step 5: -

```
float voltage()
{
  for(int a =0;a<49;a++)
    {
    volt_now = analogRead(A2);
    volt_now = volt_now/1024;
    volt_now = volt_now*5;
    volt_now = (volt_now*4.6363)*1.175;
    avg_v = volt_now + avg_v;
    }
    volt_now = avg_v/50;
    return volt_now;
  }

float current()
{
   for(int i = 0;i<49;i++)
    {
    cur_now = analogRead(A1);
    cur_now = cur_now-452;
    cur_now = (cur_now/1024);
    cur_now = cur_now/0.185;
    cur_now = cur_now*5;
    cur_now = -(cur_now-1.43)*1.07;
    avg_c = cur_now + avg_c;

    }
    cur_now = (avg_c/50.00);

    return cur_now;
  }
```

Figure 6.10 Function to calculate panel voltage and current

Below are the functions created to calculate panel voltage and current, here first of all we need to subtract the offset and the multiply with the gain if required to get the correct reading, Further to reduce the fluctuation and get accurate reading we took average of 50 values of both voltage and current.

Step 6: -

```
float voltage_op()
{
  for(int a =0;a<29;a++)
    {
    volt_op = analogRead(A3);
    volt_op = volt_op/1024;
    volt_op = volt_op*5;
    volt_op = (volt_op*2.96)*1.28;
    avg_vop = volt_op + avg_vop;
    }
    volt_op = avg_vop/30;
    return volt_op;
  }

float current_op()
{
   for(int i = 0;i<29;i++)
    {
    cur_op = analogRead(A4);
    cur_op = cur_op-452;
    cur_op = (cur_op/1024);
    cur_op = cur_op/0.185;
    cur_op = cur_op*5;
    avg_cop = cur_op + avg_cop;
    }
    cur_op = (avg_cop/30.00);
    cur_op = -((cur_op - 1.5)*2.763)
    return cur_op;
  }
```

Figure 6.11 Calibration of voltage and current sensor

Here we have created functions to calculate the MPPT output voltage and current, the calibration of the sensors is done as mentioned above.

Step 7: -

```
void lcdprint()
{
        lcd.setCursor(0,0);
        lcd.print("I=");
        lcd.print(irr,2);
        lcd.print(" V=");
        lcd.print(volt_now,2);
        lcd.setCursor(0,1);
        lcd.print("P=");
        lcd.print(power_now,2);
        lcd.print(" D=");
        lcd.print(duty);
}
```

Figure 6.12 LCD display code

To monitor the panel parameters and to analyze the performance we have interfaced a 16x2 LCD display which is used to display the panel current, voltage, power, and the operating duty cycle of buck converter. Thus, to achieve that we have created a function which is called in the main loop to display above parameters continuously.

Step 8: -

For implementation of perturb and observe algorithm we have created a function block named MPPT which is called in the main loop after the sensing of voltage, current and calculation of power.

```
int mppt()
  {
       if(power_now > power_prev)
    {
     if(volt_now > volt_prev)
      {
        duty = duty + delta;
      }
        else
        {
        duty = duty - delta;
        }
    }
  else
  {
    if(volt_now>volt_prev)
    {
      duty = duty - delta;
    }
    else
    {
    duty = duty + delta;
    }
  }
  return duty;
  }
```

Figure 6.13 Perturb & Observe Algorithm (MPPT function block)

Step 9: -

```
float TandH()
    {
    // Wait a few seconds between measurements.

    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    h = dht.readHumidity()*0.27;
    // Read temperature as Celsius (the default)
    t = dht.readTemperature()*2.04;

    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t))
    {
      Serial.println(F("Failed to read from DHT sensor!"));
      return;
    }
    Serial.print(F(" Humidity: "));
    Serial.println(h);
    Serial.print(F("%  Temperature: "));
    Serial.println(t);
    Serial.print(F("C "));
    return h;
    return t;
    }
```

Figure 6.14 Sensing of temperature and, humidity and Irradiance function sensing of temperature and humidity is done using the above-mentioned function which is called in the main loop.

Step 10: -

This function is used to sense the light intensity in W/M^2, calibration of this sensor was also performed as discussed in the current sensor part.

```
float intensity()
    {
     float a = -1.405;
    irr = analogRead(A5);
    irr =(irr/1024);
    float ldrvolt = irr*5;
    float ldrres = ((10000*ldrvolt)/(5-ldrvolt));
    irr = 12518931*pow(ldrres,a);
    irr = irr *0.0079;
    Serial.println("irr");
    Serial.println(irr);
    return irr;
    }
```

Figure 6.15 Irradiance measurement and light intensity calibration

Step 11: -

To store the data to SD card this function is called in the void setup part as this function creates a file and the file needs to be created only once.

```
void Initialize_SDcard()
{
  if (!SD.begin(chipSelect))
  {
    Serial.println("Card failed, or not present");
    return;
  }
  File dataFile = SD.open("MLDATA.txt", FILE_WRITE);
  if (dataFile)
  {
    dataFile.println("Temperature,Humidity,Irradiance,Power_ip,Power_op"); //Write the first row of the excel file
    dataFile.close();
  }
}
```

Figure 6.16 Data storing in SD card

6.5 Hardware Results

Temperature	Irradiance	Panel Power with MPPT Device	Panel power with Direct Loading
33	321.942	15.83	8.79
32.4	570.6678	21.26	11.6
32.3	570.6678	20.96	11.83
33	321.942	15.83	8.79
36.1	570.6678	17.53	9.33
36.2	570.6678	17.81	9.59
35.2	856.0913	20.22	11.4
35.4	856.0913	19.14	10.66
36	856.0913	19.82	10.95
35.1	856.0913	21.62	11.96
35.2	570.6678	22.35	12.26
35.3	856.0913	21.35	12.14
34.3	570.6678	21.23	12.31
34.4	570.6678	21.95	12.37

Table 6.1 Glimpse of hardware sensor data

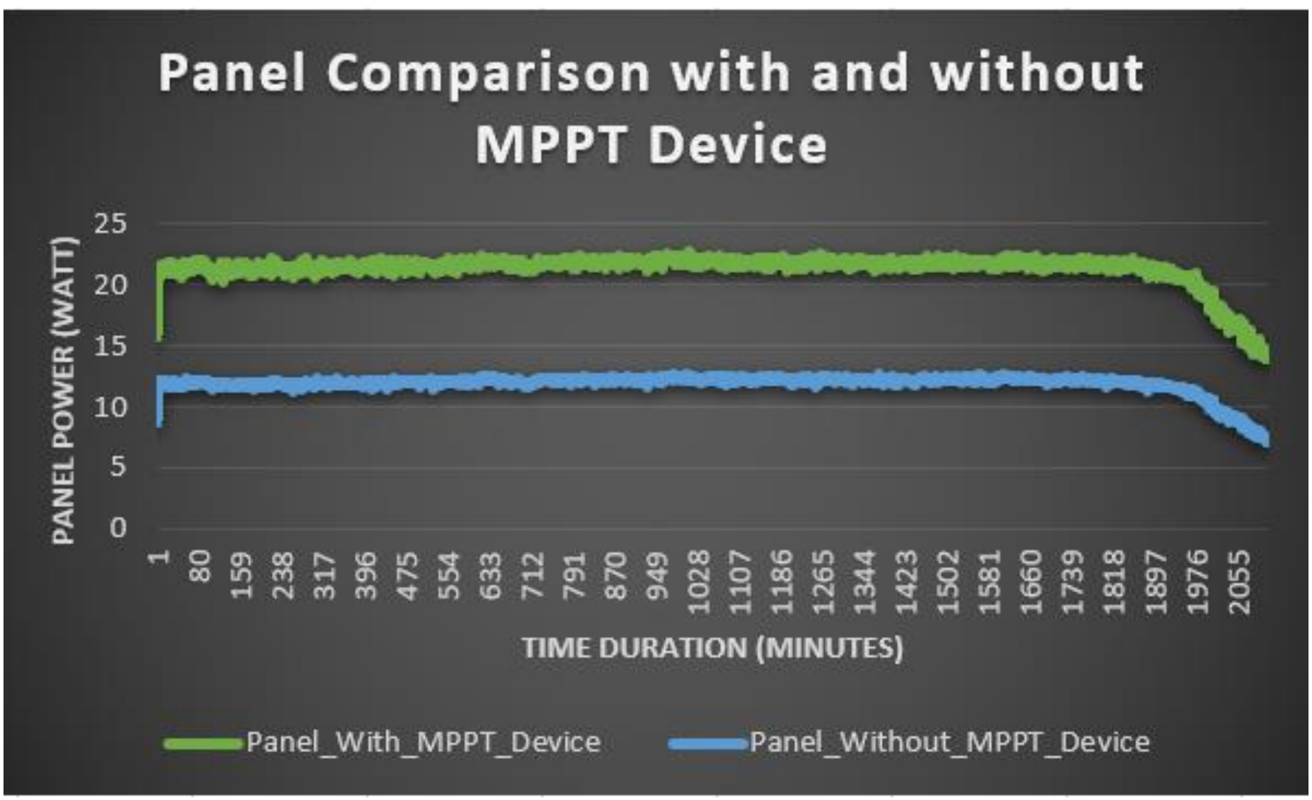

Figure 6.17 Panel output power with and without MPPT device

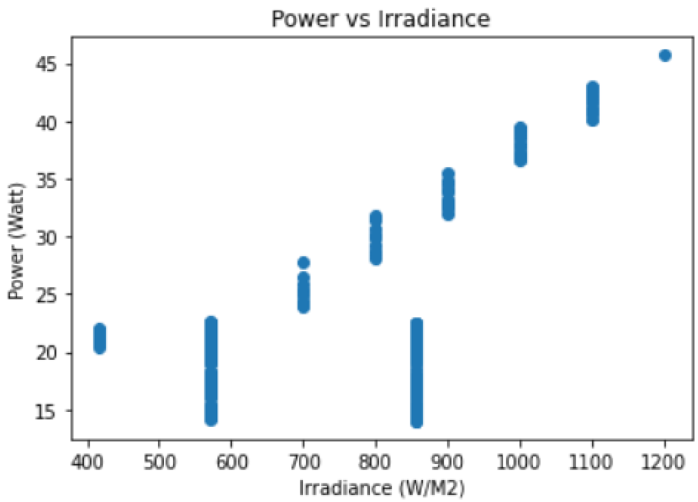

Figure 6.18 Power (Watt) vs Irradiance (Watt/Meter-Square)

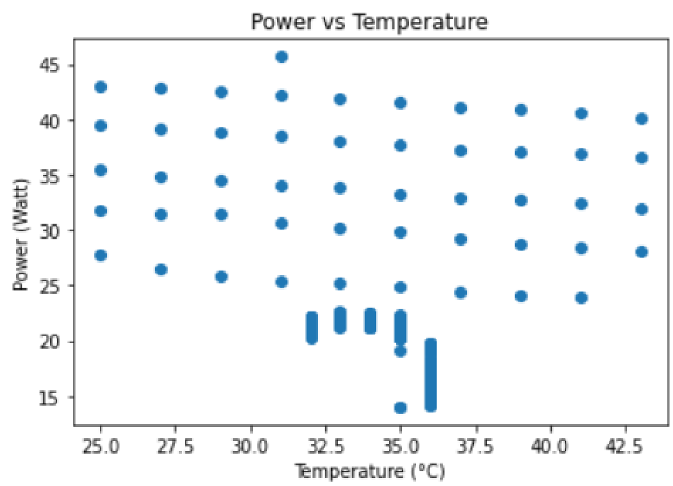

Figure6.19 Power (Watt) vs Temperature (℃)

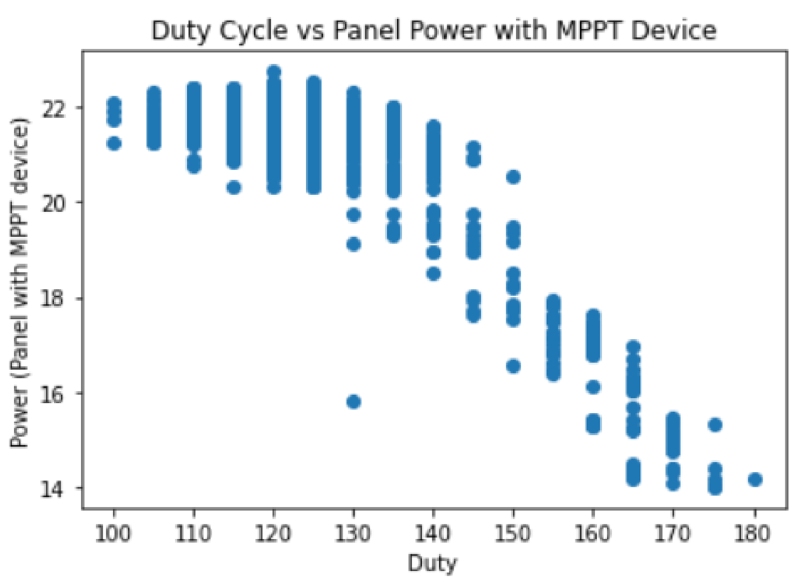

Figure 6.20 Panel output Power vs Duty Cycle

Chapter 7: Machine Learning Implementation on the Collected Data

7.1 Machine Learning Regression Models

Machine Learning is the branch of artificial Intelligence which is broadly defined as the capability of a machine to imitate intelligent human behavior.

Initially, we have decided to build a robust machine learning model on the simulation data. The data collected from Simulink through a data logging method. The data includes three columns temperature, irradiance and MPPT output power.

As the data includes the output itself in it, the problem becomes supervised machine learning problem. Here we need to use the regression models as we already have the output feature along with us. A regression model is created to predict the output power of solar MPPT.

The features of the model are Irradiance and temperature, and output is power retrieved from solar MPPT.

We discovered that the higher the irradiance, the higher the output current, and thus the higher the power generated. Furthermore, as the temperature rises, the output current rises exponentially up to 25 degree Celsius and then it falls down, while the voltage output falls linearly; heat can significantly reduce the solar panel's power production.

We have irradiance and temperature as our features and output as solar MPPT output, here we will use regression models to predict the output. Varied regressionmodels will be multiple regression model, decision tree regressor, random forest regressor and support vector machine regressor.

Before the implementation of various machine learning techniques on the collected data it becomes essential to study different regression models in order to implement them and get the best results.

Multiple Regression Model

Multiple regression model is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of machine learning model is to predict the output based on multiple input feature on which the respective model is trained.

Multiple Regression model is the extended version of linear regression model, here there are multiple features on which our target variable depends. Hence, we have to consider the weights or we can say slope of those dependent variables as well.

$$Y = B_0 + B_1X_1 + B_2X_2 + B_3X_3 + ... + B_NX_N + E$$

$Y \rightarrow$ Target Variable.

$X_1, X_2, X_3 ... X_N \rightarrow$ features / Independent Variables.

$B_0, B_1, B_2, B_3 .. B_N \rightarrow$ slope / weights of several features.

$E \rightarrow$ Error.

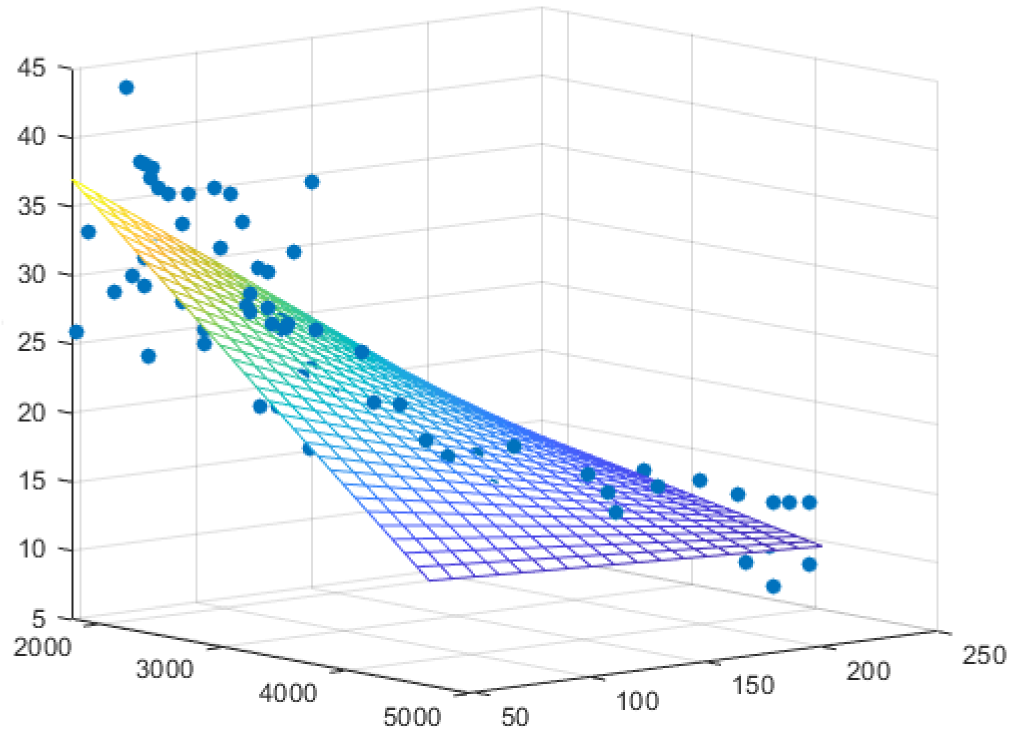

Figure 7.1 Multiple Regression Model

Support Vector Regression

Support vector regression algorithm is based on support vector machine algorithm. Support vector is basically used for both the classification and regression problems. Here there are basically boundaries on the basis of which we will differentiate the errors.

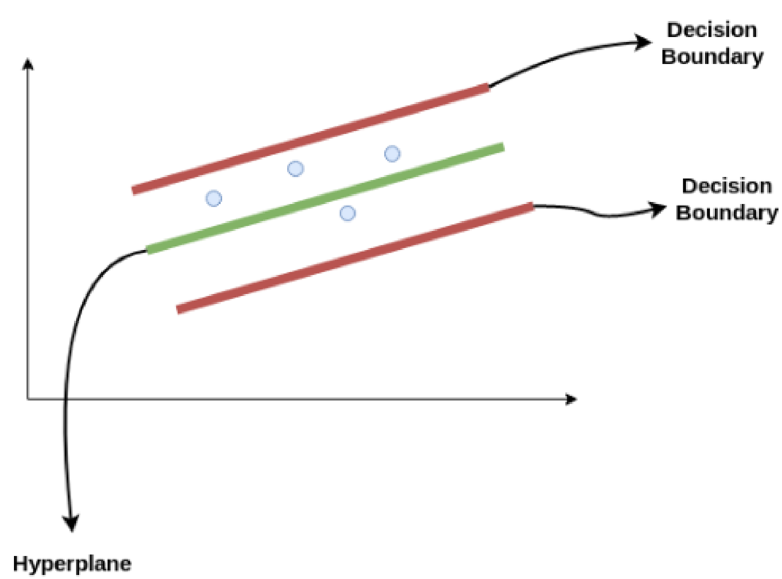

Figure 7.2 Support Vector Regression model

Here, In the above figure we can see there are two red lines which depicts the boundary which is also known as decision boundaries. The incoming or the input points should always lie in between those decision boundaries. Those lines are drawn at a particular perpendicular distance from the hyperplane and let's consider those distance to be '+a' for positive distance and '-a' for negative distance.

The equation of hyperplane,
Y = wx+b
Y → Dependent variable / Target
W → Weight of variable
B → Bias
Then, the equation of boundaries is given as,
wx+b= +a
wx+b= -a
Thus, the equation which satisfies the hyper plane is given as
-a < Y- wx+b < +a

Our main aim here is to decide a decision boundary at 'a' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

Hence, we are going to take only those points that are within the decision boundary and have the least error rate or are within the Margin of Tolerance. This gives us a better fitting model.

Decision Tree Regressor

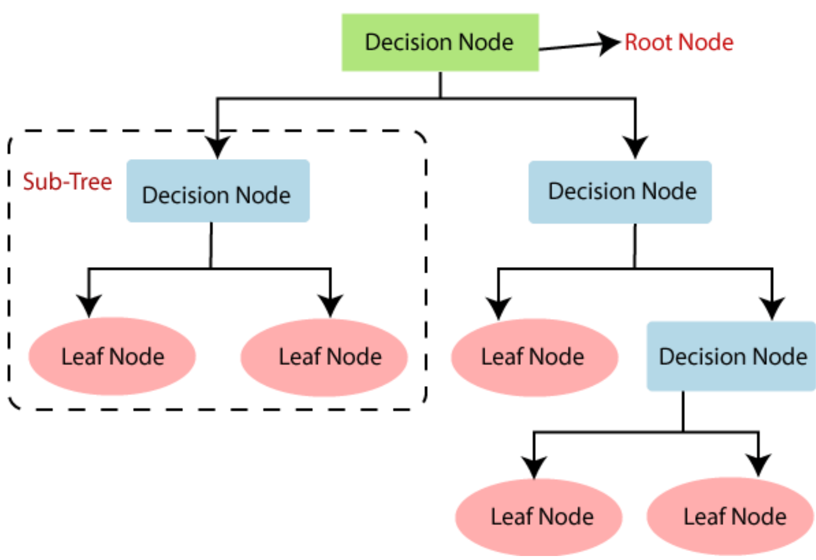

Figure 7.3 Decision Tree Regression model

Decision tree regression examines an object's features and trains a model with a tree structure to predict data in the future to generate meaningful continuous output. Continuous output indicates that the output/result is not discrete, that it is not solely represented by a discrete, well-known set of numbers or values.

Decision Tree models are widely used for classification problems, but it can also be used for regression problems also. Here the tree consists of tree node, sub node and decisions node. Our main target is to reach at the leaf node.

The decision-making rules and its respective variables are decided by Gini impurity.

Random Forest Regressor

Every decision tree has a high variance, but when we combine all of them in parallel, the resultant variance is low because each decision tree is perfectly trained on that specific sample data, and thus the output does not depend on one decision tree but on multiple decision trees. In the case of a classification problem, the majority voting classifier is used to determine the final output. The final output of a regression problem is the mean of all the outputs. This section is called Aggregation.

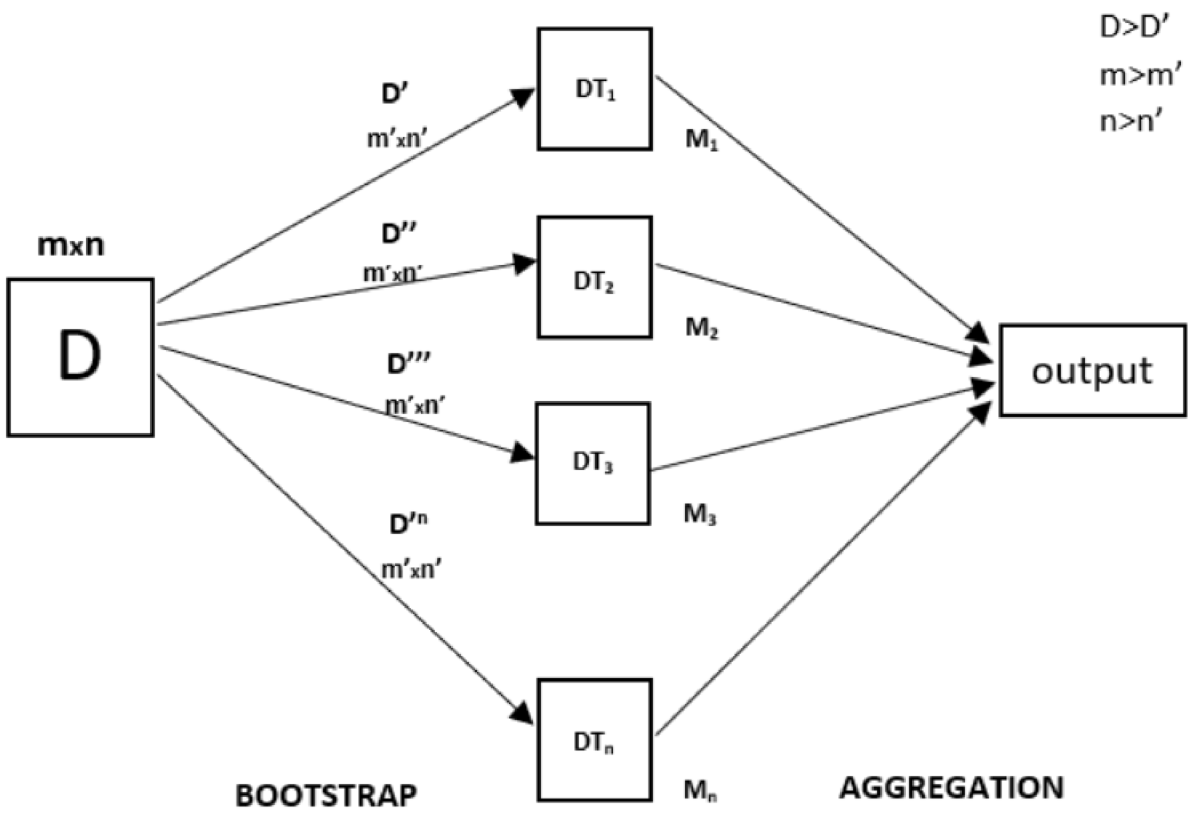

Figure 7.4 Random Forest Regression Model

A Random Forest is an ensemble technique that combines multiple decision trees and a technique known as Bootstrap and Aggregation, or bagging, to perform both regression and classification tasks.. The basic idea is to use multiple decision trees to determine the final output rather than relying on individual decision trees.

Random Forest's base learning models are multiple decision trees. We randomly select rows and features from the dataset to create sample datasets for each model. This section is known as Bootstrap.

7.2 Machine Learning Implementation on the Collected Data

We have implemented the regression models to our data as we have a target variable in our dataset. The following steps were taken in order to build a robust machine learning model.

Step 1: -
Import the necessary libraries
The libraries were utilized in order to perform the exploratory data analysis tasks and also further implementation of machine learning techniques.

Code: -
```
import numpy as np //Import Numpy Library
import pandas as pd //Import Pandas Library
```

Step 2: -
Import the data into pandas' data frame
Here we will input the data from excel/csv or text file into pandas' data frame. The primary reason for doing it, is to analyze and operate the data in an efficient manner.

Code: -
```
df = pd.read_excel("DATA_HARDWARE_ML.xlsx") //Load the excel data into pandas dataframe
```

Step 3: -
Data Cleaning
The collected data from the simulation using data logging may contain garbage values. We cannot feed such garbage values to machine learning models as it will underperform with the actual data. For the cleaning purpose of data, we use domain knowledge and try to filter data in such form that it actually makes sense. Hereby, we have removed all the NaN values present in the data, Zero and negative values of temperature and also restricted the temperature value below 45°C and Irradiance value below 1000 W/M^2.

Code:-
```
Cols =
['Temperature','Humidity','Irradiance(W/M2)','Panel_Power_MPPT_Loading','Panel_Power_Direct_Loading'] //Load all the features in cols variable.
df[cols] = df[cols].replace({'0':np.nan, 0:np.nan}) // Replace all the zero values with Nan
df.dropna() //Remove all the Nan Values
```

df = df[(df['Temperature']>0) &(df['Temperature']<46)& (df['Panel_Power_MPPT_Loading']>10)] //Condition to remove outliers and get temperature in the range of 0 to 45°C and irradiance below 1000 W/M^2.

Step 4: -

Exploratory Data Analysis

Here we will analyze the data through various plots and other method. Our primary purpose behind data analysis is to understand our data in a much better way and to find the loopholes is exists.

Code: -

```
df.describe() //To get the overall statistical summary of the data.
df.count() // To get the count of rows in each column
df.isnull().sum()  // To get the count of all the null values in respective columns.
%matplotlib inline                         // To get the histogram plot of all the respective features
import matplotlib.pyplot as plt
df.hist(bins=50,figsize=(20,15))
plt.savefig("Features_Histogram_Plot")
plt.show()
```

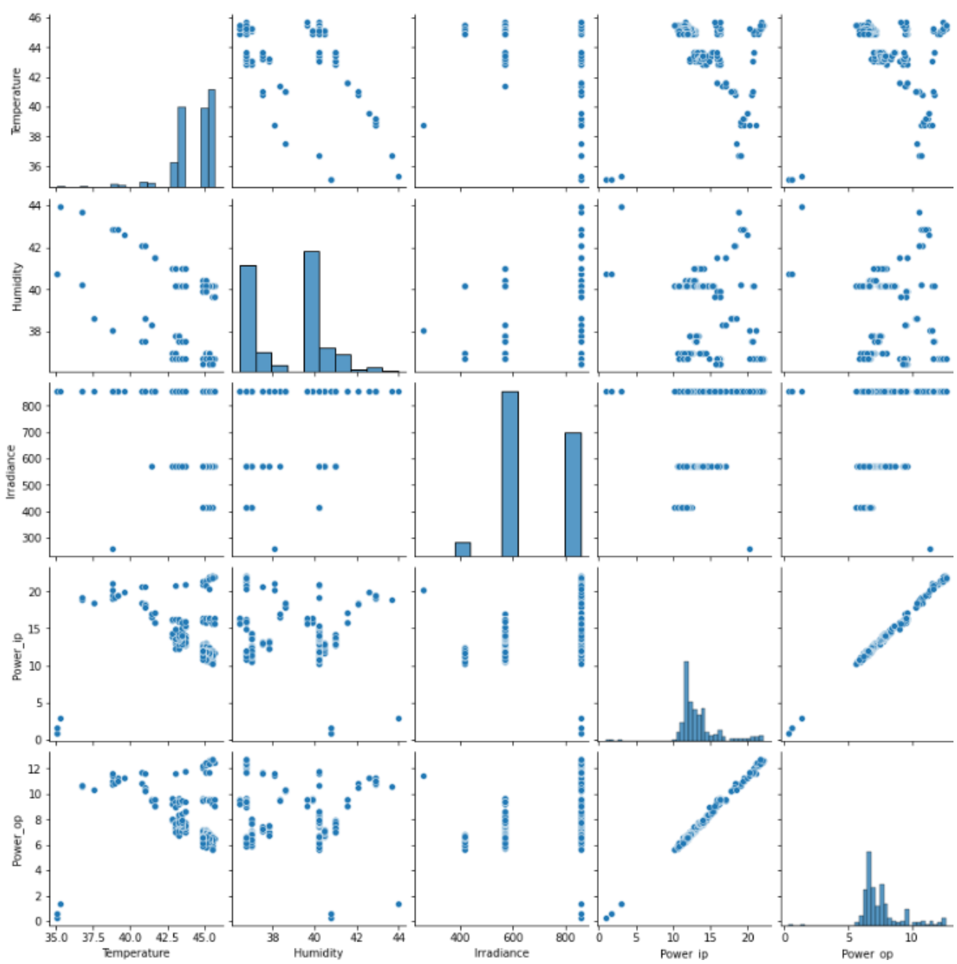

Figure 7.5 Overall relation between data features

Step 5: -

Splitting of training and test dataset

As we have cleaned and transformed the data into a proper manner, we will further move towards splitting of training and test dataset. The training dataset is utilized to train the data.

Code: -

```
from sklearn.model_selection import train_test_split //Import train_test_split function
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size = 0.3,random_state=2529)
X_train.shape, X_test.shape, Y_train.shape, Y_test.shape  //splitting of training and test dataset
```

Step 6:-

Feature Sampling

The features associated with our data have values in discrete range. Hence, it becomes necessary to transform the data in a particular scale of our choice. Sklearn Library gives us an opportunity to transform the data into a scaled one.

Code: -

```
from sklearn.preprocessing import StandardScaler //Import Standard Scaler Function
scaler = StandardScaler()
x_train = scaler.fit_transform(X_train) //Perform scaling operation on each training/test dataset
x_test = scaler.fit_transform(X_test)
y_train = scaler.fit_transform(Y_train)
y_test = scaler.fit_transform(Y_test)
```

Step 7: -

Machine Learning Model Training

As we have our data ready for training. We will further move towards the training of machine learning model. Various regression machine learning models will be used for the training purpose.

Code: -

```
from sklearn.linear_model import LinearRegression  // Import LinearRegression Library
lr = LinearRegression()
lr.fit(X_train_hum,Y_train_hum) // Fit the data into Linear Regression Model
y_pred_hum = lr.predict(X_test_hum) // Predict the values with trained model
```

Step 8: -

Evaluation of Machine Learning Model

After training the machine learning model, we will further test it with the testing data. There are various parameters used to evaluate the machine learning model.

Mean Absolute Error (MAE)

The mean absolute error is one of the most widely parameter in order to evaluate the performance of regression machine model. Here the actual value is compared with the predicted value.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

N → Total Number of Observations.
Y_i → Actual Value.
ŷ → Predicted Value.

Mean Squared Error (MSE)

The mean absoluter error contains negative value which may reduce the overall effect of error associated with our machine learning model. Hence, here we square all the values to get only the positive values and greater attention to error can be given.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

N → Total Number of Observations.
Y_i → Actual Value.
ŷ → Predicted Value.

Root Mean Squared Error (RMSE)

The most well-known regression model evaluation metric is RMSE. RMSE is calculated in the same way as MSE, except that the final value is square-rooted, as we calculated the square of errors in MSE. In MAE, we learned that any new prediction would be in the [Ypredicted-Error, Ypredicted+Error] range at the time of inference. We squared the error in MSE, so we must calculate the square root to return it to the normal stage.

$$RMSE = \sqrt{\frac{1}{N}\Sigma_i^N(y_i - \hat{y}_i)^2}$$

N → Total Number of Observations.
Y_i → Actual Value.
ŷ → Predicted Value.

R – Squared (R^2)

The strength of a relationship between two variables is explained by their correlation. R-squared, on the other hand, explains how much the variance of one variable explains the variance of the second variable. It is also known as the Determination Coefficient.

$$R^2 = 1 - \frac{\Sigma_i^N (y_i - \hat{y}_i)^2}{\Sigma_i^N (y_i - \bar{y})^2}$$

N → Total Number of Observations.
Y_i → Actual Value.
ŷ → Predicted Value.

Adjusted R – Squared Error

To address R2's issues, researchers developed a new metric known as adjusted R2 that is considered an improvement in R2. N is the total number of data samples in the equation below, and k is the number of independent variables.

$$Adjusted\_R^2 = 1 - [(\frac{N-1}{N-k-1}) * (1 - R^2)]$$

N → Total Number of Data Samples.
K → Number of Independent Variables.

7.3 Machine Learning Results

Regression Algorithm	Mean Absolute Error	Mean Squared Error	R – Squared (r2 score)
Multiple Regression Model	***0.333***	***0.195***	**0.994**
Support Vector Regressor	***3.083***	***12.845***	***0.665***
Decision Tree Regressor	***1.487***	***3.725***	***0.902***
Random Forest Regressor	***0.917***	***1.294***	***0.966***

Table 7.1 Machine Learning Regression Models Performance

We have not considered the adjusted R squared error because, it is used when the number of features is more and, in our case, we have only 2 features and 1 target variable. Hence, there is no such requirement in our case.

We found that, Multiple Linear Regression model performs best in our scenario, as the number of features is less. Hence, it is desirable to use multiple regression model.

The final equation which we get from the Multiple Regression model is given below: -

MPPT Power = (0.04 * Irradiance) + (-0.176 * Temperature) + 2.835

Here, 0.04, -0.176 are the weights associated with their respective variables and 2.835 is the bias of entire relation.

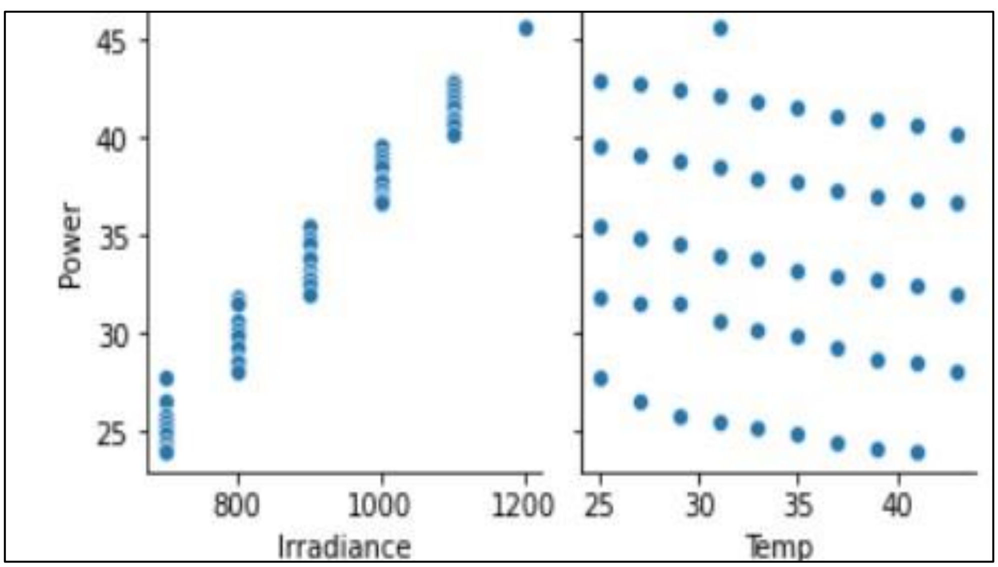

Figure 7.6 MPPT Power vs Irradiance, Temperature (Software data)

Machine Learning Model Performance on the Hardware collected data (Temperature, humidity, and Irradiance sensor data)

Regression Algorithm	Mean Absolute Error	Mean Squared Error	R – Squared (r2 score)
Multiple Regression Model	2.63	18.69	0.02
Support Vector Regressor	2.18	17.63	0.07
Decision Tree Regressor	1.02	4.62	0.75
Random Forest Regressor	0.75	2.14	**0.88**

Table 7.2 Machine Learning Regression models performance on hardware data

In the case of simulation data, we achieved maximum R-Squared Score with Multiple Regression model, the reason behind that was the data was not continuous and was steady at a multiple value of irradiance and temperature with less fluctuations. While, On the other hand in the case of sensor data there are lot of anomalies and after clearing those anomalies also, we have the data which is continuous or fluctuate more from a particular point of values. Random forest algorithm usually performs better in the regression model which contain large data that can be easily distributed among decision trees.

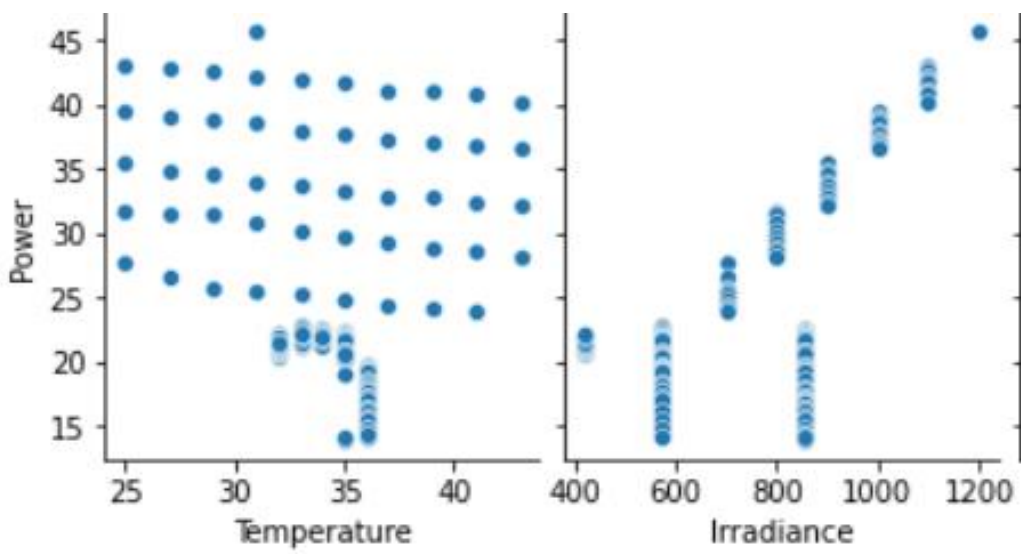

Figure 7.7 Power vs Irradiance, Temperature (Hardware Data)

7.4 Machine Learning Based User Interface System

After training and developing the model, it becomes essential to develop a system where user or power operator can operate and get the estimation of maximum output power with respect to the instance temperature and irradiance. Here the user will input irradiance and temperature and it output they will get the respective maximum output power.

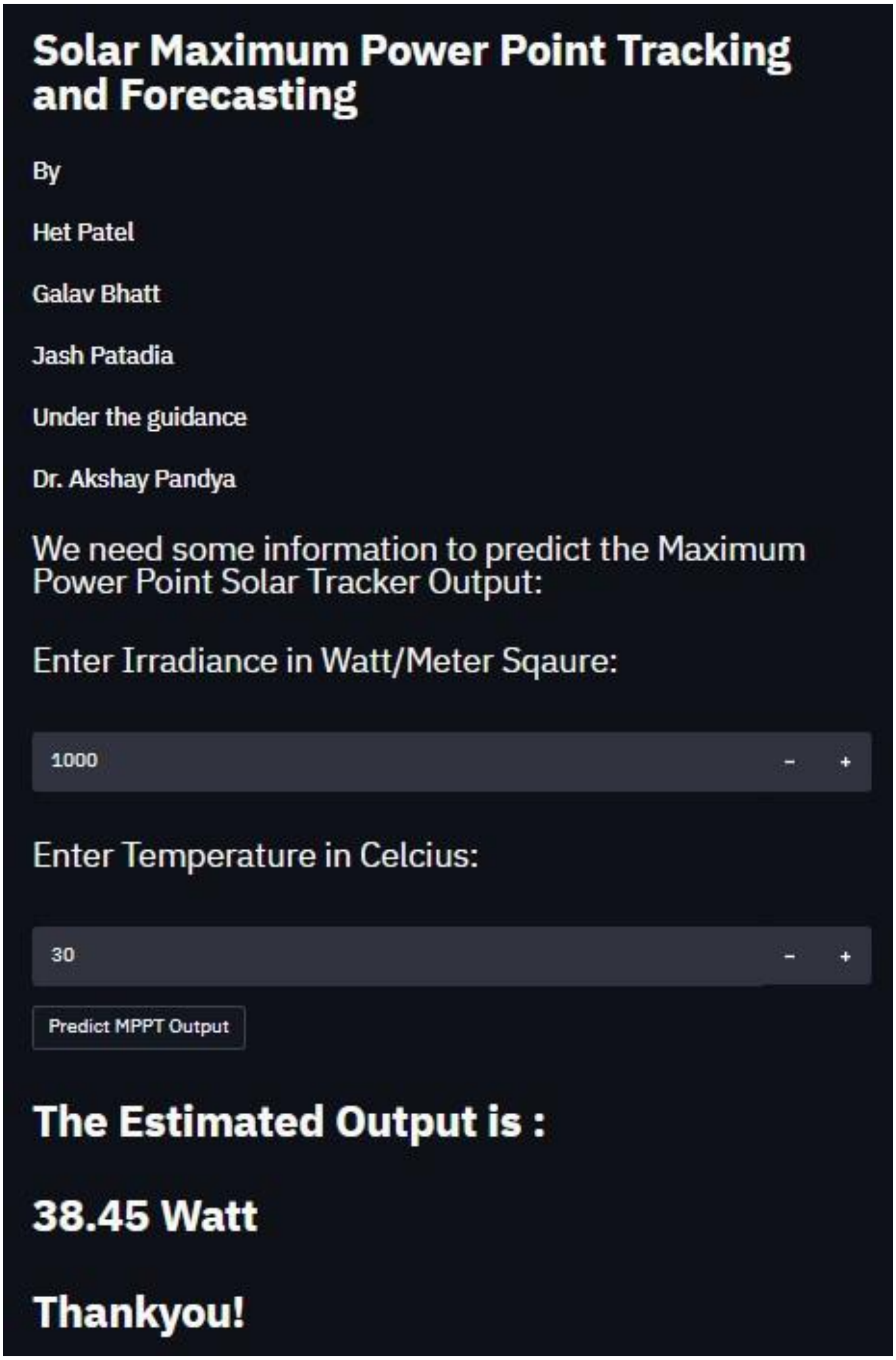

Figure 7.8 Machine Learning based User Interface System

Chapter 8: Conclusion and Future Scope

Conclusion

Solar energy is one of the solutions to pollution free energy, thus it is necessary to maximize the efficiency of the system. From the research papers studied and the basic study of solar panel operation we came across the fact that by directly loading the panel may not be able to achieve the maximum power that a panel can generate and to deal with the problem we decided to take this project. Purpose of this project was to create an intermediate stage which can help us control the operating power of the solar panel and help us achieve maximum power. To do so here we have used an algorithm **"PERTURB & OBSERVE"** to take steps to extract maximum power out of solar panel.

Along with that we have created a **machine learning model** with **88%** accuracy which will help user/solar panel operator to predict the generation with the help of parameters on which solar panel power depends which are irradiance and temperature. Further a user interface was created by using the machine learning model to predict the power. This UI was created using **"STREAMLIT"** library. This UI can be used to for forecasting of power and the user can plan the use of the energy throughout the day based on the prediction, also it can be used when we need to perform maintenance of a generating unit and using the solar plant instead of it to supply the energy.

From the test/monitoring we found out the by using MPPT as a intermediate stage the power at which the panel is operating gets almost doubled, which can be seen from the graphs attached in the hardware implementation chapter. The hardware consists of a DC/DC buck converter, Arduino Uno, Current sensor, DHT11 Sensor (Temperature & Humidity), SD Card module, Light intensity sensor and is tested for a DC bulb of 12V & 35W. According to our observation when the **dc bulb is directly connected to the panel, it operates at 10W, on the other hand after introducing MPPT the panel produces 20W power**. Further, while testing during evening (less sunlight) when the panel was directly loaded with dc bulb the panel was not able to supply enough power to glow the bulb (not even slightly) but after loading the panel through MPPT the panel was supplying enough power to glow the bulb, thus this explains the importance of the MPPT device.

As the MPPT can achieve maximum power out of the panel, we can maximize the panel power capacity that can lead us towards utilization of pollution free energy source

Future Scope

- ✓ The project can be developed using an arm controller instead of Arduino Uno.
- ✓ Current transformer can be used instead of current sensor to get more accurate results.
- ✓ Implement the project on Printed Circuit Board (PCB), so the overall size becomes compact, and it can be converted into product.
- ✓ Other techniques i.e., Incremental Conductance Method, Neural Network based Method etc. can be implemented as per the user requirement.
- ✓ Machine Learning based application can be developed where user enters the respective parameters into the app and get the respective MPPT output power.
- ✓ By implementing Battery Management system (BMS), it can be used to charge a battery with maximum possible power.

Paper Publication

After completion and successfully receiving the desired results from the project, we have further decided to write a research paper comprising of all the experiment methodologies and results. We will include all the steps associated with our experimentation process, also the problems associated with in the process and its respective solution. Further, we have sent our abstract to NATIONAL SYMPOSIUM ON ELECTROCHEMICAL SCIENCE AND TECHNOLOGY [NSEST - 2021] an Annual Event of ECSI organized by **THE ELECTROCHEMICAL SOCIETY OF INDIA Indian Institute of Science (IISc) Campus, Bengaluru**-560012 and DEPT.OF INORGANIC AND PHYSICAL CHEMISTRY Indian Institute of Science Campus, Bengaluru-560012. **Our abstract got selected by the technical committee and we were invited to submit our complete research paper and present poster at IISc Bangalore.**

References

1) Salman, S., AI, X. & WU, Z. Design of a P-&-O algorithm based MPPT charge controller for a stand-alone 200W PV system. Springer Conference on *Prot Control Mod Power Syst* **3,** 25 (2018). https://doi.org/10.1186/s41601-018-0099-8.
2) D. Beriber and A. Talha, "MPPT techniques for PV systems," *4th International Conference on Power Engineering, Energy and Electrical Drives*, 2013, pp. 1437-1442, doi: 10.1109/PowerEng.2013.6635826.
3) Villalva, Marcelo & Gozzoli, Jonas & Filho, Ernesto. (2009). Analysis and simulation of the P&O MPPT algorithm using a linearized PV array model. Conference: Power Electronics Conference, 2009. COBEP '09. Brazilian189 - 195. 10.1109/COBEP.2009.5347755.
4) https://towardsdatascience.com/predicting-solar-power-output-using-machine-learning-techniques by Article published in Towards Data Science by Abiodun Olaoye Mechanical and computational engineer (PhD @ MIT), Data scientist | Energy systems enthusiast.
5) Takruri, Maen & Farhat, Maissa & Barambones, Oscar & Ramos-Hernanz, & Turkieh, & Badawi, & AlZoubi, & Sakur, Abdus. (2020). Maximum Power Point Tracking of PV System Based on Machine Learning. Energies. 13. 692. 10.3390/en13030692.
6) https://www.enjoyalgorithms.com/blog/evaluation-metrics-regression-models by enjoy algorithms.
7) https://www.geeksforgeeks.org/machine-learning/ by geeks for geeks.
8) https://streamlit.io/ streamlit library.

Plagiarism Report

Plagiarism Checker X Originality Report

Similarity Found: 9%

Date: Thursday, May 12, 2022
Statistics: 1045 words Plagiarized / 12111 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

Date	Thursday, May 12, 2022
Words	1045 Plagiarized Words / Total 12111 Words
Sources	More than 142 Sources Identified.
Remarks	Low Plagiarism Detected - Your Document needs Optional Improvement.