OPTIMIZATION OF PHOTOVOLTAIC SYSTEM AND ALGORITHM DEVELOPMENT

BY

UWUOROYA OSAYUWAMEN JEFFERY	**ENG1704254**
ABIODUN-FALODUN DAMILOLA OMOREGBE	**ENG1704146**
OLUDOLA AYOKANMI CARLTON	**ENG1704228**
UWEZUKWE OZIOMA	**ENG1709451**
OKPAKO UFUOMA ANDREW	**ENG1303576**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN

FEBRUARY 2023

OPTIMIZATION OF PHOTOVOLTAIC SYSTEM AND ALGORITHM DEVELOPMENT

BY

UWUOROYA OSAYUWAMEN JEFFERY	**ENG1704254**
ABIODUN-FALODUN DAMILOLA OMOREGBE	**ENG1704146**
OLUDOLA AYOKANMI CARLTON	**ENG1704228**
UWEZUKWE OZIOMA	**ENG1709451**
OKPAKO UFUOMA ANDREW	**ENG1303576**

A PROJECT SUMMITED TO THE DEPARTMENT OF ELECTRICAL ELECTRONICS ENGINEERING, UNIVERSITY OF BENIN, BENIN CITY, IN PARTIAL FULFIMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING (B.ENG) DEGREE

UNIVERSITY OF BENIN

FEBRUARY 2023

CERTIFICATION

This is to certify that this project was carried out by UWUOROYA OSAYUWAMEN JEFFERY, ABIODUN-FALODUN DAMILOLA OMOREGBE, OLUDOLA AYOKANMI CARLTON, UWEZUKWE OZIOMA and OKPAKO UFUOMA ANDREW, of the department of Electrical/Electronic Engineering, University of Benin, in partial fulfillment of the requirements for the award of the Bachelor of Engineering (B.Eng.) Degree in Electrical/Electronic Engineering.

____

Engr. Dr. Omorogiuwa
Project Supervisor

____

Prof. K.O. Ogbeide
Head of Department

____
DATE

____
DATE

DEDICATION

This project is dedicated to the almighty God whose infinite mercy, grace and guidance has seen us through. Also, to our parents for their consistent prayers and support.

ACKNOWLEDGEMENT

We give thanks to Almighty God for his grace, loving-kindness, guidance and protection throughout our stay at the university of Benin.

Our sincere gratitude goes to our parents for their prayers, love, immense support and encouragement both during the execution of this project and throughout the duration of our academic pursuit at the University of Benin.

We wish to acknowledge with much appreciation the crucial contribution of our project supervisor Engr. Dr. Omorogiuwa, whose advice, suggestions, motivation, and encouragement had power worthwhile in the execution of this project.

Also big thanks to all the lecturers and staff of the department of electrical and electronic engineering, University of Beni, for the tutoring and enlightenment in the field of electrical/electronic engineering.

Completing this work would have been all the more difficult were it not for the support and friendship provided by our course-mates at the department of electrical/electronic engineering. University of Benin.

ABSTRACT

This project presents the optimization of PV systems and development of optimization algorithms for optimizing the power output of a solar PV system. The aim of the project was to investigate and understand the various existing optimization techniques and methods, then develop an algorithm that can optimize a Solar PV system.

The optimization method used was MPPT (maximum power point tracking) and P&O algorithm, INC algorithm and Genetic algorithm were developed. Development and simulation of the optimization methods and developed algorithms was achieved with the use of MATLAB SIMULINK software. For this project, the output of the optimized system with the MPPT method trained with P&O optimization, INC algorithm and Genetic algorithm were compared with the output of an unoptimized system under different atmospheric conditions and then to each other.

The results of the optimized system were compared with the result of the unoptimized system. From the result, it was found out that the optimized system has a better response, lesser error, better accuracy and tracking speed than the unoptimized system. At the end of the project, it became obvious for the need of optimization in PV system and the step-by-step process of developing an optimization algorithm was shown.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER ONE

1.0 INTRODUCTION

Due to its many benefits on photovoltaic power generation, commonly known as solar power, has attracted a lot of interest lately. It allows for the development of sustainable electricity without harmful emissions due to its clean and non-polluting energy production capabilities. Sunlight is immediately converted into electricity by photovoltaic systems, making it a dependable and sustainable source of power.

Electricity production near to the user is one of the main benefits of photovoltaic power generation. This lowers transmission losses, that occur when power is delivered over long distances, making the supply of energy more efficient. Furthermore, photovoltaic systems require less maintenance, which adds to their allure.

Additionally, solar systems have a particularly long lifespan that enables long-term electricity production. Because of their durability, engineers and investors seeking dependable and sustainable energy solutions appreciate them.

The power output of a photovoltaic (PV) grid array has been optimized using a variety of optimization approaches and control algorithms.

1.1 BACKGROUND OF THE STUDY

PV optimization is crucial for maximizing energy production, boosting system effectiveness, and increasing the PV installations' economic feasibility. PV systems may capture more sunlight and convert it into electricity by optimizing factors including system configuration, module orientation, and tracking devices, which increases energy generation and overall energy yield. Through the use of algorithms that evaluate data, simulate system behavior, and pinpoint ideal operating points, optimization also minimizes energy losses and maximizes power production. By altering system components in real-time, this ensures that PV systems run as efficiently as possible.

Genetic algorithms and machine learning are two recent examples of technology that have benefited the field of PV optimization. These cutting-edge algorithms allow for effective exploration of complicated parameter spaces while taking into account a variety of factors and restrictions to identify the ideal PV system configurations and operational approaches. By fine-tuning optimization algorithms, PV systems produce more energy, are more efficient, and are less expensive while operating at their peak capacity. PV installations can give optimum performance while overcoming obstacles and constraints thanks to this exact control and optimization.

1.2 STATEMENT OF THE PROBLEM

A viable renewable energy source to address climate change and lessen reliance on fossil fuels is solar energy. The inefficient use of solar energy is hampered by the intermittent nature of solar power production and the fluctuation of solar resources.

Solar resource collection and use by solar cells (PV cells) are maximized via solar optimization. More energy may be extracted from individual solar cells using nature-inspired algorithms, increasing system dependability, maximizing financial returns, and boosting energy output.

1.3 AIMS

This study aims is optimization of photovoltaic system and algorithm development

1.4 OBJECTIVES

The objectives of this study are

a. to explore the various methods of solar PV optimization system

b. to optimization solar PV system.

c. to develop three algorithms for solar optimization.

d. comparison of optimized system to unoptimized system

e. comparison of the various optimization to each other.

1.5 RESEARCH METHODOLOGY

a. the various methods of PV optimization will be explored theoretically

b. PV system will be optimized using MPPT optimization method

c. P&O algorithm, INC algorithm and Genetic algorithm will be used for PV system optimization, these algorithms will be developed in MATLAB Simulink

d. the response of an optimized system (P&O and GA) will be compare with an unoptimized system under different conditions using MATLAB Simulink.

e. the response of the P&O and GA algorithm was compared to each other.

The project employs a simulation-based approach to evaluate and compare the effectiveness of the optimization methods. A computational model representing a typical PV system is developed, and various scenarios are simulated to analyze the impact of each optimization technique on energy output and system efficiency. Figure 1.1 shows the means of achieving the required result. The authors review on solar optimization methods and approach for developing an effective algorithm. A comprehensive review will be discussed on this in the literature review section. The algorithm developed will be simulated using MATLAB and the results analyzed in Results and Analysis section of this study.

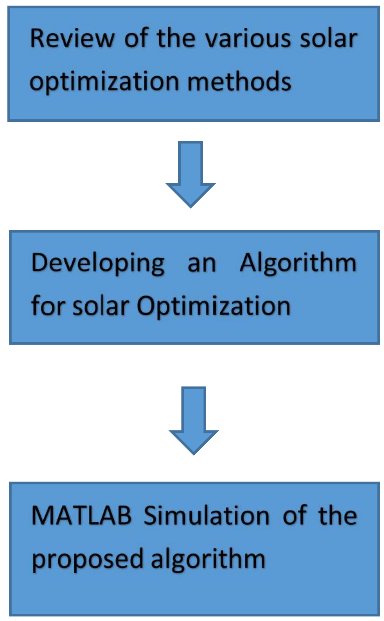


Figure 1.1: Graphical demonstration of methodology

1.6 SCOPE OF STUDY

This research involves the study of solar optimization and algorithm. The optimization methods and algorithms such as

- Maximum Power Point Tracking (MPPT) Control Method:

  Perturb and Observe (P&O) algorithm

  Incremental Conductance (INC) algorithm

- Particle Swarm Optimization (PSO) method
- Artificial Neural Network (ANN) method
- Fuzzy Logic Control (FLC)

- Genetic Algorithm (GA)
- Estimated-perturb-perturb (EPP)

And Nature-inspired optimization algorithms like

- Bat Foraging Optimization (BFO) algorithm.

Will be the main focus of this study.

CHAPTER TWO

2.0 LITERATURE REVIEW

Significant emphasis has been paid to solar energy as a clean and sustainable energy source. Researchers have created a variety of optimization algorithms and strategies to increase the effectiveness of solar energy systems. This review discusses the methodologies and conclusions of various eminent scholars who have contributed to this topic. Significant emphasis has been paid to solar energy as a clean and sustainable energy source. Researchers have created a variety of optimization algorithms and strategies to increase the effectiveness of solar energy systems. This review discusses the methodologies and conclusions of various eminent scholars who have contributed to this topic.

2.1 REVIEW OF LITERATURE

A new optimization approach for maximizing the photovoltaic panel power based on genetic algorithm and language" by El-Arini MM, Othman AM, Fathy A (2013): For the purpose of optimizing the location of solar panels, El-Arini suggested a brand-new method based on genetic algorithms. Their strategy intended to optimize the solar system's total energy production by taking into account elements including panel orientation, geography, and shade.

Modeling of photovoltaic system and design of MPPT controller using PSO algorithm" by J. Prakash, Sarat Kumar Sahoo (2013): Particle swarm optimization (PSO) is a method that J. Prakash developed for the positioning of solar panels. Real-time weather information was used in their research to dynamically improve panel location, resulting in greater energy production efficiency.

Design and implementation of a solar tracking algorithm" by Stamatescu et al. (2014): In his thorough examination of sun tracking algorithms, Stamatescu contrasted numerous strategies including azimuth-elevation, two-axis tracking, and slanted tracking. Their study assessed how various algorithms affected energy production and system cost-effectiveness.

Solar tracing by network adaptive techniques" by Ashhab, M.S (2011): With the use of machine learning methods, Ashhab suggested an adaptive control system for solar tracking that would dynamically change the orientation of the panel in response to outside factors in real time. Their approach enhanced energy production while also improving tracking accuracy.

Using a shading matrix to estimate the shading factor and irradiation in a three-dimensional model of a receiving surface in an urban environment" E.G Melo et al. (2013): Melo created a shading analysis program that evaluates the impacts of shade on solar panels using 3D modeling methods. Their study intended to maximize energy output by minimizing shadowing and optimizing the arrangement of solar panels.

2.2 OPTIMIZATION METHODS AND ALGORITHMS

In order to maximize the power output of a photovoltaic (PV) grid array, various techniques and control methods have been proposed. One such method is the Maximum Power Point Tracking (MPPT) control, which aims to find the optimal operating point at which the PV system can generate the maximum power output.

The MPPT control method involves estimating the process by perturbing the system and observing its response. Typically, the estimation and perturbation processes are repeated at regular intervals to continuously search for the maximum power output of the PV grid array. An advanced version of this method is the estimate-perturb-perturb (EPP) method, which significantly improves the tracking accuracy and speed of the MPPT control.

The "estimate-perturb-perturb" (EPP) method helps us estimate parameters in a statistical model when it's hard to directly evaluate the likelihood function. Here's how it works: First, we make an initial guess for the parameters. Then, we make some random changes to that guess. We check how well the model fits the data for each new guess. Based on these results, we assign weights to the different guesses. Finally, we calculate a new parameter estimate by combining all the guesses, taking into account their weights. The EPP method allows us to explore the different possibilities for the parameters and deal with problems like finding the best solution or dealing with uncertainty. Its success depends on the specific problem and model we're working with. Figure 2.1 shows the flow chart of this method.

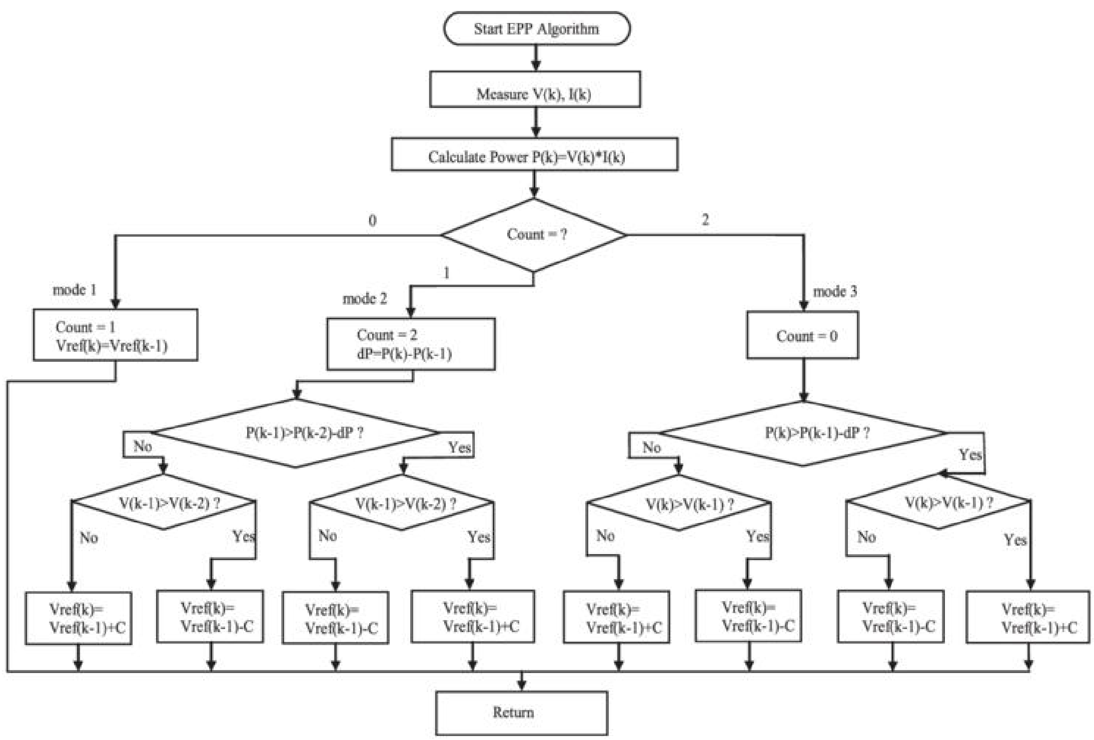


Figure 2.1: Flow chart for Estimate-Perturb-Perturb (EPP)

The use of optimization methods to improve the performance of solar systems has also been studied by researchers. The Bat Foraging Optimization (BFO) algorithm is one such method that seeks out the best answers by simulating the foraging activity of bats.

The Bat Foraging Optimization (BFO) algorithm is a nature-inspired algorithm created in response to how bats locate food. A collection of virtual bats represents potential solutions to a problem from the onset. Bats look for the best replies by moving their places and producing sounds. Better-performing bats attract more individuals and create a group. They travel aimlessly and sometimes uncover whole new places. The algorithm is terminated when a given criterion is met. BFO is a popular treatment since it is simple and effective for a wide range of situations.

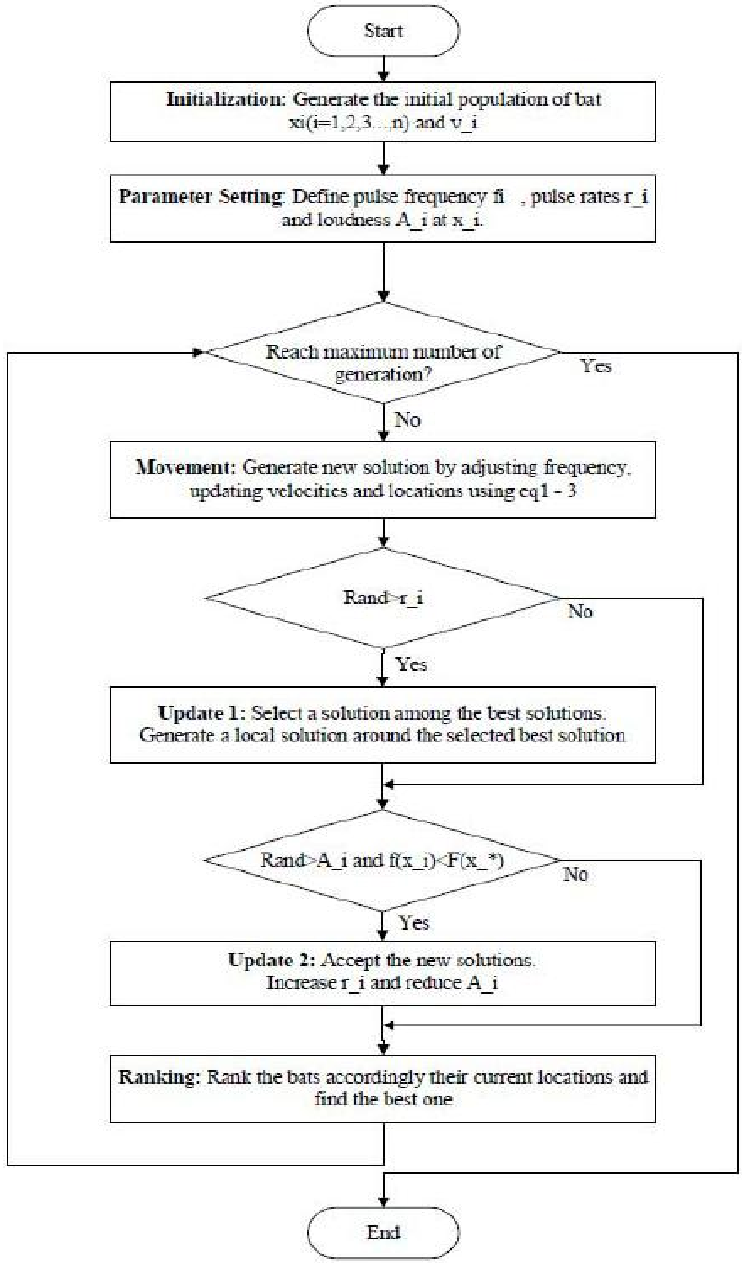


Figure2.2: Flow Chart of Bat Foraging Optimization (BFO) algorithm

The Artificial Bee Foraging Optimization (ABFO) approach, which was inspired by bee behavior, was also used. The goal of these strategies is to create well-organized forecasting models that anticipate the ideal input parameters for maximum power output.

Particle Swarm Optimization (PSO) is another nature-inspired optimization approach that replicates particle social behavior in order to find the highest power point. PSO has been used in solar systems to boost output power, particularly under different irradiation situations, including partial shadowing.

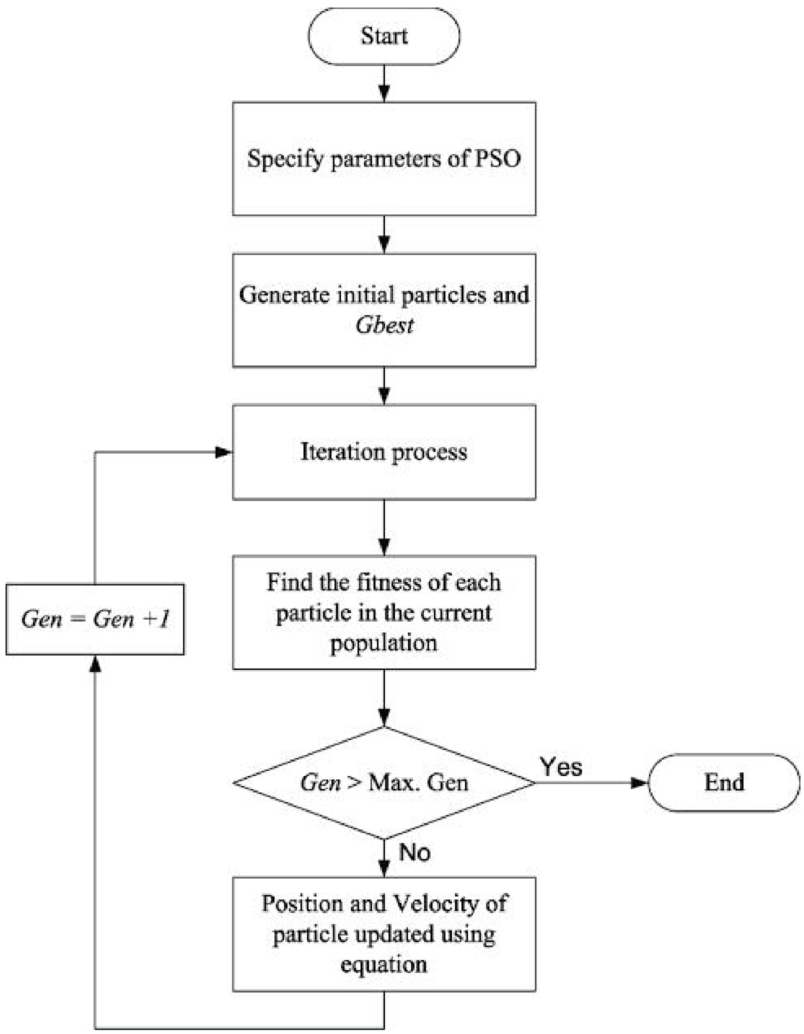


Figure 2.3: Flow Chart of Particle Swarm Optimization (PSO)

MPPT in solar systems has also been suggested using Artificial Neural Network (ANN) approaches. ANN algorithms monitor the Maximum Power Point (MPP) using feedback voltage and current data under varying solar irradiation and temperature conditions. This method allows for quick and exact searches for the MPP, allowing for more efficient power production.

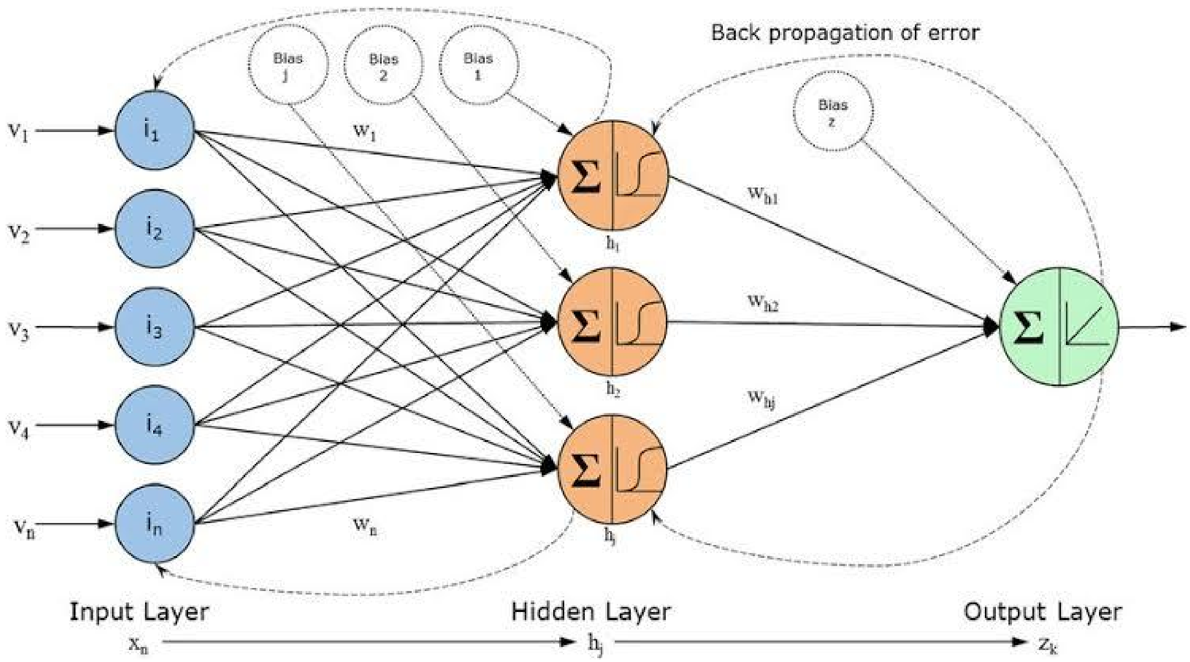


Figure 2.4: Flow Chart of Artificial Neural Network (ANN) methods

Researchers investigated the combination of Fuzzy Logic Control (FLC) with MPPT to provide steady and dependable power conversion. FLC is a control method that models and controls complicated systems using linguistic variables and fuzzy sets. When used in combination with MPPT, FLC tries to prevent variations in output power by using a discrete Proportional-Integral (PI) controller. This method aids in achieving the intended MPP even in the face of changing weather conditions.

The fuzzy logic PV control method optimizes the control of photovoltaic (PV) systems by using fuzzy sets and rules. Fuzzifying crisp inputs, assessing fuzzy rules based on degree of membership, deciding on control actions, de-fuzzifying fuzzy outputs, and executing control actions are all part of the process. The algorithm is capable of managing inaccurate and unpredictable inputs, with the goal of optimizing the PV system's power production.

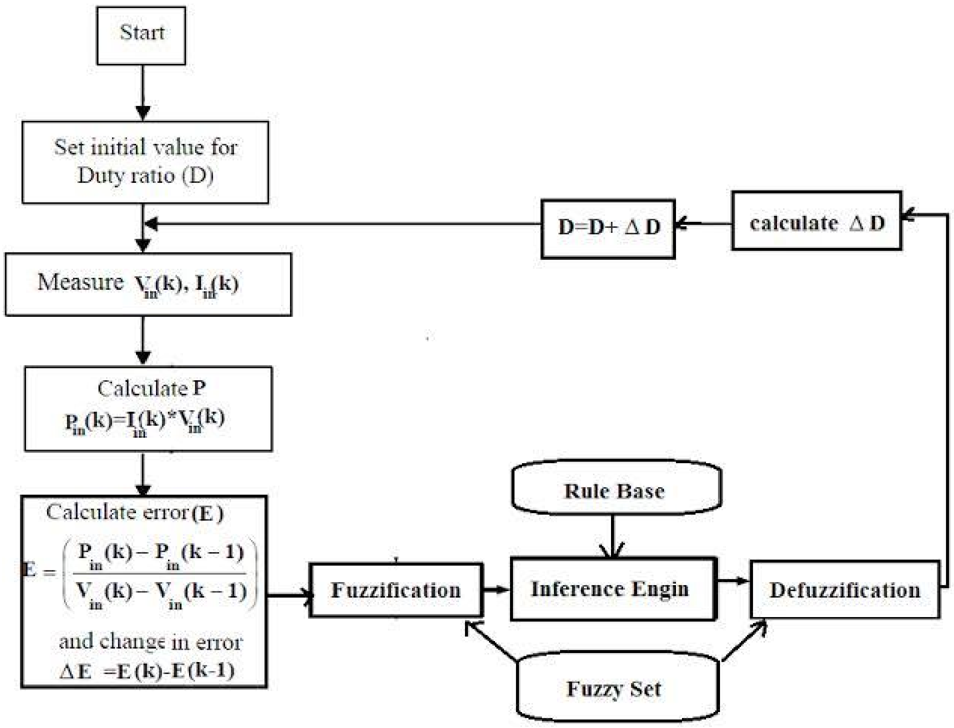
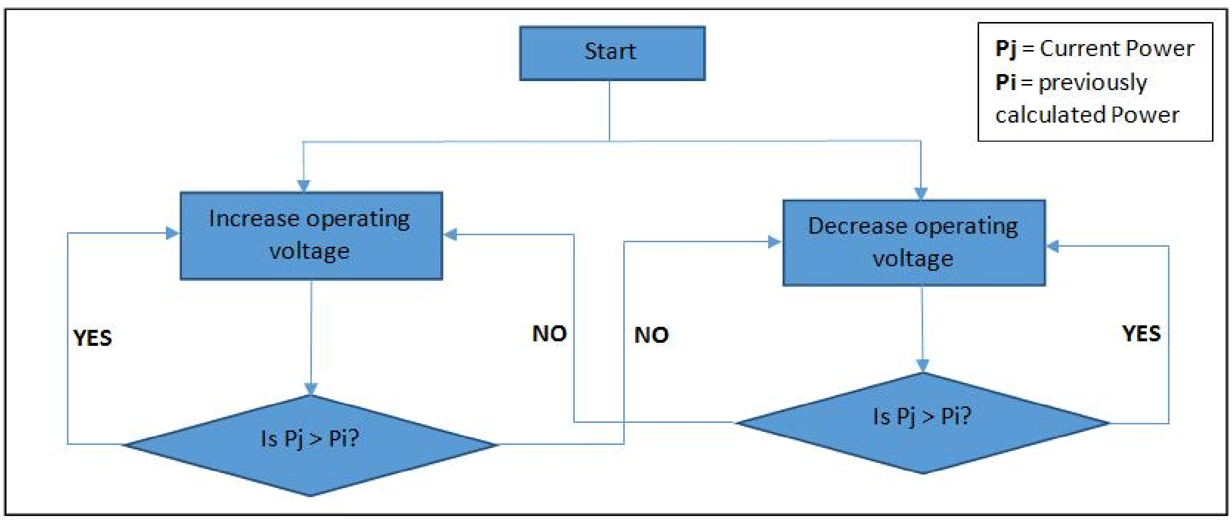


Figure 2.5: Flow Chart of Fuzzy Logic Control (FLC)

The efficiency of several optimization techniques has been studied. In one research, the Swarm Optimization Algorithm (SOA) was compared to other optimization methods such as the BFO and the Genetic Algorithm (GA). The findings showed that SOA is more successful than BFO or GA in determining optimum transient performance.

To monitor the MPPT of solar systems, swarm intelligence has been used in combination with the Perturbation and Observation (P&O) approach. The P&O approach includes changing the operating point and measuring the power output.



Flowchart for Perturb and Observe

Figure 2.6 : Flow Chart of Perturbation and Observation (P&O) method

CHAPTER THREE

RESEARCH METHODOLGY

3.0 INTRODUCTION

The various methods of PV system optimization were discussed and a methos of PV system optimization using MPPT investigated using the models from the Electrical toolbox in Simscape library in MATLAB Simulink. The P&O algorithm was developed following a step-by-step approach and used alongside the MPPT method of PV optimization chosen. This system was simulated and the result compared to an unoptimized PV system.

3.1 EXPLORING THE VARIOUS METHODS OF SOLAR PV OPTIMIZATION SYSTEM

The most popular solar PV optimization system is the MPPT, particle swarm optimization, artificial neutral network and fuzzy logic. Each optimization technique is suitable for certain level of system complexity.

3.1.1 MPPT (Maximum Power Point Tracking) Control Method

MPPT is a photovoltaic system approach that ensures that solar panels function at their maximum power point (MPP) despite shifting environmental circumstances. The MPP is the voltage and current combination that provides the most electricity from the solar panel. To identify and maintain the MPP, MPPT algorithms continually monitor changes in sun irradiance, temperature, and load. The following are the two most frequent MPPT control methods:

Perturb and Observe (P&O): This approach changes the operating point of the PV system and measures the power change. The operating point is then adjusted in the direction of power growth until it achieves the MPP.

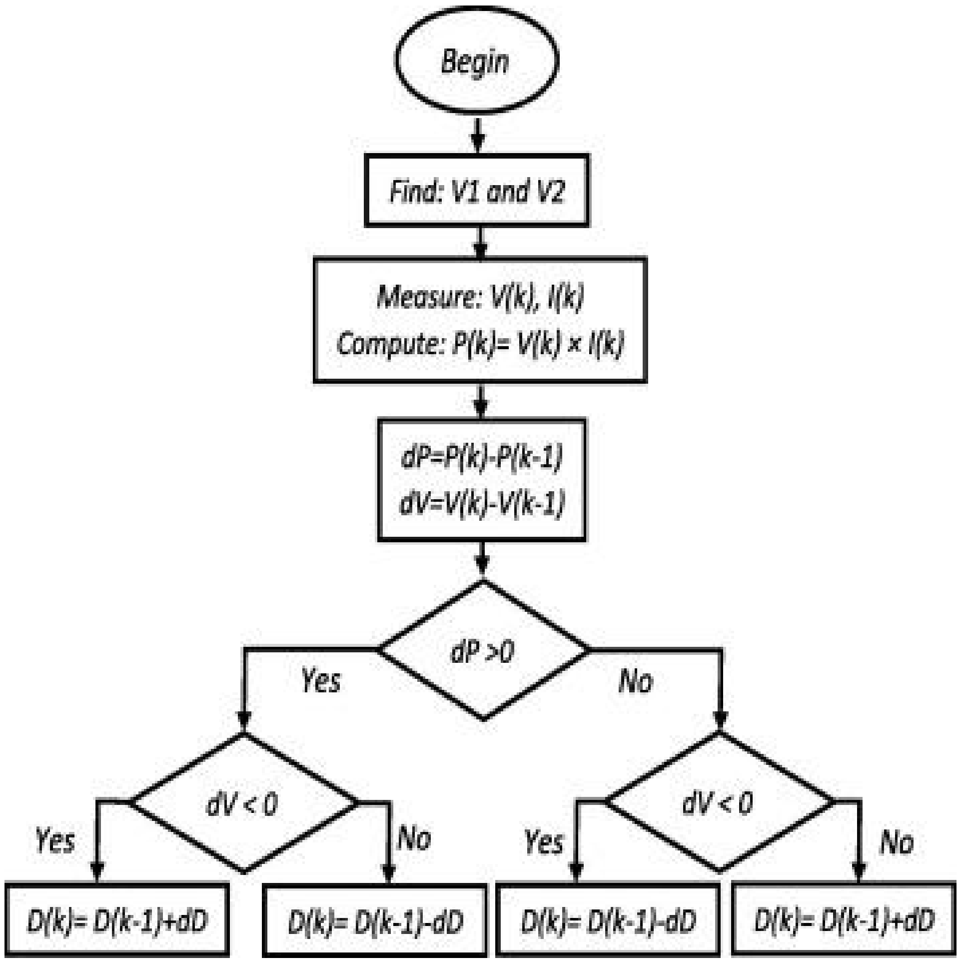


Figure 3.1: P&O flowchart

Incremental Conductance (INC): The INC technique compares the PV system's incremental conductance (dI/dV) to its instantaneous conductance. It modifies the operating point in order to maintain the PV system at the MPP.

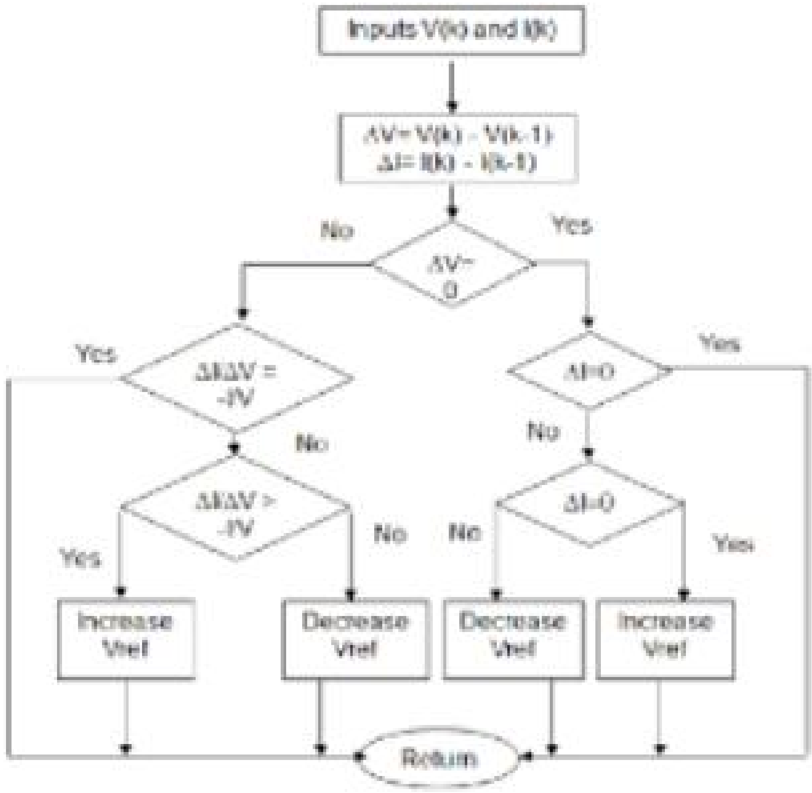


Figure 3.2: INC flowchart

MPPT management systems are critical for maximizing energy output, particularly in scenarios with variable solar irradiation or shade.

3.1.2 Particle Swarm Optimization (PSO) is a technique for optimizing particle swarms.

PSO is a nature-inspired optimization algorithm that mimics the social behavior of flocks of birds or schools of fish. In PSO, a population of particles traverses a multidimensional search space in pursuit of the best solution. Each particle modifies its location depending on its current position, best-known position, and neighbors' best-known positions. The method is repeated repeatedly until the best answer is discovered.

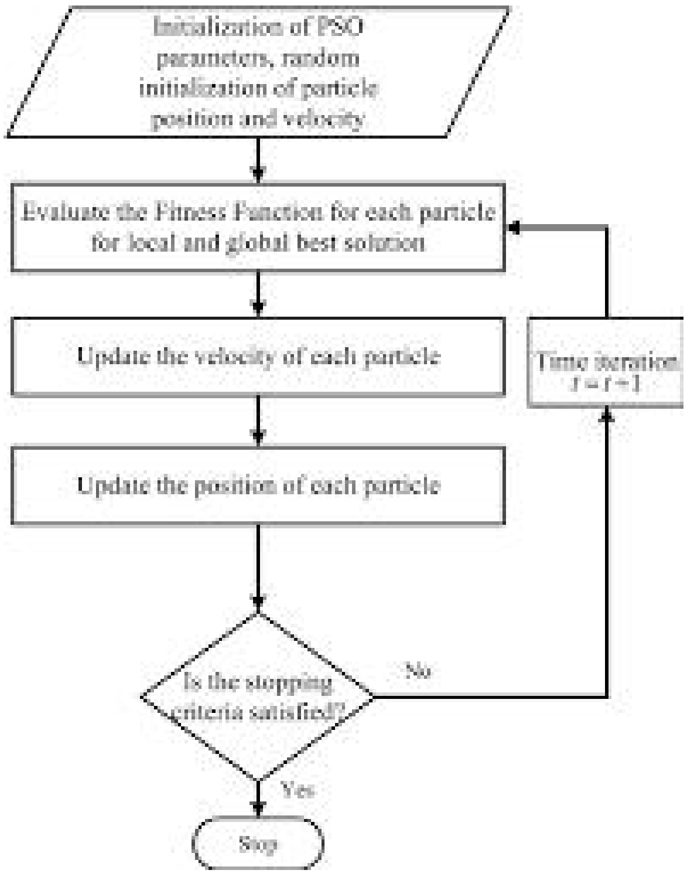


Figure 3.3: PSO flowchart

PSO may be used in PV optimization to discover the ideal operating point of solar panels to obtain maximum power production under changing environmental circumstances. PSO is a computationally efficient algorithm that can deal with nonlinear and complicated systems.

3.1.3 Methods of Artificial Neural Network (ANN)

ANNs are computer models inspired by the neural networks of the human brain. They are made up of linked nodes (neurons) arranged in layers. Each neuron gets input, processes it using an activation function, and then outputs. ANN models may recognize patterns in past data and forecast or make judgments based on fresh information.

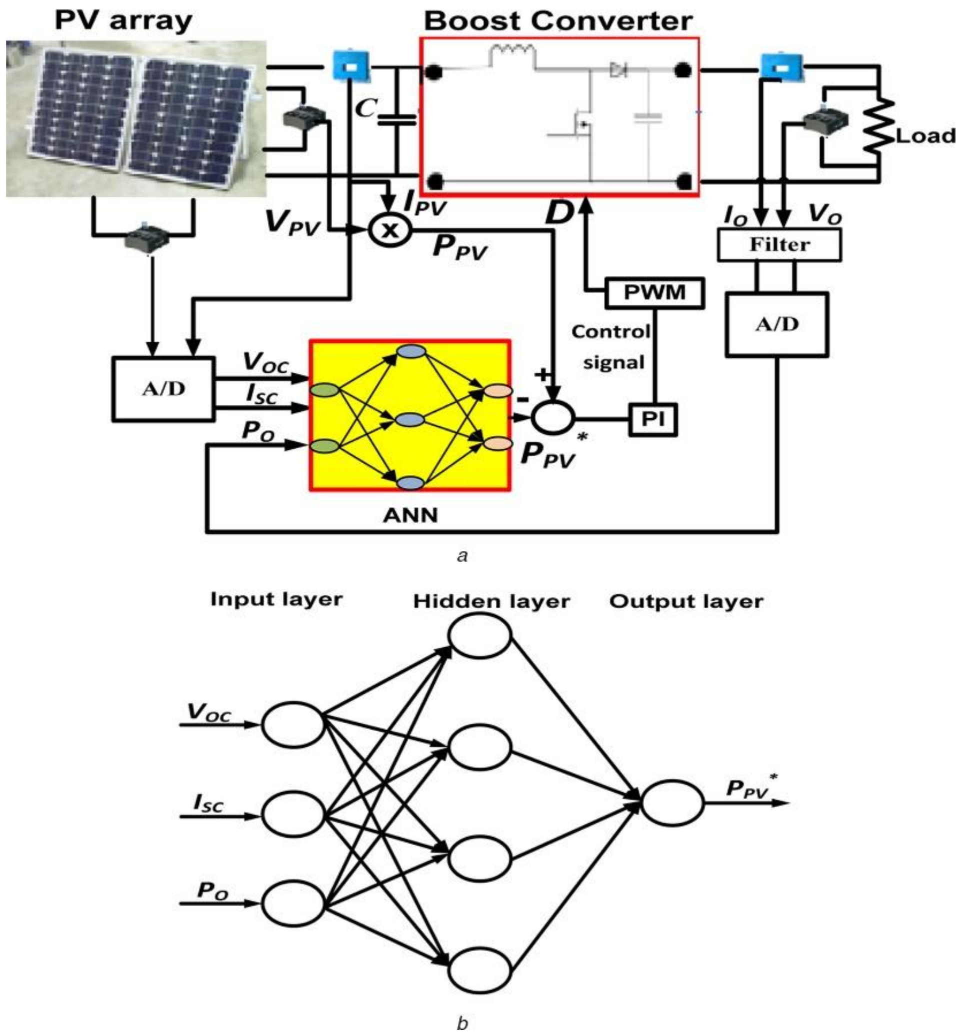


Figure 3.4: ANN system

ANNs may be utilized to construct MPPT controllers in PV optimization. The ANN can learn the link between inputs (solar irradiance, temperature, etc.) and outputs (MPP voltage and current) by training it using data from changing solar circumstances. Trained ANNs may then anticipate the MPP under different environmental circumstances, enabling real-time optimization.

3.1.4 Fuzzy Logic Control (FLC)

FLC is a fuzzy logic-based control system that handles uncertainty and imprecision in incoming data. In systems where the boundaries between distinct states are not well-defined, fuzzy logic allows for more flexible and human-like decision-making.

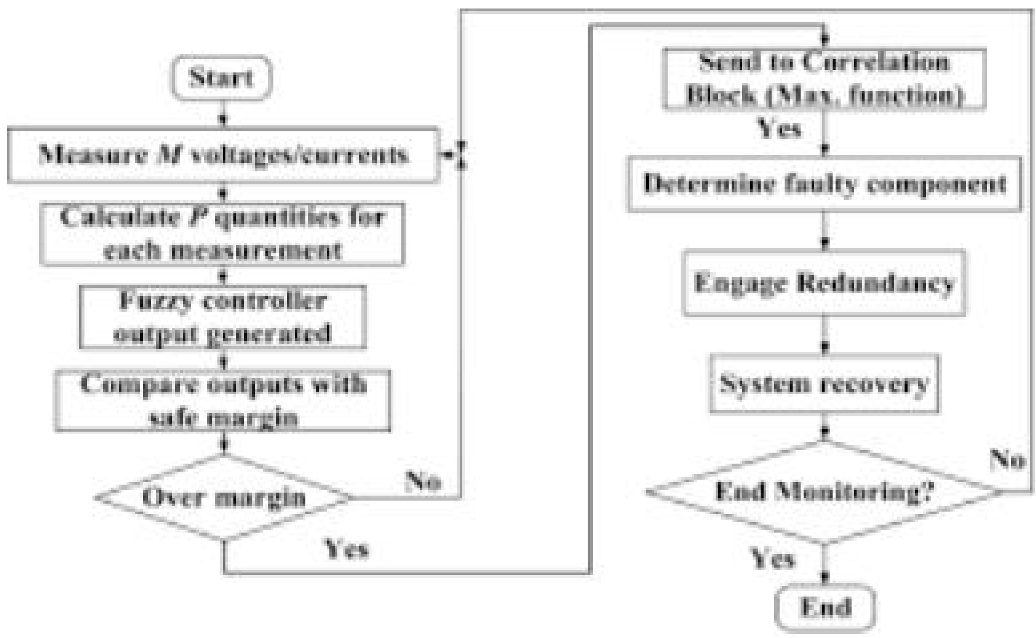


Figure 3.5: Fuzzy logic flowchart

FLC may be used in PV optimization to create MPPT controllers that respond to changing solar conditions and ambient variables. FLC-based controllers can successfully manage partial shade, variable temperature, and other dynamic circumstances.

3.2 OPTIMIZATION SOLAR PV SYSTEM USING MPPT

The growing need for clean, renewable energy has resulted in considerable breakthroughs in solar PV technology. However, solar energy production is heavily impacted by elements such as weather, shadowing, and panel temperature, all of which may have an impact on system efficiency and output. The goal of this research is to investigate and assess different PV optimization approaches in order to overcome these difficulties and improve the overall performance of solar PV systems. Among the numerous optimization approaches, this research will concentrate on an MPPT optimized system. The proposed PV system for this project is shown in figure 3.6

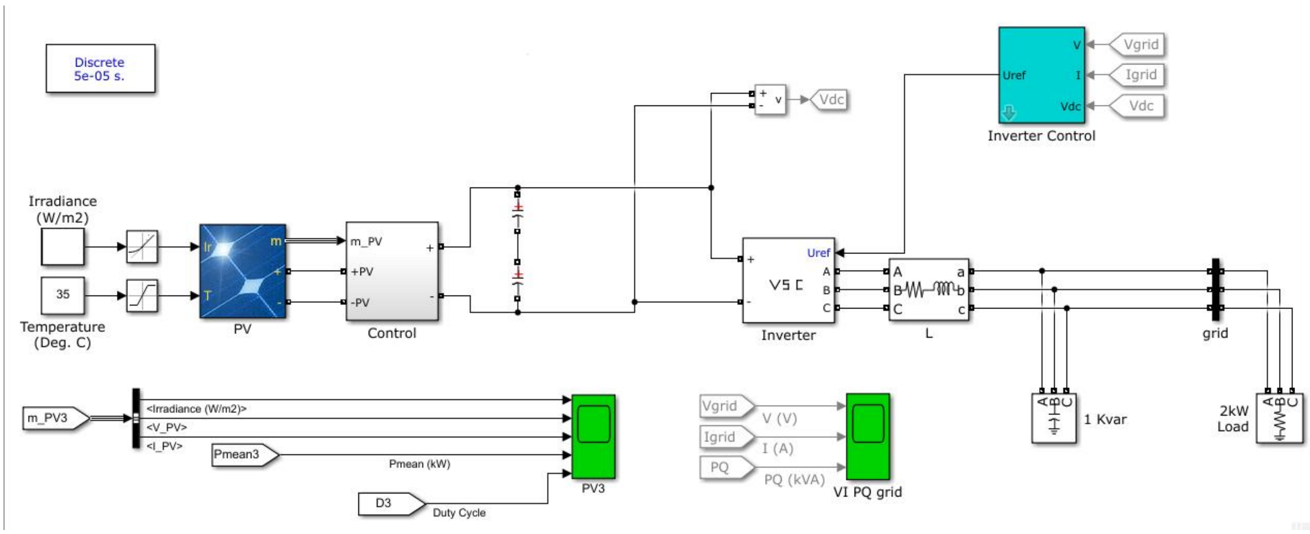


Figure 3.6: The photovoltaic system

The system uses 315W solar panels with the following specifications

Maximum power (w)	315.072
Open circuit voltage Voc (v)	64.6
Voltage at max. power point Vmp (v)	54.7
Temp. coefficient of Voc (%/deg.C)	-0.27269
Cells per Module (Ncell)	96
Short-circuit current Isc (A)	6.14
Current at max. power point Imp (A)	5.76

Temp. coefficient of Isc (%/deg.C)	0.061694

To obtain 3KW from the PV system, we will need approximately 10 solar panels

$$no.\, of\ panels = \frac{3000}{315.072}$$

$$= 10\ panels\ (approx.)$$

The output from the panel is passed through a control circuit (optimization circuit), then an inverter and finally to the load.

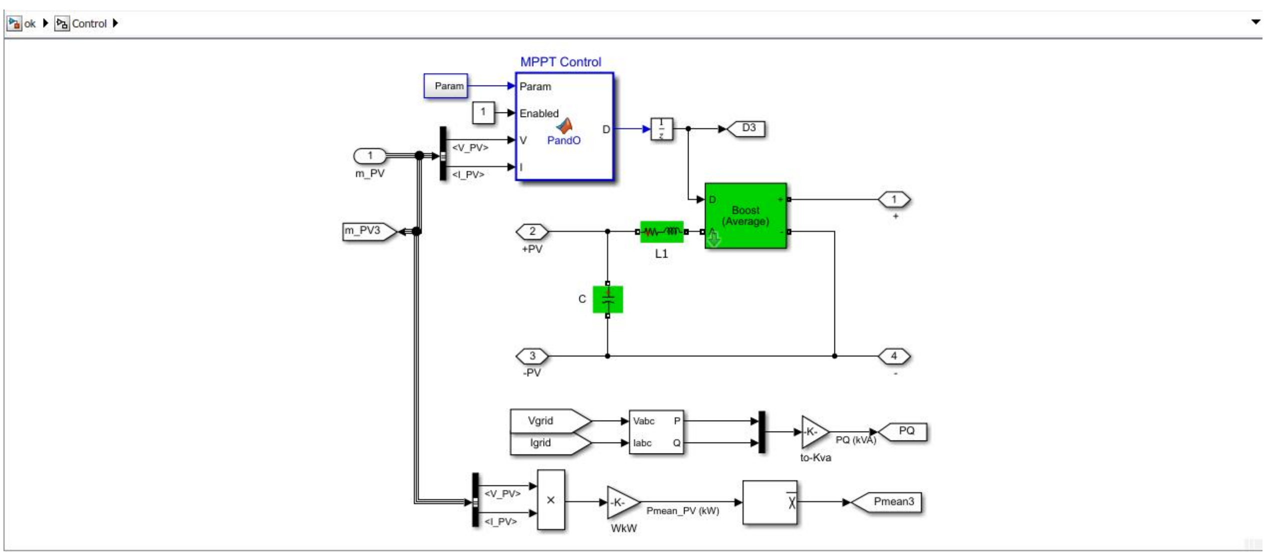


Figure 3.7: MPPT system

Figure 3.7 shows the MPPT system to be used in the simulation. The system is trained with P&O algorithm for PV optimization.

3.3 DEVELOPMENT OF ALGORITHMS FOR SOLAR OPTIMIZATION

3.3.1 DEVELOPMENT OF P&O ALGORITHM

As previous discussed, there are a number of algorithms used with MPPT controller. In this section we developed our own algorithm step by step using the perturb and observe approach (P&O). figure 3.1 shows the flowchart for the P&O algorithm.

Step 1: MATLAB System block was obtained from Simulink fundamental library. This block allows us to take measurement of the PV parameters and write the code for our optimization algorithm.

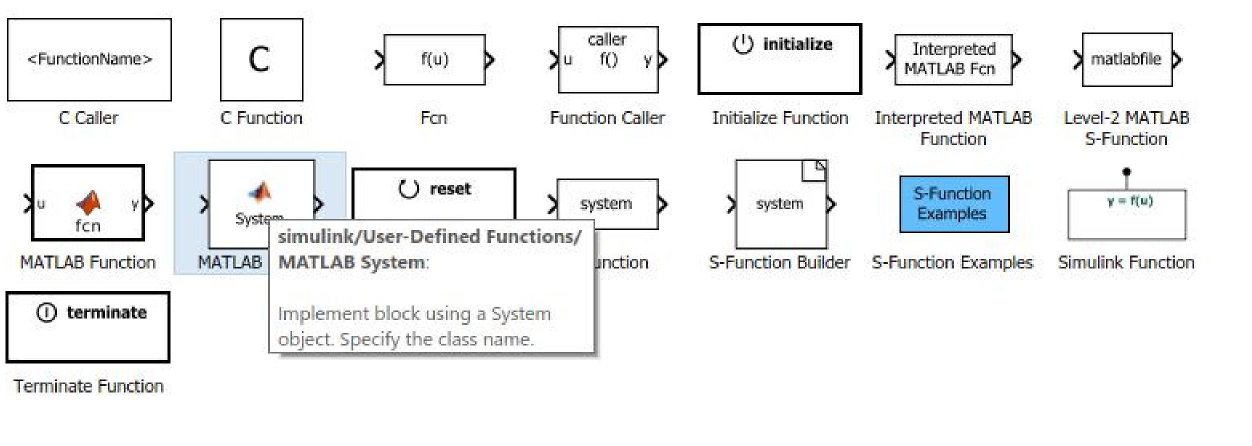


Figure 3.8: obtain MATLAB System block

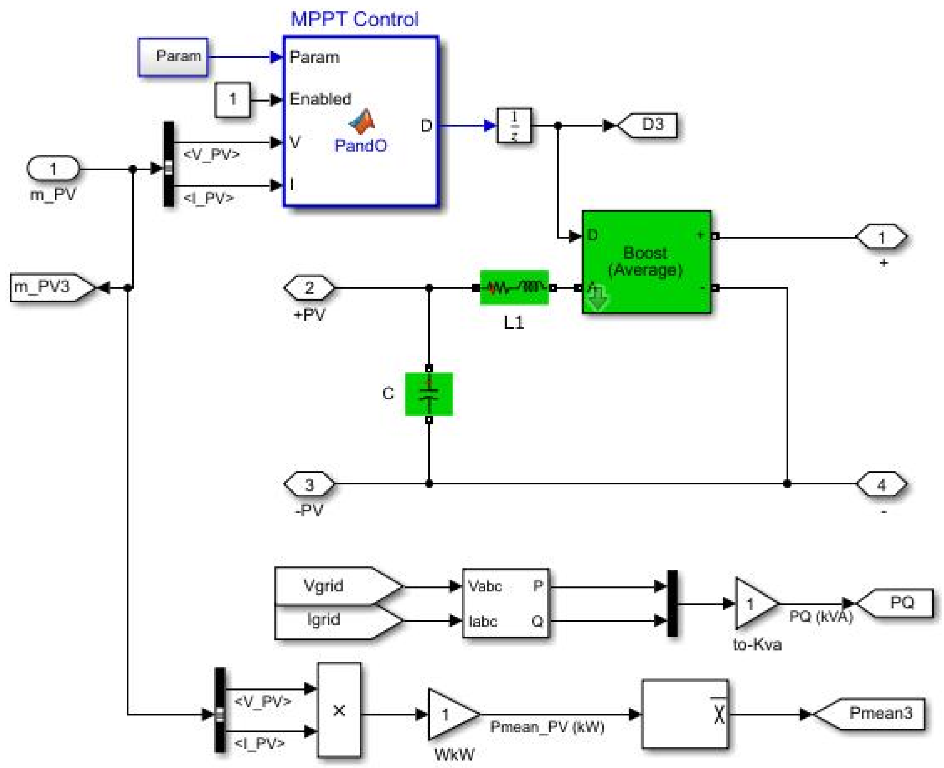


Figure 3.9: P&O System

Step 2: set constant parameters for the algorithm, the minimum duty cycle, maximum duty cycle, duty cycle steps

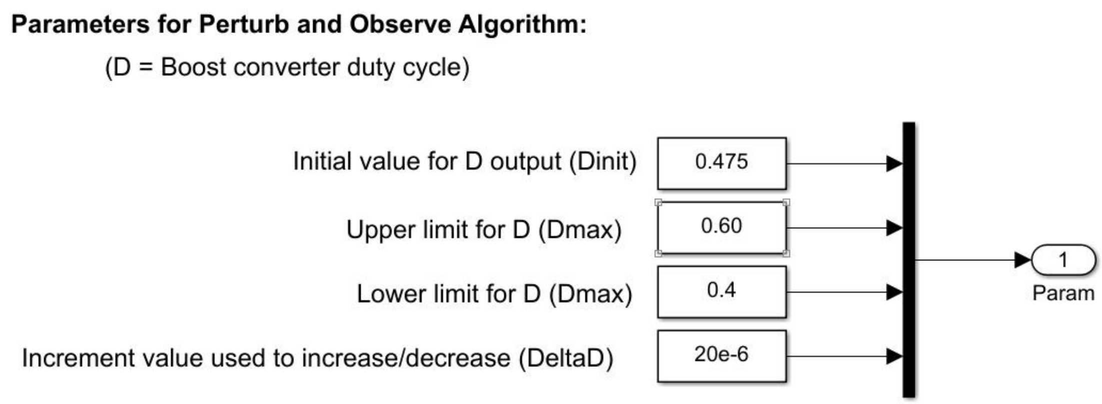


Figure 3.10: parameters for P&O algorithm

Step 3: obtained voltage and current from the PV module to calculate power. The measure power and voltage are compared to the previous reading from the PV module

```
if isempty(Vold)
    Vold=0;
    Pold=0;
    Dold=Dinit;
end
P= V*I;
dV= V - Vold;
dP= P - Pold;
```

Step 4: track maximum power of the PV module by checking to see if the newly measured voltage and power is less than the previously measured voltage and power.

```
if dP < 0
        if dV < 0
```

```
        D = Dold - deltaD;
    else
        D = Dold + deltaD;
    end
else
    if dV < 0
        D = Dold + deltaD;
    else
        D = Dold - deltaD;
    end
end
```

Refer to appendix A for the complete code of the algorithm.

3.3.2 DEVELOPMENT OF INC ALGORITHM

The INC algorithm is another popular algorithm used with MPPT controller. In this section we developed our own algorithm step by step using the incremental conductance algorithm. figure 3.2 shows the flowchart for the INC algorithm.

Step 1: obtain the function block as discussed in the previous section.

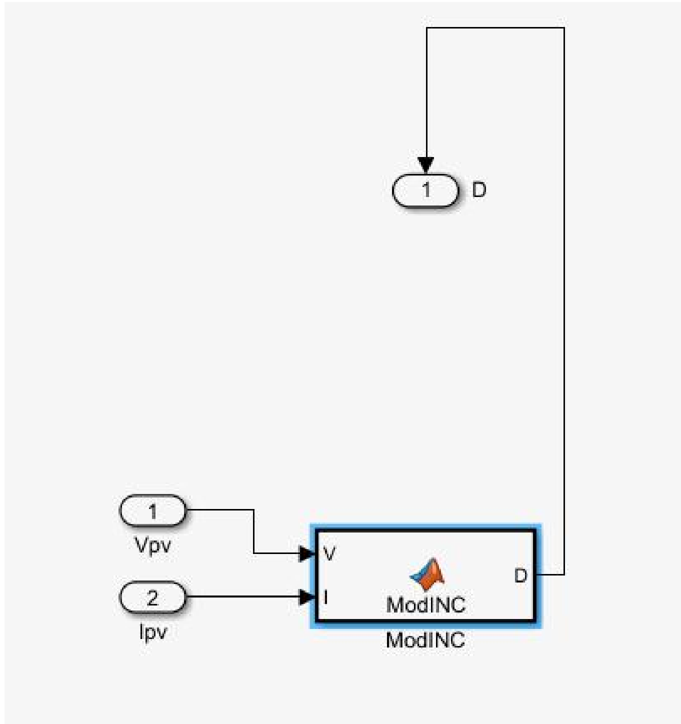


Figure 3.11: INC system

Step 2: set the limits of the algorithm and obtain the current and voltage from the PV system to be optimized

```
function D=ModINC(V, I)



Dinit = 0.6;  %Initial value for D output

Dmax = 0.65;   %Maximum value for D

Dmin = 0.1;   %Minimum value for D

deltaD = 0.002; %Increment value used to increase/decrease the duty cycle D
```

step 3: track maximum power by checking for changes in current and voltage

```
if M < 0.005
    D=Dold;
else
    if dV == 0
        if dI == 0
            D=Dold;
        elseif dI>0
            D=Dold - (M*deltaD);
        else
            D=Dold + (M*deltaD);
        end
    else
        if dI/dV == -I/V
            D=Dold;
        elseif dI/dV>-I/V
           D=Dold - (M*deltaD);
        else
           D=Dold + (M*deltaD);
        end
    end
end
```

step 4: adjust the duty cycle to maintain maximum power. The complete code can be found in the appendix A.

3.3.2 DEVELOPMENT OF A GENETIC ALGORITHM

The Genetic algorithm is another popular algorithm that is inspired by nature. In this section we developed our own algorithm step by step using the Genetic algorithm.

Step 1: obtain the function block as discussed in the previous two sections.

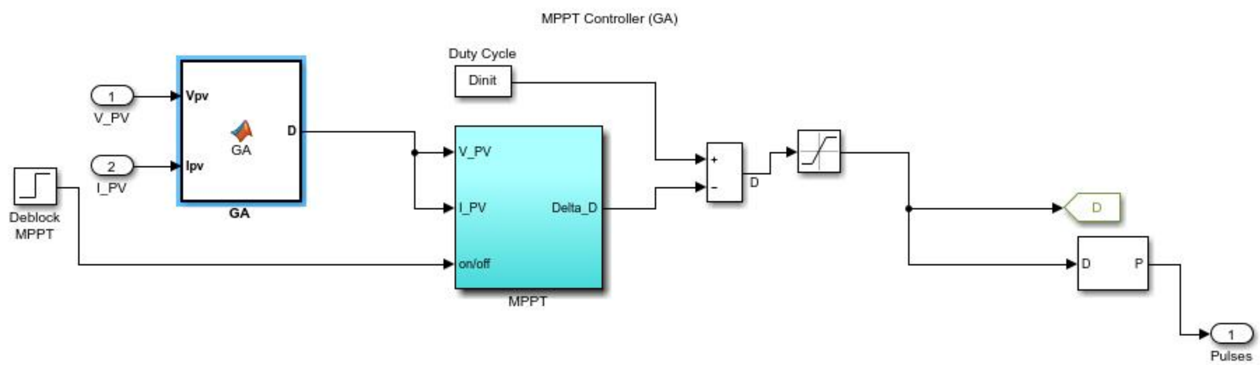


Figure 3.12: GA System

Step 2: obtain the current and voltage from the PV system to be optimized

```
function D = GA(Vpv,Ipv)

%#codegen

persistent u;

persistent dcurrent;

persistent pbest;

persistent p;
```

```
persistent dc;

persistent v;

persistent counter;

persistent gbest;

if(isempty(counter))

    counter=0;

function D = GA(Vpv,Ipv)

%#codegen

persistent u;

persistent dcurrent;

persistent pbest;

persistent p;

persistent dc;

persistent v;

persistent counter;

persistent gbest;

if(isempty(counter))

    counter=0;
```

step 3: check if the previously calculated change in counter, voltage, current and power is zero. Set to a default value if yes.

```
if(isempty(dcurrent))

    dcurrent=0.5;

end
```

```
if(isempty(gbest))
    gbest=0.5;
end
if(isempty(p))
    p=zeros(4,1);
end

if(isempty(v))
    v=zeros(4,1);
end

if(isempty(pbest))
    pbest=zeros(4,1);
end
if(isempty(u))
    u=0;
end
```

step 4: calculate the input power and compare with the previous power. Adjust output parameters accordingly.

The complete code can be found in the appendix A.

3.4 COMPARISON OF OPTIMIZED PV SYSTEM TO UNOPTIMIZED PV SYSTEM

The research implemented using two cases to assess resilience of the proposed system built in MATLAB under rapidly changing atmospheric circumstances. First, we considered the performance of the system at 25^{0}C and rapidly while varying the irradiation as a simple case. Second, we adjust he temperature and irradiation simultaneously to simulate real environmental conditions.

The performance of the unoptimized is compared to the P&O and GA optimized system under the stated conditions above. However, the time response, oscillation and stability are the three most important factors to consider when evaluating the effectiveness of any PV algorithm.

Results of these simulations are presented in section 4.3 of this study.

CHAPTER FOUR

RESULTS

4.0 ALGORITHM OF OPTIMIZATION METHODS

The code of the various optimization methods discussed can be found in Appendix A of this Report. The code was developed and debugged in MATLAB Simulink.

4.1 RESULTS OF OPTIMIZATION OF SOLAR PV SYSTEM USING MPPT

MPPT method of solar PV optimization was used in this project for PV optimization, figure 4.1a and figure 4.1b shows the result of MPPT method in solar optimization.

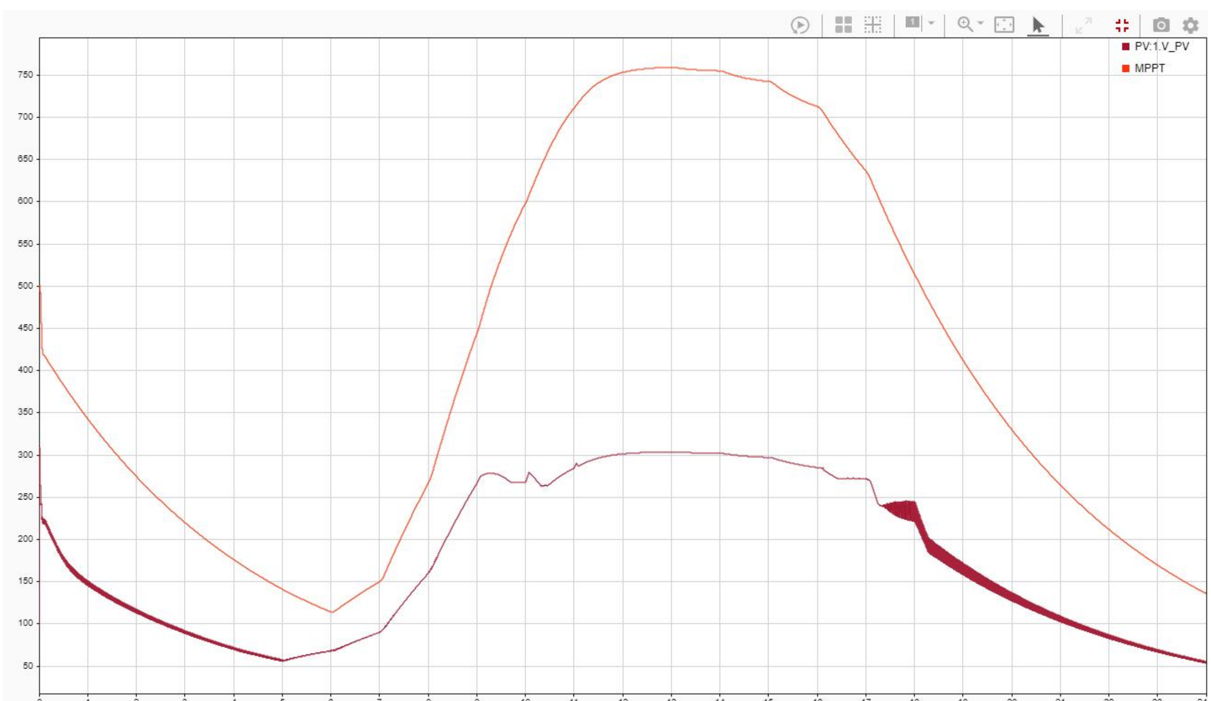


Figure 4.1a: MPPT Voltage

Figure 4.1a shows the MPPT voltage to the PV voltage from the solar cell. From the figure, it can be seen that MPPT optimization method removes disturbance seen on the PV output voltage.

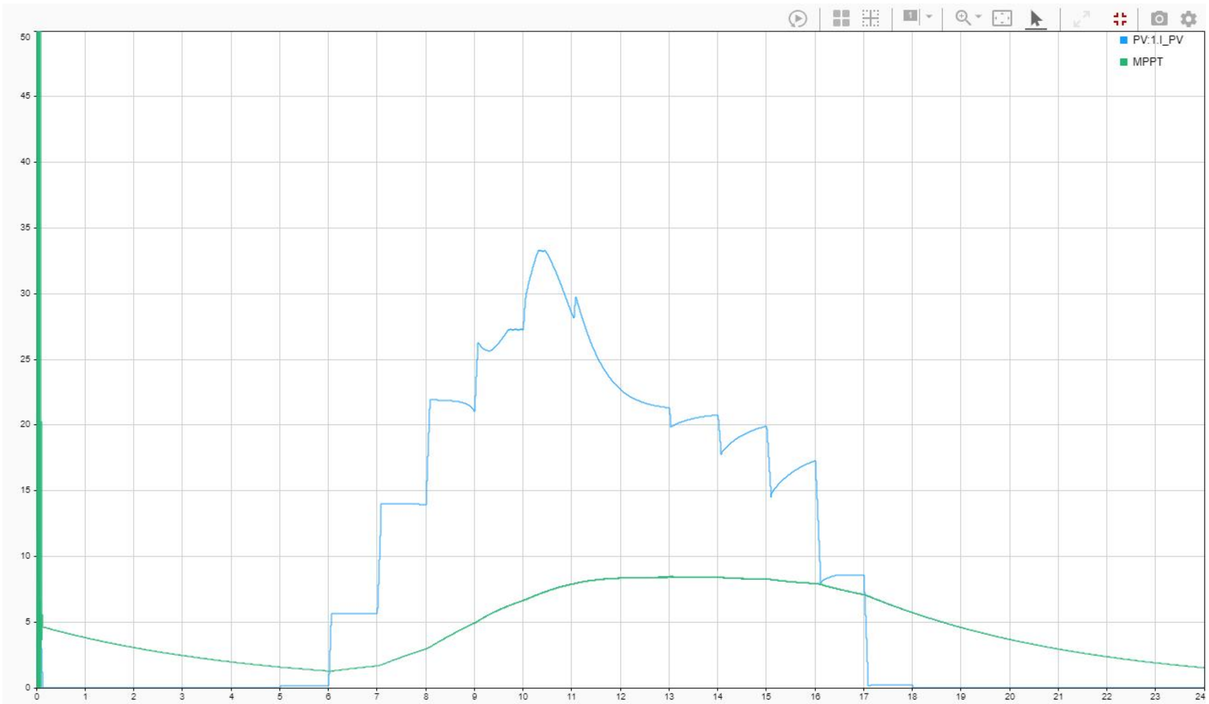


Figure 4.1b: MPPT current

4.2 RESULTS OF DEVELOPED ALGORITHMS FOR SOLAR OPTIMIZATION

4.2.1 RESULTS OF DEVELOPED P&O ALGORITHM FOR SOLAR OPTIMIZATION

Perturb and Observed algorithm was developed in this study, the complete code of the algorithm can be found in Appendix A. This algorithm was implemented into the MPPT method used.

4.2.2 RESULTS OF DEVELOPED GA ALGORITHM FOR SOLAR OPTIMIZATION

Genetic algorithm was developed in this study, the complete code of the algorithm can be found in Appendix A. This algorithm was implemented into the MPPT method used. Figure 4.2a and figure 4.2b shows the performance of the system with GA optimization.

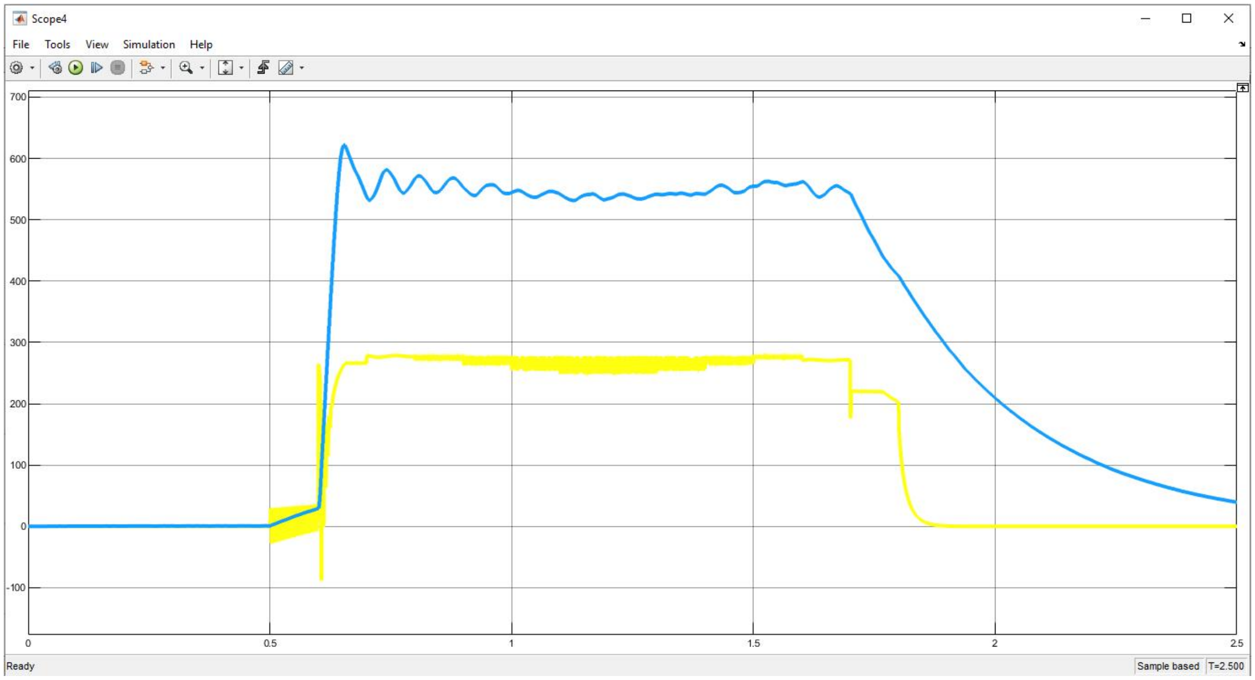


Figure 4.2a: GA optimized voltage

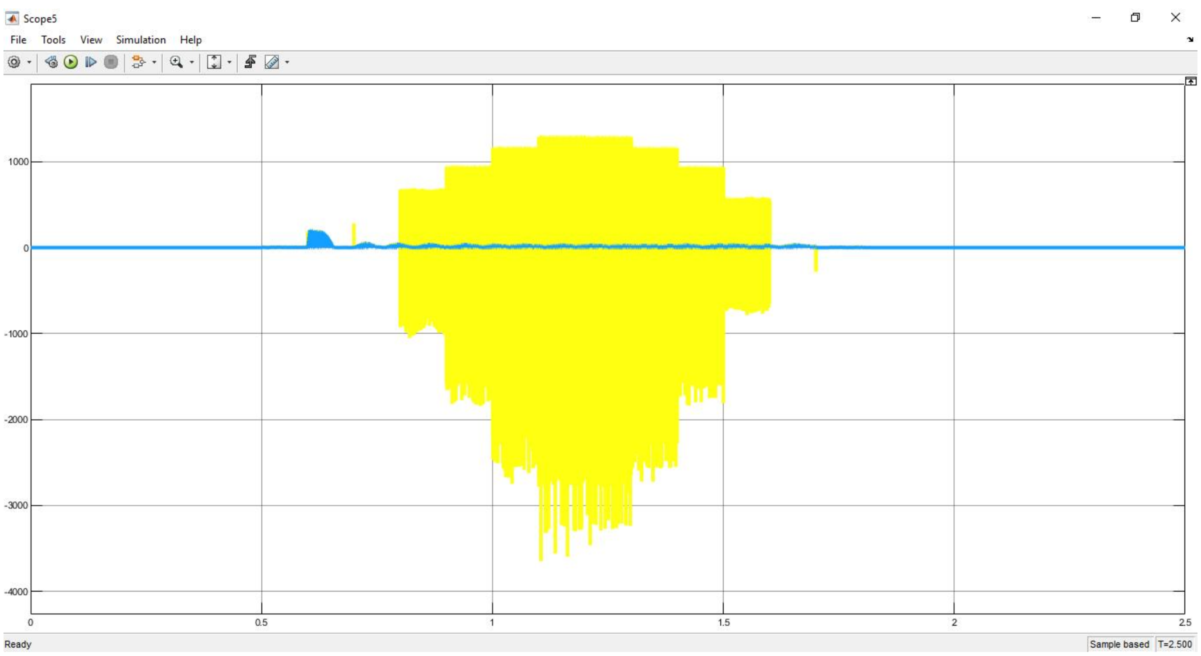


Figure 4.2b: GA optimized current

4.3 RESULTS OF COMPARISON BETWEEN OPTIMIZED PV SYSTEM TO UNOPTIMIZED PV SYSTEM

The MPPT optimized solar model developed in MATLAB Simulink is compared with an unoptimized system that uses a fixed duty cycle. Simulations was run for two cases, firstly, using a simple case where the irradiation was varied between 750W/m^2 and 1000W/m^2. The results obtained are in section 4.3.1. Secondly, simulation was performed using real life solar isolation data for edo state, the data was obtained from Global Solar Atlas official site. The results of this simulation is presented in section 4.3.2.

4.3.1 RESULTS FROM SIMPLE CASE

Initially considering only two irradiation conditions (1000W/m^2 and 750W/m^2) as simple cases and keeping the temperature constant at 25^{0}C, the irradiation was allowed to drop rapidly to 750W/m^2 at 0.5 seconds to simulate the transitory condition to determine how both the optimized and unoptimized respond to transient state.

Figure 4.3a shows result for the P&O optimized system, this result shows that P&O optimized system has a stable power tracking, the output power is above 2KW compared to the output power of the unoptimized system which has not been kept stable at a specific value and during the same condition has less value (<2KW) as shown in figure 4.3b. figure 4.3c shows the zoomed graph of figure 4.3b.

Figure 4.3d. shows the result for the GA optimized system, this result shows that the GA algorithm is superior to both P&O and the unoptimized system.

Figure 4.3a: Output Power P&O optimized system

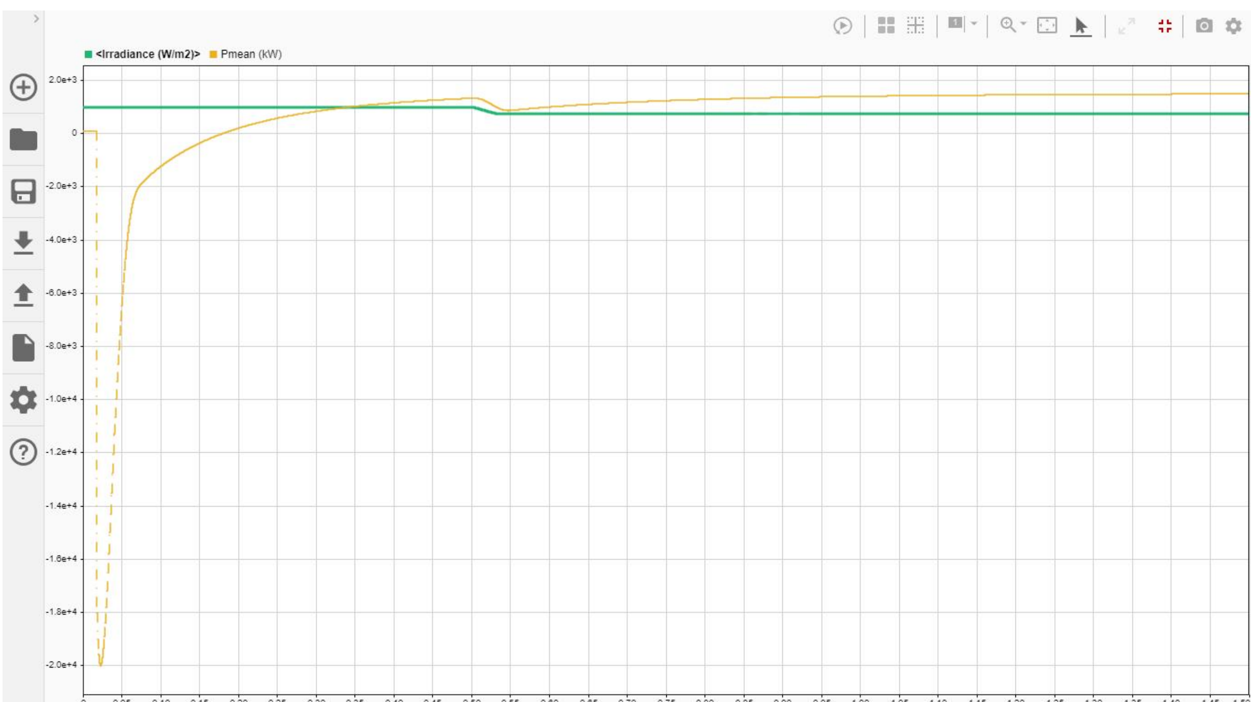


Figure 4.3b: Output Power unoptimized system

Figure 4.3c: Output Power unoptimized system (zoomed)

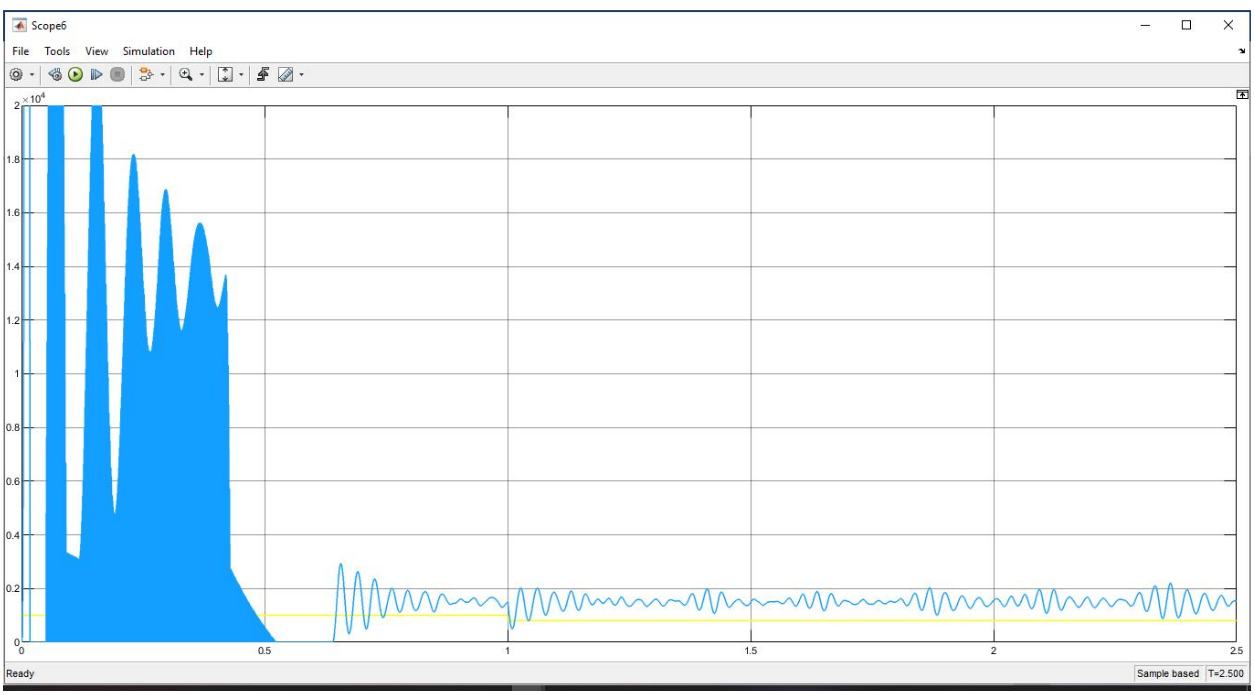


Figure 4.3d: Output power of GA optimization

When the radiation changed from 1000W/m^2 to 750W/m^2, it is observed that the time response for the P&O algorithm trained system was less than that of the unoptimized system.

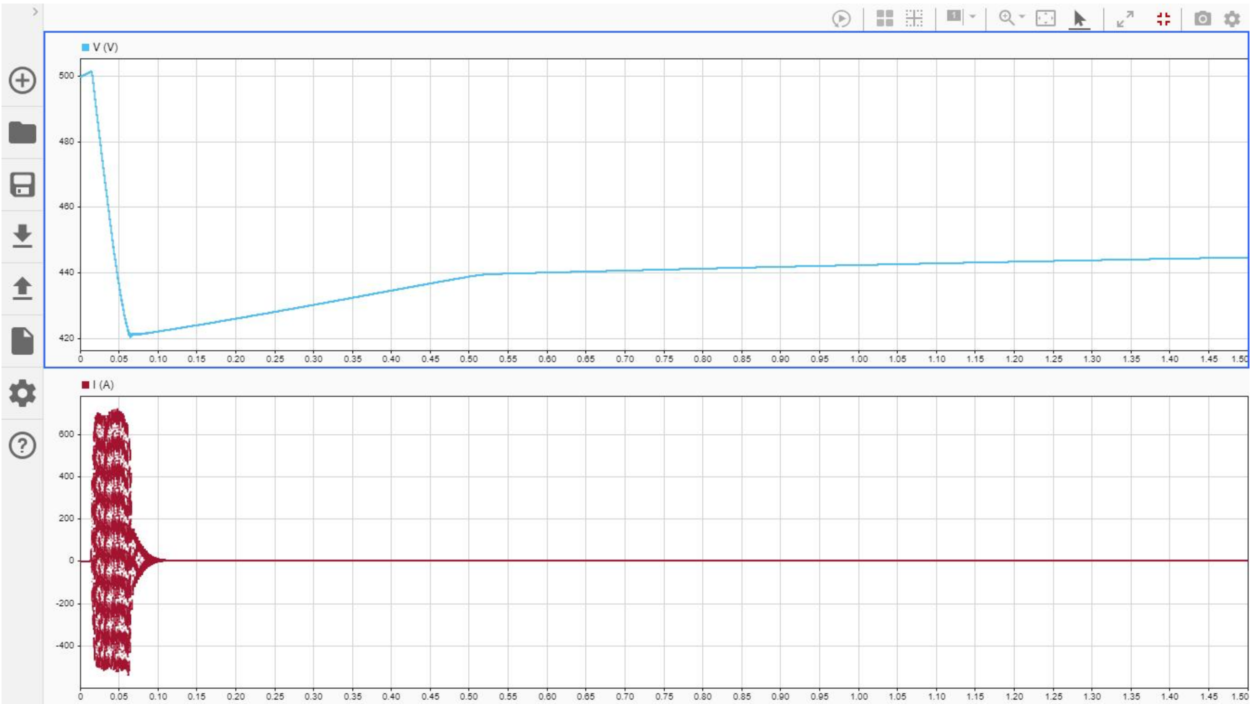


Figure 4.4a: Voltage and Current of P&O system

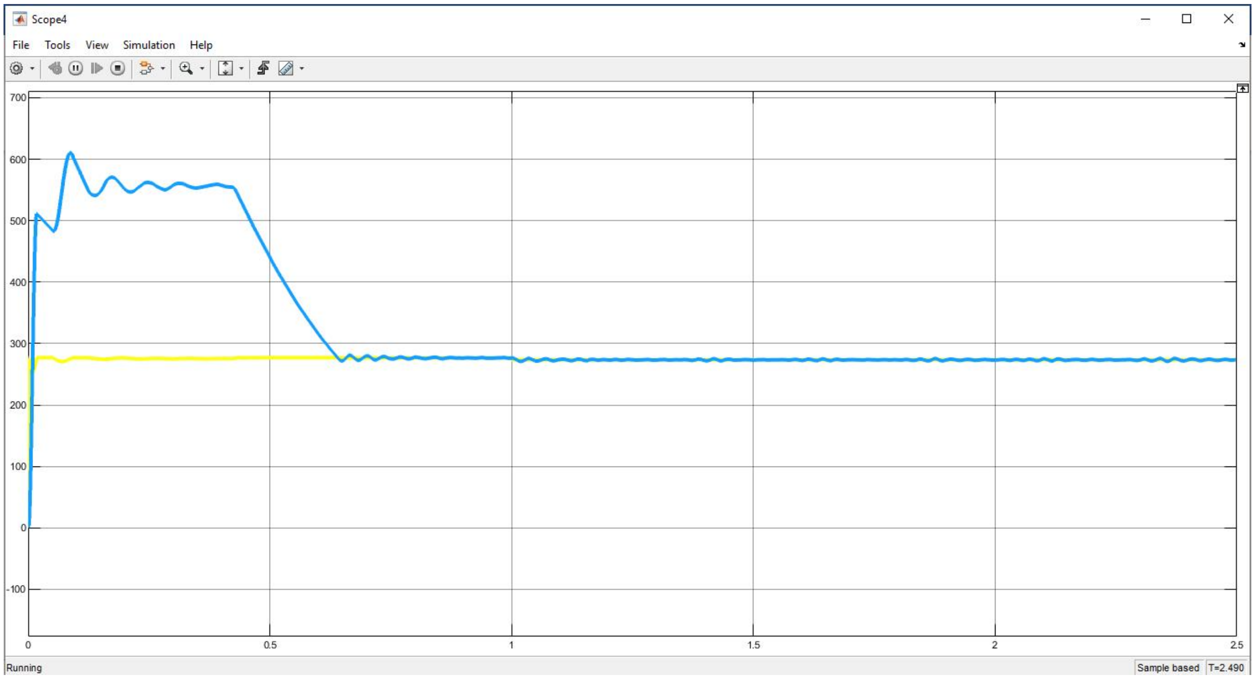


Figure 4.4b: Voltage and current of optimized system

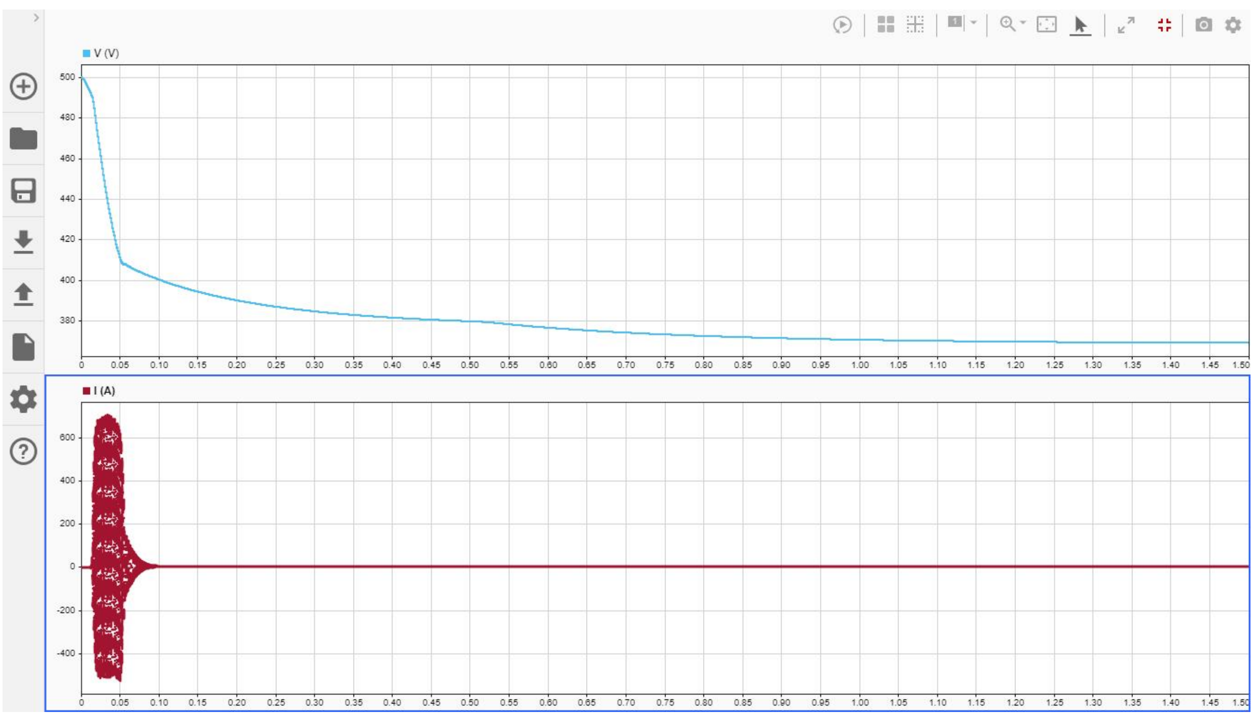


Figure 4.4c: Voltage and Current of unoptimized system

The voltage and current curves for the same case, in figure 4.4, it is observed that the voltage curve for P&O has a stable performance and recorded a constant value and current. On the other hand, the unoptimized system recorded fluctuating voltage. The current fluctuation at the beginning for P&O compared to the unoptimized system is less. Figure 4.5 shows the output voltage, current and power from the inverter.

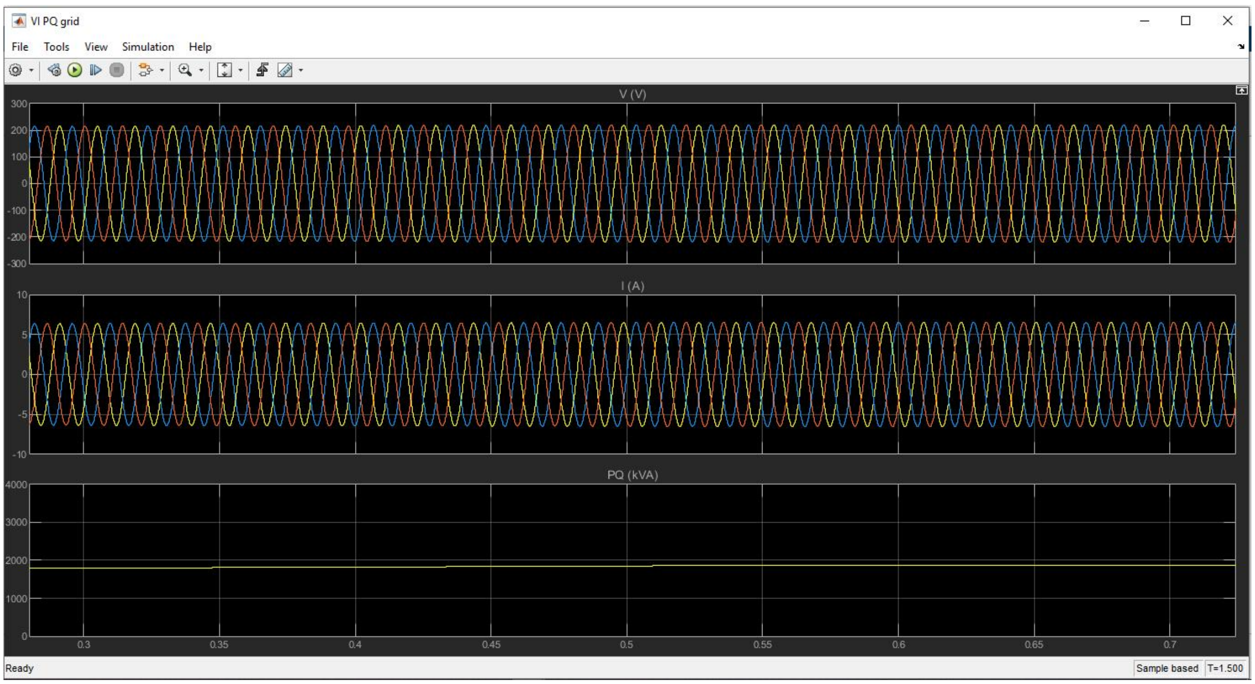


Figure 4.5: Output from inverter

4.3.2 RESULTS FROM REAL LIFE PARAMETERS

Solar data obtained for Edo state, nigeria from Global Solar Atlas official site is presented in figure 4.6 and 4.7. simulation was performed using non uniform irradiance and temperature patterns on the solar array. Radiation started rising from zero to a specific value and then back to zero. This simulation was performed for April and November using 24 seconds to map 24 hours in a day.

Figure 4.6: Average Solar irradiance for Edo State Nigeria (chart)

Average hourly profiles

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0 - 1												
1 - 2												
2 - 3												
3 - 4												
4 - 5												
5 - 6				1	3	3	2	2	8	7	2	
6 - 7	8	6	12	46	86	80	65	65	113	115	74	34
7 - 8	80	65	76	115	164	142	106	101	163	182	155	145
8 - 9	162	133	138	181	224	191	135	122	187	228	252	240
9 - 10	236	201	193	238	282	234	162	136	200	266	355	333
10 - 11	295	258	247	291	346	272	207	173	239	321	425	404
11 - 12	323	296	281	322	347	307	229	219	304	361	443	426
12 - 13	324	295	291	321	355	322	247	215	322	346	408	411
13 - 14	290	266	246	291	325	292	226	203	271	292	341	357
14 - 15	228	211	191	237	281	252	186	176	225	229	265	283
15 - 16	137	130	121	164	219	199	151	148	184	185	182	184
16 - 17	45	50	46	74	124	131	110	110	126	102	65	60
17 - 18		3	3	2	14	23	23	18	9	1		
18 - 19												
19 - 20												
20 - 21												
21 - 22												
22 - 23												
23 - 24												
Sum	2127	1914	1846	2284	2769	2448	1848	1689	2351	2635	2968	2876

Source: globalsolaratlas.info

Figure 4.7: Average Solar irradiance for Edo State Nigeria (Table)

Figure 4.8 and 4.9 shows the output power of the system after simulation using the irradiance information for April and November.

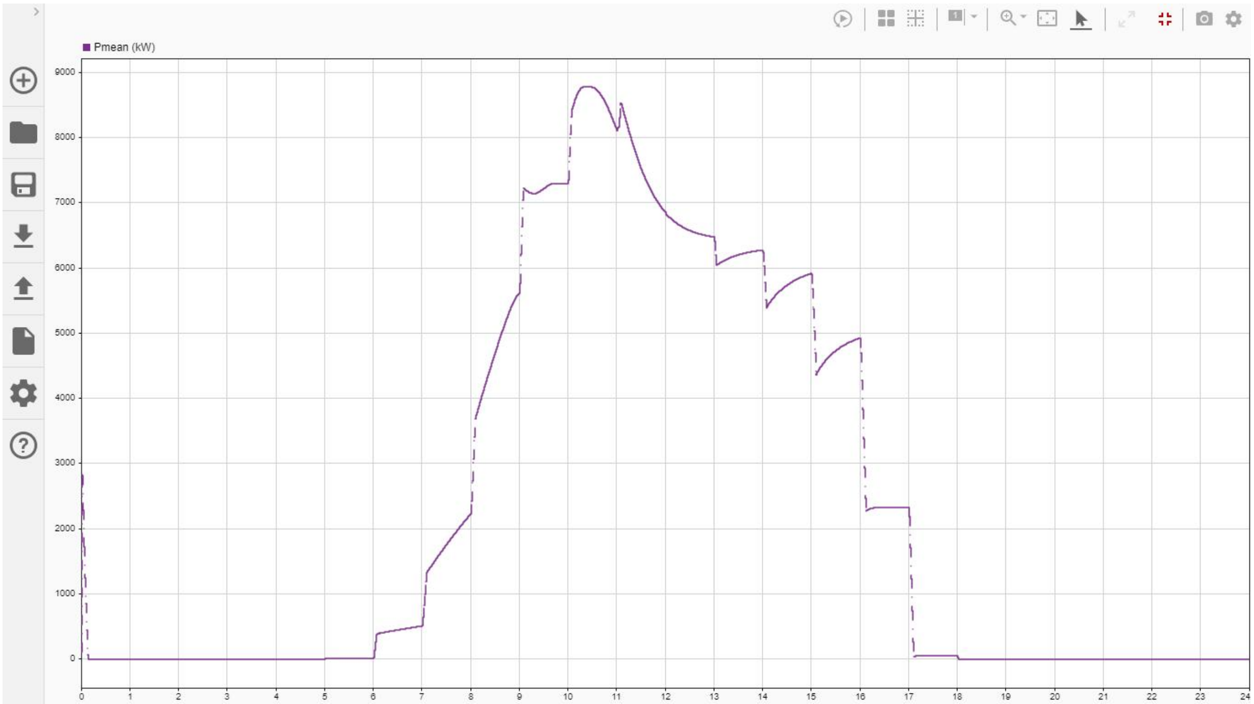


Figure 4.8: Simulated Output power for the month of April

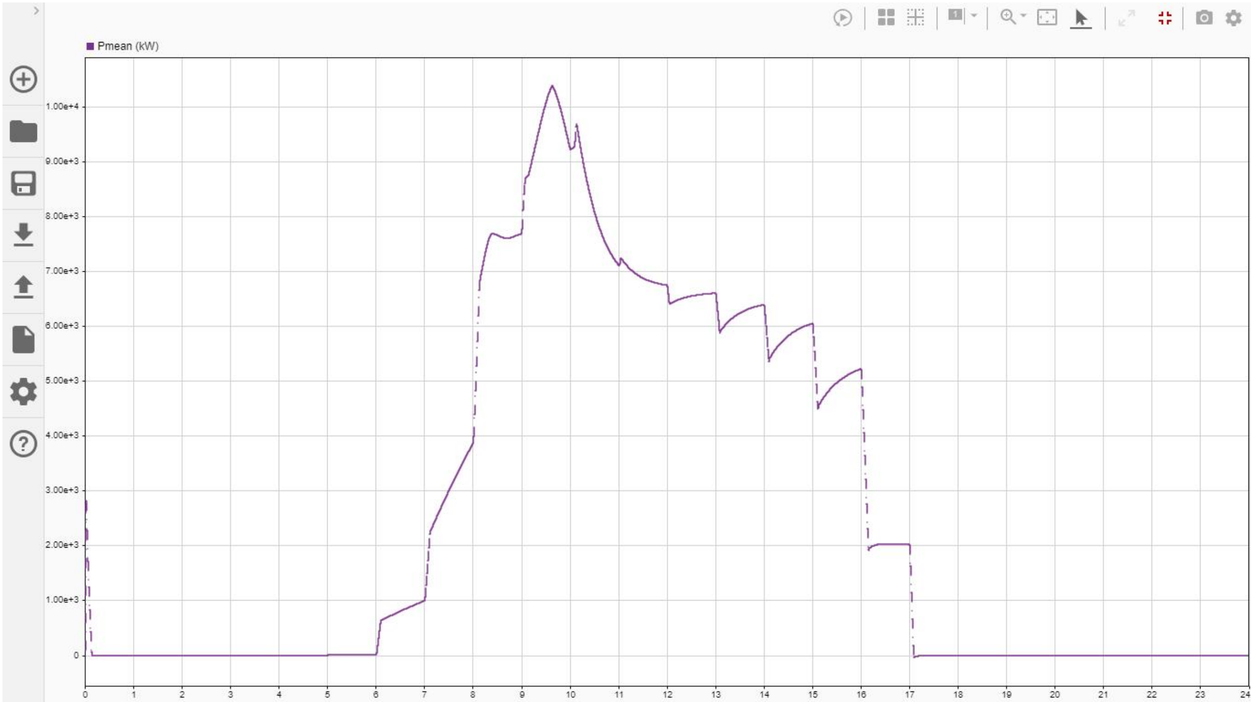


Figure 4.9: Simulated Output power for the month of November

With increasing solar irradiation and duration, the acquired power rises and decreases with decreasing power. Figure 4.10a, 4.10b and 4.10c shows the PV characteristics with P&O, GA and unoptimized system for the month of April.

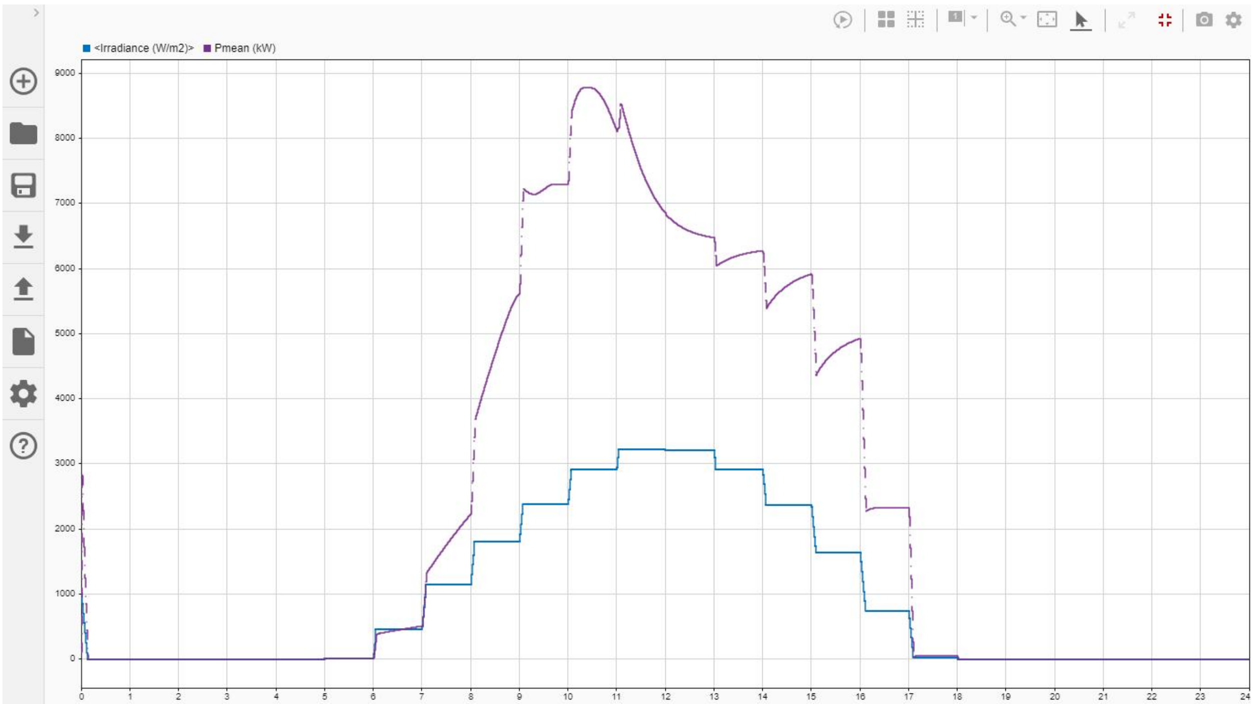


Figure 4.10a: Output for P&O optimization algorithm

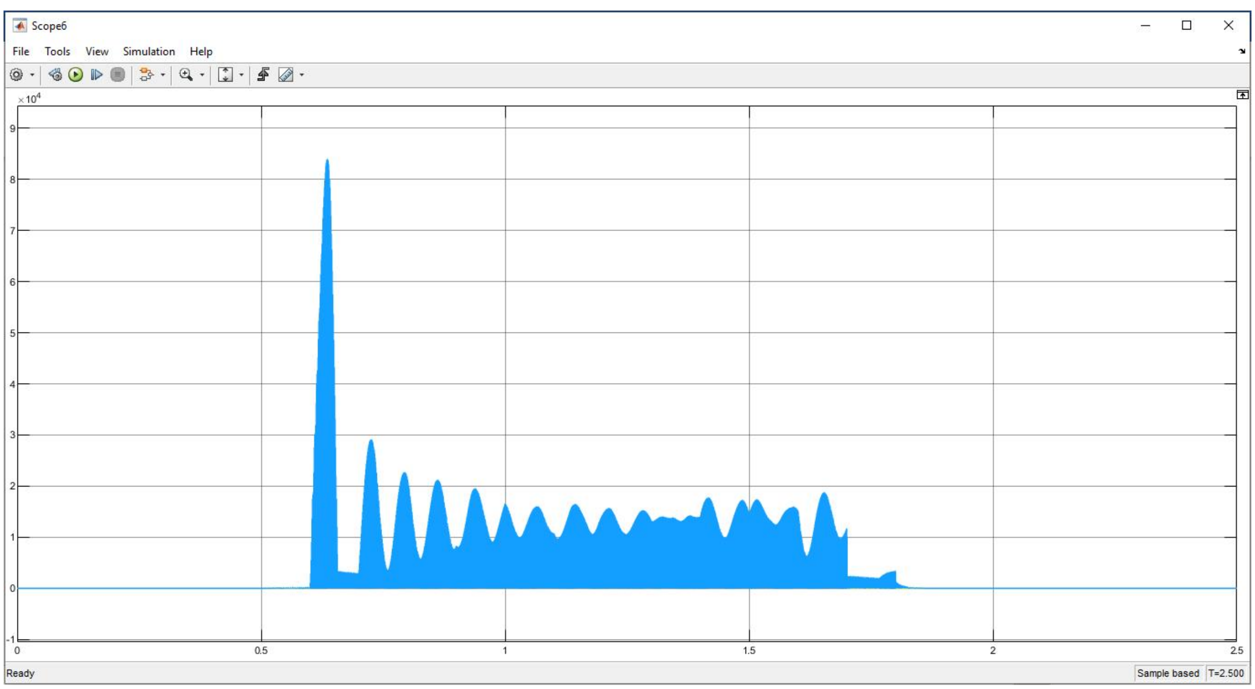


Figure 4.10b: output of GA optimized system

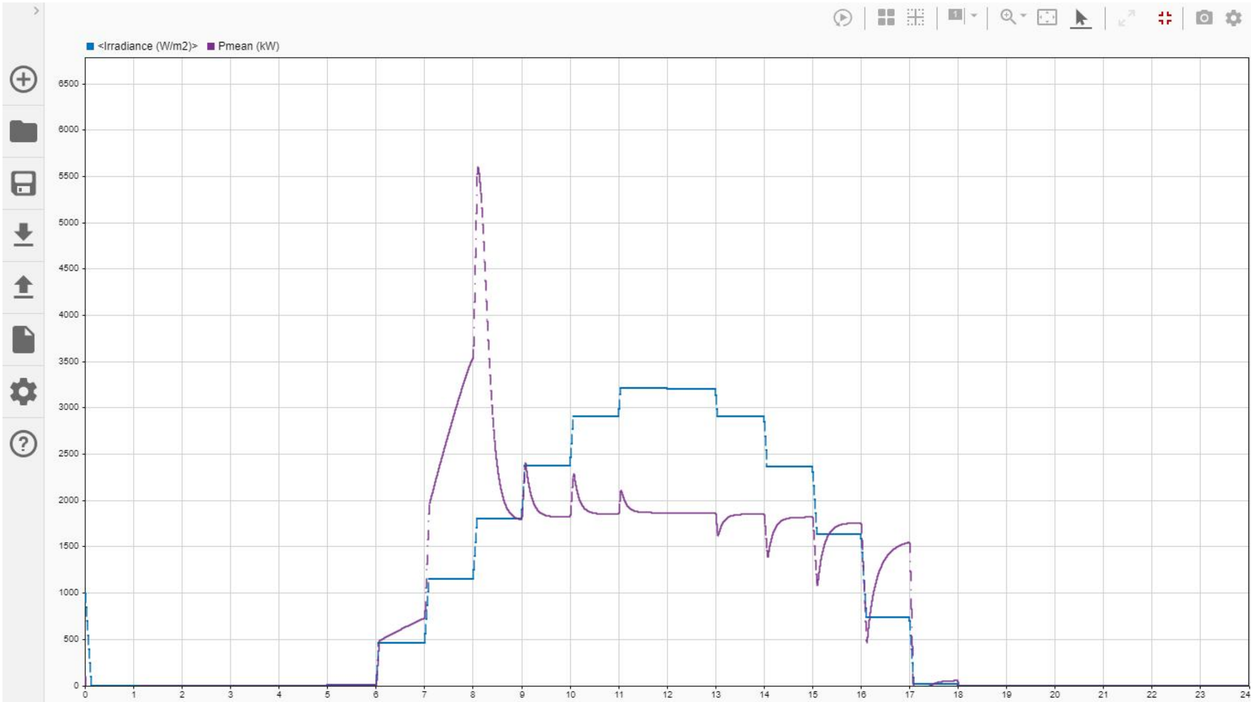


Figure 4.10c: Output for unoptimized system

Figure 4.11a, 4.11b and 4.11c shows the response of both system to the irradiation values for the month of November.

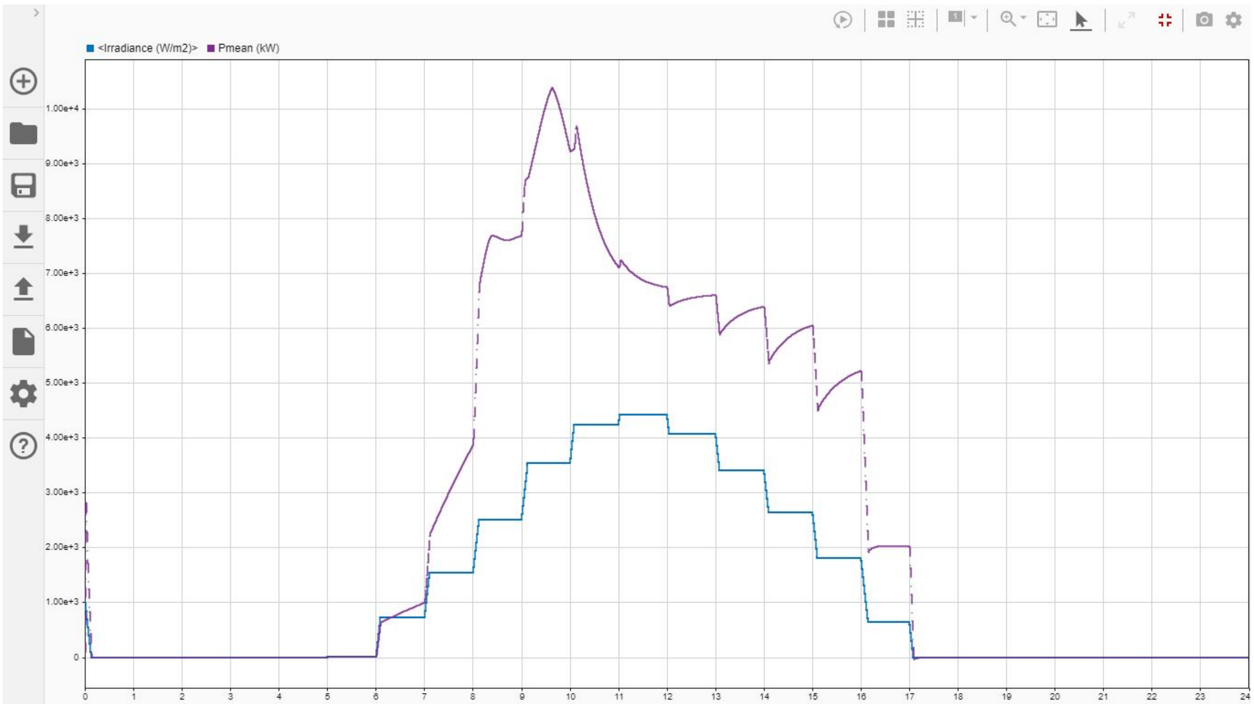


Figure 4.11a: Output for P&O optimization algorithm

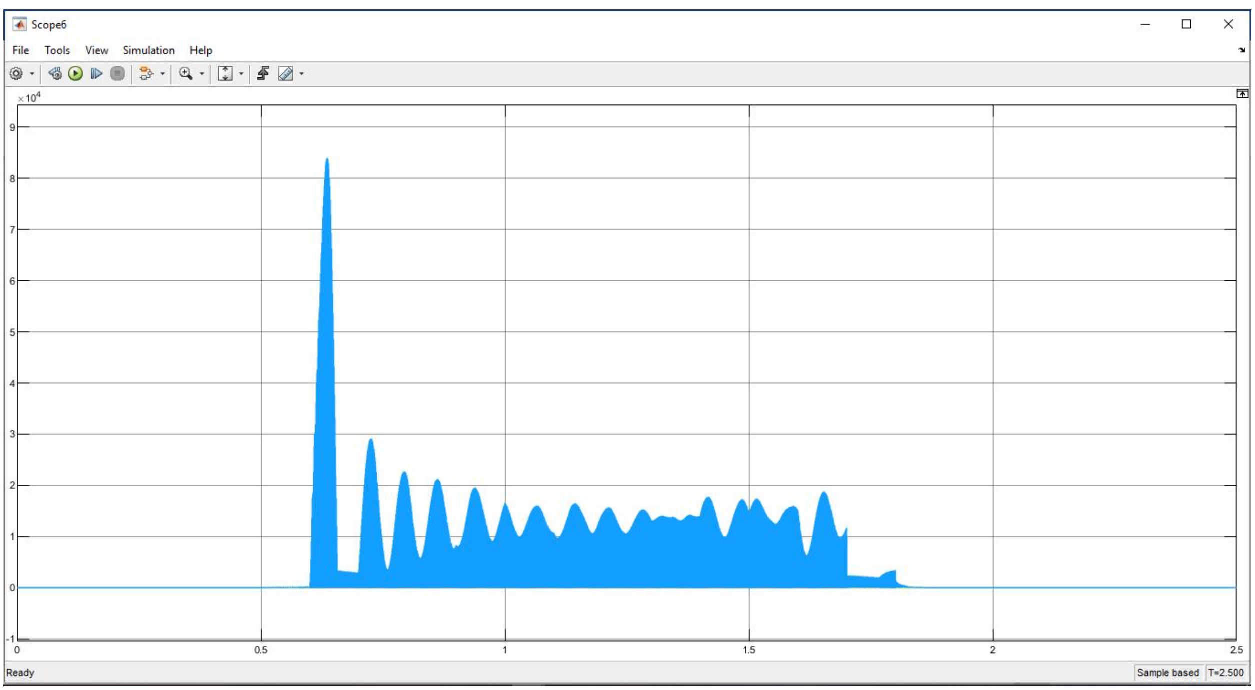


Figure 4.11b: Output of GA optimized algorithm

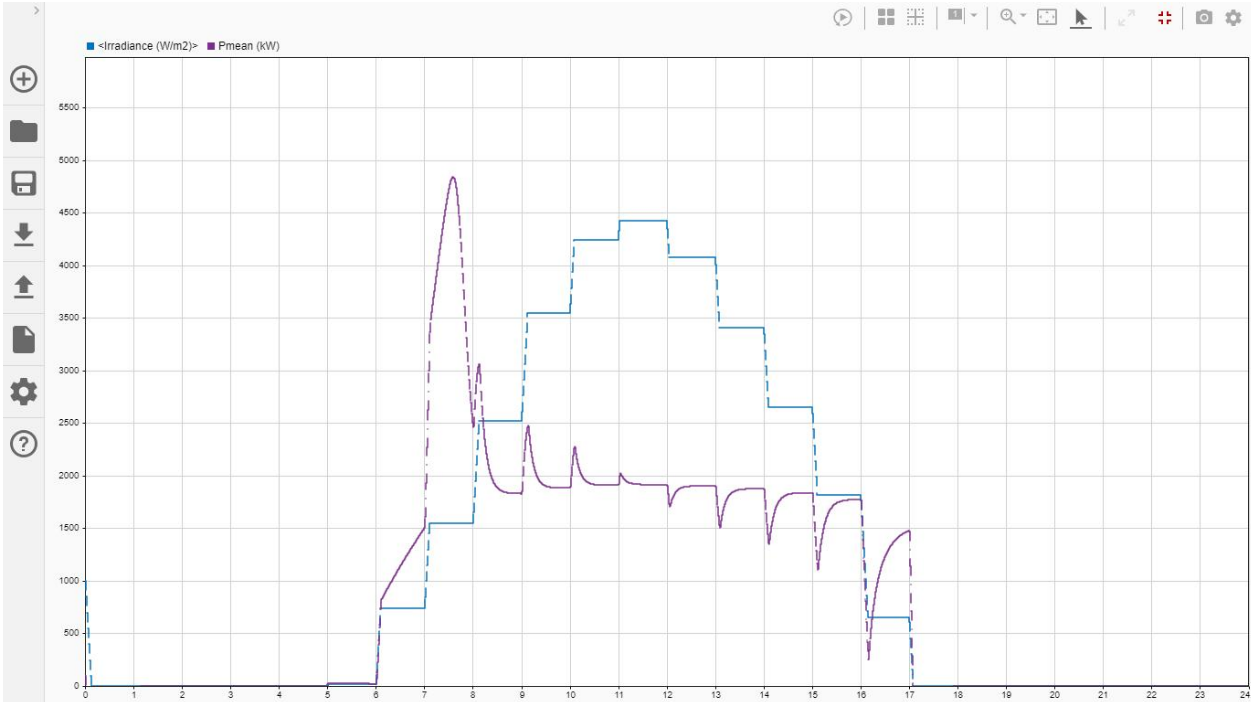


Figure 4.11c: Output for unoptimized system

4.3 ANALYSIS OF RESULT

From the simulation result, we can see that optimization help to improve the efficiency of the PV system, it does this by sensing the voltage and current variable (P&O algorithm) and alter the system output in response. Optimization improves tracking speed and tracking accuracy. Table 4.1 summarizes the performance of the optimized system in comparison to an unoptimized system.

Table 4.1 Analysis of the performance of proposed methods

Optimization Technique	Sensed variable	Steady state error	Tracking Speed	Tracking accuracy	Efficiency	Complexity
P&O	V, I	Less	Faster	Stable	High	High
GA	V, I	Less	Very Fast	Stable	Very High	Very High

Unoptimized	None	Moderate	Slow	Less stable	Moderate	Low

CHAPTER FIVE

CONCLUSION AND RECOMMENDTION

5.1 CONCLUSION

In this study, the various optimization methods and algorithm was discussed, an optimization algorithm was then developed and the effectiveness and performance of an optimized PV system based on the algorithm was examined. The simulation was implemented using two cases to assess resilience of the proposed system built in MATLAB under rapidly changing atmospheric circumstances. The output of the P&O trained by PSO algorithm was compared to the output of an un-optimized system. The P&O algorithm suggest an improved efficiency of the system and resistance to atmospheric irradiation changes

The P&O system performed better than the un-optimized system, having a faster time response, tracking speed and less oscillation of the output power. Although the un-optimized system was able to produce almost constant power, the P&O optimized system was more effective with an excellent performance.

5.2 CONTRIBUTION TO KNOWLEDGE

The study contributes to knowledge in the following ways

- Provision PV optimization strategies for solar power system for better performance and efficiency.
- Development of advance optimization algorithms for solar power systems
- Provision of evidence for the need of solar optimization

5.3 RECOMMENDATION

It is advisable to perform comprehensive testing and enhance the functionality of control parts before initialing test commissioning in photovoltaic plants. This proactive approach aims to improve accuracy and efficiency within the plant operations. Through thorough assessment of the control parts, any potential issues can be identified and ensure the plant operates at its best. This is vital for maintain performance and durability.

REFERENCES

1. Vishal Dabra, Krishan Kumar Paliwal, Narender Kumar (2017). Optimization of photovoltaic power system: a comparative study.
2. El-Arini MM, Othman AM, Fathy A (2013). A new optimization approach for maximizing the photovoltaic panel power based on genetic algorithm and language multiplier algorithm. International journal of photoenergy 2103:1-12
3. Ahhab, M.S. Solar tracking by neutral network adaptive technique. The second Annual Int. Conf. on Advance Topics in Artificial Intelligence, Singapore, 2011, vol 1, pp. 63-67.
4. Stamatescu et al. (2014). Design and implementation of a solar tracking algorithm. Procedia Eng., 69(2014), pp. 500-507.
5. J. Prakash, Sarat Kumar Sahoo (2013). Modeling of photovoltaic system and design of MPPT controller using PSO algorithm.
6. El-Arini MM, Othman AM, Fathy A (2013). A new optimization approach for maximizing the photovoltaic panel power based on genetic algorithm and language.
7. Adbalftah Hamed Ali and Atabak Najafi (2022). Optimization and Performance Improvement of Grid-connected PV plant Based on ANN-PSO and P&O Algorithm.
8. Liu, C, Wu, B, & Cheung, R. (2004). Advanced algorithm for MPPT control of photovoltaic systems. Montreal: Canadian Solar Buildings Conference.
9. Google Scholar Majhi, R, Panda, G, Majhi, B, & Sahoo, G. (2009). Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques.

10. Expert Systems with Applications, 36, 10097–10104. doi:10.1016/j.eswa.2009.01.012. Article Google Scholar Kamejima, T, Phimmasone, V, Kondo, Y, & Miyatake, M. (2011).
11. The optimization of control parametersof PSObased MPPT for photovoltaics. IEEE. doi:10.1109/PEDS.2011.6147358. Google Scholar Zhang, H, & Cheng, S. (2011).
12. A New MPPT algorithm based on ANN in solar PV systems. Advances in Computer, Communication, Control and Automation, 121, 77–84. doi:10.1007/978-3-642-25541-0_11.
13. Article Google Scholar Nabulsi, AA, & Dhaouadi, R. (2012). Efficiency optimization of a DSP-based standalone PV system using fuzzy logic and dual-MPPT control. IEEE, 8(3), 573–584. doi:10.1109/TII.2012.2192282. Google Scholar Shaw, B, Banerjee, A, Ghoshal, SP, & Mukherjee, V. (2011).
14. Comparative seeker and bio-inspired fuzzy logic controllers for power system stabilizers. International Journal of Electrical Power & Energy Systems, 33(10), 1728–1738. doi:10.1016/j.ijepes.2011.08.015.
15. Google Scholar Sundareswaran, K, Kumar, VV, & Palani, S. (2015). Application of a combined particle swarm optimization and perturb and observe method for MPPT in PV systems under partial shading conditions.
16. Renewable Energy, 75, 308–317. doi:10.1016/j.renene.2014.09.044. Article Google Scholar Qin, L, & Lu, X. (2012). Matlab/simulink-based research on maximum power point tracking of photovoltaic generation.
17. Physics Procedia, 24(A), 10–18. doi:10.1016/j.phpro.2012.02.003. Article Google Scholar Algazar, MM, AL-Monier, H, EL-Halim, H, & Salem, MEEK. (2012).

18. Maximum power point tracking using fuzzy logic control. International Journal of Electrical Power & Energy System, 39(1), 21–28. doi:10.1016/j.ijepes.2011.12.006.

19. Article Google Scholar El-Khozondar, HJ, El-Khozondar, RJ, Matter, K, & Suntio, T (2016).

APPENDIX A

Code for P&O algorithm

```
function D = PandO(Param, Enabled, V, I)


% MPPT controller based on the Perturb & Observe algorithm.


% D output = Duty cycle of the boost converter (value between 0 and 1)

% Param input:

Dinit = Param(1);  %Initial value for D output

Dmax = Param(2);   %Maximum value for D

Dmin = Param(3);   %Minimum value for D

deltaD = Param(4); %Increment value used to increase/decrease the duty cycle D

% ( increasing D = decreasing Vref )

%


persistent Vold Pold Dold;


dataType = 'double';


if isempty(Vold)

    Vold=0;

    Pold=0;

    Dold=Dinit;
```

```
end
P= V*I;
dV= V - Vold;
dP= P - Pold;

if dP ~= 0 & Enabled ~=0
   if dP < 0
      if dV < 0
         D = Dold - deltaD;
      else
         D = Dold + deltaD;
      end
   else
      if dV < 0
         D = Dold + deltaD;
      else
         D = Dold - deltaD;
      end
   end
else D=Dold;
end

if D >= Dmax | D<= Dmin
```

```
    D=Dold;
end

Dold=D;
Vold=V;
Pold=P;
```

Code for INC Algorithm

```
function D=ModINC(V, I)


Dinit = 0.6;  %Initial value for D output
Dmax = 0.65;   %Maximum value for D
Dmin = 0.1;   %Minimum value for D
deltaD = 0.002; %Increment value used to increase/decrease the duty cycle D


persistent Vold Pold Dold M Iold;

dataType = 'double';

if isempty(Vold)
```

```
    Vold=0;
    Pold=0;
    Iold=0;
    Dold=Dinit;
    M=1;
end
P= V*I;
dV= V - Vold;
dP= P - Pold;
dI= I - Iold;
M=abs(dP);

if M < 0.005
    D=Dold;
else
    if dV == 0
        if dI == 0
            D=Dold;
        elseif dI>0
            D=Dold - (M*deltaD);
        else
            D=Dold + (M*deltaD);
        end
```

```
    else
        if dI/dV == -I/V
            D=Dold;
        elseif dI/dV>-I/V
            D=Dold - (M*deltaD);
        else
            D=Dold + (M*deltaD);
        end
    end
end

if D >= Dmax | D<= Dmin
    D=Dold;
end

Dold=D;
Vold=V;
Pold=P;
Iold=I;
```

Code For ANN Algorithm

```
f=xlsread('DATA(cylindrical)')';

inputs=f(1:3,:);

targets=f(4:5,:);


%% Create a Fitting Network

hiddenLayerSize = 1;

TF={'tansig','purelin'};% Activition function

net = newff(inputs,targets,hiddenLayerSize,TF);


%% Choose Input and Output Pre/Post-Processing Functions

% For a list of all processing functions type: help nnprocess

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};

net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};



%% Setup Division of Data for Training, Validation, Testing For a list of all data division
functions type: help nndivide

net.divideFcn = 'dividerand';  % Divide data randomly

net.divideMode = 'sample';  % Divide up every sample

net.divideParam.trainRatio = 50/100;

net.divideParam.valRatio = 25/100;

net.divideParam.testRatio = 25/100;
```

```
%% For help on training function 'trainlm' type: help trainlm For a list of all training functions
type: help nntrain
net.trainFcn = 'trainlm';  % Levenberg-Marquardt

%% Choose a Performance Function For a list of all performance functions type: help
nnperformance
net.performFcn = 'mse';  % Mean squared error

%% Choose Plot Functions For a list of all plot functions type: help nnplot

net.plotFcns = {'plotperform','ploterrhist','plotregression','plotfit'};
net.trainParam.showWindow=true;
net.trainParam.showCommandLine=false;
net.trainParam.show=1;
net.trainParam.epochs=100;
net.trainParam.goal=1e-8;
net.trainParam.max_fail=20;

%% Train the Network

[net,tr] = train(net,inputs,targets);

%% Test the Network
```

```
outputs = net(inputs);

errors = gsubtract(targets,outputs);

performance = perform(net,targets,outputs);


%% Recalculate Training Performance


trainInd=tr.trainInd;

trainInputs = inputs(:,trainInd);

trainTargets = targets(:,trainInd);

trainOutputs = outputs(:,trainInd);

trainErrors = trainTargets-trainOutputs;

trainPerformance = perform(net,trainTargets,trainOutputs);


%% Recalculate Validation Performance


valInd=tr.valInd;

valInputs = inputs(:,valInd);

valTargets = targets(:,valInd);

valOutputs = outputs(:,valInd);

valErrors = valTargets-valOutputs;

valPerformance = perform(net,valTargets,valOutputs);
```

```
%% Recalculate Test Performance

testInd=tr.testInd;
testInputs = inputs(:,testInd);
testTargets = targets(:,testInd);
testOutputs = outputs(:,testInd);
testError = testTargets-testOutputs;
testPerformance = perform(net,testTargets,testOutputs);

%% View the Network

view(net);

%% Plots
% Uncomment these lines to enable various plots.

figure;
plotperform(tr);
figure;
plottrainstate(tr);
figure;
plotregression(trainTargets,trainOutputs,'Train Data',...
    valTargets,valOutputs,'Validation Data',...
```

```
    testTargets,testOutputs,'Test Data',...
    targets,outputs,'All Data')
figure;
ploterrhist(errors);
```

Code for PSO algorithm

```
%% Particle Swarm Optimization Simulation
% Find minimum of  the objective function
%% Initialization

clear
clc
iterations = 1000;
inertia = 1.0;
correction_factor = 2.0;
swarms = 5000;

% ---- initial swarm position -----
swarm=zeros(5000,7);
step = 1;
for i = 1 : 5000
swarm(step, 1:7) = i;
step = step + 1;
```

```
end


swarm(:, 7) = 1000;      % Greater than maximum possible value

swarm(:, 5) = 0;         % initial velocity

swarm(:, 6) = 0;         % initial velocity


%% Iterations

for iter = 1 : iterations


   %-- position of Swarms ---

   for i = 1 : swarms

      swarm(i, 1) = swarm(i, 1) + swarm(i, 5)/1.2  ;  %update u position

      swarm(i, 2) = swarm(i, 2) + swarm(i, 6)/1.2 ;    %update v position

      u = swarm(i, 1);

      v = swarm(i, 2);


      value = (u - 20)^2 + (v - 10)^2;          %Objective function


      if value < swarm(i, 7)           % Always True

         swarm(i, 3) = swarm(i, 1);    % update best position of u,

         swarm(i, 4) = swarm(i, 2);    % update best postions of v,

         swarm(i, 7) = value;          % best updated minimum value

      end
```

```
    end

    [temp, gbest] = min(swarm(:, 7));        % gbest position

    %--- updating velocity vectors
    for i = 1 : swarms
        swarm(i, 5) = rand*inertia*swarm(i, 5) + correction_factor*rand*(swarm(i, 3)...
            - swarm(i, 1)) + correction_factor*rand*(swarm(gbest, 3) - swarm(i, 1));   % u velocity
parameters
        swarm(i, 6) = rand*inertia*swarm(i, 6) + correction_factor*rand*(swarm(i, 4)...
            - swarm(i, 2)) + correction_factor*rand*(swarm(gbest, 4) - swarm(i, 2));   % v velocity
parameters
    end

    %% Plotting the swarm
    clf
    plot(swarm(:, 1), swarm(:, 2), 'x')   % drawing swarm movements
    axis([-1000 5000 -1000 5000])
pause(.1)
end
```

Genetic Algorithm Code

```
function D = GA(Vpv,Ipv)
```

```
%#codegen
persistent u;
persistent dcurrent;
persistent pbest;
persistent p;
persistent dc;
persistent v;
persistent counter;
persistent gbest;
if(isempty(counter))
    counter=0;
end
if(isempty(dcurrent))
    dcurrent=0.5;
end
if(isempty(gbest))
    gbest=0.5;
end
if(isempty(p))
    p=zeros(4,1);
end


if(isempty(v))
```

```
    v=zeros(4,1);
end

if(isempty(pbest))
    pbest=zeros(4,1);
end
if(isempty(u))
    u=0;
end
if(isempty(dc))
    dc=zeros(4,1);
    dc(1)=0;
    dc(2)=0.3;
    dc(3)=0.5;
    dc(4)=0.9;
end
if(counter>=1 && counter<300)
    D=dcurrent;
    counter=counter+1;
    return;
end
counter=0;
```

```
if(u>=1 && u<=4)
    if((Vpv*Ipv)>p(u))
        p(u)=Vpv*Ipv;
        pbest(u)=dcurrent;
    end
end
u=u+1;
if(u==6)
    u=1;
end
if(u==1)
    D=dc(u);
    dcurrent=D;
    counter=1;
    return;
elseif(u==2)
    D=dc(u);
    dcurrent=D;
    counter=1;
    return;
elseif(u==3)
    D=dc(u);
    dcurrent=D;
```

```
    counter=1;
    return;
elseif(u==4)
    D=dc(u);
    dcurrent=D;
    counter=1;
    return;
elseif(u==5 )
    [m,i]=max(p);
    gbest=pbest(i);
    D=gbest;
    dcurrent=D;
    counter=1;
    %update velocity and duty cycle
    v(1)=updatevelocity(v(1),pbest(1),dc(1),gbest)
    v(2)=updatevelocity(v(2),pbest(2),dc(2),gbest)
    v(3)=updatevelocity(v(3),pbest(3),dc(3),gbest)
    v(4)=updatevelocity(v(4),pbest(4),dc(4),gbest)
    %update duty cycle
    dc(1)=updateduty(dc(1),v(1))
    dc(2)=updateduty(dc(2),v(2))
    dc(3)=updateduty(dc(3),v(3))
    dc(4)=updateduty(dc(4),v(4))
```

```
    return;

else
    u
    y='ga'
    D=0.1
end
end
function vfinal=updatevelocity(velocity,pobest,d,gwbest)
w=0.4;
c1=1.2;
c2=2;
vfinal = (w*velocity)+(c1*rand(1)*(pobest-d))+(c2*rand(1)*(gwbest-d));
end

function dfinal=updateduty(d,velocity)

dup=d+velocity;
if(dup>1)
    dfinal=1;
elseif(dup<0)
    dfinal=0;
else
```

```
    dfinal=dup;
end
end
```