

Photovoltaic MPPT Based on Multilayer Extreme Learning Machine

Moh. Khatib, Lecturer Bambang Sri Kaloko, Lecturer Bambang Sujanarko

Department of Electrical Engineering,
Engineering Faculty, University of Jember, Indonesia
Correspondent: sujanarko.teknik@unej.ac.id

Abstract- This article will discuss the system design to maximize photovoltaic power, using MPPT technique based on Multilayer Extreme Learning Machines (ML-ELM). The system is designed using the training data output from the Incremental Conductance (IC) method. By including the irradiance value and temperature time function, ML-ELM produces good performance, which is characterized by a faster turnaround time and 10% higher energy gain than the IC method.

Keywords- Photovoltaic, MPPT, Incremental Conductance, Multilayer Extreme Learning Machines.

I. INTRODUCTION

The use of renewable energy is growing rapidly, with one of the most widely used energy sources being photovoltaic (PV) [1-2]. The use of PV is driven by the problem of using fossil energy which exacerbates global warming [3]. Optimizing the use of PV is needed, considering that PV has problems with low energy conversion, around 20%. The low conversion is due to material factors and factors that are not yet maximal in power transfer.

MPPT with PID method, [9-10], which uses Fuzzy as MPPT algorithm, [11-12], Artificial Neural Network (ANN) and ANFIS as MPPT algorithm [13-14]. In the MPPT, Buck-Boost Converter, SEPIC, and others are used.

Rapidly developing artificial intelligence technology has resulted in a new type of AI, namely Extreme Learning Machine (ELM). This ELM method is a family of ANN-type AI methods that use networks to process information. ELM has been widely used in various fields, such as image processing, video processing, and even control systems. This AI is in great demand by various groups because in addition to having good accuracy, it also produces a lot of performance improvements.

There has been research on ELM for MPPT, including Extreme Learning Machine as an algorithm that adopts MPPT in the network [15].

In this study also used the MPPT Perturb and Observed model, which utilizes power and voltage variables as inputs. Another research [16], applied the MPPT method with ELM as an AI method and compared the ability of Back propagation - Artificial Neural Network (BP ANN), with the results that ELM had a fairly good performance compared to BP-ANN.

Another research [17], applied the same method to track maximum power. In contrast to previous studies, ELM in this study was combined with Fuzzy. Meanwhile in the study [18], the ELM method was used only for multi-photovoltaic applications.

From all the studies mentioned, ELM has been able to control various types of converters, such as Buck Boost Converter, SEPIC, Boost Converter, and Buck Converter so as to maximize power.

By referring to the development of ELM research results, it turns out that there is an ELM type that has the advantage of minimizing errors, namely the Multilayer Extreme Learning Machine (ML-ELM).

ML-ELM has the advantage of reducing errors because it has more than one layer. With multi-layer, the error will spread to every layer. This causes ML-ELM to produce better accuracy than ELM [18]. This is the basic reason for conducting this research, namely ML-ELM which can improve MPPT performance.

II. RESEARCH METHOD

1. System Block:

This research is based on simulation operation, where the application that will be used is Simulink Matlab. To perform MPPT simulation using Multilayer Extreme Learning Machine, it is necessary to design the system to be formed. Therefore, Figure 1 shows the arrangement of the block diagrams used.

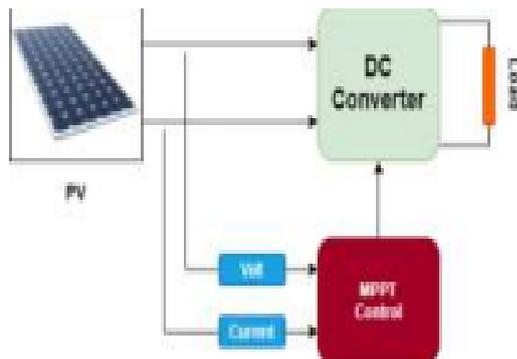


Fig 1. Block Diagram of the System.

The block diagram is like the MPPT mechanism in general. This research consists of 3 main blocks. The first is photovoltaic (PV) as an energy source, then a DC converter in the form of a Buck-Boost Converter (BBC), and the third is MPPT control as a controller to find the maximum transfer point. There are two algorithms used in this research, the first is the Incremental Conductance method, then the Multilayer Extreme Learning Machine (ML-ELM).

Two important variables are needed to obtain this function, namely (V) Voltage and (I) Current. After that, the two variables are multiplied, and the PV power curve is obtained. This curve will then be traced to find the maximum transfer value. For more details related to the simulations contained in Matlab can be seen in Figure 2.

Array	Module	Model	Display
Number of cells per module:			
66			
Open circuit voltage Voc (V):			
64.2			
Short-circuit current Isc (A):			
5.96			
Voltage at maximum power point Vmp (V):			
54.7			
Current at maximum power point Imp (A):			
5.58			

Fig 2. PV module specifications.

2. PV system:

The type of PV used is found in Simulink Matlab, namely Sun Power SPR-305-WHT. The Open Circuit Voltage (VOC) on this PV is 64.2VDC with a Short Circuit current of 5.96A. The array that is connected is five modules. For more details, the PV system used can be seen in Figure 2.

3. Buck-Boost Converter Design:

Following the design shown in the block diagram, this research will use a DC converter, namely a buck boost converter. In the series in this study, DC converter used is a buck-boost converter circuit in general. The circuit design used can be seen in Figure 3.

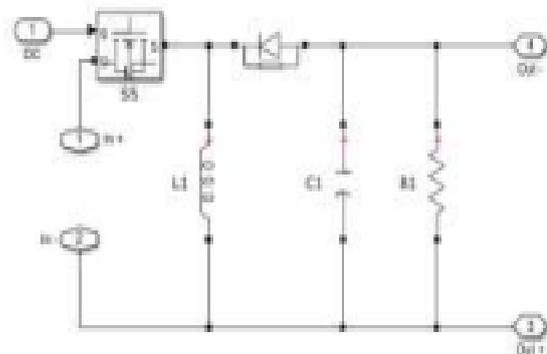


Fig 3. Buck-Boost Converter in Simulink Matlab.

To use this converter, it is necessary to calculate the components using the Buck-Boost Converter equation. From several parameters that have been set according to the type and specifications of the PV used, the following characteristics are obtained: $V_{in}/V_{PV} = 36.3V$, $V_{Out} = 100VDC$, and Frequency = 25000Hz, then the other components can be calculated using the following steps:

$$\frac{V_D}{V_{in}} = \frac{D}{1-D} \quad (1)$$

$$R = \frac{V_D}{I_D} \quad (2)$$

$$L_{min} = \frac{(1-D)^2}{2f} R \quad (3)$$

$$\Delta V_D = \frac{V_D D}{R C f} \quad (4)$$

By using equations 1 to 4, the values of each component used in the simulation are as follows: $L1 = 66.58 \mu H$, $C1 = 67.68 \mu H$, $R1 = 10$, and $Fs = 25000$ Hz.

4. MPPT Control Design:

In the MPPT control itself, there are two controls, each of which will represent the output, in the form of a duty cycle, the first control is to use the IC method, then the output data from the IC method itself will be studied by the ML EML method, then MPPT design using the ML EML method will be obtained.

4.1. Incremental Conductance method:

Incremental conductance (IC) method. This method is based on the change between the current and voltage deltas to the current and voltage conditions at that time. The maximum point in this method is defined when the delta of change is equal to the value of the ratio between current and voltage at that time.

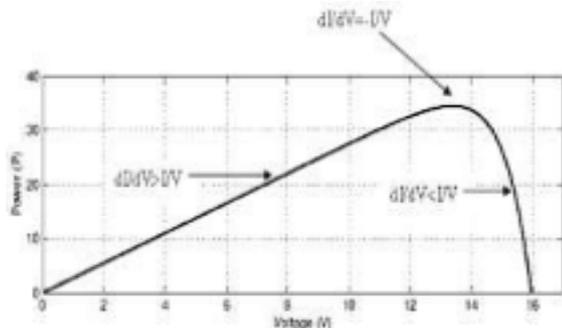


Fig 4. Maximum Power Curve in the Incremental Conductance Method.

In principle, tracking the IC method, it can be written with the following equation:

Maximum Power Point Tracking (MPPT) Theory.
 $\frac{dI}{dV} = \frac{I}{V}$ (Maximum) (5)
 $\frac{dI}{dV} < \frac{I}{V}$ (More than the maximum point) (6)
 $\frac{dI}{dV} > \frac{I}{V}$ (Less than the maximum point) (7).

The equation for the variable delta change from the MPPT . Theory

$$dI = I(0) - I(t-1) \quad (8) \quad dV = V(0) - V(t-1) \quad (9)$$

Where for each variable, represented with the following explanation:

- dI : Delta Current (A)
- dV : Delta Volt (V)
- I(0) : Current time (A)
- I(t-1) : Current at the previous time (A)
- V(0) : Voltage at current time (V)
- V(t-1) : Voltage at previous time (V)

As for the image of the flow chart on the incremental learning algorithm, it can be seen in the figure below.

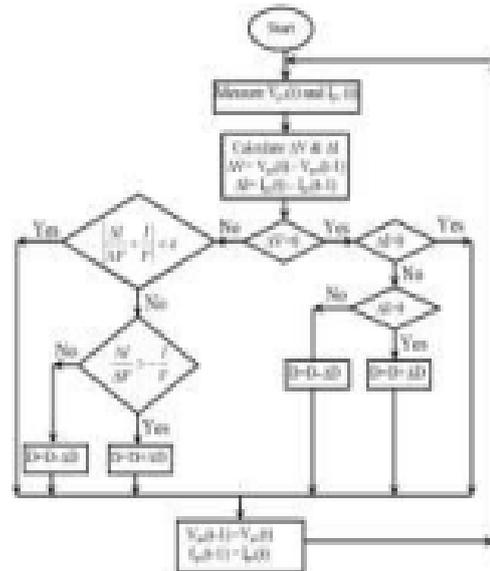


Fig 5. Algorithm on Incremental Conductance.

4.2. Multilayer Extreme Learning Machine:

Multilayer Extreme Learning Machine, known as ML ELM, is a multi-layer Extreme Learning Machine (ELM). This technique is a development of the Artificial Neural Network (ANN) which was developed in 2004 by G.B. Huang [19]. If we talk about the basis of Extreme Learning Machines, it is Single Hidden Layer Feed forward Neural Networks (SLFNs).

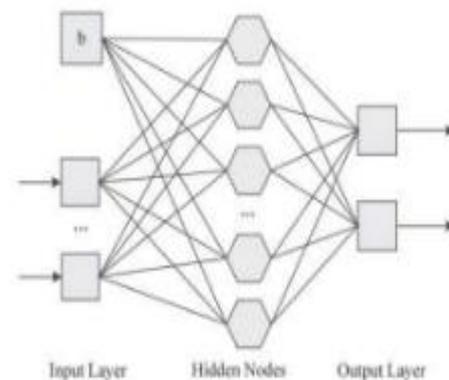


Fig 6. Extreme Learning Machine Architecture.

This technique is in great demand because it has advantages in speed and accuracy of learning compared to back propagation (BP-ANN). [20]. If the back propagation method, ANN, is required to adjust the bias and weight of each neuron using certain equations (forward and backward), then in ELM these parameters are not corrected.

In ANN, improvements are based on the accumulation of errors that occur. Meanwhile, in ELM, there are two things to do. The first is forward,

which is randomly selecting the initial value for each weight and then at the last layer, pseudo-inverse is used to determine the weight value. With this technique, ELM tends to have a fast learning speed and has good generalization performance [18].

The mathematical model of the ELM is as follows, the mathematical model of the ELM: Consider a set of n samples (x_i, y_i) , with $1 \leq i \leq n$ $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, SFLN with M as hidden neurons in the hidden layer which can be expressed by the following equation:

$$\sum_{j=1}^M \beta_j f(w_j x_i + b_j), 1 \leq i \leq n \quad (10)$$

Where b_j is the bias, f is the activation function, w_j is the input weight and b_j is the bias. While \hat{y}_i is the estimated output where the SLFN has a value close to the actual output y_i .

So when the equation is written as follows:

$$\sum_{j=1}^M \beta_j f(w_j x_i + b_j) = y_i, 1 \leq i \leq n \quad (11)$$

When written in a mathematical equation as $H\beta = y$, with:

$$H = \begin{pmatrix} f(w_1 x_1 + b_1) & \dots & f(w_m x_1 + b_m) \\ \vdots & \ddots & \vdots \\ f(w_1 x_n + b_1) & \dots & f(w_m x_n + b_m) \end{pmatrix} \quad (12)$$

$$\beta = (\beta_1, \beta_2, \dots, \beta_j)^T \text{ and } y = (y_1, y_2, \dots, y_i)^T$$

Given these equations, the earliest ELM approach is to use initialization randomly and. This process is called forwarding. Meanwhile, the next step is to calculate the output weight y using Moore Penrose in the form of a pseudo-inverse.

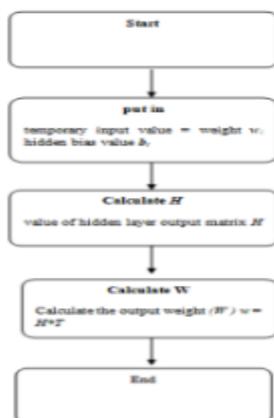


Fig 7. Extreme Learning Machine Flowchart.

4.3. ML-ELM Working Principle: The weights are assigned randomly on the first player to the n -layer (before the last layer). To be able to control MPPT using ML-ELM training data is needed. This is because ML-ELM is included in supervised learning. Therefore, the training data needed are voltage (V), current (I), and (dI) and (dV) data, the output of the ML-ELM is the Duty Cycle or PWM value used to control the Buck-Boost Converter. In this study, in the early stages of ML-ELM using 2 layers with 10 neurons in each layer. Then after the training process data is run, it will be evaluated how much similarity to the training data used.

If it is too low with an indication of the accuracy value or R-Square below 0.85%, then the number of neurons and layers will be increased. The addition will be done using a trial and error method based on changes in the R-Square value when there is an addition to the layer and also the neurons. Testing is done by trial and error method because the target of using ML-ELM is the speed of the process. Therefore, the number of neurons and layers will be limited in finding the best R-Square value. The limit is a maximum of 5 layers and 300 neurons for each layer. The flow diagrams and block diagrams for ML-ELM can be seen below.

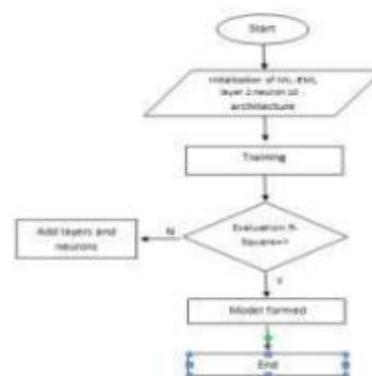


Fig 8. ML-ELM Training Process Flowchart.

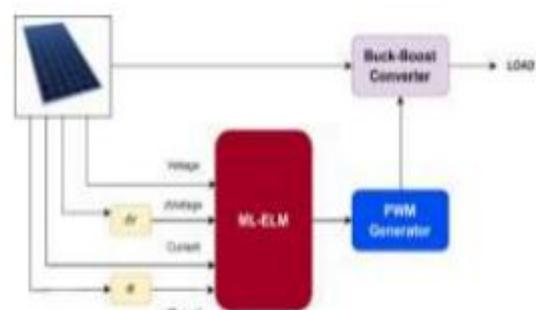


Fig 9. ML-ELM Block Diagram as Buck-Boost Converter Controller.

Where the input used is 4. The four inputs are voltage (V), current (I), voltage delta (dV) and current delta (dI). For the equations of dV and dI are the same as those found in IC or more precisely as contained in Eq. While the output of the ML-ELM is the duty cycle value not the PWM signal, a PWM generator is needed as shown in Figure 5.

5. Simulation Scenario:

To carry out the testing in this paper, it is necessary to have a scenario to get results in each method. Thus, the irradiation value of the PV used is changed in this study which is used as the PV input to test the three MPPT algorithms: IC and ML-ELM.

In order to get the value, the parameter is the point or maximum power reached by each control, settling time or travel time to reach that point. Then the result will be compared to find out the advantages and disadvantages of each MPPT algorithm. Figure of simulation circuit of Simulink using the MATLAB 2019 application shown in Figure 11.

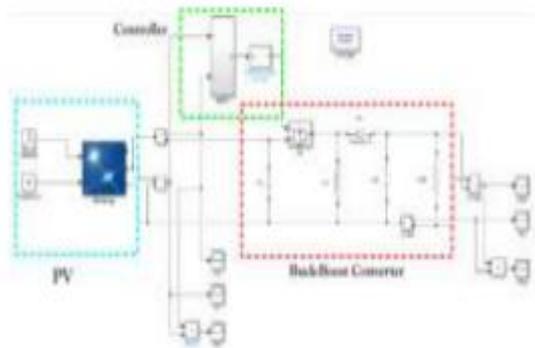


Fig 10. Simulation circuit of Simulink Matlab for large systems.

III. RESULT AND DISCUSSION

To explain the results in this section will be separated into several discussions: the first section on the performance of ML-ELM during the training process; the second is about how the power output of the MPPT method with IC and ML-ELM control; the last is to know the duty cycle obtained by IC and ML EML method.

1. ML EML Training Process:

The first discussion is the Multilayer Extreme Learning Machine working process. Where at the time of the first training experiment with 2 layers and 10 neurons in each layer, then the results will be

obtained, if the first training process the results obtained have not reached the Rsquer of 80%, then the training process will be repeated. And as for the input data from ML EML, the data taken is the PV data created by the author to be able to run the MPPT function. The data includes data on voltage (V), current (I), voltage (V(t-1)) and current (I(t-1)).

Table 1. ML EML training data.

V	I	V(t-1)	I(t-1)	DC
0.196664	39.3328	0	0	0
0.58998	39.32647	39.3328	0.196664	0
0.983193	39.32013	39.32647	0.58998	0
1.376363	39.3138	39.32013	0.983193	0
1.769469	39.30746	39.3138	1.376363	0
2.162532	39.30113	39.30746	1.769469	0
243.0959	32.91331	32.87413	243.3139	0.6462
242.8664	32.95401	32.91331	243.0959	0.64623
242.6255	32.99615	32.95401	242.8664	0.6463

Regarding the architecture that was formed during the evaluation using the dataset in the previous table, then can be seen in Table 1, where the results are quite high with an accuracy value of 0.997466. To do this, we use Python to speed up the calculation process.

This is because by using the Python language and using the google Colab facility, the ML ELM computing process will not consume a large amount of RAM. This is a weakness of all ELM families that only occurs during the training process. The code that we use in this process can be seen in Figure 12. Then from the code we get the load value for the system we designed and then we enter the load manually into the ML-ELM load.

```

import numpy as np
from sklearn.metrics import r2_score

class ELM:
    def __init__(self, architecture, activation_func, class_output):
        self.architecture = architecture
        self.activation_func = activation_func
        self.class_output = class_output
        self.train_data = None
        self.test_data = None
        self.train_weights = None
        self.test_weights = None

    def train(self, X, Y):
        # Training process
        # ... (code for training weights) ...

    def test(self, X):
        # Testing process
        # ... (code for testing weights) ...

    def predict(self, X):
        # Prediction process
        # ... (code for predicting output) ...
    
```

Fig 11. Display for Multilayer Extreme Learning Machine (ML-ELM) code in Python.

To perform the design, namely increasing and decreasing the number of layers in ML-ELM, we use the code shown in Figure 13. Where in the variable section there is an "architecture" column with two values of 10 that are lined up twice. This indicates that we are evaluating two layers with a value of 10 for each neuron. Meanwhile, for "class_type" we use a value of 1, which means that ML-ELM is in regression mode, which is a task for the problems we are facing.

```
#Regression
architecture [10, 10]
activation = 'tanh'
class_type = 1
start = time.time()
model = elm(architecture,activation,class_type)
model.fit(x, y)
end = time.time()
ETA = end - start
print('Time :', ETA)
```

Fig 12. Multilayer Extreme Learning Machine (ML ELM) in Architectural Development.

During the first experiment with 2 layers and 10 neurons in each layer, the results were quite good as shown in Figure 14. Where the results were quite high with an accuracy value of 0.997466. Therefore, the process of adding and subtracting the architecture of ML-ELM does not need to be done.

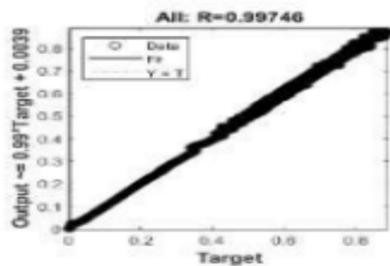


Fig 13. Linear Regression Results to check the accuracy of the ML-ELM training results.

2. MPPT IC and ML EML Output Power Response:

The second discussion is about how each MPPT control, namely Incremental Conductance (IC) and Multilayer Extreme Learning Machine can reach its maximum power. In this study, the PV used has input in the form of changes in irradiance and temperature. To facilitate the analysis, the results are shown in Figure 15.

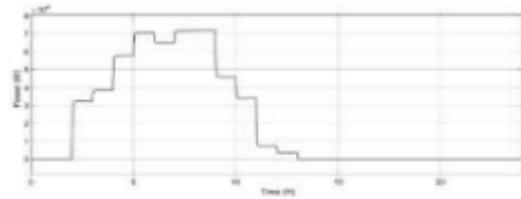


Fig 14. Power Output in Incremental and Conductance method.

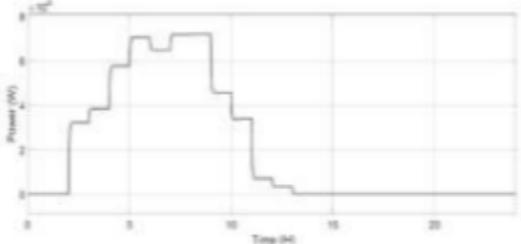


Fig 15. Power Output on the Multilayer Extreme Learning Machine method.

The results obtained on the ML-ELM control itself, where the results obtained have a response that is more or less the same as the IC method. As for the analysis in terms of speed and power generated, it cannot be seen clearly when using graphic analysis. Therefore, this analysis will be carried out using the calculation analysis in the recapitulation sub-chapter.

Table 2. Response time and power achieved when there is a change in irradiance in the Incremental Conductance method.

Time (s)	Irradian (W/m²)	Temperature (C)	Power (W)	Start time (s)	End time (s)	Difference (s)
01.00	0	26.83	0	0	0	0
06.00	0	27.27	0	1	1	0
07.00	397	27.7	3200	2	2.7	0.7
08.00	479	27.87	3800	3	3.2	0.2
09.00	716	29.83	5700	4	4.7	0.7
10.00	892	28.2	6800	5	5.4	0.4
11.00	810	28.84	6200	6	6.2	0.2
12.00	903	27.89	6980	7	7.15	0.15
13.00	984	27.75	7050	8	8.1	0.1
14.00	361	27.43	4800	9	9.3	0.3
15.00	415	27.14	3400	10	10.6	0.6
16.00	192	26.84	700	11	11.3	0.3
17.00	15	26.12	100	12	12.2	0.2
18.00	0	25.41	0	13	13.15	0.15
19.00	0	24.7	0	14	14	0
20.00	0	24.33	0	15	15	0
21.00	0	24.28	0	16	16	0
22.00	0	24.24	0	17	17	0
23.00	0	24.81	0	18	18	0
24.00	0	25.87	0	19	19	0
01.00	0	25.34	0	20	20	0
02.00	0	25.6	0	21	21	0
03.00	0	25.87	0	22	22	0
04.00	0	26.14	0	23	23	0

Table 3. Response time and power achieved when there is a change in irradiance in the ML EML method.

Time (s)	Irradiance (W/m ²)	temperature (C)	power (W)	start time (s)	End time (s)	difference (s)
00:00	0	26.65	0	0	0	0
00:00	0	27.27	0	1	1	0
07:00	997	27.7	32400	2	2.6	0.6
08:00	479	27.67	36000	3	3.15	0.15
09:00	716	28.03	37500	4	4.2	0.2
10:00	892	28.2	38000	5	5.3	0.3
11:00	819	28.04	38000	6	6.2	0.2
12:00	903	27.89	37500	7	7.2	0.2
13:00	966	27.73	35000	8	8.3	0.2
14:00	763	27.43	43000	9	9.4	0.4
15:00	413	27.14	34500	10	10.5	0.4
16:00	162	26.84	1000	11	11.2	0.2
17:00	0	26.42	3400	12	12.1	0.1
18:00	0	25.41	0	13	13.1	0.1
19:00	0	26.7	0	14	14	0
20:00	0	26.6	0	15	15	0
21:00	0	26.29	0	16	16	0
22:00	0	26.34	0	17	17	0
23:00	0	24.81	0	18	18	0
24:00	0	23.07	0	19	19	0
00:00	0	25.34	0	20	20	0
01:00	0	25.6	0	21	21	0
02:00	0	25.97	0	22	22	0
03:00	0	26.14	0	23	23	0

3. Duty Cycle output from IC and ML EML Method

The next analysis compares the duty cycle generated by the IC and ML EML methods on MPPT. Each MPPT method will record every duty cycle achieved in every irradiation change given in this experiment. This result shown in Table 2 and 3.

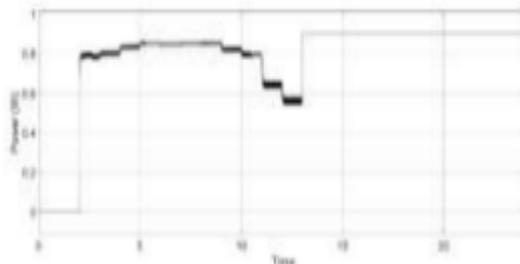


Fig 16. Duty Cycle Response to the Incremental Learning method.

From the results obtained in the previous display, it can be seen that ML-ELM has almost the same response as IC. See Figure 17 and 18. This indication is seen in the value of the duty cycle which almost looks the same for every change in irradiance.

In general, the power achieved also tends to be the same. When viewed from the apparent duty cycle response, it can be seen that the IC and ML-ELM experience oscillation changes in the same value as the test. This also indicates that the response of the two methods can be said to be quite responsive.

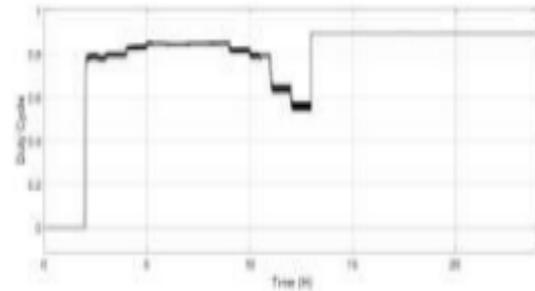


Fig 17. Duty Cycle Response to the Multilayer Extreme Learning Machine method.

IV. CONCLUSION

From the results obtained in this research, it can be concluded that the MPPT design using the ML-EML method has good performance compared to the IC method itself, both in terms of energy achievement, duty cycle and settling time with have a difference of 10%.

REFERENCE

- [1] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, "Challenges and solution technologies for the integration of variable renewable energy sources—a review," *Renew. Energy*, vol. 145, pp. 2271–2285, 2020, doi: 10.1016/j.renene.2019.06.147.
- [2] A. Q. Al-Shetwi, M. A. Hannan, K. P. Jern, M. Mansur, and T. M. I. Mahlia, "Grid-connected renewable energy sources: Review of the recent integration requirements and control methods," *J. Clean. Prod.*, vol. 253, p. 119831, 2020, doi: 10.1016/j.jclepro.2019.119831.
- [3] M. Shahbaz, C. Raghutla, K. R. Chittedi, Z. Jiao, and X. V. VO, "The effect of renewable energy consumption on economic growth: Evidence from the renewable energy country attractive index," *Energy*, vol. 207, p. 118162, 2020, doi: 10.1016/j.energy.2020.118162.
- [4] M. Kermadi et al., "Recent developments of mppt techniques for PV systems under partial shading conditions: A critical review and performance evaluation," *IET Renew. Power Gener.*, vol. 14, no. 17, pp. 3401–3417, 2020, doi: 10.1049/iet-rpg.2020.0454.
- [5] N. Hussein Selman, "Comparison Between Perturb & Observe, Incremental Conductance and Fuzzy Logic MPPT Techniques at Different Weather Conditions," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 5, no. 7, pp. 12556–12569, 2016, doi: 10.15680/ijirset.2016.0507069.

- [6] M. W. Rahman, C. Bathina, V. Karthikeyan, and R. Prasanth, "Comparative analysis of developed incremental conductance (IC) and perturb & observe (P&O) MPPT algorithm for photovoltaic applications," Proc. 10th Int. Conf. Intell. Syst. Control. ISCO 2016, no. Ic, 2016, doi: 10.1109/ISCO.2016.7726991.
- [7] M. N. Ali, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Promising mppt methods combining metaheuristic, fuzzy-logic and ann techniques for grid-connected photovoltaic," Sensors (Switzerland), vol. 21, no. 4, pp. 1–18, 2021, doi: 10.3390/s21041244.
- [8] M. K. Behera and L. C. Saikia, "A new combined extreme learning machine variable steepest gradient ascent MPPT for PV system based on optimized PI-FOI cascade controller under uniform and partial shading conditions," Sustain. Energy Technol. Assessments, vol. 42, no. February, p. 100859, 2020, doi: 10.1016/j.seta.2020.100859.
- [9] C. R. Algarín, J. T. Giraldo, and O. R. Álvarez, "Fuzzy logic based MPPT controller for a PV system," Energies, vol. 10, no. 12, 2017, doi: 10.3390/en10122036.
- [10] J. K. Shiao, Y. C. Wei, and M. Y. Lee, "Fuzzy controller for a voltage-regulated solar-powered MPPT system for hybrid power system applications," Energies, vol. 8, no. 5, pp. 3292–3312, 2015, doi: 10.3390/en8053292.
- [11] S. K. Roy, S. Hussain, and M. A. Bazaz, "Implementation of MPPT Technique for Solar PV System Using ANN," 2017 Recent Dev. Control. Autom. Power Eng., pp. 339–342, 2017.
- [12] A. G. Al-Gizi, A. Craciunescu, and S. J. Al Chlahawi, "The use of ANN to supervise the PV MPPT based on FLC," 2017 10th Int. Symp. Adv. Top. Electr. Eng. ATEE 2017, pp. 703–708, 2017, doi: 10.1109/ATEE.2017.7905128.
- [13] K. Amara et al., "Improved Performance of a PV Solar Panel with Adaptive Neuro Fuzzy Inference System ANFIS based MPPT," 7th Int. IEEE Conf. Renew. Energy Res. Appl. ICRERA 2018, vol. 5, pp. 1098–1101, 2018, doi: 10.1109/ICRERA.2018.8566818.
- [14] N. Priyadarshi, S. Padmanaban, J. B. Holm Nielsen, F. Blaabjerg, and M. S. Bhaskar, "An Experimental Estimation of Hybrid ANFIS-PSO Based MPPT for PV Grid Integration under Fluctuating Sun Irradiance," IEEE Syst. J., vol. 14, no. 1, pp. 1218–1229, 2020, doi: 10.1109/JSYST.2019.2949083.
- [15] M. K. Behera, I. Majumder, and N. Nayak, "Extreme learning forecast model for grid connected MPPT based photovoltaic station," Int. J. Pure Appl. Math., vol. 114, no. 10 Special Issue, pp. 373–381, 2017.
- [16] M. K. Behera, I. Majumder, and N. Nayak, "Solar photovoltaic power forecasting using optimized modified extreme learning machine technique," Eng. Sci. Technol. an Int. J., vol. 21, no. 3, pp. 428–438, 2018, doi: 10.1016/j.jes.tch.2018.04.013.
- [17] Y. Du, K. Yan, Z. Ren, and W. Xiao, "Designing localized MPPT for PV systems using fuzzy weighted extreme learning machine," Energies, vol. 11, no. 10, pp. 1–10, 2018, doi: 10.3390/en11102615.
- [18] K. Anam, C. Avian, and M. Nuh, "Multilayer extreme learning machine for hand movement prediction based on electroencephalography," Bull. Electr. Eng. Informatics, vol. 9, no. 6, pp. 2404–2410, 2020, doi: 10.11591/eei.v9i6.2626.