Hybrid Improved Differential Evolution and Spline-based Jaya for Photovoltaic MPPT Technique

Khusnul Hidayat
Department of Electrical Engineering
Master Student at Universitas Brawijaya and Lacturer at Universitas Muhammadiyah Malang
Malang, Indonesia
khusnulhidayat@yahoo.com

Rini Nur Hasanah*
Department of Electrical Engineering
University of Brawijaya
Malang, Indonesia
*Corresponding author:
rini.hasanah@ub.ac.id

Hadi Suyono
Department of Electrical Engineering
University of Brawijaya
Malang, Indonesia
hadis@ub.ac.id

***Abstract*—Some *Soft Computing* algorithms to solve the maximum power point tracking (MPPT) method problem of the photovoltaic system under partially shaded conditions will stop tracking Global Maxima and produce reference voltage or the best duty-cycle if the difference between the worst and the best candidate solution is smaller than the specified threshold. A large threshold value will produce fast converging, but the accuracy value will be low, and vice versa, then the determination of the threshold value will be very dilemma. Therefore, this study proposed a combination of Improved Differential Evolution (IDE) and Jaya optimization based on predictive curves using cubic spline interpolation to determine the best particles after the IDE reaches convergent criteria, so that with a large threshold value it will still get high accuracy and high convergent speed. Furthermore, the algorithm proposed in this study is known as Improved Differential Evolution and Jaya Based Spline (IDESJaya). The proposed algorithm is compared with conventional P&O, Jaya based on Spline, and IDE. Simulation results show that the IDESJaya technique is faster converging, provides a better overall tracking efficiency and higher accuracy.**

Keywords—maximum power point tracking; differential evolution; jaya algorithm; cubic spline interpolation; photovoltaic system.

I. INTRODUCTION

Photovoltaic (PV) is a source of current whose amplitude depends on exposure to sunlight received by the PV surface. The relationship between power and voltage, current and voltage of photovoltaic are not linear. Maximum Power Point (MPP) location of the PV is varied depending on irradiation and temperature changes [1]. This MPP occupies the top position on the photovoltaic Power-Voltage (P-V) curve. The P-V curve in terms of the irradiation conditions received by photovoltaic divided into two types, Uniform Conditions (UC) and Partially Shaded Conditions (PSC). In uniform conditions, the P-V curve has only one MPP, as all solar cells have the same irradiation and temperature.

Meanwhile, in partially shaded conditions, the P-V curve has more than one MPP. The highest MPP is known as Global Maximum Power Point (GMPP), and other MPPs are known as Local Maximum Power Point (LMPP) [2]. This condition occurs because radiation and temperature in each solar cell are not uniform. Such conditions can occur because some photovoltaic is block by buildings, clouds, towers, and trees [3].

The characteristic curve of PV is not linear, and the PV efficiency is very small, about 23% for silicon material [4], the Maximum Power Point Tracker (MPPT) technique is often used to find or track GMPP values so that the PV power can be maximized. This algorithm requires a power converter that is inserted between solar panels and loads. The power converter acts to control the flow of power from the photovoltaic panel to the load. A good MPPT criterion are (i) can track GMPP quickly, (ii) have a small oscillation during tracking, and (iii) high accuracy.

The most popular conventional MPPT methods are Perturbing and Observing (P&O) [5] and Incremental Conductance (IC) [6], both are very simple algorithms so they can be implemented easily. The concept of these two algorithms is to decrease or increase the reference voltage or duty-cycle directly on the power converter [7]. However, both techniques have oscillations when steady-state around MPP and can only work well when solar modules are in normal conditions. When photovoltaic solar panels are partially shaded, MPP obtained by the method above may not be GMPP but LMPP, so that the power released by photovoltaic is not optimal [8], then tracking accuracy is very low.

The latest research progress related to the design of the MPPT algorithm that can work better is Soft Computing (SC) based on the heuristic tracking method to track GMPP under partially shaded conditions quickly. One of the heuristic search techniques that can be easily implemented to track GMPP is Particle Swarm Optimization (PSO) [9], [10], [11]. The application of the PSO method successfully solved the problem of oscillation around MPP when it was steady-state and successfully tracked the GMPP. However, the initialization of three PSO parameters for MPPT, which is rapidly converging and accurate is quite tedious [8]-[12]. To increase the tracking speed of GMPP by a simple method, [12] have proposed the Improved Differential Evolution (IDE) method. Conventional Differential Evolution (DE) modified its mutation strategy so that it can accelerate convergence. However, a *differential mutation base* with a random solution causes high oscillation when tracking, so tracking efficiency is very low. To improve the efficiency of GMPP tracking, [13] proposed MPPT with Jaya algorithm, which was guided by predictive models based on cubic spline interpolation to obtain a better solution, the method known as S-Jaya. The application of this method has succeeded in reducing oscillation problems when tracking to improve tracking efficiency. However, this method loses its diversity when the level of randomness decreases. Another disadvantage to this method is that when GMPP and better solutions are based on predictive curves among the best and worst particles. The basic idea of the Jaya method is to avoid the worst candidates, and then the best particle updates will never find a better solution so that the best particle update process based on the prediction curve model will be repeated as much as the maximum recurring value specified. Too many predetermined repetitions will slow down convergence.

This study proposed MPPT based on Improved Differential Evolution optimization for photovoltaic systems under uniform and partially shaded conditions. The DE algorithm mutation strategy uses the best solution as a *differential mutation base* in order to increase convergent speed and reduce oscillation when searching GMPP. Some of the advantages of conventional DE are (i) requiring only two parameters that must be regulated and (ii) extremely robust so that it is not easily trapped in a local solution, to maintain these advantages, mutation factor parameters have high values at the beginning of the iteration then decrease with increasing number of iterations.

The proposed SC algorithms above (PSO, S-Jaya, and IDE), will stop tracking Global Maxima and produce the best reference voltage or duty-cycle particle if the difference between the worst and the best solution candidate is less than the threshold that has been determined. A large threshold value will produce fast convergence, but the accuracy will be low, and vice versa [13], then the determination of the threshold value will be very dilemma. Therefore, this study proposed a combination of IDE and Jaya algorithms based on cubic spline interpolation as an objective function to determine the best particles after the IDE reach convergent criteria, so that with a large threshold value it still gets high accuracy and high convergent speed. The basic concept of Jaya method is that the solution obtained must move towards the best solution and avoid the worst solution. One of the advantages of the Jaya method is that no specific parameters need to be determined, unlike the Genetic Algorithm (GA), DE, PSO, and Firefly Algorithm (FA). Furthermore, the algorithm proposed in this study is known as Improved Differential Evolution and Jaya Based on Spline (IDESJaya).

II. ELECTRIC CHARACTERISTICS OF PHOTOVOLTAIC

A. Uniform Conditions

The solar cell current I_{PV} depends on the voltage of the solar cell terminal V_{PV}, as shown in Fig. 1. Photovoltaic current during short circuit I_{SC} and voltage in open circuit V_{OC} are two parameters that are often used to describe the performance of solar cell electricity.

Electrical characteristics of solar cells are represented in the form of P-V and current-voltage (I-V) curves in certain environmental conditions. Fig. 2(a) and Fig. 2(b) shows the characteristics of the curve of solar cells at different levels of irradiation. From Fig. 2(a), it can be seen that the change in I_{SC} is much greater than V_{OC} along with changes in irradiation. Whereas Fig. 2(c), the decrease in V_{OC} is much greater than the increase in I_{SC} when the surface temperature of the solar cell increases. Based on Fig. 2 can be concluded, irradiation changes greatly affect the photovoltaic output current, and temperature changes affect the output voltage of PV.

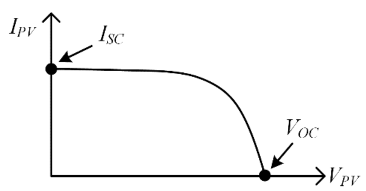


Fig. 1. Power-against-voltage (P-V) curve

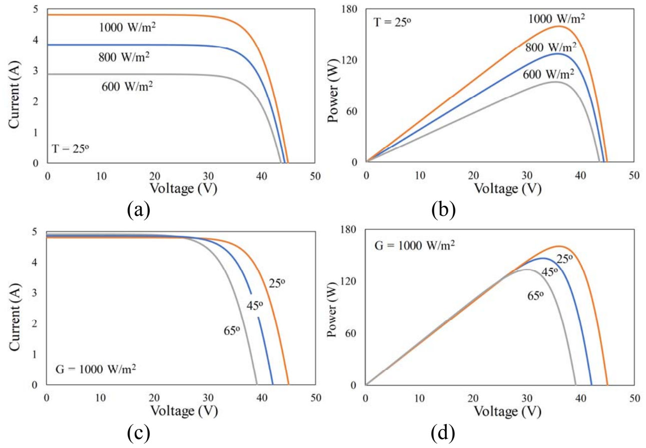


Fig. 2. PV curves under normal conditions: (a) I-V and (b) P-V curves with various irradiations. (c) I-V and (d) P-V curves with various temperatures

B. Partially Shaded Conditions

In general, a solar panel module consists of several PV cells connected in series. Photovoltaic modules can be connected in series or in parallel to get the arrangement of photovoltaic solar panels according to the desired capacity of power, current, and voltage. Partially shaded conditions are conditions when solar cells in photovoltaic solar panels do not receive the same solar radiation. Fig. 3 shows the I-V curves characteristic for two photovoltaic modules connected in series and one module is in uniform condition, and the other is partially shaded. The shaded PV module will operate in the reverse-biased region, while the panel module that is not shaded will operate in the forward-biased region if the solar panel modules connected in series and operate at the current level of I_A. As a result, a shaded PV module will be a burden on modules that are not shaded, causing heat to PV cells, and extreme heat can make the module damaged [14].

A bypass diode can be added to the photovoltaic modules to reduce the problems caused by PSC. How to install a bypass diode can be seen in Fig. 4, where bypass diodes are installed in parallel with half the solar cells installed in series. The output power of the photovoltaic module will be at point *A* if the photovoltaic voltage is at level $\pm 0.4V_{OC,}$ and the current is at level $I_{A,}$ as shown in the P-V curve Fig. 4. The power of the solar panel module will be at point *B* in the P-V curve if the photovoltaic module operates at the level of the I_B current. The current flows the bypass diode on PV shaded when the output power PV operates at points *A* and *B*. In these conditions, shaded photovoltaic cells do not release the power. Both solar cells are in the forward bias region if the PV power output operates at point C, so the

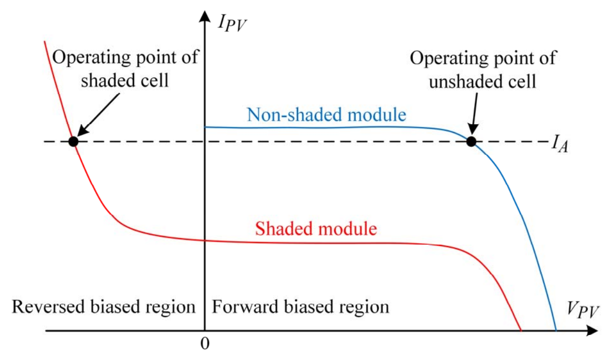


Fig. 3. I-V curve of two photovoltaic modules connected in series without bypass diode

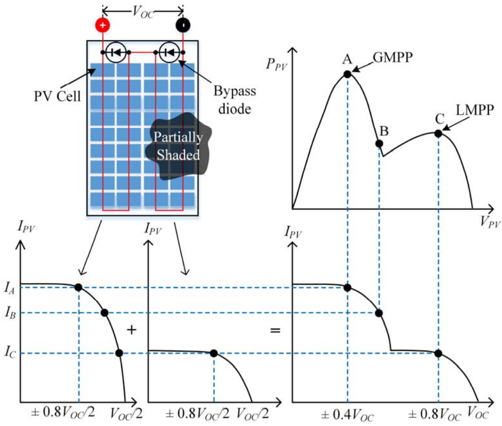


Fig. 4. Partially shaded PV module

generated power comes from all photovoltaic cells. As a result of the solar panel module when partially shaded conditions have several peaks in the power versus voltage curve due to the presence of bypass diodes.

III. Solution Algorithms

In this study, the MPPT method was proposed with a combination of Improved Differential Evolution and Jaya optimization based on cubic spline interpolation as a prediction model for uniform and partially shaded conditions, known as IDESJaya. The Improved Differential Evolution algorithm was executed first before Jaya optimization, the IDE has the role of finding power and voltage data points (P_{PV} & V_{PV}) around the highest power of the photovoltaic. After the IDE algorithm succeeds in meeting convergent criteria, all photovoltaic power and voltage data points are used as a piece of small-scale polynomial functions with three degrees (cubic spline interpolation) as the objective function of Jaya algorithm to find predictions of the best candidate. The best candidate between the IDE and Jaya algorithms is specified as the dc-dc converter reference voltage. The speed of convergence and oscillation tracking algorithm proposed in this study depends on the IDE algorithm, while the level of accuracy depends on the Jaya algorithm.

This study uses a reference voltage based MPPT method using a PI (Proportional & Integral) control to control a dc-dc converter. The method has several advantages compared to the direct duty-cycle based method, e.g., 1) the steady-state response is faster when there is a level change in the dc-dc converter, 2) the candidate solution does not affect the load changes in the dc-dc converter when tracking process, and 3) do not need additional algorithms when there is a load change after convergence.

A. Improved Differential Evolution

Differential Evolution algorithm used for global optimization and was first written by Price and Storn in 1995 [15]. DE works with a simple process, and it is very suitable for solving nonlinear, multidimensional, and many local maxima problems. Unlike traditional EA methods, DE uses different vectors (*genomes*) in order to explore objective functions. The concept of DE comes from the Genetic Annealing algorithm, and the algorithm is based on perturbation in a mutant vector to form a new mutant population. The tracking space will be wide if there is a large population Np, but the speed of convergence will be slower, and vice versa. In this study, the output voltage of the PV as a vector (particle), and the PV output power as a solution candidate. A good population number for PV systems has been discussed in previous studies [9, 10, 11, 14, 15], in this study the population size $Np = 4$, $V_{i,G}$ ($i = 1, \ldots, Np = 4$), where $V_{i,G}$ (known as the *mother*) is the representation of the i^{th} candidate solution at G^{th} iteration. Three evolutionary operations, e.g., differential mutations, crossover, and selection, are executed sequentially [16].

1) Differential Mutation: Differential mutations will generate mutant vectors $U_{i,G}$ for each $V_{i,G}$. The mutation strategy in conventional DE (DE/rand/1) is very strong in terms of exploring most objective functions, but the speed of convergence is very slow. Therefore, this study uses the DE/best/1 strategy to accelerate convergence and minimize oscillations when tracking with the following formula:

$$U_{i,G} = V_{best,G} + F(V_{r1,G} - V_{r2,G}),\ 1 \le i \ne r1 \ne r2 \le Np \quad (1)$$

where $V_{best,G}$ (*differential mutation base*) is the best vector for the solution $P_{best,G}$ at G^{th} iteration, $r1$ and $r2$ random integers [1, Np]. F parameter used to give weight to the difference between two vectors, known as the mutation factor. Mutation factors in conventional DE are fixed between [0, 1], in this study, the mutation factor parameters have high values at the beginning of the iteration then decrease with an increasing number of iterations [12] with the following formula:

$$F = F_{\max} - \left(\frac{G\left(F_{\max} - F_{\min}\right)}{G_{\max}} \right) \quad (2)$$

where G is the number of ongoing iterations, and $G_{\max}$ is the maximum iteration for executing the IDE algorithm, the maximum iteration is determined $G_{\max} = 10$ [13], $F_{\max}$ and $F_{\min}$ are the limits of the highest and lowest mutations parameter with values 0.4 and 0.2.

2) Crossover: The crossover process in DE is done by crossing each vector $V_{i,G}$ with a mutant vector $U_{i,G}$, to make a melting vector $V'_{i,G}$ (known as a child). The most widely used crossover method in DE is the binomial and exponential method [17]. This study uses the binomial crossover method with the following formula:

$$V'_{i,G} = \begin{cases} U_{i,G}, & \text{if } r_{i,G} \le Cr \\ V_{i,G}, & \text{if the other} \end{cases} \quad (3)$$

where $r_{i,G}$ is a random number with a range of [0, 1], and Cr is a crossover probability in the range [0, 1].

In this study, if evolution does not occur or $V'_{i,G}$ does not inherit from mutant vector $U_{i,G}$, then one vector randomly selected from *child* $V'_{i,G}$ will be replaced by vector mutant $U_{i,G}$.

3) Selection: The selection process is often known as mother-child competition. If the trial vector $V'_{i,G}$ has a solution better than the target vector solution $V_{i,G}$, then $V'_{i,G}$ will replace position $V_{i,G}$ in the population for the next generations. If instead, the target will remain in its position in the population, as shown in (4).

$$V_{i,G+1} = \begin{cases} V'_{i,G}, & \text{if } P'_{i,G} > P_{i,G} \\ V_{i,G}, & \text{if the other} \end{cases} \quad (4)$$

B. Jaya Optimization

Jaya method is one of the variations in intelligence optimization algorithms introduced by Rao [18]. The basic principle of this technique is to update the solution repeatedly for a particular problem by moving it to the best candidate solution and avoiding the worst candidate solution. Jaya algorithm does not need to specify parameters, not like PSO, GA, FA, and DE. To maximize solar panel power, in order to get the best solution v_{best}, the first step of Jaya algorithm is to initialize the solution candidate and then update it with (5) and (6) in each iteration [18].

$$v'_{i,G} = v_{i,G} + r_{1,G}\left(v_{best,G} - |v_{i,G}|\right) - r_{2,G}\left(v_{worst,G} - |v_{i,G}|\right) \quad (5)$$

$$v_{i,G+1} = \begin{cases} v'_{i,G}, & \text{if } p'_{i,G} > p_{i,G} \\ v_{i,G}, & \text{if the other} \end{cases} \quad (6)$$

where, $v_{best,G}$ is the particle i^{th} for the best solution candidate and $v_{worst,G}$ is the particle i^{th} for the worst solution candidate. $v'_{i,G}$ is an update particle of $v_{i,G}$, and $r_{1,G}$ and $r_{2,G}$ are two random numbers for variable i^{th} when iteration G^{th} with range [0, 1]. As shown in (6), the update particle $v'_{i,G}$ will be maintained for the next generation if it gives a better solution value than $v_{i,G}$.

C. Cubic Spline Interpolation

Cubic spline interpolation *S(v)* is a function of a three-order small polynomial piece that connects two adjacent data points [19], as shown in (7) and Fig. 5.

$$S(v) = \begin{cases} S_1(v), & \text{if } v_1 \le v \le v_2 \\ \vdots & \vdots \\ S_{n-1}(v), & \text{if } v_{n-1} \le v \le v_n \end{cases} \quad (7)$$

where,

$$S_i(v) = \sum_{q=1}^{4} a_{i,p}\left(v - v_i\right)^{q-1}, \text{ for } i = 1, \cdots, n-1 \quad (8)$$

where n is the number of photovoltaic voltage data points that have been stored as long as the IDE algorithm is executed. Function *S(v)* is built with the following conditions:

- $S_i(v_i) = P_i$ and $S_i(v_{i+1}) = P_{i+1}$, for $i = 1, ..., n$-1
- $S_i(v_{i+1}) = S_{i+1}(v_{i+1})$, for $i = 1, ..., n$-2
- $S'_i(v_{i+1}) = S'_i(v_{i+1})$, for $i = 1, ..., n$-2
- $S''_i(v_{i+1}) = S''_i(v_{i+1})$, for $i = 1, ..., n$-2
- $S''_1(v_1) = S''_{n-1}(v_n)$

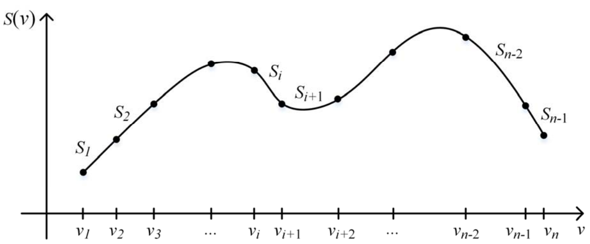


Fig. 5. The polynomial cubic spline approach curve

D. IDESJaya Algorithm

The flow chart for the IDESJaya technique for MPPT is explained in Fig. 6. The proposed IDESJaya algorithm flow chart is described as follows:

Step 1: the initialization of IDE algorithm population can be done randomly or can be determined between [0, V_{OC}]. To ensure the GMPP search covers most P-V curves, the initialization of four vectors applied to the power converter is determined as follows $V'_{i,G}$ = 10, 20, 30, and 40, where $V'_{i,G}$ is an update from $V_{i,G}$. During the initialization process, all $V_{i,G}$ and $P_{i,G}$ are zero.

Step 2: calculate all candidate power solutions $P'_{i,G}$ for all $V'_{i,G}$, $P'_{i,G}$ calculated from the multiplication of the output current end voltage of the photovoltaic. Current and voltage and sensor readings are carried out after 50 ms candidate $V'_{i,G}$ applied to the power converter [12, 13], this method is done so that the parameters to be measured have reached steady-state so that measurements are more accurate [13].

Step 3: evaluate for all candidate solutions using (4).

Step 4: find the best vector $V_{best,G}$ for the best solution $P_{best,G}$.

Step 5: The IDE technique will stop tracking the best solution if the difference between the worst $P_{worst,G}$ and best $P_{best,G}$ solution and the highest $V_{high,G}$ and the lowest particle $V_{low,G}$ is smaller from the threshold ε, the threshold is set at 5% [12], or the number of iterations has reached maximum

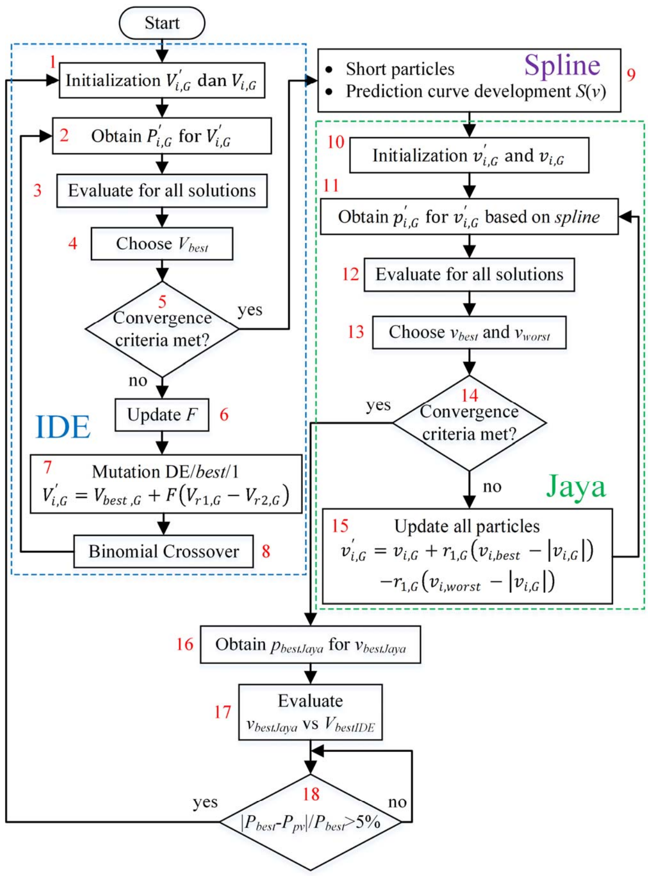


Fig. 6. Flow chart of the IDESJaya algorithm

Algorithm 1: Cubic spline interpolation

Input: $\{(v_i, p_i)\}_{i=1}^{n}$, where $v_1 < ... < v_n$;
for $i = 1: n - 1$
$h_i = v_{i+1} + v_i$;
end
for $i = 2: n - 1$
$\alpha i = 3(p_{i+1} - p_i)/h_i - 3(p_i - p_{i-1})/h_{i-1}$;
end
$l_1 = 1$; $\mu_1 = 0$; $z_1 = 0$;
for $i = 2: n - 1$
$l_i = 2(v_{i+1}\text{-}v_{i-1})\text{-}h_{i-1}\mu_{i-1}$;
$\mu_i = h_i/l_i$;
$z_i = (\alpha_i\text{-}h_{i-1}z_{i-1})/l_i$;
end
$a_{i,3} = 0$;
for $i = n\text{-}1: \text{-}1: 1$
$a_{i,3} = z_i\text{-}(\mu_i a_{i+1,3})$;
$a_{i,1} = p_i$;
$a_{i,2} = (p_{i+1}\text{-}p_i)/h_i\text{-}h_i(a_{i+1,3}+2a_{i,3})/3$;
$a_{i,4} = (a_{i+1,3}\text{-}a_{i,3})/(3h_i)$;
end
Output: $a_{i,q}$ for $i = 1, ..., n - 1$ and $q = 1, ..., 4$;

$_{max}$ = 10 [13], if the convergent criteria have not been met, then execute step 6. But if it is fulfilled, then execute the 9th step.

Step 6: update mutation parameter F each iteration with (2).

Step 7: mutation process using (1).

Step 8: crossover process using (3), then re-execute step 2. The crossover probability is $Cr = 7.3$ [3].

Step 9: prediction curve model based on cubic spline interpolation is developed according to the recording of the photovoltaic voltage and current data points that have been explored by the IDE algorithm. The procedure for obtaining $a_{i,p}$ (8) based on the conditions given, is explained in Algorithm 1.

Step 10: the population of Jaya algorithm is twice that of the IDE, $v'_{i,G}$ (i = 1, … , 8), so that the solution provided has higher accuracy than the IDE algorithm.

Step 11: Find all power $p'_{i,G}$ for all candidate update solutions $v'_{i,G}$. Replace all fitness functions $p_{i,G} = S(v_{i,G})$ with the fitness function $p'_{i,G} = S(v'_{i,G})$, if $p'_{i,G} > p_{i,G}$. Which, $S(v_{i,G})$ and $S(v'_{i,G})$ is a function of cubic spline interpolation (8).

Step 12: evaluate all candidate solutions using (6).

Step 13: find the best particle $v_{best,,G}$ for the best solution $p_{best,G}$.

Step 14: Jaya technique will stop tracking the best solution if the difference between the worst p_{worst} and best p_{best} candidate solutions, and the highest v_{high} and lowest particles v_{low} is smaller than the threshold ε = 5%, or the number of iterations G has met the maximum limit. If the convergent criteria have not been fulfilled, then the 15th step is executed, but if it is fulfilled, then the 16th step is executed.

Step 15: particle updates in Jaya algorithm use equation (5). After the update process is complete for all particles, then the 11th step is executed.

Step 16: apply $v_{best\text{-}jaya}$ to the dc-dc converter, then calculate $p_{best\text{-}jaya}$.

Step 17: the reference voltage to be applied to the buck converter is the best solution between the IDE and SJaya algorithm solutions.

Step 18: the IDE algorithm will re-track or re-initialization (step 1) when there is a change in photovoltaic power with conditions such as in the following equation:

$$\frac{\left|P_{best} - P_{now}\right|}{P_{best}} > 5\% \qquad (9)$$

IV. Simulation Studies and Analyses

A. Simulation Setup

The MPPT block diagram for a voltage reference-based photovoltaic system with a PI control for controlling a buck converter is shown in Fig. 7. A simulation study using MATLAB/Simulink. The circuit parameters are as follows: load the converter is a resistor 2 ohms, the switching frequency of semiconductor devices on the power converter is 5 kHz, the inductor is 2 mH, and the capacitors of the converter is 500 uF.

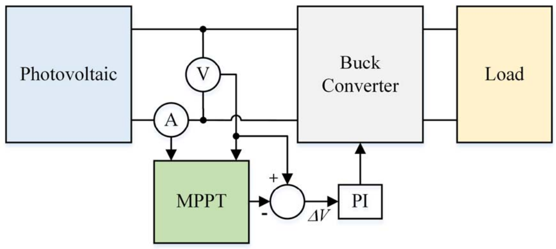


Fig. 7. Block diagram of the PV control system

A simulation study is carried out with four different scenarios, one PV scenario in uniform condition and three subsequent scenarios under partially shaded conditions, as seen in Fig. 8. Photovoltaic temperatures in all four scenarios are worth the same. The maximum power under uniform conditions is 119.1 W, while GMPP in PSC-1, PSC-2, and PSC-3 are 74.72 W. 64.45 W, and 54.54 W.

As a comparison of the performance of the proposed MPPT technique, other algorithms were also tested, e.g., generic P&O, S-Jaya, and IDE. The algorithm parameters are generally determined as follows: the population size for all algorithms is $Np = 4$, step size duty-cycle for P&O algorithm is ΔD = 0.2, sampling time for all algorithm Ts = 0.05 seconds, convergent criteria ε = 5%. Crossover parameter for IDE comparison algorithm Cr = 67 [12]. The IDE comparison algorithm mutation strategy uses the following

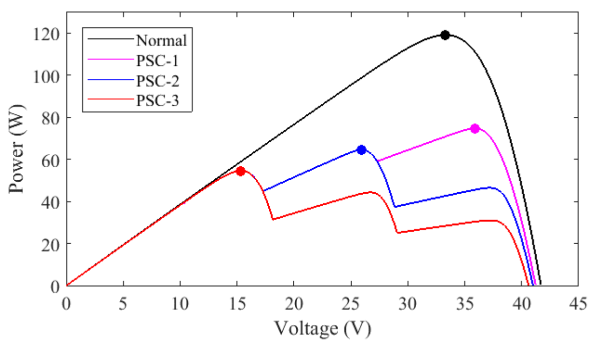


Fig. 8. PV curves under UC and PSC

formula:

$$V'_{i,G} = \begin{cases} V_{r1,G} - F\left|V_{r2,G} - V_{r3,G}\right|, & \text{if } V_{r1,G} \geq V_{best,G} \\ V_{r1,G} + F\left|V_{r2,G} - V_{r3,G}\right|, & \text{if the other} \end{cases} \quad (10)$$

B. Result and Analysis

The overall performance of the MPPT algorithm in simulation study was analyzed based on three parameters, e.g., convergent speed, tracking efficiency from the beginning to the completion of the simulation, and accuracy. The efficiency of overall tracking is formulated using (11).

$$\eta = \frac{\int P(t)dt}{\int P_{MPP}(t)\,dt} \times 100\% \quad (11)$$

where $P_{MPP}(t)$ and $P(t)$ are the maximum power point on the power versus voltage curve and the measured power at time t. The efficiency of the MPPT algorithm at steady state is determined by Equation (12), which will produce tracking accuracy.

$$\eta o = \frac{Po}{P_{MPP}} \times 100\% \quad (12)$$

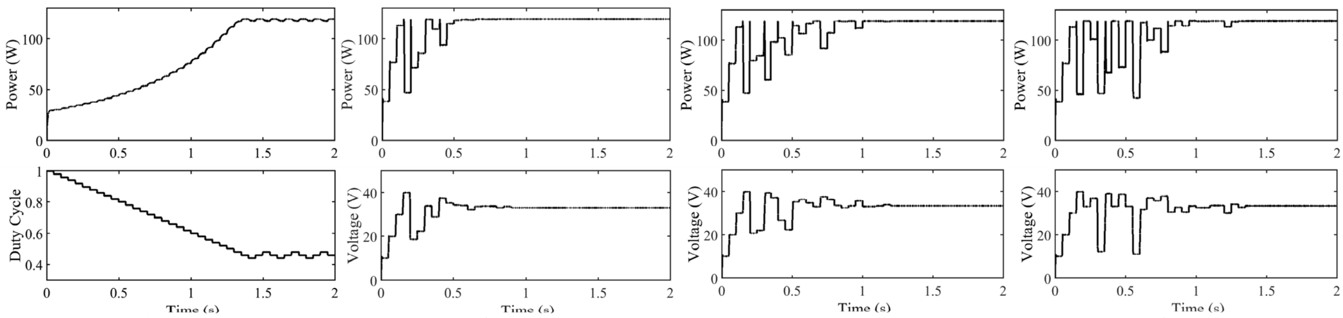
where Po is the average power of the photovoltaic output after the algorithm reaches a convergent or steady-state in the P&O algorithm, while P_{MPP} is the maximum power point on the power versus voltage curve.

The P&O algorithm is tested once for each scenario, the three optimization methods (IDESJaya, S-Jaya, and IDE) are stochastic, then tracking for the optimization algorithm will be repeated 20 times to get objective results. Simulation results for all scenarios and algorithms can be seen in Table 1. Based on Table 1, it can be observed that the proposed algorithm can outperform comparison algorithms in terms of convergent speed, tracking efficiency, and accuracy. Standard deviation (SD) which is low value on convergent speed, search efficiency, and accuracy for the IDESJaya algorithm, demonstrate its better to maintain performance.

TABLE I. SUMMARY OF SIMULATION RESULTS

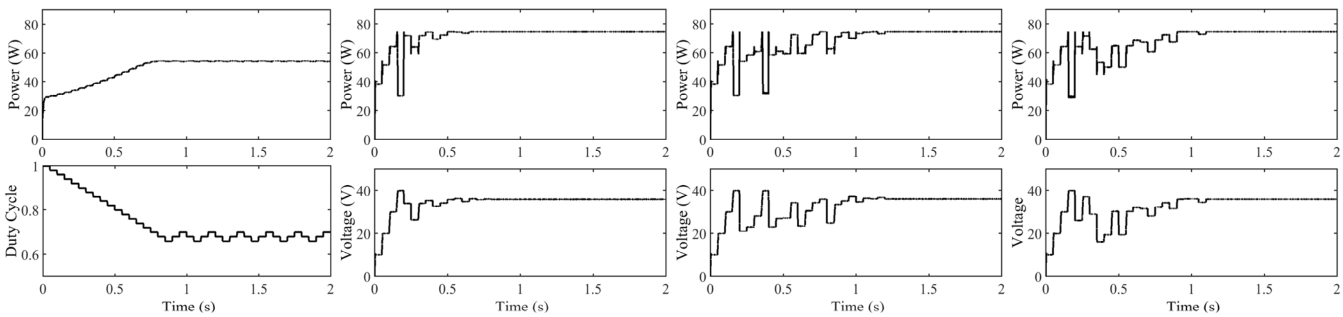
Scenario	Algorithm	Convergence Time (s)		η		ηo	
		Mean	SD	Mean	SD	Mean	SD
Normal	P&O	1.4	-	77.45%	-	99.40%	-
	IDESJaya	0.83	0.15	96.00%	0.01	99.99%	0.0002
	S-Jaya	1.16	0.33	95.26%	0.01	99.88%	0.003
	IDE	1.14	0.31	92.94%	0.03	99.97%	0.0008
PSC-I	P&O	0.85	-	67.85%	-	72.79%	-
	IDESJaya	0.86	0.17	96.16%	0.01	99.99%	0.0001
	S-Jaya	1.05	0.24	96.09%	0.01	99.95%	0.0008
	IDE	1.22	2.40	92.34%	0.88	99.34%	0.02
PSC-II	P&O	0.85	-	78.65%	-	84.36%	-
	IDESJaya	0.91	0.13	95.56%	0.01	99.98%	0.0001
	S-Jaya	1.17	0.25	94.62%	0.01	99.75%	0.005
	IDE	1.48	0.28	90.22%	0.04	99.14%	0.03
PSC-III	P&O	0.85	-	92.94%	-	99.69%	-
	IDESJaya	0.84	0.17	95.75%	0.01	99.99%	0.0001
	S-Jaya	1.24	0.23	94.21%	0.02	99.78%	0.008
	IDE	1.24	0.38	89.40%	0.06	97.50%	0.06

Based on Table 1, the overall tracking efficiency for IDESJaya and S-Jaya algorithms is better than the IDE algorithm. The fitness value of the candidate update of the S-Jaya technique is calculated based on the predictive curve based on cubic spline interpolation, if the fitness value prediction is worse than the previous fitness value, then the particle update process is repeated again until the fitness prediction value is better or has reached the recurrence limit determined [13], the opportunity to get a better solution becomes high, so that it can improve tracking efficiency and accelerate convergence, that is why the S-Jaya algorithm proposed by [13] has higher efficiency than the IDE algorithm. While the idea of the best vector-based *differential mutation base* (DE/*best*/1/bin) in the IDESJaya algorithm is to create new mutants around the best solution,



(a) (b) (c) (d)

Fig. 9. Simulated power curve for (a) P&O, (b) IDESJaya, (c) S-Jaya, dan (d) IDE algorithm at uniform condition



(a) (b) (c) (d)

Fig. 10. Simulated power curve for (a) P&O, (b) IDESJaya, (c) S-Jaya, dan (d) IDE algorithm under PSC-1

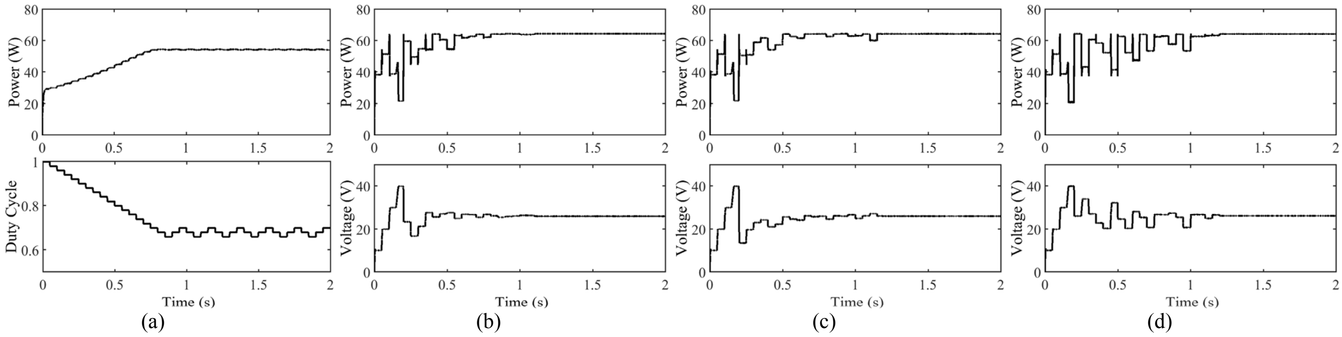


Fig. 11. Simulated power curve for (a) P&O, (b) IDESJaya, (c) S-Jaya, dan (d) IDE algorithm under PSC-2

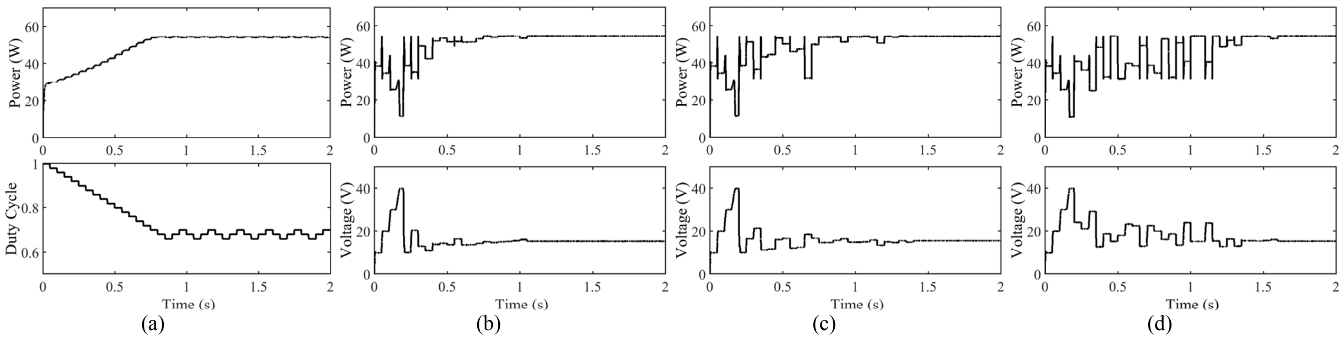


Fig. 12. Simulated power curve for (a) P&O, (b) IDESJaya, (c) S-Jaya, dan (d) IDE algorithm under PSC-3

and proven to be able to increase convergence speed and search efficiency, compared to *differential mutation base* based on random vectors (DE/rand/1/bin) in the IDE algorithm.

The best solution prediction generated by cubic spline interpolation as the Jaya algorithm prediction curve that is executed after the convergent IDE algorithm can outperform two comparison algorithms in terms of accuracy, even though the convergent criteria in the three algorithms are set the same.

Fig. 9 is the characteristic waveform tracking of the MPPT method when PV under uniform conditions, all algorithms successfully track GMPP. Fig. 10, and Fig. 11 are the characteristics of the MPPT method tracking waveform when PV under PSC-1 and PSC-2, all optimization-based

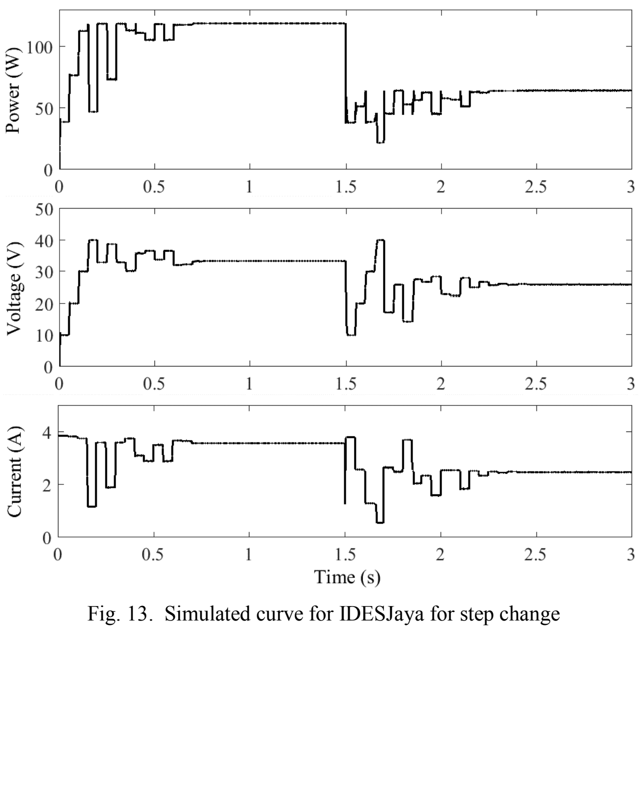


Fig. 13. Simulated curve for IDESJaya for step change

successfully track GMPP, while the P&O algorithm is stuck on local solutions. The P&O algorithm tracking process when trapped in a local solution. Fig. 12 is a characteristic waveform tracking of the MPPT method when PV under PSC-3, all algorithms successfully track GMPP.

C. Curve Change from UC to PSC

Previous simulation studies, IDESJaya techniques were tested on static irradiation conditions, because in practice PV modules can be exposed to cloud shadows so that the irradiation received by PV will change, so this time to find out the strength of IDESJaya technique also needs to be tested under dynamic irradiation conditions. A step change from uniform to partially shading condition (PSC-2). The simulation results for irradiation change cases can be seen in Fig. 13. In this case, every time there is a change in the PV output power either caused by changes in irradiation or partially shaded PV, the reference voltage of IDESJaya technique are re-initialized to track the new GMPP again. The simulation results of IDESJaya techniques when irradiation changes prove their proficiency in achieving fast convergence at 0.8 seconds for the first pattern and 0.95 seconds for each second pattern to reach GMPP. Overall tracking efficiency of IDESJaya for curve change, as shown in Fig. 13 is 91.45%.

V. CONCLUSIONS

A new evolutionary technique named IDESJaya was proposed in this study for photovoltaic systems under uniform and partially shaded conditions. This IDESJaya algorithm is a combination of modified Differential Evolution techniques and Jaya methods with predictive curve models based on cubic spline interpolation. The best vector is chosen as a *differential-mutation-base* useful to accelerate convergence and minimize tracking oscillation. After the Differential Evolution algorithm succeeded in achieving convergent criteria, all PV power, and voltage data points from DE tracking was developed into pieces of third-order polynomial functions (cubic spline interpolation) as predictive curves used by Jaya algorithm to track the best solutions. The best solution between Differential Evolution and Jaya algorithms is applied as a reference voltage for the power converter.

The results of the overall study in simulation both during uniform and partly shaded conditions indicate that the algorithm proposed in this study can outperform comparison algorithms (P&O, S-Jaya, and IDE) in terms of convergent speed, low oscillation when tracking, and accuracy. Low standard deviation values at convergent speeds, search efficiency, and accuracy for the IDESJaya algorithm indicate that the ability to maintain performance is better then S-Jaya and IDE. Simulation test of IDESJaya method when irradiation changes are also able to prove its proficiency in achieving convergence at 0.8 seconds for the first pattern (uniform condition) and 0.95 seconds for the second pattern (PSC-2) to reach GMPP.

REFERENCES

[1] H. A. Sher, A. F. Murtaza, A. Noman, K. E. Addoweesh dan K. Al-Haddad, "A New Sensorless Hybrid MPPT Algorithm Based on Fractional Short-Circuit Measurement and P&O MPPT," *IEEE Transactions On Sustainable Energy*, vol. 6, no. 4, pp. 1426-1434, 2015.

[2] H. Patel dan V. Agarwal, "MATLAB-Based Modeling to Study the Effects of Partial Shading on PV Array Characteristics," *IEEE Transactions on Energy Conversion*, vol. 23, no. 1, pp. 302-310, 2008.

[3] N. Kumar, I. Hussain, B. Singh and B. K. Panigrahi, "Rapid MPPT for Uniformly and partial Shaded PV System by using JayaDE Algorithm in Highly Fluctuating Atmospheric Conditions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2406-2416, 2017.

[4] D. Rekioua and E. Matagne, Optimization of Photovoltaic Power Systems, London: Springer, 2012.

[5] F. Yuniar, R. N. Hasanah dan O. Setyawati, "Pengendalian MPPT Berbasis Metode P&O Menggunakan Boost Converter," *Jurnal Arus Elektro Indonesia* (*JAEI*), vol. 3, no. 1, pp. 1-6, 2017.

[6] A. Safari dan S. Mekhilef, "Simulation and Hardware Implementation of Incremental Conductance MPPT With Direct Control Method Using Cuk Converter," *IEEE Transaction On Industrial Electronics*, vol. 58, no. 4, pp. 1154-1161, 2011.

[7] J.-K. Shiau, Y.-C. Wei dan B.-C. Chen, "A Study on the Fuzzy-Logic-Based Solar Power MPPT Algorithms Using Different Fuzzy Input Variables," *Algorithm*, vol. 8, pp. 100-127, 2015.

[8] J. P. Ram and N. Rajasekar, "A novel Flower Pollination based Global Maximum Power Point method for Solar Maximum Power Point Tracking," *IEEE Transactions on Power Electronics*, vol. 32, no. 11, pp. 8486-8499, 2017.

[9] Y.-H. Liu, S.-C. Huang, J.-W. Huang dan W.-C. Liang, "A Particle Swarm Optimization-Based Maximum Power Point Tracking Algorithm for PV Systems Operating Under Partially Shaded Conditions," *IEEE Transactions On Energy Conversion*, vol. 27, no. 4, pp. 1027-1035, 2012.

[10] K. Ishaque dan Z. Salam, "A Deterministic Particle Swarm Optimization Maximum Power Point Tracker for Photovoltaic System Under Partial Shading Condition," *IEEE Transactions On Industrial Electronics*, vol. 60, no. 8, pp. 3195-3206, 2013.

[11] K. L. Lian, J. H. Jhang dan I. S. Tian, "A Maximum Power Point Tracking Method Based on Perturb-and-Observe Combined With Particle Swarm Optimization," *IEEE Journal of Photovoltaics*, vol. 4, no. 2, pp. 626-633, 2014.

[12] K. S. Tey, S. Mekhilef, M. Seyedmahmoudian, B. Horan, A. T. Oo and A. Stojcevski, "Improved Differential Evolution-based MPPT Algorithm using SEPIC for PV Systems under Partial Shading Conditions and Load Variation," *IEEE Transactions on Industrial Informatics*, pp. 4322 - 4333, 2018.

[13] C. Huang, L. Wang, R. . S.-c. Yeung, Z. Zhang, H. S.-h. Chung and A. Bensoussan, "A Prediction Model Guided Jaya Algorithm for the PV System Maximum Power Point Tracking," *IEEE Transactions on Sustainable Energy*, vol. PP, no. 99, pp. 1-1, 2017.

[14] Y. Wang, Y. Li and X. Ruan, "High-Accuracy and Fast-Speed MPPT Methods for PV String Under Partially Shaded Conditions," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 235 - 245, 2016.

[15] R. Storn dan K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces," ICSI Technical Report TR-95-012, Berkeley, 1995.

[16] A. Qing, Differential Evolution: Fundamentals and Applications in Electrical Engineering, Clementi: Clementi, 2009.

[17] C. Lin, A. Qing and Q. Feng, "A comparative study of crossover in differential evolution," *Springer J Heuristics*, 2010.

[18] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, p. 19–34, 2016.

[19] Supriyanto, "Interpolasi Cubic Spline," Lab. Komputer, Departemen Fisika, Universitas Indonesia, Jakarta, 2006.