International Journal of Scientific Engineering and Science
Volume 8, Issue 7, pp. 136-141, 2024. ISSN (Online): 2456-7361

# PID-MPSO Based Dual Axis System Design for Sun Tracking Control

Arya Sula Cakra Buana[1], Bustanul Arifin[2], Muhammad Khosyi'in[3]
[1, 2, 3]Department of Electrical Engineering, Universitas Islam Sultan Agung, Semarang, Indonesia

***Abstract***— *The research aims to design and develop an optimized PID controller using the Modified Particle Swarm Optimization (PID-MPSO) algorithm on a dual-axis solar tracking system. The algorithm is designed to increase the accuracy of PID parameters to improve the performance of photovoltaic control. Performance evaluation showed that PID-MPSO provides better performance than uncontrolled, auto-tuned, and PSO methods. On the horizontal axis, the PID-MPSO algorithm produced an overshoot of 45.17% and an undershoot of 9.51%, compared to uncontrolled, which produces an overshoot of 34.64% and an undershoot of 9.23%; auto tuning, which produces an overshoot of 57.41% and an undershoot of 26.64%; and PSO, which produces an overshoot of 56.06% and an undershoot of 20.27%. On the vertical axis, PID-MPSO produced an overshoot of 41.68% and an undershoot of 9.29%, compared to uncontrolled, which produced an overshoot of 219.62% and an undershoot of 96.65%; auto tuning, which produced an overshoot of 44.91% and an undershoot of 19.35%; and PSO, which produced an overshoot of 38.82% and an undershoot of 8.78%. Overall, the PID-MPSO algorithm proved to be effective in reducing overshoot and undershoot and the most stable among other methods. PID-MPSO can significantly improve the performance of PID controllers, making it a superior choice for photovoltaic applications in dual-axis solar tracking systems.*

***Keywords***— *MPSO, Photovoltaic, PSO, Uncontrolled.*

## I. Introduction

Electrical energy is the primary need of modern society. This is because electricity is the main energy source for electronic equipment that supports social aspects such as places of worship, housing, and industries such as large-scale factories and homes. In its development, the modern era is an era towards clean energy, namely electrical energy obtained from renewable energy sources. A large amount of natural energy can be used, such as solar energy and other energies. Solar energy uses solar energy to generate electricity by utilizing photovoltaic panels as a medium for capturing solar energy. Photovoltaic panels are gaining popularity due to their effectiveness, affordability, and scalability. Common problems in capturing energy from sunlight are the geographical location of an area, weather, and shifts in the direction of sunlight. In countries that have sufficient sunlight, photovoltaic panels will be designed to have a control system that follows the direction of the sun's movement.

The maximum absorption of solar energy in photovoltaic panels is highly dependent on the system's ability to track and follow the direction of sunlight. Previous research has tried various methods to overcome this problem. P. V. Mahesh used a regression-based machine learning algorithm to improve MPPT efficiency to more than 95% [1]. Imam Abadi implemented MPPT with the ANFIS, which resulted in an increase in PV energy of up to 46.19843% compared to a fixed PV system [2]. Sryang T. Sarena developed a solar tracking system (STS) with a microcontroller and zero-order fuzzy sugeno to increase the efficiency of sunlight capture [3]. However, challenges in terms of control accuracy and stability still exist. This research proposes the use of the Modified Particle Swarm Optimization algorithm to optimize the angle and orientation of solar panels more efficiently. MPSO is designed to overcome the weaknesses of previous methods and improve the control performance of a dual-axis solar tracking system. The main contribution of this research is the development of an MPSO algorithm that provides more accurate and stable control, thereby increasing the efficiency of solar energy absorption.

## II. Theoretical Review

Solar energy is one example of alternative energy and suitable for tropics, but a simple solar panel system is not optimum enough to get energy from the constantly moving and changing sun, so the development of a dual-axis solar tracker with intelligent controls needs to continue.

### A. DC Motor

A common actuator in control systems is the DC motor. These motors directly provide rotary motion and, coupled with wheels, drums, and cables, can provide translational motion. To perform system simulation, an appropriate model must be created. Therefore, a model based on the specifications of the motor needs to be obtained. Figure 1 shows a DC motor circuit with torque and rotor angle considerations.

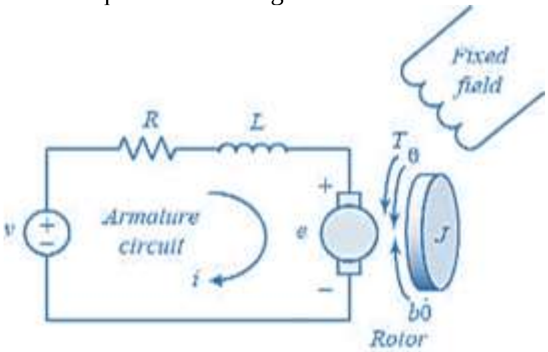

Fig. 1. Schematic Diagram of DC Motor

Based on Figure 1, the motor torque T is related to the armature current, i, with a torque constant K. Some parameters of the DC motor are moment of inertia (J), motor friction-viscosity constant (b), electromotive force constant (Kb), motor torque constant (Kt), resistance (R), and inductance (L). The input of this system is the voltage source (V), and the output is the shaft position (ɵ) In this case, the assumption is made that the rotor and shaft are rigid. The friction torque is proportional to the angular velocity of the shaft. we can write equations based on Newton's law combined with Kirchhoff's law, as shown in Equations (1) and (2).

$$J\frac{d^2\theta}{d^2t}+b\frac{d\theta}{dt}=Ki \qquad (1)$$

$$L\frac{di}{dt}+Ri=V-K\frac{d\theta}{dt} \qquad (2)$$

J is the symbol of the moment of inertia, d is the dimension, b acts as the friction constant of the motor, V acts as the voltage source, and Ri is the current resistance value. L acts as inductance, and K is a constant [4].

### B. Transfer Function

In the field of control theory, transfer functions are commonly used to describe the relationship between input and output elements in a system that can be represented by a time-independent linear differential equation. The transfer function of a time-independent linear differential equation is defined as the ratio of the Laplace transform of the system output (referred to as the response function) to the Laplace transform of the input, assuming that the initial conditions are set to zero [5].

### C. Transfer Function of DC Motors

In a direct current motor, the Laplace transformation value has been obtained by performing a mathematical derivation of a motor model into equation (3), and the mathematical value of the direct current without load will be converted into an s function in order to enter the MATLAB simulation stage as shown in equation (4).

$$L.sI(s)+RI(s) = V(s) - K.\ s\ \theta(s) \qquad (3)$$

$$\frac{\theta(s)}{V(s)}=\frac{K}{s\left((Js+b)(Ls+R)+K^2\right)} \qquad (4)$$

L is the inductance value, RI is the current resistance value, V is the DC motor voltage value, Ksɵ is the angular constant value, and b is the motor friction viscosity constant value [6].

### D. Axis-Twisting Moment of Inertia

In tracking, the torque value of the photovoltaic load will definitely be obtained from the moment of inertia of the photovoltaic panel multiplied by the rotating angular acceleration. The horizontal rotating-axis photovoltaic moment of inertia can be defined in equations (5) and (6) [7].

$$J_1=\frac{1}{12}m_{pv}L^2\left(\frac{N_2}{N_1}\right)^2\ [kg.m^2] \qquad (5)$$

$$J_{T1}=J_{st}+J_1\ [kg.m^2] \qquad (6)$$

The vertical rotating axis can be formulated in equations (7) and (8).

$$J_2=\frac{1}{2}m_{pv}(L^2+W^2)\left(\frac{N_2}{N_1}\right)^2\ [kg.m^2] \qquad (7)$$

$$J_{T2}=J_{st}+J_2\ [kg.m^2] \qquad (8)$$

J has a role as the moment of inertia, L is the length of the PV, and mPV has a role or represents the mass of the PV. Jst is the moment of inertia of the solar tracker without a load. The symbol W symbolizes the width of the photovoltaic [8].

### E. Dual Axis Rotation Transfer Function

In the moment of inertia of the dual-axis rotation of photovoltaic solar tracking, changes have been made to the MATLAB simulation stage, namely the transfer function. Dual axis rotation are horizontal axis and vertical axis. For we get dual axis transfer function, we will substitution moment of inertia at equation (4). In horizontal axis, we substitution equation (6) to equation (4) and then we get horizontal axis transfer function at equation (9). Likewise with the vertical axis, do substitution equation (8) to equation (4), so we get vertical axis transfer function in equation (10).

$$\frac{\theta(s)}{v(s)}=\frac{K}{s\left((JT1s+b)(Ls+R)+K^2\right)} \qquad (9)$$

$$\frac{\theta(s)}{v(s)}=\frac{K}{s\left((JT2s+b)(Ls+R)+K^2\right)} \qquad (10)$$

S is the time function in Laplace notation format, J is the moment of inertia in gear 1 or 2, R is the electrical resistance, L is the electrical inductance, b is the frictional viscosity constant of the motor, K is the (back) electromotive force constant, and V is the voltage [9].

### F. Particle Swarm Optimization (PSO)

The PSO algorithm is a scalable computing technology that simulates the foraging activities of a flock of birds. It seeks optimal solutions through cooperation and information sharing among individuals in the group. The particles in PSO have only two attributes: speed and position, where speed represents the speed of movement and position represents the direction of movement. Each particle searches individually for the optimal solution in the search space, which is characterized as the individual optimal value of the particle. The individual optimal value of each particle in the particle swarm is shared with each other, and the optimal individual optimal value is found from it, which is characterized as the global optimal value [10].

### G. Modified Particle Swarm Optimization (MPSO)

MPSO stands for modified particle swarm optimization and is an improved form of the more common PSO technique. The convergence speed and search efficiency of the basic algorithm are both significantly improved by adding new features in MPSO. The goal of modified PSO is to improve the efficiency and convergence rate of the conventional PSO algorithm by using several modifications. Adaptive inertia weights, velocity clamping, as well as regional topology are some examples of this algorithm. This algorithm can achieve a good balance between exploration and exploitation with the help of adaptive inertial weights, which modify the particle velocity in real-time. To keep the particle from spreading too far, we can "pinch" its velocity at a narrow range. Particle interactions and information propagation are characterized by the topology of the environment in which they occur. As a result of these changes, Modified PSO can effectively investigate the search space and identify the best solution [11].

## III. Methods

### A. Research Model

The research begins with the collection of technical data, motor specification data, and transfer function data. The data that has been collected is used as a basis or foothold in applying the Uncontrolled, PID-Auto, PID-PSO, and PID-MPSO methods to the dual-axis solar tracking simulation. In general, the flowchart of the dual-axis tracking system design with MPSO and its comparison method can be described as a flowchart in Figure 2.

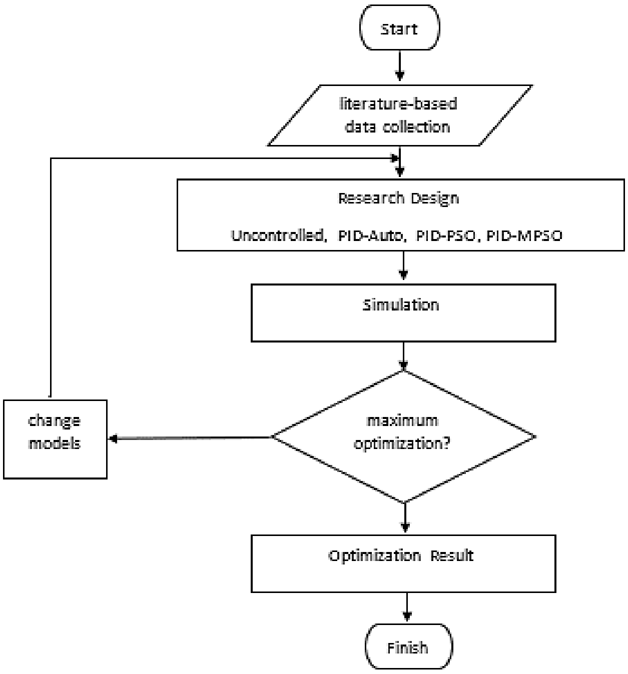

Fig. 2. Research Model Flow of Dual Axis Solar Tracking Based on PID-MPSO with Comparison Method

Based on the flowchart in Figure 2, it begins with the collection of literature or data that includes an understanding of solar tracking PV systems with dual axes: Uncontrolled, PID-Auto, PID with PSO, and PID with MPSO. Data is collected in the form of DC motor parameters, which include friction constant, moment of inertia, emf constant, resistance, inductance, and torque constant. Other parameters are photovoltaic model parameters in terms of mass and dimensions, and then the next parameter is gear parameters in the form of mass and diameter. Next, the angular position modeling of the DC motor combination with the dual-axis solar tracking system is carried out. The second step is to design the PSO and MPSO algorithms as a means of optimizing the parameters of the PID, then continue at the trial or running stage with the MATLAB R2023a program. The results of data processing from tuning with PSO and MPSO are continued where it will get a successful response or not; if not successful, it will enter the evaluation phase or failure analysis and continue in the same process. When there is a response, it will continue at the result analysis stage, and the data analysis process will be continued in the discussion process. From all existing processes, complete research data will be generated, which will be concluded.

### B. Data Collection

Photovoltaic (PV) is a load from a tracking system that is placed in such a way that the PV position is always perpendicular to the sun. In this research, use a simulation of the spur gear or gear unit. The gear is a spur gear consisting of two types, namely type M1B12 and type M1A20, as shown in Table 1. This study uses a PV model of type STM 40-50 as shown in Table 2. Technical specifications in the form of system parameters used have specification data that can be seen in Table 3.

TABLE 1. Unit Gears [12]

| No. | Spurs Gear Models | Number of teeth | Mass (g) |
|---|---|---|---|
| 1 | Model M1A20 | 120 | 1320 |
| 2 | Model M1B12 | 12 | 10 |

TABLE 2. STM40-50 PV Model Parameters [7]

| No. | Parameter | Value |
|---|---|---|
| 1 | Dimension (mm) | 637X545X35 |
| 2 | Mass (Kg) | 4.5 |
| 3 | $J_1$ (Kg.m$^2$) | 0.0015216 |
| 4 | $J_{T1}$(Kg.m$^2$) | 0.0015488 |
| 5 | $J_2$ (Kg.m$^2$) | 0.0158129 |
| 6 | $J_{T2}$ (Kg.m$^2$) | 0.01584 |
| 7 | L (m) | 0.637 |
| 8 | W (m) | 0.545 |

TABLE 3. DC Motor Model Parameters [12]

| No. | Parameter | Value |
|---|---|---|
| 1 | J (Kg.m$^2$) | 3,2284x10$^{-6}$ |
| 2 | b (N.m.s) | 3,5077 x10$^{-6}$ |
| 3 | Kb (Vsec.rad$^{-1}$) | 0,0274 |
| 4 | Kt (Nm.Amp$^{-1}$) | 0,0274 |
| 5 | R (Ω) | 4 |
| 6 | L (H) | 2,75 x10$^{-6}$ |

### C. Transfer Function: Horizontal and Vertical Axis

To obtain the transfer function of PV on the horizontal and vertical axes, it is necessary to find the rotating angular velocity ratio, as shown in equations (11) and (12).

$$wR = \frac{N_2}{N_1} \quad (11)$$

$$wR = \frac{12}{120} = 0.1 \quad (12)$$

The rotary angular velocity ratio equation is multiplied by the dual axis rotation transfer function so that it becomes equations (13) and (15) to get the transfer function that will be used in the simulation, namely equations (14) and (16).

$$TF_{Hor} = \frac{\theta(s)}{V(s)} = wR.\frac{K}{s((J_{T_1}s+b)(Ls+R)+K^2} \quad (13)$$

$$TF_{Hor} = \frac{\theta(s)}{V(s)} = \frac{0.00274}{4.259x10^{-9}s^3+0.0061952\,s^2+0.0007648\,s} \quad (14)$$

$$TF_{Ver} = \frac{\theta(s)}{V(s)} = wR.\frac{K}{s((J_{T_2}s+b)(Ls+R)+K^2} \quad (15)$$

$$TF_{Ver} = \frac{\theta(s)}{V(s)} = \frac{0.00274}{4.356\,x10^{-8}s^3+0.06336\,s^2+0.0007648\,s} \quad (16)[13]$$

## IV. Results and Discussion

In this simulation, there is a simplification of the transfer function that will be used by shrinking the zero value, which is too much to be simpler and easier to write without reducing the function and simulation results. The transfer function value will be multiplied by a value of 1000/1000 where 1000/1000 is 1, which means the multiplier is 1. The multiplier value of 1 can be stated that the transfer function value remains the same so that equations (14) and (16) can still be used in the basic simulation model Figure 3.

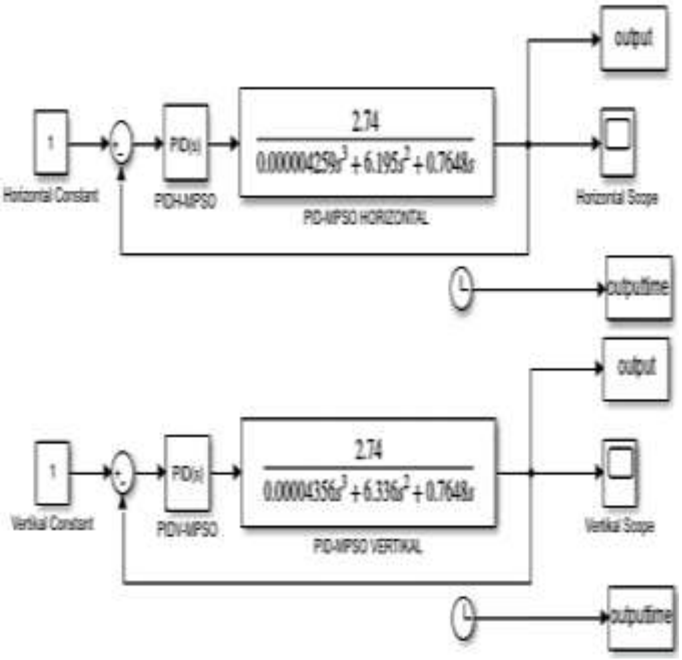

Fig. 3. Simulink design of PID-MPSO method

The results of the horizontal and vertical axis responses for Uncontrolled, PID-Auto, PID-PSO, and PID-MPSO can be seen in Figures 4, 5, Table 4, and 5.

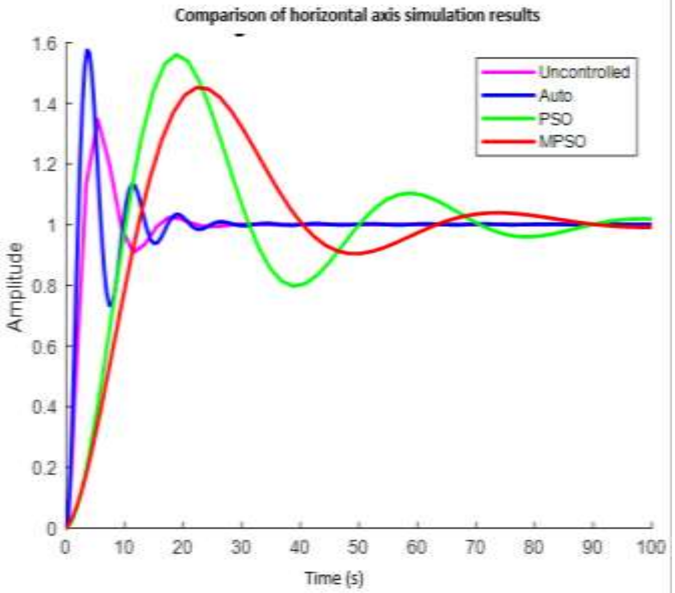

Fig. 4. Comparison of horizontal axis simulation results

In Figure 4, it is clear that the uncontrolled horizontal axis simulation results are quite stable, followed by auto simulation results, which clearly have a high overshoot at the beginning of the simulation, then followed by the PSO method, which has an overshoot almost the same as auto but slightly lower, and then simulation results with MPSO, which has an overshoot close to the uncontrolled version. In terms of the response to the increase in simulation results, it can be seen that the increase in overshoot in the uncontrolled and auto methods occurs earlier when compared to the simulation results of the PSO and MPSO methods.

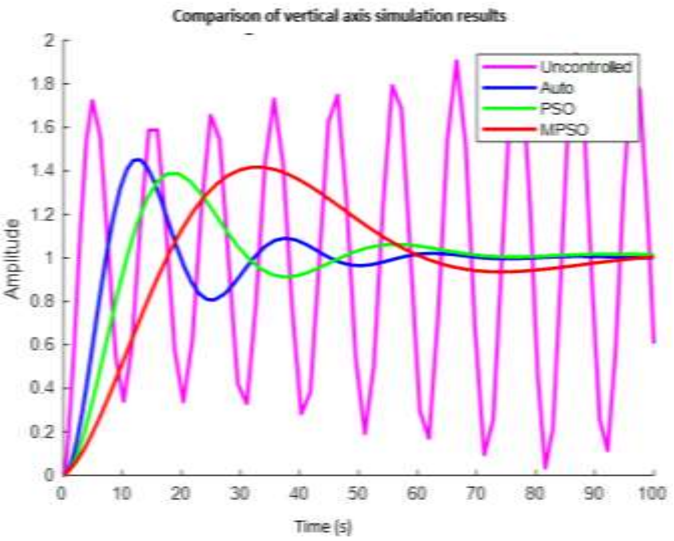

Fig. 5. Comparison of vertical axis simulation results

TABLE 4. Horizontal axis simulation result comparison data

| Description | Uncontrolled | Auto | PSO | MPSO |
|---|---|---|---|---|
| Kp | - | 1.108738051<br>69029 | 0.0706 | 0.0513 |
| Ki | - | 0.007892959<br>09834757 | 0.0032 | 0.0021 |
| Kd | - | 0.923885804<br>516481 | 0.0277 | 0.0548 |
| N | - | 0.818491886<br>754977 | 100 | 100 |
| Risetime | 2.17 s | 1.35 s | 7.03 s | 8.93 s |
| Transient Time | 19.47 s | 20.18 s | 85.51 s | 83.24 s |
| Settling Time | 19.47 s | 20.18 s | 85.51 s | 83.24 s |
| Settling Min | 0.91 s | 0.73 s | 0.80 s | 0.90 s |
| Settling Max | 1.35 s | 1.58 s | 1.56 s | 1.45 s |
| Overshoot | 34.64 % | 57.41 % | 56.06% | 45.17% |
| Undershoot | 9.23 % | 26.64 % | 20.27% | 9.51 % |
| Peak | 1.35 s | 1.58 s | 1.56 s | 1.45 s |
| Peak Time | 5.41 s | 3.56 s | 18.81 s | 22.25 s |

TABLE 5. Vertical axis simulation result comparison data

| Description | Uncontrolled | Auto | PSO | MPSO |
|---|---|---|---|---|
| Kp | - | 0.147069348<br>292425 | 0.0749 | 0.0297 |
| Ki | - | 0.000312338<br>570016394 | 0.0010 | 0.0011 |
| Kd | - | 0.042701856<br>4885878 | 0.0194 | 0.0427 |
| N | - | 0.242859295<br>278801 | 100 | 100 |
| Risetime | 1.28 s | 4.86 s | 7.43 s | 12.75 s |
| Transient Time | 99.95 s | 54.24 s | 67.99 s | 91.04 s |
| Settling Time | 99.98 s | 54.24 s | 67.99 s | 91.04 s |
| Settling Min | 0.03 s | 0.81 s | 0.91 s | 0.91 s |
| Settling Max | 1.93 s | 1.45 s | 1.38 s | 1.42 s |
| Overshoot | 219.62 % | 44.91 % | 38.82 % | 41.68 % |
| Undershoot | 96.65 % | 19.35 % | 8.78 % | 9.29 % |
| Peak | 1.94 s | 1.45 s | 1.39 s | 1.45 s |
| Peak Time | 86.78 s | 13.06 s | 19.50 s | 31.94 s |

Figure 5 can be clearly seen that the vertical axis simulation version for the uncontrolled method is very unstable; the simulation results have very high overshoot and undershoot, and there is no sign of heading towards system stability. Auto simulation results appear to have overshoot and undershoot earlier when compared to PSO and MPSO methods. In the simulation results, it can be seen that the overshoot results of auto, PSO, and MPSO are not too significant a difference, but the undershoot version is very clear that the results of the auto simulation are quite high when compared to the very small PSO and MPSO simulation results.

In Table 4, it can be seen that the auto method gives the fastest risetime (1.35 s), while PSO and MPSO have much longer risetime. The auto (20.18 s) and uncontrolled (19.47 s) methods have similar transient times, while PSO and MPSO show much higher values. The uncontrolled and MPSO methods showed minimum and maximum settling values that were closest to the set point value, indicating better stability. Uncontrolled has the lowest overshoot (34.64%), which indicates better performance in terms of stability of value increase. Uncontrolled (9.23%) and MPSO (9.51%) have the lowest undershoot, indicating better performance in terms of downward stability. All the peak values of the four methods are very similar, namely uncontrolled (1.35 s), auto (1.56 s), PSO (1.56 s), and MPSO (1.45 s). Auto (3.56 s) has the fastest peak time, with uncontrolled (5.41) slower but much faster than PSO and MPSO. It can be concluded that auto's overshoot is very high compared to PSO and MPSO. The effect of high overshoot can cause instability and potential damage to the system. MPSO has offered lower overshoot than auto, which means it is more stable. Auto has a lower settling time when compared to PSO and MPSO; however, this must be balanced with a higher overshoot. The simulation results with MPSO have a very low undershoot and are almost the same as uncontrolled, which indicates better stability after overshoot.

Table 5 shows that uncontrolled has the fastest rise time (1.28 s), followed by (4.86 s). Auto has the lowest transient and settling time (54.24 s), which indicates faster response. PSO and MPSO have minimal and maximal settling values that are closest to the set point value. PSO has the lowest overshoot (30.82%), followed by MPSO (41.68%), which shows better performance in terms of stability. Uncontrolled has a very large overshoot (219.62%), so it is declared very unstable. PSO (8.78%) and MPSO (9.29%) have the lowest undershoot, which shows better performance. Uncontrolled has a very large undershoot value (96.65%), making it unstable. The lowest peak values are PSO (1.39 s) and MPSO (1.42 s). Auto has the fastest peak time (13.06), followed by PSO (19.50 s). Based on the analysis, overall, the total of the horizontal and vertical axes shows that the PID-MPSO method is the most stable but has the disadvantage of a slower response time.

## V. Conclusion

The design of the dual-axis solar tracking system runs well; this is based on the overall overshoot and undershoot values generated, and the system successfully goes to a steady state. Overall, the response of the dual-axis solar tracking system with the PID-MPSO method shows slower response time but better stability after overshoot. Although it takes longer to reach stability, the system tends to reach more stable values with lower overshoot and undershoot. The PID-MPSO method has better performance in handling input changes and approaching the set point value of 1 radian; the overall overshoot and undershoot values are better than the other methods. Quantitative data, processed and compared, supports this conclusion.

## VI. Limitation

In conducting this research, the problem limitations are that the dual-axis solar tracking simulation with the PID-MPSO method is not influenced by time or environmental factors. This research is conducted in simulation using MATLAB Simulink. This research only focuses on testing the response of the PID-MPSO system with test parameter fix data in the form of gear unit parameter data, DC motor, and STM 40-50 PV Model.

## Acknowledgment

The authors are grateful to my family, all the lecturers, supervisor, staff, friends, and students of the Magister Program of Electrical Engineering Department, Universitas Islam Sultan Agung, Semarang, Indonesia.

## References

[1] V. Mahesh, S. Meyyappan, and R. R. Alla, “Maximum Power Point Tracking with Regression Machine Learning Algorithms for Solar PV systems,” Int. J. Renew. Energy Res., vol. 12, no. 3, pp. 1327–1338, 2022, doi: 10.20508/ijrer.v12i3.13249.g8517.

[2] I. Abadi, C. Imron, Mardlijah, and R. D. Noriyati, “Implementation of Maximum Power Point Tracking (MPPT) Technique on Solar Tracking System Based on Adaptive Neuro-Fuzzy Inference System (ANFIS),” E3S Web Conf., vol. 43, 2018, doi: 10.1051/e3sconf/20184301014.

[3] S. T. Sarena, R. Y. Adhitya, N. Rinanto, and D. Hartono, “Pengembangan PV Solar Tracking System Dua Sumbu Putar Berbasis Model Fuzzy Sugeno Oorde Nol,” J. IPTEK, vol. 24, no. 1, pp. 1–10, May 2020, doi: 10.31284/j.iptek.2020.v24i1.549.

[4] K. Sao, D. Kumar Singh, A. Agrawal, P. Scholar, A. Professor, and D. Raman, “Study of DC Motor Position Control using Root Locus and PID Controller in MATLAB,” *IJSRD-International J. Sci. Res. Dev.*, vol. 3, no. 05, pp. 2321–0613, 2015, [Online]. Available: www.ijsrd.com

[5] N. Putri and M. Muhafzan, “Konvensional Realisasi Dari Fungsi Transfer Dalam Bentuk Kanonik Terkontrol,” *J. Pendidik. Mat.*, vol. 10, no. 1, p. 12, 2019, doi: 10.36709/jpm.v10i1.5640.

[6] M. Ali, H. Suyono, M. A. Muslim, M. R. Djalal, Y. M. Safarudin, and A. A. Firdaus, “Determination of the parameters of the firefly method for PID parameters in solar panel applications,” *Sinergi*, vol. 26, no. 2, p. 265, 2022, doi: 10.22441/sinergi.2022.2.016.

[7] A. Adhim and A. Musyafa, “Optimization of PID Controller Based on PSO for Photovoltaic Dual Axis Solar Tracking in Gresik Location – East Java,” *Int. J. Eng. Technol. IJET-IJENS*, vol. 16, no. 01, pp. 65–72, 2016.

[8] M. Ali and R. Rukslin, “Hybrid System of Dual Axis Photovoltaic Tracking System Using ANFIS-ACO,” *Front. Energy Syst. Power Eng.*, vol. 3, no. 2, p. 1, 2021, doi: 10.17977/um049v3i2p1-12.

[9] M. Ali, R. Rukslin, and C. Hasyim, “Hybrid System of Dual Axis Photovoltaic Tracking System Using PID-CES-ACO,” *JEEMECS (Journal Electr. Eng. Mechatron. Comput. Sci.*, vol. 4, no. 2, pp. 59–68, 2021, doi: 10.26905/jeemecs.v4i2.6138.

[10] Q. Song, L. Yu, S. Li, N. Hanajima, X. Zhang, and R. Pu, "Energy Dispatching Based on an Improved PSO-ACO Algorithm," *Int. J. Intell. Syst.*, vol. 2023, pp. 1–17, 2023, doi: 10.1155/2023/3160184

[11] S. Verma, I. Sahu, A. D. Prasad, and M. K. Verma, "Modified Particle Swarm Optimization for the Optimum Use of Multi-Reservoir Systems: MRP Complex, Chhattisgarh," *J. Environ. Informatics Lett.*, vol. 10, no. 2, pp. 132–144, 2023, doi: 10.3808/jeil.202300111

[12] M. Ali *et al.*, "The comparison of dual axis photovoltaic tracking system using artificial intelligence techniques," *IAES Int. J. Artif. Intell.*, vol. 10, no. 4, pp. 901–909, 2021, doi: 10.11591/IJAI.V10.I4.PP901-909.

[13] A. Adhim, "PID Auto Tuning Menggunakan PSO Pada Sistem Fotovoltaik Penjejak Matahari Dua-Sumbu," M.S. thesis, Department of Physics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2016.