



QGFSMC: a Q-learning-assisted fuzzy integral sliding MPPT controller for PV power boost conversion systems

Ammar K. Al Mhdawi , Akram Bati & Hamed Al-Raweshidy

To cite this article: Ammar K. Al Mhdawi , Akram Bati & Hamed Al-Raweshidy (2026) QGFSMC: a Q-learning-assisted fuzzy integral sliding MPPT controller for PV power boost conversion systems, International Journal of Sustainable Energy, 45:1, 2740876, DOI: [10.1080/14786451.2026.2740876](https://doi.org/10.1080/14786451.2026.2740876)

To link to this article: <https://doi.org/10.1080/14786451.2026.2740876>



© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 29 Oct 2026.



Submit your article to this journal [↗](#)

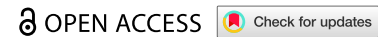


View related articles [↗](#)



View Crossmark data [↗](#)

RESEARCH ARTICLE



QGFSMC: a Q-learning-assisted fuzzy integral sliding MPPT controller for PV power boost conversion systems

Ammar K. Al Mhdawi^a, Akram Bati^a and Hamed Al-Raweshidy^b

^aSchool of Engineering, University of Greater Manchester, Bolton, United Kingdom; ^bDepartment of Electronic and Electrical Engineering, Brunel University of London, Uxbridge, United Kingdom

ABSTRACT

With increasing renewable-energy demand, efficient maximum power point tracking (MPPT) remains essential for photovoltaic (PV) systems operating under nonlinear and varying conditions. This paper proposes a hybrid Q-learning-based fuzzy sliding mode controller (QGFSMC), integrating Q-learning adaptation, fuzzy decision-making, and gain-scheduled sliding mode control to improve dynamic tracking and disturbance rejection. The coordinated framework provides rapid convergence while maintaining robust operation under irradiance variations and load disturbances. Simulation results demonstrate superior performance compared with *P*&*O*, incremental conductance (INC), and model predictive control (MPC). QGFSMC achieved an effective startup rise time of approximately 0.05 s, startup settling time of 0.06 s, and post-disturbance settling time of approximately 0.02 s following disturbances at $t = 1$ s and $t = 1.5$ s. Furthermore, QGFSMC maintained the highest minimum transient PV power, demonstrating improved disturbance rejection, faster recovery, and enhanced dynamic MPPT performance under varying operating conditions.

ARTICLE HISTORY

Received 11 July 2026
Accepted 14 September 2026

KEYWORDS

Maximum power point tracking (MPPT); Fuzzy logic control (FLC); photovoltaic systems (PV); dc–dc converter; sliding mode control; RL control

1. Introduction

Energy challenges associated with fossil-fuel depletion, environmental concerns, increasing demand, and continued economic growth have accelerated global interest in renewable-energy systems. Among the available technologies, photovoltaic (PV) systems have attracted considerable attention because of their wide solar availability, simple structure, low maintenance requirements, and ease of operation. However, the electrical output of PV systems is highly dependent on irradiance, temperature, and operating conditions (Chandola et al. 2026). The power–voltage (*P*–*V*) characteristic contains a maximum power point (MPP), whose location continuously varies under irradiance fluctuations and partial shading. Since PV installation costs remain relatively high, extracting the maximum available power is essential, creating a strong need for robust MPPT techniques capable of accurately identifying the global MPP under varying operating conditions (Reguieg et al. 2026).

Over recent years, several MPPT approaches have been developed to improve tracking speed, accuracy, stability, and convergence performance (Muñoz-Rodríguez et al. 2026; Xiang et al. 2026). The Incremental Conductance (INC) method offers improved control precision and relatively fast response (Zanib et al. 2022); however, its tracking process may become slower under rapidly changing environmental conditions (Khan et al. 2020). Constant Voltage Transfer (CVT) methods are simple to implement but often suffer from low efficiency and inaccurate MPP localisation (Mukhatov, Thao, and Do 2022). Artificial Neural Network (ANN)-based approaches can improve tracking accuracy and stability (Aly et al. 2021), although their performance strongly depends on training quality and dataset construction (Pires et al. 2023; Sahu and Dey 2022). Other techniques, including *P*&*O*, GA, ACO, and PSO, have also demonstrated acceptable MPPT performance but frequently involve high computational complexity, oscillatory behaviour, or extensive parameter tuning requirements (Maeng et al. 2024;

CONTACT Akram Bati  a.bati@bolton.ac.uk

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Article highlights

- A hybrid Q-learning-assisted fuzzy integral sliding-mode MPPT controller is proposed for photovoltaic boost-conversion systems.
- The developed QGFSMC framework integrates reinforcement learning, fuzzy logic adaptation, and sliding-mode control within a unified supervisory structure.
- The proposed controller demonstrates faster convergence, reduced oscillation, and improved transient stability under varying irradiance conditions.
- Comparative MATLAB/Simulink validation against P&O, INC, and MPC controllers confirms enhanced disturbance rejection and tracking performance.
- The developed controller achieved approximately 0.05 s startup rise time and nearly 0.02 s post-disturbance settling response.

Mamouri, Mesbahi, and Frick 2024). Hybrid approaches reported in (Değermenci, Yalman, and Olcay 2026; Jain et al. 2023; Yue et al. 2024) improve tracking performance under varying conditions; however, they may still exhibit oscillation, increased computational burden, and slower convergence toward the global MPP during irradiance variation and partial shading.

To address these limitations, this paper proposes a hybrid MPPT framework implemented in MATLAB/Simulink. The proposed controller integrates fuzzy logic, sliding-mode control (SMC), and Q-learning reinforcement learning to enhance tracking speed, convergence accuracy, disturbance rejection, and dynamic stability while maintaining a simple and low-cost control structure. The main contributions of this work are summarised as follows:

- Development of a novel Q-learning-assisted fuzzy integral sliding-mode MPPT controller (QGFSMC) for photovoltaic boost-conversion systems operating under dynamic environmental conditions.
- Integration of reinforcement learning, fuzzy adaptation, and sliding-mode control within a unified supervisory framework to improve transient response and tracking robustness.
- Introduction of an adaptive gain-scheduling mechanism to regulate fuzzy, sliding, and integral control actions according to the operating condition and tracking state.
- Comprehensive MATLAB/Simulink validation under varying irradiance and temperature conditions using P&O, INC, and MPC benchmark controllers for performance comparison.
- Demonstration of improved startup convergence, reduced oscillation, faster post-disturbance recovery, and more stable maximum power point tracking compared with conventional MPPT approaches.

2. Related work

Extensive research has been conducted in MPPT control for PV systems under varying irradiance and partial-shading conditions. Ullah et al. (2023) proposed a fuzzy-based MPPT controller using MATLAB/Simulink, where fuzzy logic was combined with P&O control to improve tracking under irradiance and temperature variations. The method demonstrated improved stability, reduced output-power fluctuation, and an average efficiency of approximately 97%. However, the work remained primarily simulation-based and lacked detailed validation under hardware implementation and partial-shading conditions. Melhaoui et al. (2025) developed a hybrid InC–FLC MPPT framework using SInC and CSI as fuzzy input variables. The method improved tracking under abrupt irradiance and load variations, achieving 97.7% MPPT efficiency and 53.5 ms convergence time. Nevertheless, the controller remained dependent on fuzzy membership-function design and was mainly validated through simulation studies. Hole and Goswami (2024) proposed an ensemble predictive MPPT framework integrating PI, PID, FLC, VARMA, LSTM-GRU prediction, bio-inspired optimisation, and a SEPIC converter. The framework demonstrated advanced predictive capability and proactive energy-management behaviour; however, its structure is highly complex and may require significant computational resources and tuning effort for practical implementation. Karimi, Siadatan, and Rezaei-Zare (2025) introduced a hybrid P&O–FLC method for PV systems under partial shading using both simulation and laboratory validation. The method improved search behaviour under non-uniform irradiance while maintaining relatively simple implementation compared with heavier AI-based approaches. However, controller performance remained dependent on

fuzzy-rule and membership-function tuning. Manna et al. (2023) proposed an MRAC-based MPPT controller for PV systems using a boost converter and compared it with *P&O* and *INC* approaches. The method achieved fast convergence of 3.6 ms with tracking efficiencies of 99.77% and 99.69% under irradiance and temperature variations. However, the controller was not validated under multi-peak partial-shading conditions. Girgis and Elkhateeb (2024) improved *P&O* MPPT using an adaptive PID controller with recursive least-squares gain updating. The approach improved dynamic voltage-reference tracking under changing irradiance, temperature, and load conditions while preserving the simplicity of conventional *P&O* control. Nevertheless, the validation remained simulation-based and relied on simplified PV-converter modelling. Kathe et al. (2023) reviewed conventional, intelligent, optimisation, and hybrid MPPT methods, highlighting that conventional approaches are simple and low-cost but often suffer from oscillation and weak partial-shading performance, whereas hybrid methods improve tracking capability at the expense of increased complexity and implementation cost. Nwachukwu, Folly, and Awodele (2025) investigated SAC and DDPG reinforcement-learning MPPT methods under partial shading. The study demonstrated that SAC reduced oscillation near the MPP because of entropy-based exploration and twin Q-network learning. However, deep reinforcement-learning approaches still require extensive training and careful simulation-to-hardware transfer. Mazumdar et al. (2024) reviewed conventional and AI-based MPPT approaches under uniform and partial-shading conditions. Their study showed that intelligent methods can improve global MPP tracking but frequently suffer from high computational complexity and implementation difficulty. The review emphasised the need for practical hybrid MPPT frameworks that combine low implementation cost with improved robustness and tracking performance under real operating conditions. Table 1 summarises representative intelligent, learning-based, and metaheuristic MPPT approaches, highlighting their operating principles, training requirements, and principal implementation considerations.

We can summarise the list of contribution as follows:

- A hierarchical MPPT framework that separates intelligent supervisory adaptation from continuous converter control.

Table 1. General comparison of intelligent, learning-based, and metaheuristic MPPT methods reported in the literature.

Method	Category	Operating principle	Training requirement	Main consideration
FLC	Intelligent control	Rule-based fuzzy inference	No	Membership-function and rule design
ANN	Machine learning	Learned nonlinear input–output mapping	Yes	Training-data dependency
ANFIS	Neuro-fuzzy learning	Adaptive fuzzy inference	Yes	Training and fuzzy-model complexity
LSTM	Deep learning	Recurrent sequence learning	Yes	Neural-network inference
BiLSTM	Deep learning	Bidirectional recurrent learning	Yes	Higher network complexity
RL	Reinforcement learning	State–action policy learning	Yes	Reward and state–action design
DQN	Deep reinforcement learning	Neural approximation of Q-values	Yes	Deep-network training and inference
DDPG	Deep reinforcement learning	Continuous actor–critic policy	Yes	Actor–critic network complexity
PSO	Metaheuristic	Population-based particle search	No	Iterative search process
GWO	Metaheuristic	Population-based hierarchical search	No	Iterative candidate evaluation
GA	Evolutionary	Population evolution and selection	No	Multiple iterative evaluations
ACO	Metaheuristic	Pheromone-based search	No	Iterative probabilistic search
DE	Evolutionary	Population-based differential search	No	Iterative candidate optimisation
CS	Metaheuristic	Cuckoo-search optimisation	No	Population-based iterative search
FPA	Metaheuristic	Pollination-inspired global/local search	No	Iterative optimisation
ANN–PSO	Hybrid AI–metaheuristic	Neural learning with swarm optimisation	Yes	Combined training and optimisation burden
FLC–PSO	Hybrid intelligent–metaheuristic	Fuzzy control with PSO optimisation	No*	Additional optimisation stage

Conventional supervised training is not required; however, PSO performs iterative optimisation of the selected controller variables. The table provides a qualitative literature-level comparison rather than a numerical performance ranking because published MPPT studies use different PV configurations, converters, environmental profiles, partial-shading patterns, and validation conditions.

- A state-dependent gain-scheduling strategy that improves MPP convergence while reducing oscillation near the optimum operating point.
- A compact learning-based supervisory policy that adapts controller aggressiveness without requiring online optimisation or neural-network inference.
- Improved transient response and post-disturbance recovery compared with *P&O*, *INC*, and *MPC* under identical operating conditions.

3. Methods used in validation process

3.1. *Perturb and Observe*

The Perturb and Observe (*P&O*) method operates by introducing a small perturbation into the PV system and then observing the resulting change in output power. This deliberate disturbance shifts the operating point slightly, and the subsequent variation in power is used to determine whether the search direction should be maintained or reversed (Jabbar et al. 2023). If the applied perturbation causes the extracted PV power to increase, the controller continues perturbing in the same direction. In contrast, if the power decreases, the perturbation direction is reversed so that the operating point moves back toward the maximum power point (MPP). Because this process is repeated continuously, the algorithm ultimately converges to the vicinity of the MPP and then oscillates around it in steady state. The magnitude of the perturbation therefore plays a key role: a smaller perturbation step reduces oscillation and improves tracking accuracy, whereas a larger step can accelerate convergence but may increase steady-state ripple (Ali et al. 2023; Necaibia et al. 2026). The instantaneous output power of the PV module is obtained from the measured voltage and current as

$$P_{PV}(k) = V_{PV}(k) I_{PV}(k) \quad (1)$$

where $V_{PV}(k)$ and $I_{PV}(k)$ are the sampled PV voltage and current at the k th instant. The algorithm evaluates the changes in power and voltage between two consecutive sampling instants as

$$\Delta P(k) = P_{PV}(k) - P_{PV}(k-1) \quad (2)$$

$$\Delta V(k) = V_{PV}(k) - V_{PV}(k-1) \quad (3)$$

Using these increments, the operating condition of the PV module relative to the MPP can be inferred. In a voltage-perturbation interpretation of *P&O*, the decision rule can be expressed as

$$\begin{cases} \Delta P(k) > 0 \Rightarrow \text{continue perturbation in the same direction} \\ \Delta P(k) < 0 \Rightarrow \text{reverse the perturbation direction} \\ \Delta P(k) = 0 \Rightarrow \text{the operating point is at or very near the MPP} \end{cases} \quad (4)$$

Equivalently, when the sign of both $\Delta P(k)$ and $\Delta V(k)$ is considered, the algorithm may be written as

$$\begin{cases} \Delta P(k) > 0, \Delta V(k) > 0 \Rightarrow \text{move toward a higher OV} \\ \Delta P(k) > 0, \Delta V(k) < 0 \Rightarrow \text{move toward a lower OV} \\ \Delta P(k) < 0, \Delta V(k) > 0 \Rightarrow \text{reverse toward a lower OV} \\ \Delta P(k) < 0, \Delta V(k) < 0 \Rightarrow \text{reverse toward a higher OV} \end{cases} \quad (5)$$

In converter-based MPPT implementation, the perturbation is commonly applied through the duty cycle of the DC–DC converter. For an ideal boost converter, the relationship between the PV-side voltage and the load-side voltage is given by (Deshi et al. 2026; Djilali et al. 2025)

$$V_L = \frac{V_{PV}}{1-D} \quad (6)$$

which can be rearranged to express the duty ratio as

$$D = 1 - \frac{V_{PV}}{V_L} \quad (7)$$

where D is the duty cycle, V_{PV} is the PV voltage, and V_L is the output or load voltage. The duty cycle is also defined in terms of the switch conduction intervals by

$$D = \frac{T_{on}}{T_{on} + T_{off}} \quad (8)$$

where T_{on} and T_{off} denote the switch on-time and off-time, respectively.

Accordingly, the duty cycle can be updated iteratively as

$$D(k) = D(k-1) + u(k) \Delta D \quad (9)$$

where ΔD is a small perturbation step and $u(k) \in \{+1, -1\}$ represents the perturbation direction. The direction term may be determined from the observed power variation as

$$u(k) = \begin{cases} u(k-1), & \Delta P(k) > 0 \\ -u(k-1), & \Delta P(k) < 0 \\ u(k-1), & \Delta P(k) = 0 \end{cases} \quad (10)$$

Thus, when the measured electrical variables indicate that the perturbation has moved the operating point toward a higher power level, the controller increases or decreases the duty cycle in the same direction so that the search point continues progressing toward the MPP. On the other hand, when the PV power decreases while the voltage changes in the opposite trend, the duty-cycle update is reversed, and the operating point is driven back toward the optimum region (Douifi et al. 2026; Xu et al. 2023).

3.2. Incremental conductance MPPT method

The Incremental Conductance (INC) method is a widely used maximum power point tracking (MPPT) technique for photovoltaic (PV) systems because it determines the operating point from the slope of the power-voltage characteristic (Çakmak, Aydoğmuş, and Tür 2025). Its fundamental principle is that the slope of the PV power curve is zero at the maximum power point (MPP), positive on the left side of the MPP, and negative on the right side. Unlike conventional perturbation-based methods, the INC algorithm can identify when the operating point has reached the MPP and can therefore reduce steady-state oscillation around the optimal point (Bakar Siddique et al. 2023; Ibrahim et al. 2023).

The instantaneous output power of the PV module is expressed as

$$P_{PV} = V_{PV} I_{PV} \quad (11)$$

where V_{PV} and I_{PV} are the PV voltage and current, respectively.

Differentiating (11) with respect to voltage gives

$$\frac{dP_{PV}}{dV_{PV}} = \frac{d(V_{PV} I_{PV})}{dV_{PV}} \quad (12)$$

Applying the product rule yields

$$\frac{dP_{PV}}{dV_{PV}} = I_{PV} + V_{PV} \frac{dI_{PV}}{dV_{PV}} \quad (13)$$

At the maximum power point,

$$\frac{dP_{PV}}{dV_{PV}} = 0 \quad (14)$$

Thus, from (13) and (14),

$$I_{PV} + V_{PV} \frac{dI_{PV}}{dV_{PV}} = 0 \quad (15)$$

which gives

$$\frac{dI_{PV}}{dV_{PV}} = -\frac{I_{PV}}{V_{PV}} \quad (16)$$

Equation (16) is the basis of the INC method. It shows that, at the MPP, the incremental conductance $\frac{dI}{dV}$ is equal in magnitude and opposite in sign to the instantaneous conductance $\frac{I}{V}$. Accordingly, the operating region of the PV module can be identified as

$$\frac{dP_{PV}}{dV_{PV}} > 0 \quad (\text{left of MPP}) \quad (17)$$

$$\frac{dP_{PV}}{dV_{PV}} = 0 \quad (\text{at MPP}) \quad (18)$$

$$\frac{dP_{PV}}{dV_{PV}} < 0 \quad (\text{right of MPP}) \quad (19)$$

Using (13), these conditions may also be written as

$$\frac{dI_{PV}}{dV_{PV}} > -\frac{I_{PV}}{V_{PV}} \quad (\text{left of MPP}) \quad (20)$$

$$\frac{dI_{PV}}{dV_{PV}} = -\frac{I_{PV}}{V_{PV}} \quad (\text{at MPP}) \quad (21)$$

$$\frac{dI_{PV}}{dV_{PV}} < -\frac{I_{PV}}{V_{PV}} \quad (\text{right of MPP}) \quad (22)$$

In digital implementation, the derivatives are approximated by finite differences between two consecutive sampling instants:

$$\Delta I(k) = I_{PV}(k) - I_{PV}(k-1) \quad (23)$$

$$\Delta V(k) = V_{PV}(k) - V_{PV}(k-1) \quad (24)$$

Hence, the incremental conductance is approximated by

$$\frac{dI_{PV}}{dV_{PV}} \approx \frac{\Delta I(k)}{\Delta V(k)} \quad (25)$$

The practical INC decision law becomes

$$\frac{\Delta I(k)}{\Delta V(k)} > -\frac{I_{PV}(k)}{V_{PV}(k)} \quad (\text{left of MPP}) \quad (26)$$

$$\frac{\Delta I(k)}{\Delta V(k)} = -\frac{I_{PV}(k)}{V_{PV}(k)} \quad (\text{at MPP}) \quad (27)$$

$$\frac{\Delta I(k)}{\Delta V(k)} < -\frac{I_{PV}(k)}{V_{PV}(k)} \quad (\text{right of MPP}) \quad (28)$$

Based on these conditions, the PV operating voltage reference can be updated as

$$V_{ref}(k+1) = \begin{cases} V_{ref}(k) + \Delta V_{step}, & \text{if } \frac{\Delta I(k)}{\Delta V(k)} > -\frac{I_{PV}(k)}{V_{PV}(k)} \\ V_{ref}(k), & \text{if } \frac{\Delta I(k)}{\Delta V(k)} = -\frac{I_{PV}(k)}{V_{PV}(k)} \\ V_{ref}(k) - \Delta V_{step}, & \text{if } \frac{\Delta I(k)}{\Delta V(k)} < -\frac{I_{PV}(k)}{V_{PV}(k)} \end{cases} \quad (29)$$

where ΔV_{step} is the selected voltage step size.

In practical implementation, a special case occurs when

$$\Delta V(k) = 0 \quad (30)$$

because direct division in (25) becomes undefined. In that case, the decision is made from the current variation only:

$$\Delta V(k) = 0, \quad \Delta I(k) = 0 \quad \Rightarrow \quad \text{at MPP} \quad (31)$$

$$\Delta V(k) = 0, \quad \Delta I(k) > 0 \quad \Rightarrow \quad \text{left of MPP} \quad (32)$$

$$\Delta V(k) = 0, \quad \Delta I(k) < 0 \quad \Rightarrow \quad \text{right of MPP} \quad (33)$$

For a PV system connected to an ideal boost converter, the voltage conversion relation is

$$V_o = \frac{V_{PV}}{1 - D} \quad (34)$$

or equivalently

$$D = 1 - \frac{V_{PV}}{V_o} \quad (35)$$

where V_o is the converter output voltage and D is the duty ratio.

Therefore, if duty-cycle control is used directly for MPPT, the INC duty update law may be expressed as

$$D(k+1) = \begin{cases} D(k) - \Delta D, & \text{if } \frac{\Delta I(k)}{\Delta V(k)} > -\frac{I_{PV}(k)}{V_{PV}(k)} \\ D(k), & \text{if } \frac{\Delta I(k)}{\Delta V(k)} = -\frac{I_{PV}(k)}{V_{PV}(k)} \\ D(k) + \Delta D, & \text{if } \frac{\Delta I(k)}{\Delta V(k)} < -\frac{I_{PV}(k)}{V_{PV}(k)} \end{cases} \quad (36)$$

where ΔD is the duty-cycle increment.

The main advantage of the INC method is that it can theoretically stop perturbing once the MPP is reached, thereby reducing the steady-state oscillation that is common in simpler tracking approaches. However, in practical implementation, its performance still depends on the chosen sampling interval and step size, since a large step improves convergence speed but may reduce accuracy, while a small step improves precision at the expense of slower dynamic response (Poothullil and Saleem 2026; Xiao, Dong, and He 2025).

3.3. Fuzzy logic control MPPT method

Fuzzy Logic Control (FLC) is widely used in photovoltaic (PV) maximum power point tracking (MPPT) because it can handle the nonlinear behaviour of the PV source without requiring an exact mathematical model of the plant. In MPPT applications, the fuzzy controller receives electrical information from the PV array, interprets the operating condition through a linguistic rule base, and generates a control action that updates the duty cycle of the DC-DC converter (Değermenci, Yalman, and Olcay 2026; Türkay and Yüksek 2025).

A common FLC-based MPPT structure consists of three main stages: fuzzification, inference, and defuzzification. In a widely used formulation, the controller has two inputs, namely the error $E(k)$ and the change in error $CE(k)$, and one output, which is the increment in duty cycle $\Delta D(k)$ (Hasheminasab, Alzayed, and Chaoui 2025; Hazzab et al. 2024).

The instantaneous PV power is first computed as

$$P_{PV}(k) = V_{PV}(k)I_{PV}(k) \quad (37)$$

where $V_{PV}(k)$ and $I_{PV}(k)$ are the PV voltage and current at sampling instant k .

The fuzzy input error is commonly defined as the slope of the power–voltage characteristic,

$$E(k) = \frac{P_{PV}(k) - P_{PV}(k-1)}{V_{PV}(k) - V_{PV}(k-1)} = \frac{\Delta P(k)}{\Delta V(k)} \quad (38)$$

where

$$\Delta P(k) = P_{PV}(k) - P_{PV}(k-1) \quad (39)$$

and

$$\Delta V(k) = V_{PV}(k) - V_{PV}(k-1) \quad (40)$$

The second fuzzy input is the change in error, defined by

$$CE(k) = E(k) - E(k-1) \quad (41)$$

The physical meaning of these quantities is important. The error $E(k)$ represents the slope of the PV power curve. At the maximum power point (MPP),

$$\frac{dP_{PV}}{dV_{PV}} = 0 \quad (42)$$

and therefore, in the discrete implementation,

$$E(k) \approx 0 \quad (43)$$

Thus, $E(k)$ indicates whether the operating point is located to the left or right of the MPP, while $CE(k)$ indicates the direction of motion of the operating point between two consecutive samples.

In practical digital implementation, the fuzzy inputs are often normalised before fuzzification:

$$e(k) = K_e E(k) \quad (44)$$

$$ce(k) = K_{ce} CE(k) \quad (45)$$

where K_e and K_{ce} are input scaling factors selected so that the normalised variables lie inside the universe of discourse of the membership functions.

The fuzzification stage maps the crisp inputs $e(k)$ and $ce(k)$ into fuzzy sets. A common choice is to use triangular membership functions because of their simplicity and ease of implementation. For a triangular membership function with parameters $[a, b, c]$, the degree of membership is

$$\mu(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c \end{cases} \quad (46)$$

where a and c define the feet of the triangle and b defines its peak.

For MPPT design, the normalised inputs and output are typically partitioned into linguistic terms such as

$$\{NB, NS, ZE, PS, PB\}$$

representing negative big, negative small, zero, positive small, and positive big, respectively. Accordingly, the fuzzy sets for $e(k)$, $ce(k)$, and the normalised output $d(k)$ may be written as

$$e(k) \in \{A_1, A_2, \dots, A_m\}, \quad ce(k) \in \{B_1, B_2, \dots, B_n\} \quad (47)$$

$$d(k) \in \{C_1, C_2, \dots, C_r\} \quad (48)$$

where m , n , and r are the numbers of fuzzy subsets used for each variable.

The inference engine evaluates a rule base of the form

$$R_{ij} : \text{IF } e(k) \text{ is } A_i \text{ AND } ce(k) \text{ is } B_j \text{ THEN } d(k) \text{ is } C_{ij} \quad (49)$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.

For a Mamdani-type fuzzy controller using the minimum operator for conjunction, the firing strength of the ij -th rule is

$$w_{ij}(k) = \min(\mu_{A_i}(e(k)), \mu_{B_j}(ce(k))) \quad (50)$$

The output fuzzy set contributed by the ij -th rule is then obtained by implication:

$$\mu_{ij}^{out}(z) = \min(w_{ij}(k), \mu_{C_{ij}}(z)) \quad (51)$$

where z is the output universe variable.

After all rules are evaluated, the individual rule outputs are aggregated into a single fuzzy output set. For max-aggregation,

$$\mu_{agg}(z) = \max_{i,j} \mu_{ij}^{out}(z) \quad (52)$$

The defuzzification stage converts the aggregate fuzzy output into a crisp control signal. The centroid method is commonly used, for which the normalised crisp output is

$$d^*(k) = \frac{\int_{\Omega} z \mu_{agg}(z) dz}{\int_{\Omega} \mu_{agg}(z) dz} \quad (53)$$

where Ω is the output universe of discourse.

In discrete implementation, the centroid can also be written as

$$d^*(k) = \frac{\sum_{\ell=1}^N z_{\ell} \mu_{agg}(z_{\ell})}{\sum_{\ell=1}^N \mu_{agg}(z_{\ell})} \quad (54)$$

where z_{ℓ} are the discrete output points.

The duty-cycle increment is then obtained by output denormalization:

$$\Delta D(k) = K_d d^*(k) \quad (55)$$

where K_d is the output scaling factor.

Finally, the converter duty cycle is updated as

$$D(k) = D(k-1) + \Delta D(k) \quad (56)$$

In practice, the duty cycle is limited to the admissible switching range:

$$D(k) = \text{sat}(D(k-1) + \Delta D(k)), \quad D_{\min} \leq D(k) \leq D_{\max} \quad (57)$$

When the PV system is connected through a boost converter, the ideal voltage conversion relation is

$$V_o = \frac{V_{PV}}{1 - D} \quad (58)$$

or equivalently

$$V_{PV} = V_o(1 - D) \quad (59)$$

where V_o is the converter output voltage. Therefore, by adjusting the duty ratio D , the fuzzy controller shifts the PV operating voltage and drives the operating point toward the MPP.

It is worth noting that some FLC-based MPPT implementations use ΔP and ΔV directly as the fuzzy inputs instead of E and CE . In that case, the controller input vector becomes

$$\mathbf{x}(k) = \begin{bmatrix} \Delta P(k) \\ \Delta V(k) \end{bmatrix} \quad (60)$$

and the same fuzzification–inference–defuzzification procedure is followed to obtain $\Delta D(k)$. Both formulations are valid, but the $E - CE$ representation is especially attractive because $E(k)$ is directly related to the slope of the $P - V$ characteristic and therefore to the MPP condition.

The main advantage of FLC-based MPPT is that it can provide adaptive duty-cycle variation and reduced steady-state oscillation without requiring an exact model of the PV source. Its performance, however, depends on the appropriate design of the membership functions, scaling factors, and rule base (Aiboudi et al. 2025).

4. Proposed system modelling and design

This section presents the modelling and design of the proposed hybrid MPPT framework implemented in MATLAB/Simulink. The developed controller combines a fitted nonlinear photovoltaic (PV) array model, a continuous-conduction-mode (CCM) averaged boost converter, a five-set fuzzy logic controller (FLC), an integral sliding-mode correction term, and a Q-learning layer that adaptively schedules the controller aggressiveness. The complete control loop is executed in discrete time with measurement averaging, while the plant is simulated using the averaged continuous-time state model.

4.1. PV array modelling

The PV array is formed using $N_s = 5$ series-connected modules per string and $N_p = 10$ parallel strings. The module-level datasheet values used in the implementation are summarised in Table 8. The aggregate array open-circuit voltage, short-circuit current, nominal maximum-power voltage, and nominal maximum-power current are given by

$$V_{oc} = N_s V_{oc,m} \quad (61)$$

$$I_{sc} = N_p I_{sc,m} \quad (62)$$

$$V_{mp} = N_s V_{mp,m} \quad (63)$$

$$I_{mp} = N_p I_{mp,m} \quad (64)$$

where the subscript m denotes a single PV module.

Instead of using the full single-diode equation, the implemented code adopts a fitted exponential current–voltage model constrained by the datasheet points:

$$I_{pv}(V) = I_{sc,eff} \left[1 - C_1 \left(\exp \left(\frac{V/V_{oc,eff}}{C_2} \right) - 1 \right) \right], \quad (65)$$

with saturation applied such that

$$0 \leq V \leq V_{oc,eff}, \quad I_{pv}(V) \geq 0. \quad (66)$$

The coefficients C_1 and C_2 are identified from the normalised datasheet ratios

$$r_V = \frac{V_{mp,m}}{V_{oc,m}}, \quad r_I = \frac{I_{mp,m}}{I_{sc,m}}. \quad (67)$$

The parameter C_2 is obtained numerically from

$$1 - \frac{\exp(r_V/C_2) - 1}{\exp(1/C_2) - 1} - r_I = 0, \quad (68)$$

and then

$$C_1 = \frac{1}{\exp(1/C_2) - 1}. \quad (69)$$

To account for operating-condition changes, the effective open-circuit voltage and short-circuit current are updated with irradiance G and temperature T as

$$V_{oc,eff} = V_{oc} [1 + T_{c,Voc} (T - 25)], \quad (70)$$

$$I_{sc,eff} = I_{sc} \left(\frac{G}{1000} \right) [1 + T_{c,Isc} (T - 25)]. \quad (71)$$

The normalised power function used in the code to locate the theoretical MPP is

$$p_n(x) = x \max \left(1 - C_1 \left(\exp \left(\frac{x}{C_2} \right) - 1 \right), 0 \right), \quad x \in [0, 1]. \quad (72)$$

The optimal normalised MPP location is then

$$x_{mpp} = \arg \max_{x \in [0,1]} p_n(x). \quad (73)$$

Hence, at each operating condition,

$$V_{mpp,eff} = x_{mpp} V_{oc,eff}, \quad (74)$$

$$I_{mpp,eff} = I_{pv}(V_{mpp,eff}), \quad (75)$$

$$P_{mpp,eff} = V_{mpp,eff} I_{mpp,eff}. \quad (76)$$

The imposed irradiance–temperature profile is piecewise defined as

$$G(t) = \begin{cases} 1000 \text{ W/m}^2, & 0 \leq t < 1.0 \text{ s} \\ 1500 \text{ W/m}^2, & 1.0 \leq t < 1.5 \text{ s} \\ 2000 \text{ W/m}^2, & 1.5 \leq t \leq 2.0 \text{ s} \end{cases} \quad (77)$$

and

$$T(t) = \begin{cases} 25^\circ\text{C}, & 0 \leq t < 1.0 \text{ s} \\ 35^\circ\text{C}, & 1.0 \leq t < 1.5 \text{ s} \\ 45^\circ\text{C}, & 1.5 \leq t \leq 2.0 \text{ s}. \end{cases} \quad (78)$$

4.2. Averaged boost converter model

The PV array feeds a DC–DC boost converter represented by a large-signal averaged CCM model. The state variables are the PV-side capacitor voltage V_{pv} , the inductor current I_L , and the output capacitor voltage V_{out} . With duty cycle D , input capacitor C_{in} , output capacitor C_{out} , inductance L , load resistance R , and inductor parasitic resistance r_L , the plant equations implemented in the code are

$$C_{in} \frac{dV_{pv}}{dt} = I_{pv}(V_{pv}) - I_L, \quad (79)$$

$$L \frac{dI_L}{dt} = V_{pv} - (1 - D)V_{out} - r_L I_L, \quad (80)$$

$$C_{out} \frac{dV_{out}}{dt} = (1 - D)I_L - \frac{V_{out}}{R}. \quad (81)$$

The model is numerically integrated using a fixed simulation step dt , yielding

$$V_{pv}(k + 1) = \max(V_{pv}(k) + \dot{V}_{pv}(k) dt, 0), \quad (82)$$

$$I_L(k + 1) = \max(I_L(k) + \dot{I}_L(k) dt, 0), \quad (83)$$

$$V_{out}(k + 1) = \max(V_{out}(k) + \dot{V}_{out}(k) dt, 0). \quad (84)$$

The converter and simulation parameters are given in Table 2.

At each controller update, the measured PV voltage, current, and power are averaged over the preceding N_{ctrl} simulation samples:

$$\begin{aligned} \bar{V}_{pv}(k) &= \frac{1}{N_m} \sum_{j=1}^{N_m} V_{pv}(j), \\ \bar{I}_{pv}(k) &= \frac{1}{N_m} \sum_{j=1}^{N_m} I_{pv}(j), \\ \bar{P}_{pv}(k) &= \frac{1}{N_m} \sum_{j=1}^{N_m} P_{pv}(j). \end{aligned} \quad (85)$$

where N_m is the current measurement count within the control window.

4.3. Architecture of the proposed QGFSMC MPPT controller

The proposed controller is the third controller in the simulation framework and is denoted as QGFSMC. It is composed of four coordinated layers:

1. a filtered MPP voltage reference obtained from the fitted PV model,
2. a fuzzy logic duty-cycle correction,
3. an integral sliding-mode correction,
4. a Q-learning supervisor that schedules controller aggressiveness online.

Table 2. Simulation and converter parameters.

Parameter	Symbol	Value
Simulation step	dt	$5 \times 10^{-5} \text{s}$
Simulation duration	t_{end}	2 s
Controller update period	T_{ctrl}	$1 \times 10^{-3} \text{s}$
Controller steps/update	N_{ctrl}	20
Inductance	L	2 mH
Input capacitance	C_{in}	470 μF
Output capacitance	C_{out}	470 μF
Load resistance	R	20 Ω
Inductor resistance	r_L	0 Ω
Duty-cycle limits	D	$0.05 \leq D \leq 0.95$
Training episodes	N_{train}	50
Final evaluation episodes	–	1

Unlike the benchmark *P&O* and *INC* controllers, which directly update the duty cycle from measured ΔP , ΔV , ΔI , and conductance information, the proposed controller regulates the PV voltage toward a filtered estimate of the theoretical MPP voltage and uses hybrid intelligent/robust compensation to generate the duty-cycle increment.

4.3.1. Reference and signal filtering

The MPP voltage reference used by the proposed controller is low-pass filtered as

$$V_{ref}(k) = 0.70 V_{ref}(k-1) + 0.30 \bar{V}_{mpp}(k), \quad (86)$$

where $\bar{V}_{mpp}(k)$ is the averaged theoretical MPP voltage over the current control window. The measured PV voltage is also filtered:

$$V_{pv,f}(k) = 0.55 V_{pv,f}(k-1) + 0.45 \bar{V}_{pv}(k). \quad (87)$$

The voltage tracking error is then defined as

$$e(k) = V_{ref}(k) - V_{pv,f}(k). \quad (88)$$

The raw derivative of the error is

$$\dot{e}_{raw}(k) = \frac{e(k) - e(k-1)}{T_{ctrl}}, \quad (89)$$

and the filtered derivative is

$$\dot{e}_f(k) = 0.45 \dot{e}_f(k-1) + 0.55 \dot{e}_{raw}(k). \quad (90)$$

The integral error state is updated as

$$e_i(k) = 0.99 e_i(k-1) + e(k) T_{ctrl}, \quad (91)$$

with saturation

$$-0.15 \max(V_{ref}(k), \varepsilon) \leq e_i(k) \leq 0.15 \max(V_{ref}(k), \varepsilon), \quad (92)$$

where ε is a small positive number used to avoid division by zero.

4.3.2. Normalisation

The controller operates on normalised signals clipped to $[-1, 1]$:

$$e_n(k) = \text{sat}_{[-1,1]} \left(\frac{e(k)}{\max(0.08 V_{oc,eff}, \varepsilon)} \right), \quad (93)$$

$$de_n(k) = \text{sat}_{[-1,1]} \left(\frac{\dot{e}_f(k)}{\max(0.12 V_{oc,eff} / T_{ctrl}, \varepsilon)} \right), \quad (94)$$

$$ei_n(k) = \text{sat}_{[-1,1]} \left(\frac{e_i(k)}{\max(0.15 V_{oc,eff}, \varepsilon)} \right). \quad (95)$$

4.4. Fuzzy logic layer

The fuzzy stage takes the normalised error e_n and normalised error change de_n as its two inputs and generates a normalised correction u_f . The code defines five membership functions for each input over the interval $[-1, 1]$. Since the uploaded implementation does not explicitly assign linguistic names, the ordered sets are denoted here as

Table 3. Five membership functions used for each fuzzy input.

Label	Type	Parameters	Support/transition range
NB	Left shoulder	(−1.0, −0.5)	(−∞, −0.5] with full membership for $x \leq -1.0$
NS	Triangular	(−1.0, −0.5, 0.0)	[−1.0, 0.0]
ZE	Triangular	(−0.5, 0.0, 0.5)	[−0.5, 0.5]
PS	Triangular	(0.0, 0.5, 1.0)	[0.0, 1.0]
PB	Right shoulder	(0.5, 1.0)	[0.5, ∞) with full membership for $x \geq 1.0$

$$\{NB, NS, ZE, PS, PB\},$$

which correspond to the left shoulder, two intermediate triangles, the zero-centred triangle, the positive triangle, and the right shoulder, respectively.

The triangular membership function used in the code is

$$\mu_{tri}(x; a, b, c) = \begin{cases} 0, & x \leq a \text{ or } x \geq c, \\ \frac{x-a}{b-a}, & a < x < b, \\ 1, & x = b, \\ \frac{c-x}{c-b}, & b < x < c. \end{cases} \quad (96)$$

The left-shoulder membership function is

$$\mu_{LS}(x; a, b) = \begin{cases} 1, & x \leq a, \\ \frac{b-x}{b-a}, & a < x < b, \\ 0, & x \geq b, \end{cases} \quad (97)$$

and the right-shoulder function is

$$\mu_{RS}(x; a, b) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x < b, \\ 1, & x \geq b. \end{cases} \quad (98)$$

The membership functions used for both e_n and de_n are listed in Table 3.

For the i -th fuzzy set of e_n and the j -th fuzzy set of de_n , the rule firing strength is computed by product inference:

$$w_{ij}(k) = \mu_i(e_n(k))\mu_j(de_n(k)), \quad i, j \in \{1, \dots, 5\}. \quad (99)$$

The implemented fuzzy controller is a weighted-average singleton system. Let r_{ij} denote the singleton consequent associated with rule (i, j) . The crisp fuzzy output is

$$u_f(k) = \frac{\sum_{i=1}^5 \sum_{j=1}^5 w_{ij}(k) r_{ij}}{\sum_{i=1}^5 \sum_{j=1}^5 w_{ij}(k)}. \quad (100)$$

If the denominator becomes numerically too small, the code sets $u_f(k) = 0$.

The 25 singleton consequents used in the controller are given in Table 4. These are exact numeric values taken from the MATLAB implementation.

The raw fuzzy output is further smoothed before entering the duty-cycle law:

$$u_{f,f}(k) = 0.45 u_{f,f}(k-1) + 0.55 u_f(k). \quad (101)$$

4.5. Integral sliding-mode layer

The proposed robust correction is based on a normalised integral sliding surface defined as

$$s(k) = e_n(k) + 0.20 de_n(k) + 0.10 ei_n(k). \quad (102)$$

Table 4. Singleton consequents of the fuzzy controller.

$e_n \backslash de_n$	NB	NS	ZE	PS	PB
NB	-0.70	-0.55	-0.35	-0.15	0.00
NS	-0.55	-0.35	-0.15	0.00	0.15
ZE	-0.35	-0.15	0.00	0.15	0.35
PS	-0.15	0.00	0.15	0.35	0.55
PB	0.00	0.15	0.35	0.55	0.70

This choice combines proportional, derivative-like, and integral error information into a single sliding variable.

A distance-to-MPP weighting factor is defined as

$$w_{far}(k) = \min\left(\frac{|e_n(k)|}{0.06}, 1\right), \quad (103)$$

which makes the controller more aggressive when the operating point is far from the MPP and softer when the operating point approaches the optimum region.

The gains are scheduled according to

$$K_f(k) = (1.1 \times 10^{-3} + 3.5 \times 10^{-3} w_{far}(k)) m_{K_f}, \quad (104)$$

$$K_s(k) = (1.2 \times 10^{-4} + 4.5 \times 10^{-4} w_{far}(k)) m_{K_s}, \quad (105)$$

$$K_i(k) = (4.0 \times 10^{-5}) m_{K_i}, \quad (106)$$

where m_{K_f} , m_{K_s} , and m_{K_i} are action-dependent gain multipliers selected by the Q-learning supervisor. The three duty-cycle correction components are then defined as

$$\Delta D_f(k) = -K_f(k) u_{f,f}(k), \quad (107)$$

$$\Delta D_s(k) = -K_s(k) \tanh(2.6 s(k)), \quad (108)$$

$$\Delta D_i(k) = -K_i(k) e_i(k). \quad (109)$$

Thus, the unsmoothed composite duty-cycle increment is

$$\Delta D_{raw}(k) = \Delta D_f(k) + \Delta D_s(k) + \Delta D_i(k). \quad (110)$$

The sign convention in the code is important: when $e > 0$, the measured PV voltage is below the estimated MPP voltage, so the boost-converter duty cycle must be decreased to raise V_{pv} . This is why all three terms in (107)–(109) enter with negative sign.

To limit oscillation, the raw duty-cycle increment is filtered as

$$\alpha_d(k) = (0.82 - 0.62 w_{far}(k)) + \Delta\alpha, \quad (111)$$

with clipping

$$\alpha_d(k) = \text{sat}_{[0.05, 0.95]}(\alpha_d(k)), \quad (112)$$

where $\Delta\alpha$ is the action-dependent smoothing shift selected by Q-learning.

The filtered duty-cycle command is

$$\Delta D_{cmd}(k) = \alpha_d(k) \Delta D_{cmd}(k-1) + (1 - \alpha_d(k)) \Delta D_{raw}(k). \quad (113)$$

A variable duty-increment bound is applied:

$$\Delta D_{lim}(k) = (2.0 \times 10^{-4} + 2.8 \times 10^{-3} w_{far}(k)) m_{Dlim}, \quad (114)$$

and the final command is saturated as

$$\Delta D_{cmd}(k) = \text{sat}_{[-\Delta D_{lim}(k), \Delta D_{lim}(k)]}(\Delta D_{cmd}(k)). \quad (115)$$

Finally, the converter duty cycle is updated according to

$$D(k) = \text{sat}_{[0.05, 0.95]}(D(k-1) + \Delta D_{cmd}(k)). \quad (116)$$

4.6. Q-learning supervisory layer

The supervisory reinforcement-learning layer does not directly output the duty cycle. Instead, it selects one of three controller modes—aggressive, balanced, or soft—that adapt the gains and smoothing terms of the hybrid FLC–SMC controller. Hence, Q-learning acts as a gain scheduler over the lower-level control law.

4.6.1. State construction

The Q-learning state is constructed from the magnitudes of the normalised error and normalised error derivative:

$$a_e(k) = |e_n(k)|, \quad a_{de}(k) = |de_n(k)|. \quad (117)$$

The magnitude of e_n is quantised into three bins:

$$b_e(k) = \begin{cases} 1, & a_e(k) < 0.02, \\ 2, & 0.02 \leq a_e(k) < 0.06, \\ 3, & a_e(k) \geq 0.06, \end{cases} \quad (118)$$

while the magnitude of de_n is quantised into two bins:

$$b_{de}(k) = \begin{cases} 1, & a_{de}(k) < 0.03, \\ 2, & a_{de}(k) \geq 0.03. \end{cases} \quad (119)$$

The scalar state index is then

$$s_{QL}(k) = (b_e(k) - 1)n_{de} + b_{de}(k), \quad n_{de} = 2, \quad (120)$$

which yields 6 discrete states in total. Table 5 shows the Q states used.

4.6.2. Action space

The available actions are

$$\mathcal{A} = \{1, 2, 3\} = \{\text{aggressive}, \text{balanced}, \text{soft}\}.$$

Each action selects a different set of multipliers for the lower-level controller, as given in Table 6.

4.6.3. Reward and Q-table update

The reward is formulated to penalise voltage-tracking error, error variation, integral accumulation, and excessive duty-cycle activity:

$$r(k) = -[14.0 |e_n(k)| + 1.5 |de_n(k)| + 0.3 |ei_n(k)| + 60.0 |\Delta D_{cmd}(k-1)|]. \quad (121)$$

Table 5. Q-learning state definition used in the proposed controller.

State index	$ e_n $ bin	$ de_n $ bin	Interpretation
1	<0.02	<0.03	near-calm
2	<0.02	≥ 0.03	near-changing
3	$0.02 \leq e_n < 0.06$	<0.03	medium-calm
4	$0.02 \leq e_n < 0.06$	≥ 0.03	medium-changing
5	≥ 0.06	<0.03	far-calm
6	≥ 0.06	≥ 0.03	far-changing

Table 6. Q-learning actions and scheduled controller multipliers.

Action	m_{K_f}	m_{K_s}	m_{K_i}	Δa	m_{Dlim}
Aggressive	3.20	2.80	1.50	-0.35	3.20
Balanced	1.35	1.20	1.10	-0.10	1.40
Soft	0.35	0.45	0.70	0.20	0.35

This reward encourages fast convergence to the MPP with low oscillation and limited duty-cycle chattering.

During training, the Q-table is updated using

$$Q(s_{k-1}, a_{k-1}) \leftarrow Q(s_{k-1}, a_{k-1}) + \alpha_{QL} [r(k) + \gamma_{QL} \max_a Q(s_k, a) - Q(s_{k-1}, a_{k-1})]. \quad (122)$$

where the learning rate and discount factor are

$$\alpha_{QL} = 0.35, \quad \gamma_{QL} = 0.95. \quad (123)$$

The code uses an ε -greedy policy during training. For training episode ep , the exploration probability is

$$\varepsilon(ep) = \max\left(\varepsilon_{min}, \varepsilon_0 \left[1 - \frac{ep - 1}{\max(N_{train} - 1, 1)}\right]\right), \quad (124)$$

with

$$\varepsilon_0 = 0.25, \quad \varepsilon_{min} = 0.02. \quad (125)$$

During the final evaluation episode,

$$\varepsilon = 0, \quad (126)$$

so the learned greedy policy is applied.

The initial Q-table is not zero-uniform; it is biased so that near states initially prefer softer control and far states initially prefer aggressive control:

$$Q_0 = \begin{bmatrix} 0 & 0.2 & 0.5 \\ 0 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.1 \\ 0.4 & 0.3 & 0.0 \\ 1.8 & 0.4 & 0.0 \\ 2.2 & 0.3 & 0.0 \end{bmatrix}. \quad (127)$$

4.7. Overall duty-cycle synthesis

Combining the preceding stages, the proposed QGFSMC algorithm can be summarised as follows:

Algorithm 1. Proposed QGFSMC MPPT control procedure.

Require: Irradiance $G(t)$, temperature $T(t)$, measured PV signals

Ensure: Updated converter duty cycle $D(k)$

- 1: Update the operating-condition-dependent PV model using $G(t)$ and $T(t)$
 - 2: Compute the theoretical MPP voltage $V_{mpp,eff}$ from the fitted PV model
 - 3: Filter the MPP reference and the measured PV voltage using (86) and (87)
 - 4: Generate the normalised tracking signals e_n , de_n , and ei_n
 - 5: Construct the Q-learning state from $|e_n|$ and $|de_n|$
 - 6: Select the control action and schedule the controller gains
 - 7: Compute the fuzzy correction u_f using (100)
 - 8: Form the sliding surface and integral correction terms using (102)–(110)
 - 9: Smooth and saturate the duty-cycle increment using (113)–(115)
 - 10: Update the boost-converter duty cycle using (116)
 - 11: **Return:** $D(k)$
-

Therefore, the proposed controller is not a pure Q-learning MPPT law and not a conventional standalone FLC or SMC either. Rather, it is a hierarchical hybrid controller in which Q-learning supervises the aggressiveness of an adaptive fuzzy–integral sliding-mode duty-cycle law, while the PV operating reference is continuously refreshed from the fitted nonlinear array model. This structure is intended to improve rise-time behaviour, reduce oscillation near the MPP, and preserve robustness under abrupt irradiance and temperature changes.

4.8. Hybrid controller design of the proposed QGFSMC scheme

The proposed controller, denoted as QGFSMC, is a hierarchical hybrid MPPT strategy that combines a filtered voltage-reference tracker, a fuzzy logic controller (FLC), an integral sliding-mode correction term, and a Q-learning-based supervisory gain scheduler. The objective of the lower control layer is to regulate the PV-side operating voltage toward the instantaneous maximum-power-point voltage while maintaining bounded duty-cycle variation and reducing oscillatory behaviour under abrupt irradiance and temperature changes.

The PV array is interfaced with the DC–DC boost converter described by the large-signal averaged continuous-conduction-mode model

$$C_{in} \frac{dV_{pv}}{dt} = I_{pv}(V_{pv}) - I_L \quad (128)$$

$$L \frac{dI_L}{dt} = V_{pv} - (1 - D)V_{out} - r_L I_L \quad (129)$$

$$C_{out} \frac{dV_{out}}{dt} = (1 - D)I_L - \frac{V_{out}}{R} \quad (130)$$

where V_{pv} , I_L , and V_{out} are the PV voltage, inductor current, and output voltage, respectively, and D is the converter duty cycle.

The PV current is generated by the fitted nonlinear exponential model

$$I_{pv}(V) = I_{sc,eff} \left[1 - C_1 \left(\exp \left(\frac{V/V_{oc,eff}}{C_2} \right) - 1 \right) \right] \quad (131)$$

subject to the clamping constraints

$$0 \leq V \leq V_{oc,eff}, \quad I_{pv}(V) \geq 0. \quad (132)$$

The effective open-circuit voltage and short-circuit current are updated from irradiance G and temperature T as

$$V_{oc,eff} = V_{oc}(1 + T_{c,Voc}(T - 25)) \quad (133)$$

$$I_{sc,eff} = I_{sc} \left(\frac{G}{1000} \right) (1 + T_{c,Isc}(T - 25)). \quad (134)$$

The theoretical MPP voltage is obtained from the normalised PV model through

$$V_{mpp,eff} = x_{mpp} V_{oc,eff} \quad (135)$$

where x_{mpp} is the normalised optimum point obtained offline from the fitted PV characteristic.

To reduce abrupt reference variations, the MPP voltage reference is low-pass filtered as

$$V_{ref}(k) = 0.70 V_{ref}(k - 1) + 0.30 V_{mpp,avg}(k) \quad (136)$$

while the measured PV voltage is filtered according to

$$V_{pv,f}(k) = 0.55 V_{pv,f}(k-1) + 0.45 V_{pv,avg}(k). \quad (137)$$

The tracking error is then defined by

$$e(k) = V_{ref}(k) - V_{pv,f}(k) \quad (138)$$

and the filtered error derivative is computed as

$$\dot{e}_{raw}(k) = \frac{e(k) - e(k-1)}{T_{ctrl}} \quad (139)$$

$$\dot{e}_f(k) = 0.45 \dot{e}_f(k-1) + 0.55 \dot{e}_{raw}(k). \quad (140)$$

An integral error state is also introduced:

$$e_i(k) = 0.99 e_i(k-1) + e(k) T_{ctrl} \quad (141)$$

with saturation

$$\begin{aligned} -0.15 \max(V_{ref}(k), 10^{-6}) &\leq e_i(k) \\ &\leq 0.15 \max(V_{ref}(k), 10^{-6}). \end{aligned} \quad (142)$$

The error signals are normalised to the interval $[-1, 1]$ as

$$e_n(k) = \text{sat}_{[-1,1]} \left(\frac{e(k)}{\max(0.08 V_{oc,eff}, 10^{-6})} \right) \quad (143)$$

$$de_n(k) = \text{sat}_{[-1,1]} \left(\frac{\dot{e}_f(k)}{\max(0.12 V_{oc,eff} / T_{ctrl}, 10^{-6})} \right) \quad (144)$$

$$ei_n(k) = \text{sat}_{[-1,1]} \left(\frac{e_i(k)}{\max(0.15 V_{oc,eff}, 10^{-6})} \right). \quad (145)$$

The fuzzy controller uses e_n and de_n as its two inputs and produces the fuzzy correction u_f . Five membership functions are used for each input: one left shoulder, three triangular sets, and one right shoulder. The triangular membership function is given by

$$\mu_{tri}(x; a, b, c) = \begin{cases} 0, & x \leq a \text{ or } x \geq c \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x = b \\ \frac{c-x}{c-b}, & b < x < c \end{cases} \quad (146)$$

while the left- and right-shoulder functions are

$$\mu_{LS}(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases} \quad (147)$$

$$\mu_{RS}(x; a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b. \end{cases} \quad (148)$$

The fuzzy firing strength of the (i, j) -th rule is

$$w_{ij}(k) = \mu_i(e_n(k))\mu_j(de_n(k)) \quad (149)$$

and the crisp fuzzy output is generated by weighted averaging:

$$u_f(k) = \frac{\sum_{i=1}^5 \sum_{j=1}^5 w_{ij}(k) r_{ij}}{\sum_{i=1}^5 \sum_{j=1}^5 w_{ij}(k)} \quad (150)$$

where r_{ij} is the singleton consequent in the rule table. The fuzzy output is further smoothed as

$$u_{f,f}(k) = 0.45 u_{f,f}(k-1) + 0.55 u_f(k). \quad (151)$$

The integral sliding surface used by the robust correction term is defined as

$$s(k) = e_n(k) + 0.20 de_n(k) + 0.10 ei_n(k). \quad (152)$$

To make the controller stronger when the operating point is far from the MPP, a distance factor is defined as

$$w_{far}(k) = \min\left(\frac{|e_n(k)|}{0.06}, 1\right). \quad (153)$$

The gain-scheduled lower-layer control gains are

$$K_f(k) = (1.1 \times 10^{-3} + 3.5 \times 10^{-3} w_{far}(k)) m_{K_f} \quad (154)$$

$$K_s(k) = (1.2 \times 10^{-4} + 4.5 \times 10^{-4} w_{far}(k)) m_{K_s} \quad (155)$$

$$K_i(k) = 4.0 \times 10^{-5} m_{K_i}. \quad (156)$$

The fuzzy, sliding, and integral duty-cycle corrections are then

$$\Delta D_{fuzzy}(k) = -K_f(k) u_{f,f}(k) \quad (157)$$

$$\Delta D_{smc}(k) = -K_s(k) \tanh(2.6 s(k)) \quad (158)$$

$$\Delta D_{int}(k) = -K_i(k) ei_n(k) \quad (159)$$

and the total raw duty-cycle correction is

$$\Delta D_{raw}(k) = \Delta D_{fuzzy}(k) + \Delta D_{smc}(k) + \Delta D_{int}(k). \quad (160)$$

To suppress chattering near the MPP, the duty-cycle command is filtered using

$$\alpha_d(k) = \text{sat}_{[0.05, 0.95]}(0.82 - 0.62 w_{far}(k) + \Delta\alpha) \quad (161)$$

and

$$\Delta D_{cmd}(k) = \alpha_d(k) \Delta D_{prev}(k) + (1 - \alpha_d(k)) \Delta D_{raw}(k). \quad (162)$$

The duty-cycle increment is bounded by

$$\Delta D_{lim}(k) = (2.0 \times 10^{-4} + 2.8 \times 10^{-3} w_{far}(k)) m_{Dlim} \quad (163)$$

and the final limited increment is

$$\Delta D_{cmd}(k) = \text{sat}_{[-\Delta D_{lim}(k), \Delta D_{lim}(k)]}(\Delta D_{cmd}(k)). \quad (164)$$

The boost duty cycle is updated according to

$$D(k) = \text{sat}_{[0.05, 0.95]} (D(k-1) + \Delta D_{cmd}(k)). \quad (165)$$

4.9. Objective function, Q-learning operation, and supervisory role

The Q-learning layer does not directly generate the duty cycle. Instead, it acts as a supervisory decision mechanism that selects one of three operating modes for the lower-level fuzzy-integral sliding controller: aggressive, balanced, or soft. In this way, reinforcement learning is used as an adaptive gain scheduler whose role is to modify the control aggressiveness according to the current tracking condition.

The Q-learning state is constructed from the absolute values of the normalised error and normalised error derivative,

$$a_e(k) = |e_n(k)|, \quad a_{de}(k) = |de_n(k)|. \quad (166)$$

The error magnitude is quantised into three bins:

$$eBin(k) = \begin{cases} 1, & a_e(k) < 0.02 \\ 2, & 0.02 \leq a_e(k) < 0.06 \\ 3, & a_e(k) \geq 0.06 \end{cases} \quad (167)$$

while the derivative magnitude is quantised into two bins:

$$deBin(k) = \begin{cases} 1, & a_{de}(k) < 0.03 \\ 2, & a_{de}(k) \geq 0.03. \end{cases} \quad (168)$$

The resulting scalar state index is

$$s_{QL}(k) = (eBin(k) - 1)n_{de} + deBin(k) \quad (169)$$

with $n_{de} = 2$, yielding a total of six discrete states.

The reinforcement-learning objective is encoded through the reward function

$$r(k) = -[14.0 a_e(k) + 1.5 a_{de}(k) + 0.3 |ei_n(k)| + 60.0 |\Delta D_{prev}(k)|] \quad (170)$$

This reward penalises four undesirable behaviours simultaneously: large voltage tracking error, rapid error variation, excessive integral accumulation, and excessive duty-cycle activity. Therefore, the Q-learning agent is driven to prefer controller modes that reduce tracking error quickly while maintaining a smooth duty-cycle evolution.

The Q-table is updated during training according to the standard Q-learning rule

$$Q(s_{k-1}, a_{k-1}) \leftarrow Q(s_{k-1}, a_{k-1}) + \alpha_{QL} [r(k) + \gamma_{QL} \max_a Q(s_k, a) - Q(s_{k-1}, a_{k-1})] \quad (171)$$

Action selection during training is based on an ε -greedy policy. The exploration factor decreases with the episode index according to

$$\varepsilon(ep) = \max \left(\varepsilon_{min}, \varepsilon_0 \left[1 - \frac{ep - 1}{\max(N_{train} - 1, 1)} \right] \right) \quad (172)$$

where

$$\varepsilon_0 = 0.25, \quad \varepsilon_{min} = 0.02. \quad (173)$$

During the final evaluation episode,

Table 7. Q-learning action modes and their scheduled controller parameters.

Action	m_{K_f}	m_{K_s}	m_{K_i}	$\Delta\alpha$	m_{Dlim}
Aggressive	3.20	2.80	1.50	-0.35	3.20
Balanced	1.35	1.20	1.10	-0.10	1.40
Soft	0.35	0.45	0.70	0.20	0.35

$$\varepsilon = 0 \quad (174)$$

so the learned greedy policy is used without exploration.

The action set is

$$A = \{1, 2, 3\} = \{\text{aggressive, balanced, soft}\}. \quad (175)$$

Each action modifies the lower-level controller through gain multipliers and smoothing parameters. Thus, the Q-learning policy directly affects the fuzzy gain, sliding gain, integral gain, duty smoothing, and duty increment bound.

The Q-table is initialised with a deliberate bias such that near-MPP states prefer softer action while far states prefer aggressive action:

$$Q_0 = \begin{bmatrix} 0 & 0.2 & 0.5 \\ 0 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.1 \\ 0.4 & 0.3 & 0.0 \\ 1.8 & 0.4 & 0.0 \\ 2.2 & 0.3 & 0.0 \end{bmatrix}. \quad (176)$$

The exact gain-scheduling actions selected by the Q-learning agent are summarised in Table 17. These values are directly used in the controller code (Table 7).

The operational role of Q-learning in the proposed design can therefore be summarised as follows. When the normalised tracking error is large or rapidly changing, the agent tends to select a more aggressive action that increases the fuzzy and sliding gains and relaxes the duty-rate limitation, thereby accelerating movement toward the MPP. Conversely, when the operating point approaches the optimum region, the agent can select a balanced or soft mode that increases smoothing and decreases gain magnitude, thereby reducing oscillation and duty-cycle chatter. In this sense, Q-learning does not replace the lower-level MPPT controller; rather, it supervises and reshapes its behaviour online according to the instantaneous tracking condition.

5. Simulation setup

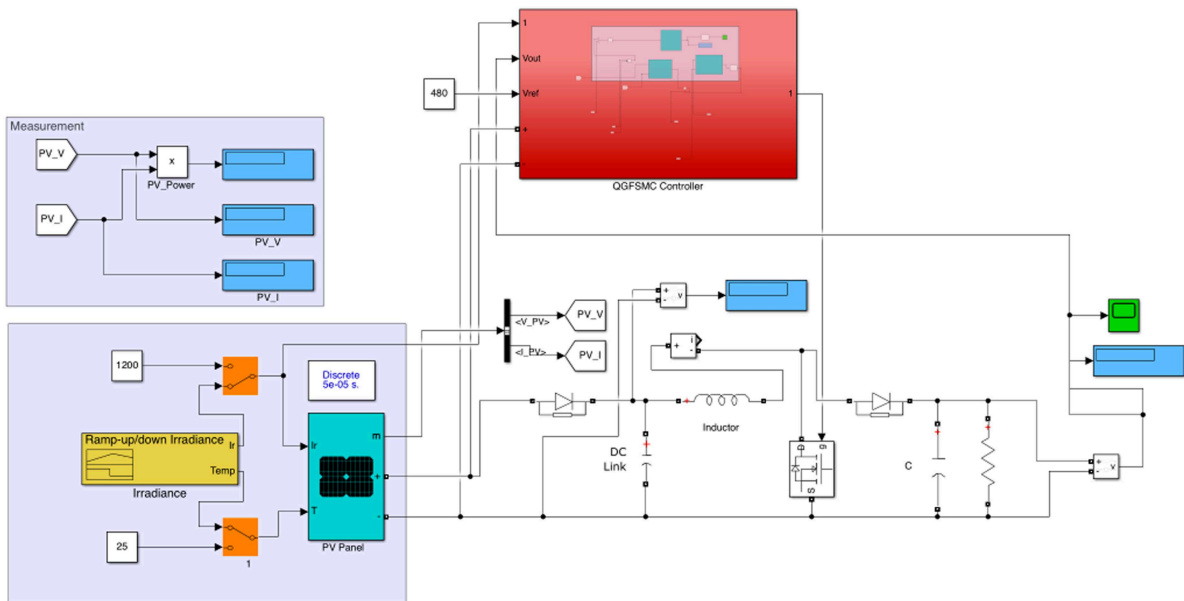
The proposed PV boost-converter MPPT system was evaluated in a MATLAB/Simulink environment, while the numerical settings, controller parameters, and supervisory logic were defined by the implemented simulation script used in the study. The simulated plant is a large-signal averaged continuous-conduction-mode boost converter driven by a photovoltaic source model fitted to the module datasheet points. The study compares four controllers within the same simulation framework, namely the classical Perturb and Observe (P&O) method, the Incremental Conductance (INC) method, the proposed Q-learning-assisted fuzzy sliding-mode controller (QGFSMC), and a benchmark model predictive control (MPC) strategy. The simulation uses a fixed time step of $dt = 5 \times 10^{-5}$ s, a total simulation duration of 2 s, and a controller update period of $T_{ctrl} = 10^{-3}$ s, corresponding to 20 simulation samples per control update.

The photovoltaic array is constructed from $N_s = 5$ series-connected modules per string and $N_p = 10$ parallel strings. The module power rating, current-voltage data, and temperature coefficients are directly embedded in the model and are then scaled to the array level. These parameters are used to determine the effective open-circuit voltage, short-circuit current, and moving MPP under changing irradiance and temperature. The exact PV module and array data employed in the simulation are summarised in Table 8.

The full proposed Simulink model is illustrated in Figure 1. Moreover, the converter and simulation settings are listed in Table 9. These values define the electrical energy-storage elements, load resistance,

Table 8. PV module and array parameters used in the proposed model.

Parameter	Symbol	Value
Series modules per string	N_s	5
Parallel strings	N_p	10
Module maximum power	$P_{max,m}$	220.5 W
Module open-circuit voltage	$V_{oc,m}$	36.8 V
Module short-circuit current	$I_{sc,m}$	8.08 A
Module MPP voltage	$V_{mp,m}$	30.0 V
Module MPP current	$I_{mp,m}$	7.35 A
Voltage temperature coefficient	$T_{c,Voc}$	-0.3364%/C
Current temperature coefficient	$T_{c,Isc}$	0.038465%/C
Array open-circuit voltage	V_{oc}	184.0 V
Array short-circuit current	I_{sc}	80.8 A
Array nominal MPP voltage	V_{mp}	150.0 V
Array nominal MPP current	I_{mp}	73.5 A
Array nominal maximum power	P_{mp}	11025 W

**Figure 1.** Overall Simulink implementation of the proposed Q-learning-assisted fuzzy integral sliding-mode MPPT control system for the PV boost converter, including the PV panel, measurement subsystem, FLC block, Q-policy controller, ISMC boost controller, PWM generation, and power-stage components.**Table 9.** Main simulation and converter parameters used by the proposed QGFSMC controller.

Parameter	Symbol	Value
Simulation step	dt	$5 \times 10^{-5} s$
Simulation time	t_{end}	2 s
Control update period	T_{ctrl}	$1 \times 10^{-3} s$
Control samples/update	N_{ctrl}	20
Inductance	L	2 mH
Input capacitance	C_{in}	470 μF
Output capacitance	C_{out}	470 μF
Load resistance	R	20 Ω
Inductor resistance	r_L	0 Ω
Duty-cycle bounds	D	$0.05 \leq D \leq 0.95$
Training episodes	N_{train}	50
Evaluation episodes	—	1

admissible duty-cycle range, and the reinforcement-learning training length used for the proposed supervisory layer. The converter is represented by the averaged CCM equations and therefore preserves both startup MPPT search and re-tracking after abrupt operating-condition changes. :contentReference [oaicite:7]index = 7.

To test the controller under non-stationary operating conditions, the simulation applies three operating cases over the 2-s window. The Simulation Setup that the 2 s horizon is a controller-dynamics test window and is not intended to represent daily harvested-energy evaluation. The irradiance and temperature are changed in a stepwise manner so that the theoretical MPP is displaced

twice during the simulation. This setup forces each controller to perform both startup tracking and post-disturbance re-tracking within the same run. The exact environmental profile used in the simulation is given in Table 10.

The controller comparison is not based on identical initial operating points. Instead, each controller is assigned a different initial PV voltage and duty-cycle guess, whereas the inductor current and output voltage are initialised to zero in all cases. This creates a more demanding startup test, since each controller begins from a different region of the P–V characteristic. The exact initial conditions are listed in Table 11.

5.1. Controller parametrization

The implemented simulation compares four MPPT controllers, and their functional roles in the Simulink study are summarised in Table 12. P&O and INC act as classical benchmarks, MPC serves as an advanced predictive benchmark, and QGFSMC is the proposed hybrid controller combining fuzzy logic, sliding-mode action, integral compensation, and Q-learning-based supervision.

The fuzzy part of the proposed controller is defined by five membership functions for each input. Both normalised error e_n and normalised error change de_n use one left-shoulder set, three triangular sets, and one right-shoulder set over the interval $[-1, 1]$. The exact membership-function definitions used in the implementation are given in Table 13.

The singleton fuzzy rule table used by the function `fuzzy5_controller` is shown in Table 14. The controller computes the membership grades of e_n and de_n , forms the weighted products of the corresponding fuzzy sets, and then produces the crisp control output through weighted averaging of these singleton consequents (Table 15).

The Q-learning supervisory layer is configured with three error bins, two derivative-error bins, six discrete states, and three available actions. Its learning rate, discount factor, exploration schedule, and training length are listed in Table 16. These values determine how the proposed controller adapts its lower-layer behaviour during training and final evaluation. Furthermore, The Q-learning supervisory component

Table 10. Irradiance and temperature profile used in the simulation.

Case	Time interval (s)	Irradiance G (W/m ²)	Temperature T (°C)
1	$0 \leq t < 1.0$	1000	25
2	$1.0 \leq t < 1.5$	1500 (stress-test)	35
3	$1.5 \leq t \leq 2.0$	2000 (stress-test)	45

Table 11. Initial conditions assigned to the compared controllers.

Controller	Initial PV voltage	Initial duty cycle	Common initial states
P&O	$0.95 V_{oc}$	0.15	$i_L(0) = 0, V_{out}(0) = 0$
INC	$0.90 V_{oc}$	0.20	$i_L(0) = 0, V_{out}(0) = 0$
QGFSMC	$0.85 V_{oc}$	0.28	$i_L(0) = 0, V_{out}(0) = 0$
MPC	$0.75 V_{oc}$	0.42	$i_L(0) = 0, V_{out}(0) = 0$

Table 12. Controllers implemented in the simulation study.

Controller	Core mechanism	Role in the study
P&O	$\Delta P - \Delta V$ -based duty perturbation	Classical MPPT benchmark
INC	Incremental versus instantaneous conductance comparison	Conductance-based benchmark
QGFSMC	Q-learning-supervised fuzzy sliding-mode MPPT with integral compensation	Proposed hybrid intelligent controller
MPC	Predictive duty-cycle optimisation using finite candidate set	Advanced benchmark controller

Table 13. Membership functions used for each fuzzy input in the proposed controller.

Label	Type	Parameters	Range
<i>NB</i>	Left shoulder	$(-1.0, -0.5)$	$x \leq -0.5$
<i>NS</i>	Triangular	$(-1.0, -0.5, 0.0)$	$[-1.0, 0.0]$
<i>ZE</i>	Triangular	$(-0.5, 0.0, 0.5)$	$[-0.5, 0.5]$
<i>PS</i>	Triangular	$(0.0, 0.5, 1.0)$	$[0.0, 1.0]$
<i>PB</i>	Right shoulder	$(0.5, 1.0)$	$x \geq 0.5$

Table 14. Singleton fuzzy rule table used by `fuzzy5_controller`.

e_n/de_n	<i>NB</i>	<i>NS</i>	<i>ZE</i>	<i>PS</i>	<i>PB</i>
<i>NB</i>	-0.70	-0.55	-0.35	-0.15	0.00
<i>NS</i>	-0.55	-0.35	-0.15	0.00	0.15
<i>ZE</i>	-0.35	-0.15	0.00	0.15	0.35
<i>PS</i>	-0.15	0.00	0.15	0.35	0.55
<i>PB</i>	0.00	0.15	0.35	0.55	0.70

Table 15. Cloud server computational configuration for Q-learning training.

Parameter	Configuration
Computing platform	Google Colab (paid)
Runtime	Cloud-based
Processor	Multi-core CPU
Memory	High-memory runtime
Hardware accelerator	GPU-enabled runtime
RL method	Tabular Q-learning
Number of states	6
Number of actions	3
Q-table dimension	6×3
Q-table entries	18
Learning rate, α	0.35
Discount factor, γ	0.95
Initial exploration, ϵ_0	0.25
Minimum exploration, $\epsilon_{\min}$	0.02
Training episodes	50
Evaluation episodes	1
Controller update period	1×10^{-3} s
Simulation time step	5×10^{-5} s

Table 16. Q-learning hyperparameters and supervisory settings used in the proposed controller.

Parameter	Symbol	Value
Number of error bins	$nErrBins_{QL}$	3
Number of derivative bins	$nDeBins_{QL}$	2
Number of states	$nStates_{QL}$	6
Number of actions	$nActions_{QL}$	3
Learning rate	α_{QL}	0.35
Discount factor	γ_{QL}	0.95
Initial exploration	ϵ_0	0.25
Minimum exploration	$\epsilon_{\min}$	0.02
Training episodes	N_{train}	50
Final evaluation episodes	–	1

was additionally trained and evaluated using a paid Google Colab cloud-computing environment. Colab provides access to accelerated CPU, high-memory, GPU, and TPU resources, with premium GPU access. Because Colab allocates hardware dynamically according to availability, the exact processor and accelerator configuration may vary between runtime sessions rather than being fixed.

The three supervisory actions selected by the Q-learning block correspond to aggressive, balanced, and soft controller modes. Each action changes the gain multipliers, the smoothing shift, and the duty-limit multiplier applied to the lower fuzzy sliding-mode layer. The exact action-dependent settings are listed in Table 17. In addition, the initial Q-table bias used before training is given in Table 18, showing that far

Table 17. Q-learning action modes and scheduled controller multipliers used by the proposed QGFSMC algorithm.

Action mode	m_{K_f}	m_{K_s}	m_{K_i}	$\Delta\alpha$	m_{Dlim}
Aggressive	3.20	2.80	1.50	−0.35	3.20
Balanced	1.35	1.20	1.10	−0.10	1.40
Soft	0.35	0.45	0.70	0.20	0.35

Table 18. Initial Q-table bias used before training.

State	Aggressive	Balanced	Soft
1 (near-calm)	0.0	0.2	0.5
2 (near-changing)	0.0	0.3	0.4
3 (medium-calm)	0.2	0.4	0.1
4 (medium-changing)	0.4	0.3	0.0
5 (far-calm)	1.8	0.4	0.0
6 (far-changing)	2.2	0.3	0.0

states were initialised to prefer more aggressive action, whereas near states were initialised to prefer softer or balanced behaviour.

The Q-learning component in the proposed QGFSMC operates as a supervisory decision layer rather than directly generating the boost-converter duty cycle. Its role is to observe the instantaneous tracking condition through the normalised error and error variation and select one of three supervisory actions: aggressive, balanced, or soft. The selected action schedules the fuzzy, sliding-mode, and integral control gains and the associated duty-cycle constraints of the lower control layer. Therefore, Q-learning provides adaptive supervisory intelligence while the fuzzy-integral sliding-mode controller retains responsibility for the continuous MPPT control action.

The training and online execution stages are separated. During training, the Q-values associated with the six states and three actions are updated according to the defined learning rule and reward function. After convergence of the training process, the resulting 6×3 Q-table is retained as the control policy. During online operation, the normalised error variables are used to identify the current discrete state, the three corresponding Q-values are retrieved, and the selected action determines the gain-scheduling mode applied to the lower controller. Since the trained policy contains only 18 state-action values, its execution does not require neural-network inference, backpropagation, or computationally intensive online optimisation.

For practical hardware deployment, the trained Q-table can be stored directly in the memory of a digital controller such as a microcontroller, DSP, or real-time control processor. At each sampling instant, measured PV voltage and current signals are acquired through the controller ADC interface and used to determine the tracking variables and corresponding Q-learning state. The stored Q-table is then accessed to select the supervisory action, after which the scheduled QGFSMC parameters are used to calculate the duty-cycle command supplied to the PWM module of the DC–DC boost converter. Consequently, the computationally intensive development and training environment is separated from the final embedded implementation, while the deployed controller requires only signal processing, state classification, a small table lookup, gain scheduling, and execution of the lower-level control law. This structure provides a practical pathway from the simulation-trained Q-learning supervisor to real-time PV converter control hardware.

From a computational perspective, the Q-learning supervisor introduces a compact discrete decision layer rather than a computationally intensive function approximator. Online policy evaluation requires state quantisation and comparison of three Q-values for the identified state. The remaining operations correspond to the fuzzy inference, sliding-mode correction, integral compensation, filtering, and duty-cycle limiting operations of the lower control layer. Nevertheless, real-time execution time, memory utilisation, and processor loading have not yet been experimentally profiled on embedded control hardware. These quantities therefore remain subjects for subsequent hardware-in-the-loop and physical implementation studies which we intend to do it as future work.

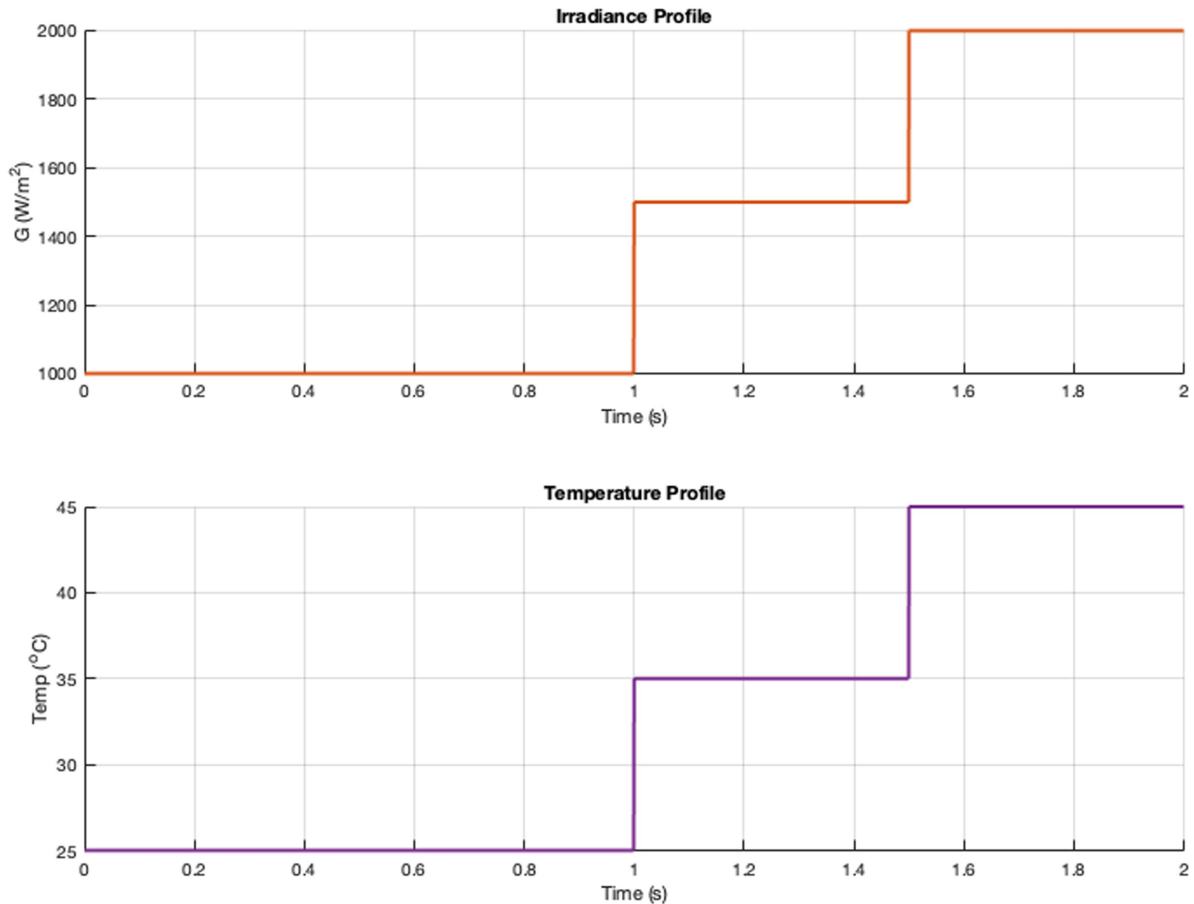
For completeness, the discrete-state definition used by the supervisory Q-learning layer is summarised in Table 19. This state construction is based on the magnitudes of the normalised tracking error and

Table 19. Discrete Q-learning state definition used in the proposed controller.

State	$ e_n $ bin	$ de_n $ bin	Interpretation
1	<0.02	<0.03	near-calm
2	<0.02	≥ 0.03	near-changing
3	$0.02 \leq e_n < 0.06$	<0.03	medium-calm
4	$0.02 \leq e_n < 0.06$	≥ 0.03	medium-changing
5	≥ 0.06	<0.03	far-calm
6	≥ 0.06	≥ 0.03	far-changing

Table 20. Reward-function coefficients and their control objectives.

Coefficient	Value	Penalised quantity	Control objective
w_e	14.0	Normalised MPP tracking error, $ e_n $	Prioritises rapid convergence toward the MPP and penalises large tracking deviations.
$w_{\Delta e}$	1.5	Normalised error variation, $ \Delta e_n $	Penalises rapid error variation to improve transient smoothness and damping.
w_l	0.3	Accumulated tracking error	Provides a lower-weight penalty on persistent tracking error while avoiding excessive integral influence.
$w_{\Delta d}$	60.0	Duty-cycle variation, $(\Delta d)^2$	Strongly penalises excessive duty-cycle movement to suppress oscillation and chattering.

**Figure 2.** Applied irradiance and temperature profiles used to test the controllers under two successive disturbances at $t = 1$ s and $t = 1.5$ s.

normalised error derivative and provides a compact supervisory state space for online controller-mode selection. The MPC benchmark also has an explicit internal configuration in the implementation. This table is useful because it makes the benchmark controller definition explicit rather than leaving it implicit.

The reward coefficients were selected empirically through iterative controller tuning. The weighting was chosen to prioritise MPP tracking accuracy while maintaining transient smoothness, limiting accumulated

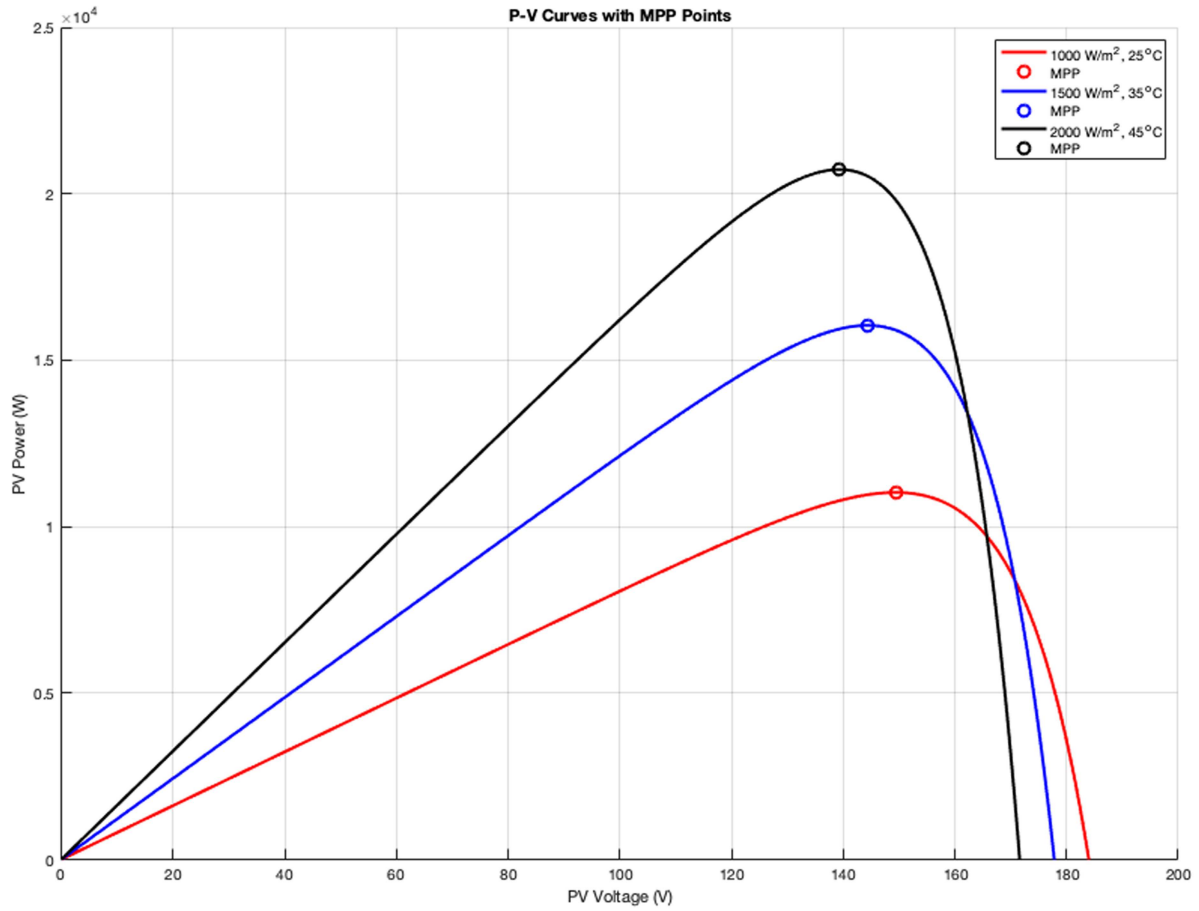


Figure 3. Theoretical P–V characteristics and corresponding maximum power points for the three operating cases: 1000 W/m², 25°C, 1500 W/m², 35°C, and 2000 W/m², 45°C.

error, and suppressing excessive duty-cycle activity. The relatively large value of $w_{\Delta d}$ reflects the small numerical magnitude of the duty-cycle increment and provides an effective penalty against unnecessary control movement. A formal parametric sensitivity analysis of these reward coefficients was not performed in the present study.

The Tables 21 and 20 above fully define the simulation configuration used for the proposed study: the PV-array data, converter constants, environmental disturbance schedule, initial conditions, compared controllers, fuzzy inference structure, Q-learning hyperparameters, supervisory action modes, initial Q-table bias, MPC benchmark settings, and the local-function input-output mapping. This establishes the simulation framework clearly before any later discussion of comparative dynamic performance.

6. Results and discussion

Figure 2 shows the operating conditions used in the Simulink evaluation. The irradiance increases from 1000 W/m² to 1500 W/m² at $t = 1$ s, and then to 2000 W/m² at $t = 1.5$ s. At the same time, the temperature rises from 25°C to 35°C, and then to 45°C. These changes force the PV operating point to move repeatedly, thereby creating a demanding MPPT test involving both startup tracking and post-disturbance re-tracking.

Figure 3 shows the theoretical P–V curves associated with these three operating conditions. As the irradiance increases, the available PV power rises significantly, while the simultaneous temperature increase shifts the maximum power point toward a slightly lower operating voltage. Consequently, the controller is required not only to extract higher power after each disturbance, but also to relocate the PV operating point to a new MPP region. This figure therefore provides the theoretical reference against which the dynamic controller trajectories can be interpreted.

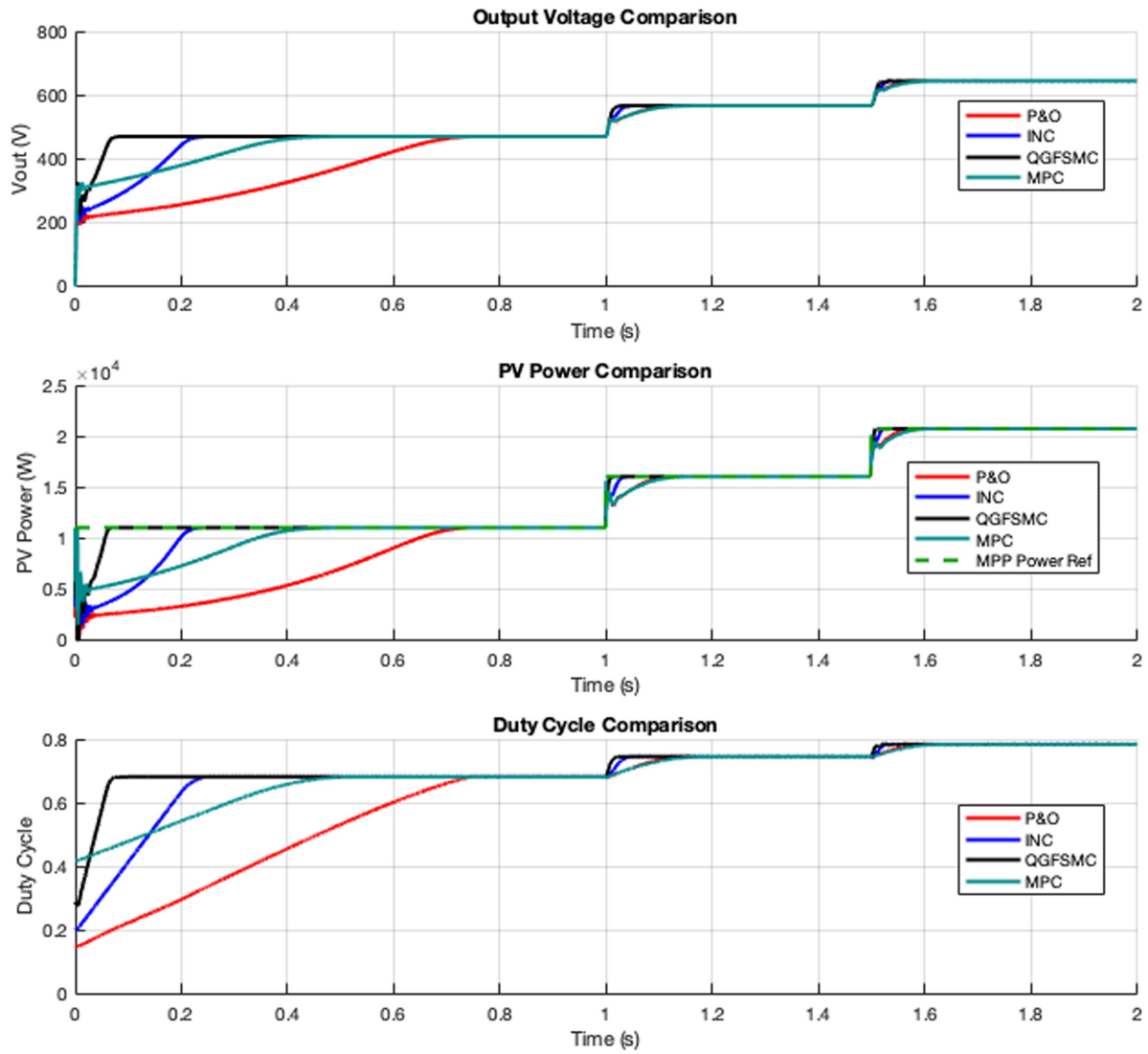


Figure 4. Overall converter-level comparison of the four controllers in terms of output voltage, PV power, and duty cycle.

Figure 4 compares the converter-level responses of *P&O*, *INC*, *QGFSMC*, and *MPC* in terms of output voltage, PV power, and duty cycle. The proposed *QGFSMC* controller reaches the steady operating region faster than the other controllers during startup, and this is especially clear in Figure 5. In the startup interval, *QGFSMC* exhibits the fastest rise of the output voltage and the quickest increase of PV power toward the theoretical MPP level. Its duty cycle also moves more rapidly to the appropriate operating region, which explains the faster establishment of the desired PV operating point. By comparison, *P&O* is the slowest controller, showing a gradual duty-cycle ramp and a delayed power build-up. *INC* is substantially faster than *P&O*, but its transient takes longer than *QGFSMC* to settle. *MPC* shows smooth behaviour, but its rise is slower than the proposed controller. The response to the first disturbance at $t = 1$ s is shown more clearly in Figure 6. When the available PV power increases, *QGFSMC* reacts immediately and drives the converter toward the new operating region faster than the other methods. The output voltage and PV power recover more quickly, and the duty cycle reaches its new steady band earlier. *INC* is again the next best among the benchmark methods, but its transient remains longer than that of *QGFSMC*. *MPC* is smoother but slower, whereas *P&O* requires the longest interval to re-establish the new operating condition. A similar pattern is also visible in Figure 4 at the second disturbance at $t = 1.5$ s, where *QGFSMC* again shows the fastest transition toward the new steady state. Overall, the converter-level comparison demonstrates that the proposed controller has the shortest rise time, faster settling, and stronger disturbance-handling capability among the four evaluated methods.

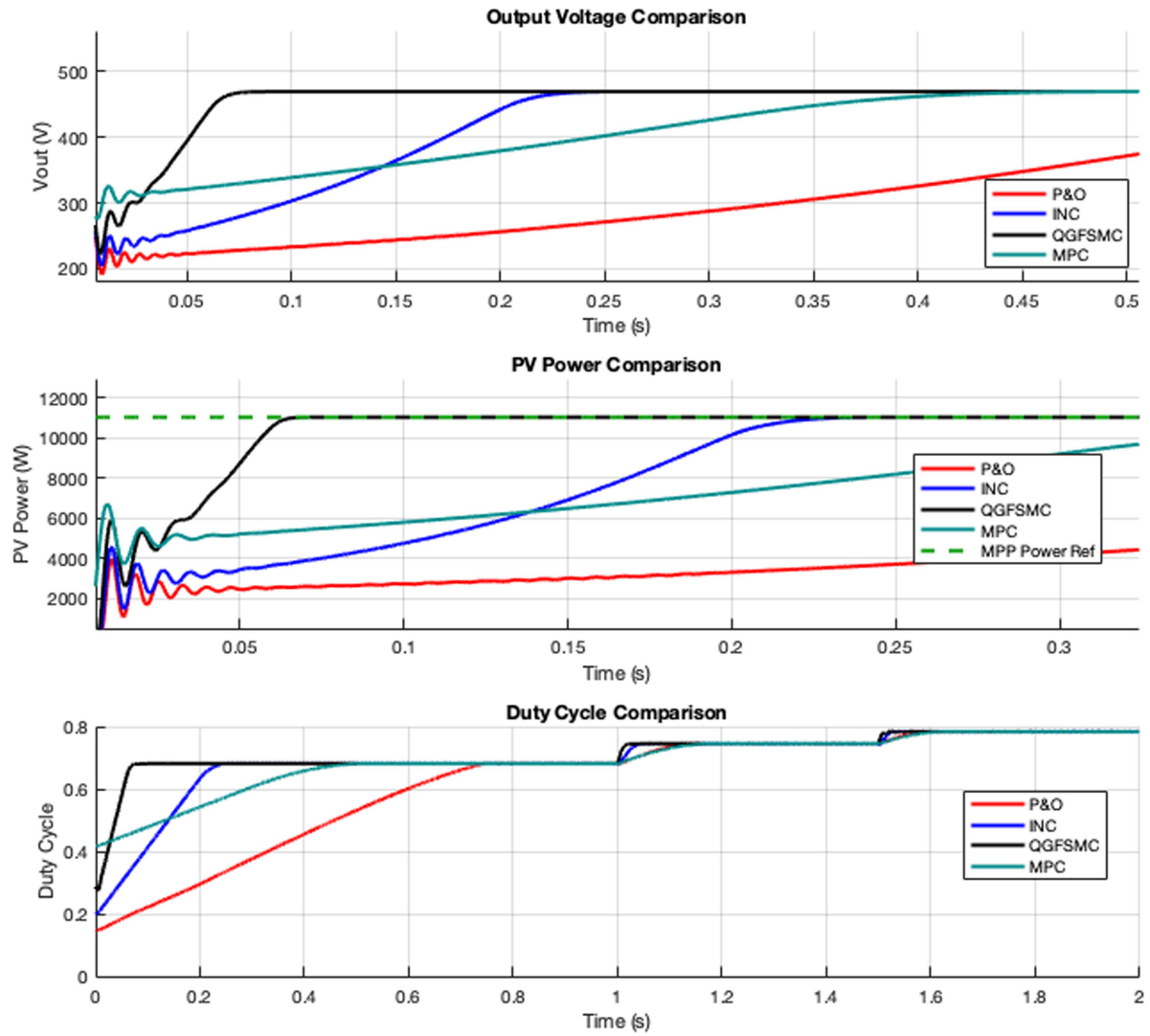


Figure 5. Magnified startup response.

Moreover, Figure 7 presents the PV-side variables V_{pv} , I_{pv} , and P_{pv} together with the theoretical reference trajectories. This figure directly shows how well each controller tracks the MPP after startup and after the two imposed disturbances. The proposed QGFSMC controller moves the PV voltage toward the reference more quickly than the other controllers and simultaneously brings the current and power to their new reference levels with less delay. This behaviour is evident during the initial transient and again after both disturbance instants. In contrast, P&O remains the slowest method, with the largest delay before its voltage, current, and power trajectories align with the new MPP region. MPC exhibits smooth but comparatively slow convergence, and INC performs better than P&O and MPC but still does not match the response speed of QGFSMC.

The superiority of the proposed controller around the first disturbance is highlighted in Figure 8. Immediately after $t = 1$ s, the theoretical MPP voltage shifts downward while the MPP current and power increase. QGFSMC responds to this change very rapidly: the PV voltage is redirected toward the new reference in a shorter interval, and the corresponding current and power recover to their new levels sooner than in the other controllers. The transient excursion of the proposed method is brief and quickly damped, which indicates faster settling. INC shows a slower recovery than QGFSMC, while P&O and MPC remain further away from the updated references for a longer duration. The same trend is visible again at $t = 1.5$ s in Figure 7, where the proposed method again adapts more rapidly to the new MPP condition.

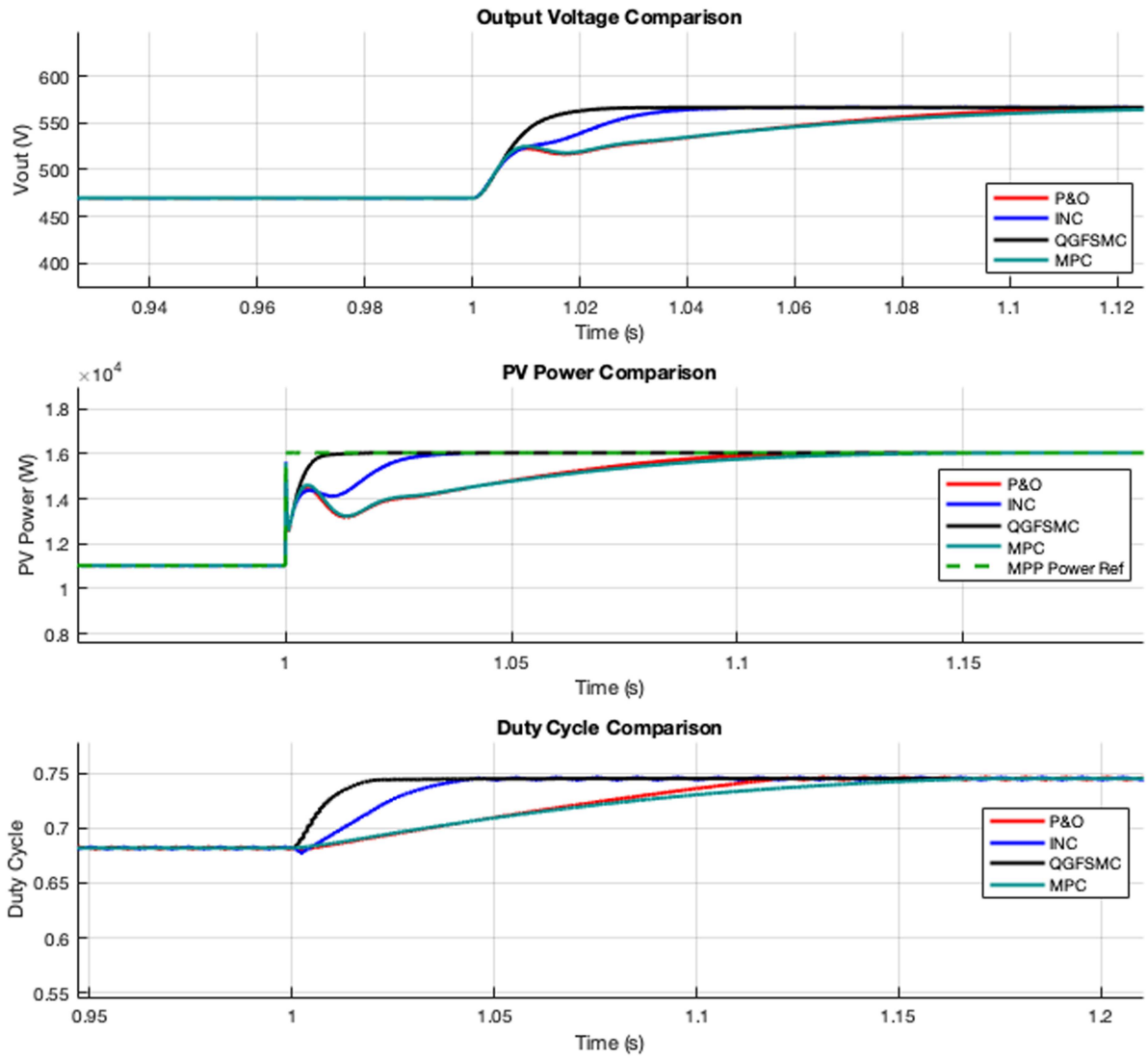


Figure 6. Magnified response around the disturbance at $t = 1$ s.

Figure 9 provides an important complementary interpretation by showing the actual controller motion on the P–V plane. The theoretical P–V curves and MPP locations define the desired operating targets, while the controller trajectories indicate how each MPPT method moves through the nonlinear characteristic in practice. The proposed QGFSMC trajectory reaches the MPP region more directly and with less prolonged wandering away from the optimum region. In contrast, *P&O* shows the longest excursion before converging, which is consistent with its slower time-domain response. INC follows a more effective path than *P&O*, but its transition remains less direct than that of the proposed method. MPC also converges to the correct region, although its path is more gradual. These P–V trajectories therefore support the earlier time-domain observations by showing that QGFSMC not only converges faster in time, but also follows a more efficient route on the nonlinear PV characteristic.

The six-state space is intentionally compact because Q-learning acts as a supervisory gain scheduler, not as the primary MPPT law. The states combine three magnitudes of normalised tracking error (near, medium, and far from the reference) with two error-rate conditions (calm and changing). This distinguishes proximity to the MPP from transient movement. The three actions—aggressive, balanced, and soft—then provide the required control response: stronger correction when far or rapidly changing, intermediate action during transition, and reduced control activity near the MPP. Thus, the 6×3 structure captures the operating information required by the supervisory task while avoiding an unnecessarily large Q-table and associated training burden.

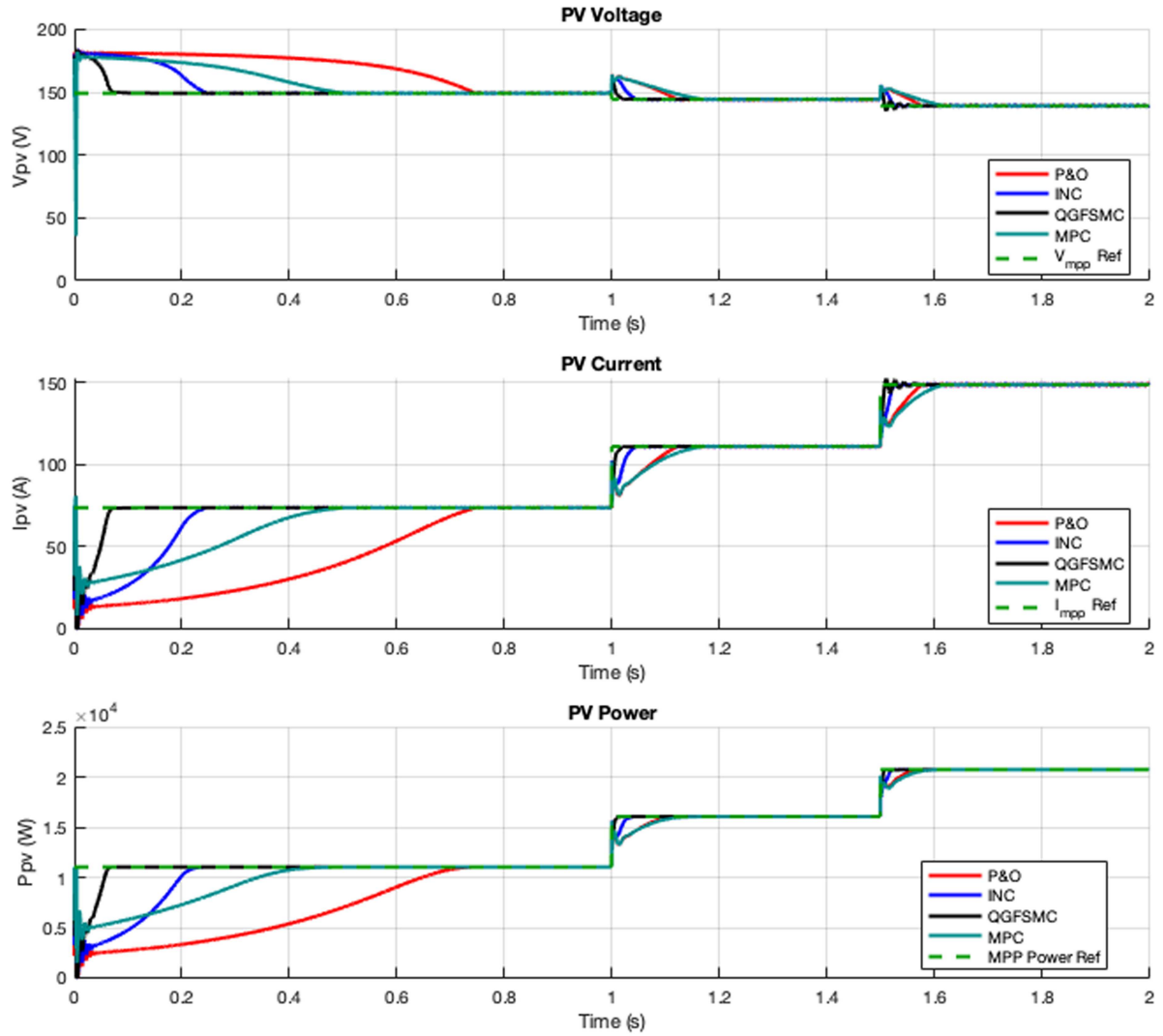


Figure 7. Overall PV-side tracking performance in terms of PV voltage, PV current, and PV power relative to their theoretical references.

Figure 10 explains why the proposed controller is able to respond more effectively. The selected-gain-mode plot shows that the supervisory layer does not remain fixed; instead, it changes the controller mode according to the tracking condition. During quiescent intervals, the policy stays mainly in a balanced mode, whereas short switches occur when the startup transient or the disturbances at $t = 1$ s and $t = 1.5$ s appear. The reward plot drops sharply only during these demanding intervals and then rapidly returns toward its nominal level, which indicates that the controller quickly suppresses the large tracking deviations. This is consistent with the normalised error and normalised error-change plots, both of which exhibit brief spikes at startup and at the two disturbance instants, followed by fast decay. The reward-versus-error scatter and the final Q-table further confirm that the learned policy establishes a clear state-dependent supervisory pattern. Thus, the Q-learning layer contributes directly to faster rise, reduced settling time, and stronger disturbance handling by adjusting the aggressiveness of the lower fuzzy sliding-mode controller online. Taken together, Figures 4–10 consistently show that the proposed QGFSMC controller outperforms P&O, INC, and MPC in the considered Simulink study. The proposed method exhibits the fastest startup rise, the shortest settling behaviour around the new operating regions, and the most effective recovery after the strong disturbances introduced at $t = 1$ s and $t = 1.5$ s. While INC provides the closest benchmark performance among the conventional approaches, QGFSMC remains superior in both transient speed and disturbance rejection. P&O is clearly the slowest, and MPC, although smooth, is less responsive than the proposed hybrid controller. Therefore, the combined action of Q-

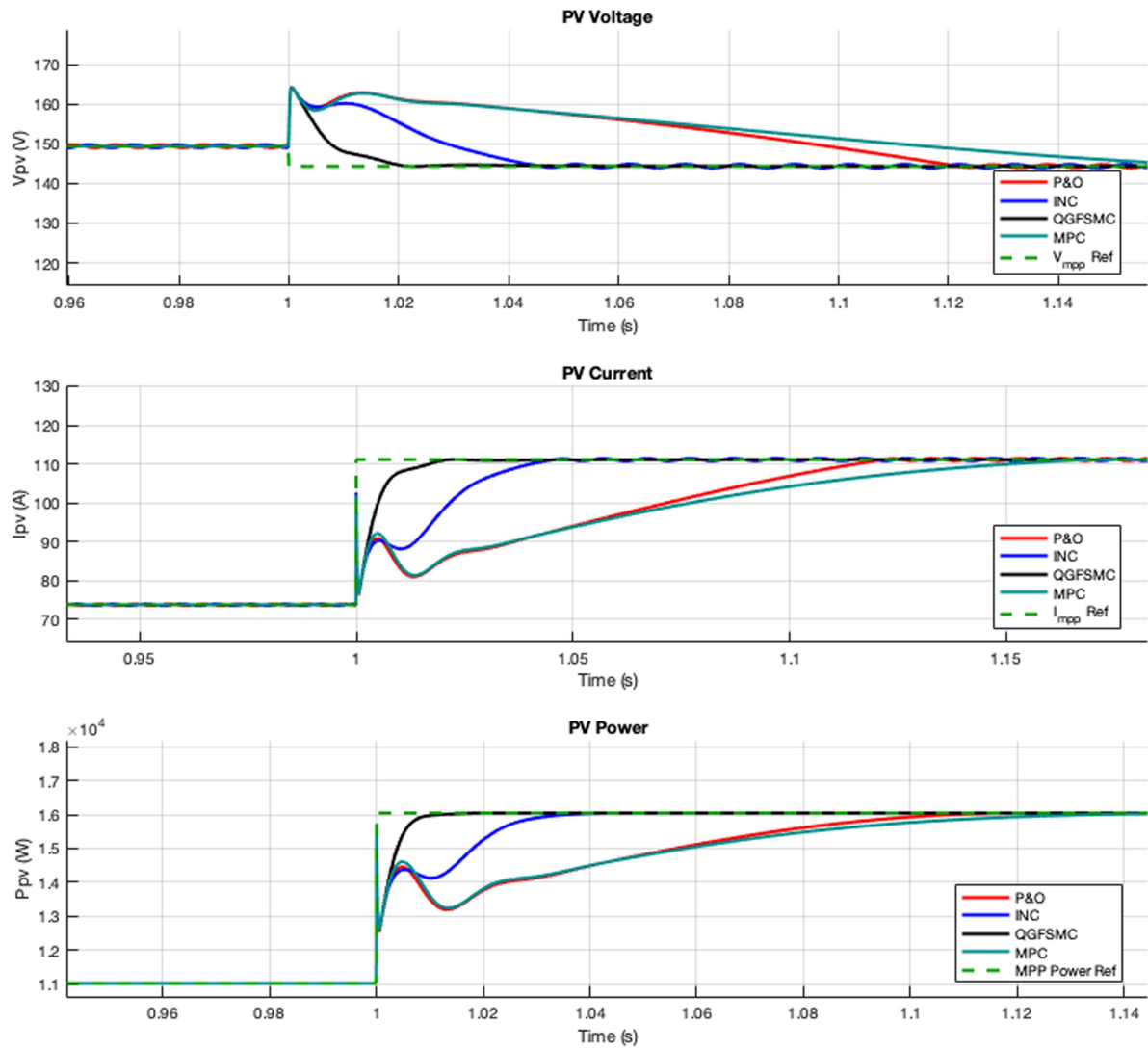


Figure 8. Magnified PV-side tracking response around the disturbance at $t = 1$ s.

learning supervision, fuzzy inference, and sliding-mode correction yields a more effective MPPT strategy for PV boost-converter operation under severe operating changes. Refer to appendix; [Tables 21](#) and [Table 22](#) for complete abbreviation description.

The proposed QGFSMC was validated through a comprehensive MATLAB/Simulink framework and cloud-based computational evaluation, providing a rigorous and reproducible basis for assessing its control performance. The nonlinear PV source and dynamic boost-converter model enable the controller to be evaluated under precisely defined operating transitions, environmental variations, and disturbance conditions. Furthermore, identical plant parameters and test scenarios were applied to QGFSMC, *P&O*, *INC*, and *MPC*, enabling a direct and technically consistent benchmark comparison. The intelligent Q-learning component was additionally developed, trained, and evaluated using dedicated cloud-computing resources. Together, these validation stages provide substantial evidence of the proposed controller's convergence capability, dynamic performance, robustness, and computational feasibility under the operating conditions investigated in this study.

The principal limitation of the present study is that the proposed QGFSMC has been validated within a comprehensive simulation environment rather than on a physical PV power-conversion testbed. Although MATLAB/Simulink provides a controlled, repeatable, and well-established engineering environment for evaluating QGFSMC against *P&O*, *INC*, and *MPC* under identical operating and disturbance conditions, practical implementation may introduce additional effects associated with switching devices, sensing,

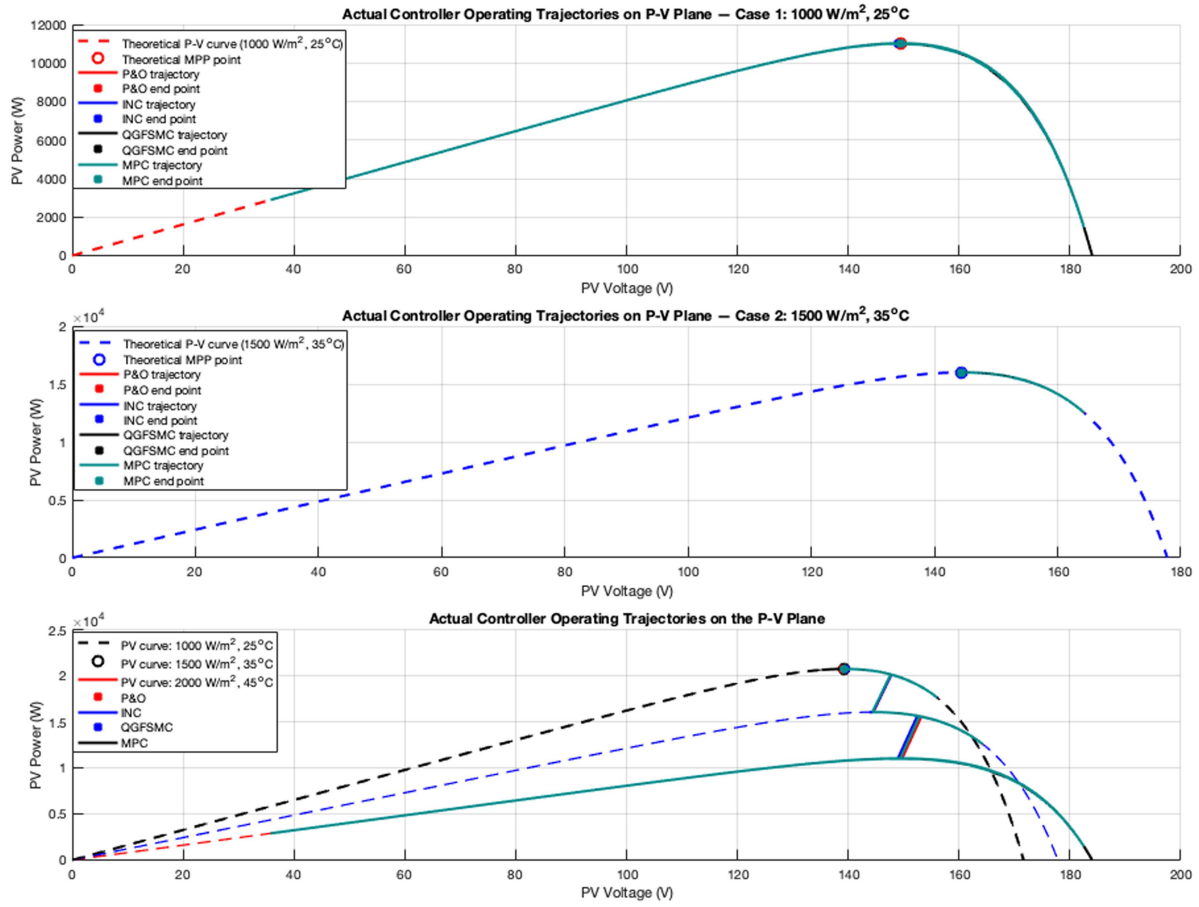


Figure 9. Actual controller operating trajectories on the P–V plane for the considered operating cases.

sampling, computation delays, component tolerances, and converter non-idealities. In addition, while the compact tabular Q-learning structure has low computational requirements during online execution, its execution time and processor loading have not yet been experimentally quantified on embedded control hardware. These aspects define the current scope of the study and will be addressed in future work through hardware-in-the-loop evaluation and experimental implementation on a dedicated PV boost-converter testbed, including real-time computational assessment under practical operating and environmental conditions. Refer to Table A1 and Table A2 in Appendix section for abbreviations and coefficient values used. Table 22 shows quantitative comparison while Table 23 shows qualitative comparison of the used controllers.

This table showcase the computational status of the controllers.

The controller parameters were selected through a structured iterative tuning procedure within the same PV boost-converter model and operating conditions used throughout the study. The tuning process was performed according to the functional role of each parameter group rather than adjusting all coefficients simultaneously. The reference and measurement filters were first adjusted to provide sufficient signal smoothing without introducing excessive tracking delay, followed by tuning of the derivative filtering and integral-memory terms to obtain a stable error response. The fuzzy and sliding-mode gains were subsequently adjusted to achieve rapid convergence while limiting oscillation around the MPP. Duty-cycle smoothing and increment limits were then selected to maintain bounded control movement and suppress excessive switching activity. Finally, the Q-learning action multipliers were tuned to provide distinct aggressive, balanced, and soft supervisory responses for large-error, transitional, and near-MPP operating conditions, respectively. The final parameter set was retained once stable convergence, reduced near-MPP oscillation, bounded duty-cycle evolution, and consistent recovery following the imposed operating changes were achieved. All controller evaluations were subsequently performed using this fixed parameter set.

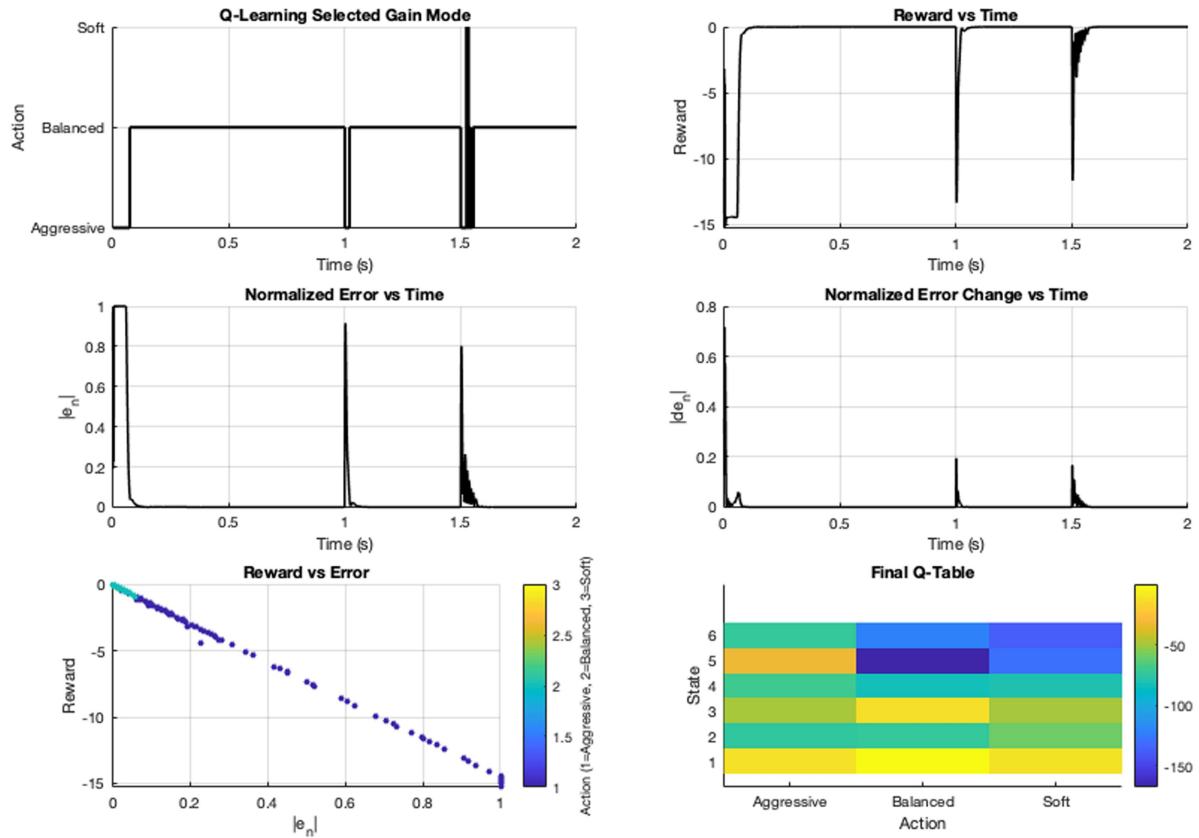


Figure 10. Q-learning supervisory behaviour, including selected gain mode, reward evolution, normalised error measures, reward-error relation, and final Q-table.

Table 21. Model predictive control parameters and tuning settings.

Parameter	Symbol	Value
Reference filter coefficient	$V_{ref,f}$	0.82 previous + 0.18 new
PV-voltage filter coefficient	$V_{pv,f}$	0.72 previous + 0.28 new
Prediction horizon	H	14
Candidate duty set	ΔD	$[-3, -2, -1, 0, 1, 2, 3] d_{base}$
Base duty step	d_{base}	$1.5 \times 10^{-4} + 7.0 \times 10^{-4} w_{far,mpc}$
Voltage-tracking weight	–	5.0
Output-voltage weight	–	1.0×10^{-4}
Duty-movement penalty	–	120
Final duty smoothing	–	$0.75 D_{prev} + 0.25 D_{best}$

Table 22. Quantitative performance comparison of the investigated MPPT controllers.

Controller	ESRT (s)	SST (s)	DST–1 (s)	DST–2 (s)	$P_{min,1}$ (kW)	Post-step power retention (%)	Rank
P&O	0.72	0.75	0.13	0.16	13.2	82.5	4
INC	0.20	0.23	0.04	0.05	14.1	88.1	2
QGFSMC	0.05	0.06	0.02	0.02	15.9	99.4	1
MPC	0.44	0.48	0.12	0.14	13.3	83.1	3

ESRT: effective startup rise time; SST: startup settling time; DST–1 and DST–2: settling times following the operating changes at $t = 1$ s and $t = 1.5$ s, respectively. $P_{min,1}$ denotes the minimum PV power immediately following the $t = 1$ s operating transition. Post-step power retention is calculated relative to the 16 kW MPP reference after the first operating transition.

7. Future work

Future work will transfer the complete QGFSMC framework from the present simulation environment to a laboratory-scale experimental PV power-conversion platform while preserving the same controller structure and operating scenarios. The proposed testbed will comprise a commercial PV module or programmable PV-array emulator, a laboratory DC–DC boost converter equipped with suitable MOSFET/

Table 23. Qualitative computational comparison of the evaluated MPPT controllers.

Method	Main online operations	Stored information	Online optimisation	Online training
P&O	Power comparison and duty-cycle perturbation	Previous operating values	No	No
INC	Incremental voltage/current calculation, conductance comparison, and duty update	Previous operating values	No	No
MPC	Plant prediction, candidate duty-cycle evaluation, cost-function calculation, and selection	Plant/model parameters	Yes	No
QGFSMC	State classification, Q-table lookup, fuzzy inference, sliding/integral control, gain scheduling, and duty update	18 Q-values, fuzzy rules, and controller parameters	No	No

IGBT switching devices and gate drivers, voltage and Hall-effect current sensors, and a real-time digital controller such as a TI C2000 DSP or Speedgoat/Simulink Real-Time platform for executing the trained QGFSMC policy and generating the PWM duty cycle. A programmable DC electronic load, digital oscilloscope, power analyser, and appropriate isolation and protection circuitry will support measurement and safe operation. The experimental programme will reproduce the irradiance, temperature, and disturbance profiles considered in the simulation study and compare QGFSMC with P&O, INC, and MPC using the same performance indicators, thereby enabling direct simulation-to-hardware assessment of MPPT convergence, transient response, tracking efficiency, computational latency, and robustness under practical operating conditions.

8. Conclusion

This paper presented a novel hybrid MPPT control framework for PV boost energy conversion systems based on the integration reinforcement learning control (Q-learning), with hybrid fuzzy-SMC architecture. The proposed controller was developed to improve the dynamic regulation of the boost converter under rapidly changing operating conditions, with particular emphasis on faster convergence toward the maximum power point, reduced steady-state oscillation, and stronger robustness against severe disturbances. By combining the adaptive decision-making capability of Q-learning with the nonlinear reasoning of fuzzy logic and the robustness of integral sliding-mode compensation, the proposed strategy provides a coordinated control structure that is both intelligent and resilient. The simulation results confirmed that the proposed controller offers superior performance compared with the other MPPT algorithms considered in this study. In particular, the proposed method achieved faster rise and settling characteristics, improved tracking smoothness, and better stability of the PV operating point during abrupt irradiance and temperature variations. Unlike conventional and benchmark algorithms, whose performance degrades under strong transients due to oscillation, slower convergence, or limited disturbance rejection, the developed controller maintained more reliable tracking of the maximum power point and produced a more stable duty-cycle evolution in the boost converter. This demonstrates that the proposed approach is more effective for handling strong disturbances and complex nonlinear operating conditions. Overall, the developed Q-learning–fuzzy–integral sliding MPPT controller constitutes a more advanced and effective solution for boost-converter-based PV systems than the state-of-the-art algorithms implemented in the simulation study. The results show that the proposed method not only improves energy extraction performance, but also enhances converter stability, dynamic robustness, and operational reliability, making it a promising candidate for high-performance photovoltaic energy conversion applications.

Author contributions

CRediT: **Ammar K. Al Mhdawi**: Conceptualization, Data curation, Formal analysis, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Akram Bati**: Investigation, Supervision, Validation, Writing – review & editing; **Hamed Al-Raweshidy**: Investigation, Project administration, Supervision, Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

No funding was received for this research.

Data availability statement

The data supporting the findings of this study are available within the article. Additional data are available from the corresponding author upon reasonable request.

AI declaration

The authors confirm that all scientific concepts, methodology, experimental design, simulations, results, analysis, and conclusions presented in this manuscript are their original work and were not generated by artificial intelligence. Generative AI (ChatGPT, OpenAI) was used solely to assist with language refinement, grammar correction, wording, and improving the clarity and readability of the manuscript. All AI-assisted text was carefully reviewed, verified, and edited by the authors, who take full responsibility for the accuracy, originality, and integrity of the manuscript.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used AI-assisted language tools for grammar refinement, language improvement, formatting support, and manuscript wording enhancement. All modelling, controller development, simulation implementation, experimental validation, performance analysis, and technical results presented in this work were independently developed, implemented, analysed, and verified by the authors. The authors carefully reviewed and edited the manuscript and take full responsibility for the scientific accuracy and content of the published article.

References

- Aiboudi, J., H. Rizki, R. Chaibi, N. Jebbor, E.-M. Boufounas, and A. El Hajjaji. 2025. "Integral-Type Terminal Sliding Mode Mppt Controller for a Standalone Pv System with a dc-dc Boost Converter under Real-Time Varying Weather Conditions." *IEEE Access* 14: 2335–2350. <https://doi.org/10.1109/ACCESS.2025.3649148>.
- Ali, Z., S. Z. Abbas, A. Mahmood, S. W. Ali, S. B. Javed, and C.-L. Su. 2023. "A Study of a Generalized Photovoltaic System with Mppt Using Perturb and Observer Algorithms under Varying Conditions." *Energies* 16 (9): 3638. <https://doi.org/10.3390/en16093638>.
- Aly, M., E. M. Ahmed, H. Rezk, and E. A. Mohamed. 2021. "Marine Predators Algorithm Optimized Reduced Sensor Fuzzy-Logic Based Maximum Power Point Tracking of Fuel Cell-Battery Standalone Applications." *IEEE Access* 9: 27987–28000. <https://doi.org/10.1109/ACCESS.2021.3058610>.
- Bakar Siddique, M. A., A. Asad, R. M. Asif, A. U. Rehman, M. T. Sadiq, and I. Ullah. 2023. "Implementation of Incremental Conductance Mppt Algorithm with Integral Regulator By Using Boost Converter in Grid-Connected Pv Array." *IETE Journal of Research* 69 (6): 3822–3835. <https://doi.org/10.1080/03772063.2021.1920481>.
- Çakmak, F., Z. Aydoğmuş, and M. R. Tür. 2025. "Analyses of Po-Based Fuzzy Logic-Controlled Mppt and Incremental Conductance Mppt Algorithms in Pv Systems." *Energies* 18 (2): 233. <https://doi.org/10.3390/en18020233>.
- Chandola, J., S. Pundir, A. Sharma, Y. Song, G. Fekete, and T. Singh. 2026. "Recent Advances in Mppt Techniques for Photovoltaic Systems: A Review of Classical (p&o, Ic), Intelligent (Ann), Optimization (Pso) and Hybrid (Ann-Pso) Methods." *Results in Engineering* 29: 109395. <https://doi.org/10.1016/j.rineng.2026.109395>.
- Değermenci, M., Y. Yalman, and K. Olcay. 2026. "Mppt Algorithms for Grid-Connected Solar Systems Including Deep Learning Approaches." *Scientific Reports* 16: 6189. <https://doi.org/10.1038/s41598-026-36321-8>.
- Deshi, M., E. Anene, J. Agee, and S. Hassan. 2026. "Optimal Pi Controller Design for High-Efficiency dc-dc Boost Converters with Enhanced Perturb-And-Observe Mppt in Photovoltaic Systems." *Journal of Electrical Systems and Information Technology* 13 (1): 17. <https://doi.org/10.1186/s43067-026-00321-5>.
- Dijlali, A. B., E. Bounadja, A. Yahdou, H. Benbouhenni, Z. Elbarbary, I. Colak, and S. F. Al-Gahtani. 2025. "Enhanced Variable Step Sizes Perturb and Observe Mppt Control to Reduce Energy Loss in Photovoltaic Systems." *Scientific Reports* 15 (1): 11700. <https://doi.org/10.1038/s41598-025-95309-y>.
- Douifi, N., A. Abbadi, K. K. Mohammed, and S. Mekhilef. 2026. "Fuzzy Logic Power Management with Enhanced Pcr-Po Mppt for a Hybrid Renewable Energy System." *Electric Power Systems Research* 258: 113102. <https://doi.org/10.1016/j.ejpsr.2026.113102>.
- Girgis, M. E., and N. A. Elkhateeb. 2024. "Enhancing Photovoltaic Mppt with p&o Algorithm Performance Based on Adaptive Pid Control Using Exponential Forgetting Recursive Least Squares Method." *Renewable Energy* 237: 121801. <https://doi.org/10.1016/j.renene.2024.121801>.

- Hasheminasab, S., M. Alzayed, and H. Chaoui. 2025. "An Adaptive Neuro-Fuzzy Controller to Enhance Power Sharing in Distributed Energy Resources Applications." *IEEE Access* 13: 94667–94680. <https://doi.org/10.1109/ACCESS.2025.3575039>.
- Hazzab, A., H. Gouabi, M. Habbab, M. Rezkallah, A. Chandra, and H. Ibrahim. 2024. "Fuzzy Logic Scheduling of the Duty Cycle Perturbation for Optimized Mppt Controller of Pv/Wind Hybrid System." *IEEE Transactions on Industry Applications* 61 (1): 642–652. <https://doi.org/10.1109/TIA.2024.3481197>.
- Hole, S. R., and A. D. Goswami. 2024. "Epcmsdb: Design of an Ensemble Predictive Control Model for Solar Pv Mppt Deployments Via Dual Bioinspired Optimizations." *Science and Technology for Energy Transition* 79: 8. <https://doi.org/10.2516/stet/2024002>.
- Ibrahim, M. H., S. P. Ang, M. N. Dani, M. I. Rahman, R. Petra, and S. M. Sulthan. 2023. "Optimizing Step-Size of Perturb & Observe and Incremental Conductance Mppt Techniques Using Pso for Grid-Tied Pv System." *IEEE Access* 11: 13079–13090. <https://doi.org/10.1109/ACCESS.2023.3242979>.
- Jabbar, R. I., S. Mekhilef, M. Mubini, and K. K. Mohammed. 2023. "A Modified Perturb and Observe Mppt for a Fast and Accurate Tracking of Mpp under Varying Weather Conditions." *IEEE Access* 11: 76166–76176. <https://doi.org/10.1109/ACCESS.2023.3297445>.
- Jain, S., S. V. S. P. K. Ch, V. Sonti, A. Keskar, J. X. Johnson, S. Ghosh, and R. N. Patel. 2023. "High-Performance Hybrid MPPT Algorithm Based Single-Stage Solar PV Fed Induction Motor Drive for Standalone Pump Application." *IEEE Transactions on Industry Applications* 59 (6): 7103–7115. <https://doi.org/10.1109/TIA.2023.3310490>.
- Karimi, H., A. Siadatan, and A. Rezaei-Zare. 2025. "A Hybrid P&O-Fuzzy-Based Maximum Power Point Tracking (Mppt) Algorithm for Photovoltaic Systems under Partial Shading Conditions." *IEEE Access* 13: 86046–86056. <https://doi.org/10.1109/ACCESS.2025.3533314>.
- Katche, M. L., A. B. Makokha, S. O. Zachary, and M. S. Adaramola. 2023. "A Comprehensive Review of Maximum Power Point Tracking (Mppt) Techniques Used in Solar Pv Systems." *Energies* 16: 2206. <https://doi.org/10.3390/en16052206>.
- Khan, A. A., Y. W. Lu, W. Eberle, L. Wang, U. A. Khan, M. Agamy, and H. Cha. 2020. "Single-Stage Bidirectional buck–boost Inverters Using a Single Inductor and Eliminating the Common-Mode Leakage Current." *IEEE Transactions on Power Electronics* 35 (2): 1269–1281. <https://doi.org/10.1109/TPEL.2019.2918349>.
- Maeng, J., J. Jeong, I. Park, M. Shim, and C. Kim. 2024. "A Time-Based Direct MPPT Technique for Low-Power Photovoltaic Energy Harvesting." *IEEE Transactions on Industrial Electronics* 71 (5): 5375–5380. <https://doi.org/10.1109/TIE.2023.3288183>.
- Mamouri, L., T. Mesbahi, and V. Frick. 2024. "MPPT Technique with Improved FOCV Control Applied to a Full Active Rectifier for Low-Voltage Piezoelectric Energy Harvesting." *IEEE Transactions on Circuits and Systems II: Express Briefs* 71 (2): 532–536.
- Manna, S., D. K. Singh, A. K. Akella, H. Kotb, K. M. AboRas, H. M. Zawbaa, and S. Kamel. 2023. "Design and Implementation of a New Adaptive Mppt Controller for Solar Pv Systems." *Energy Reports* 9: 1818–1829. <https://doi.org/10.1016/j.egy.2022.12.152>.
- Mazumdar, D., C. Sain, P. K. Biswas, P. Sanjeevikumar, and B. Khan. 2024. "Overview of Solar Photovoltaic Mppt Methods: A State of the Art on Conventional and Artificial Intelligence Control Techniques." *International Transactions on Electrical Energy Systems* 2024: 8363342. <https://doi.org/10.1155/2024/8363342>.
- Melhaoui, M., M. Rhiat, M. Oukili, I. Atmane, K. Hirech, B. Bossoufi, M. M. Almalki, T. A. H. Alghamdi, and M. Alenezi. 2025. "Hybrid Fuzzy Logic Approach for Enhanced Mppt Control in Pv Systems." *Scientific Reports* 15: 19235. <https://doi.org/10.1038/s41598-025-03154-w>.
- Mukhatov, A., N. G. M. Thao, and T. D. Do. 2022. "Linear Quadratic Regulator and Fuzzy Control for Grid-Connected Photovoltaic Systems." *Energies* 15 (4): 1286. <https://doi.org/10.3390/en15041286>.
- Muñoz-Rodríguez, F. J., J. L. Sánchez-Jiménez, C. Robles, C. Ventura, C. Gilabert, C. Rus, and G. Jiménez-Castillo. 2026. "String Level Dc Performance Analysis in Rooftop Photovoltaic Systems with Multi-Mppt Inverters." *Solar Energy* 308: 114359. <https://doi.org/10.1016/j.solener.2026.114359>.
- Necaibia, S., A. Laib, B. Kanouni, and A. Krama. 2026. "Experimental Validation of a Region-Segmented Mppt with Disturbance Detection and Lyapunov-Based Damping for Enhanced Pv Energy Extraction." *Electric Power Systems Research* 258: 113056. <https://doi.org/10.1016/j.epsr.2026.113056>.
- Nwachukwu, S. E., K. A. Folly, and K. O. Awodele. 2025. "A Comparative Study between Soft Actor-Critic (Sac) and Deep Deterministic Policy Gradient (Ddpq) Algorithms for Solar Pv Mppt Control under Partial Shading Conditions." *IEEE Access* 13: 71738–71754. <https://doi.org/10.1109/ACCESS.2025.3561807>.
- Pires, V. F., A. Cordeiro, D. Foito, A. J. Pires, H. Chen, J. F. Martins, and R. Castro. 2023. "A Cascaded Dual Four-Leg Inverter for Photovoltaic Systems with Capability to Compensate Unbalanced Distribution Networks." *IEEE Journal of Emerging and Selected Topics in Industrial Electronics* 4 (3): 960–968. <https://doi.org/10.1109/JESTIE.2023.3262449>.
- Poothullil, N. G., and A. Saleem. 2026. "Pandagonal Magic Square Reconfiguration of Partially Shaded Photovoltaic Arrays for Simplified Mppt and Power Enhancement." *IEEE Access* 14: 23194–23210. <https://doi.org/10.1109/ACCESS.2026.3661978>.
- Reguieg, Z., F. Mehedi, I. Bouyakoub, W. M. Kacemi, F. Saidi, and S. Mekhilef. 2026. "Ann-Based Pv-Integrated Power Conditioning System for Mppt Optimization and Power Quality Enhancement in Hybrid Microgrids." *Energy Reports* 15: 109129. <https://doi.org/10.1016/j.egy.2026.109129>.

- Sahu, P., and R. Dey. 2022. "Maximum Power Point Tracking Using Adjustable Gain Based Model Reference Adaptive Control." *Journal of Power Electronics* 22 (1): 138–150. <https://doi.org/10.1007/s43236-021-00336-3>.
- Türkey, Y., and A. G. Yüsek. 2025. "Investigating the Potential of an Anfis-Based Maximum Power Point Tracking Controller for Solar Photovoltaic Systems." *IEEE Access* 13: 41768–41784. <https://doi.org/10.1109/ACCESS.2025.3547954>.
- Ullah, K., M. Ishaq, F. Tchier, H. Ahmad, and Z. Ahmad. 2023. "Fuzzy-Based Maximum Power Point Tracking (Mppt) Control System for Photovoltaic Power Generation System." *Results in Engineering* 20: 101466. <https://doi.org/10.1016/j.rineng.2023.101466>.
- Xiang, Q., L. Xu, C. Yuan, and J. Cai. 2026. "An Endogenous Adaptive Slime Mould Algorithm for Curved Photovoltaic Systems Mppt under Dynamic Self-Shading." *Energy Conversion and Management* 354: 121289. <https://doi.org/10.1016/j.enconman.2026.121289>.
- Xiao, W., N. Dong, and K. He. 2025. "A Hybrid Global Wolf Pack Algorithm-Based Incremental Conductance Method under Partial Shading Conditions." *Solar Energy* 291: 113388. <https://doi.org/10.1016/j.solener.2025.113388>.
- Xu, D., H. Chen, X. Wang, V. Pires, J. Martins, A. Anuchin, X. Li, R. Palka, and J. Gu. 2023. "Coupling Analysis of Differential Power Processing-Based PV System and Its Decoupling Implementation of Synchronous MPPT Control." *IEEE Transactions on Industrial Electronics* 70 (7): 6973–6983. <https://doi.org/10.1109/TIE.2022.3201277>.
- Yue, X., S. Javvaji, Z. Tang, K. A. A. Makinwa, and S. Du. 2024. "A Bias-Flip Rectifier with Duty-Cycle-Based MPPT for Piezoelectric Energy Harvesting." *IEEE Journal of Solid-State Circuits* 59 (6): 1771–1781. <https://doi.org/10.1109/JSSC.2023.3313733>.
- Zanib, N., M. Batool, S. Riaz, and F. Nawaz. 2022. "Performance Analysis of Renewable Energy Based Distributed Generation System Using ANN Tuned Upqc." *IEEE Access* 10: 110034–110049. <https://doi.org/10.1109/ACCESS.2022.3213948>.

Appendix

Table A1. Summary of principal symbols, converter parameters, adaptive gains, and reinforcement-learning variables used in the proposed QGFSMC MPPT framework.

Symbol	Meaning	Value/Range	Description
V_{pv}	PV voltage	Dynamic	Measured photovoltaic input voltage.
I_{pv}	PV current	Dynamic	Measured photovoltaic array current.
P_{pv}	PV power	$V_{pv}I_{pv}$	Extracted PV power used for MPPT tracking.
V_{oc}	Open-circuit voltage	184 V	Maximum PV voltage at zero current.
I_{sc}	Short-circuit current	80.8 A	Maximum PV current at zero voltage.
V_{mpp}	MPP voltage	150 V	Voltage corresponding to maximum power point.
I_{mpp}	MPP current	73.5 A	Current corresponding to maximum power point.
D	Duty cycle	$0.05 \leq D \leq 0.95$	Boost-converter PWM switching ratio.
L	Converter inductance	2 mH	Energy-storage inductance reducing current ripple.
C_{in}	Input capacitor	470 μ F	Stabilises PV-side voltage ripple.
C_{out}	Output capacitor	470 μ F	Stabilises DC output voltage.
R	Load resistance	20 Ω	Equivalent converter output load.
dt	Simulation step size	5×10^{-5} s	Numerical integration step used in simulation.
T_{ctrl}	Control update period	1×10^{-3} s	Sampling period of MPPT controller.
$e^{(k)}$	Tracking error	Dynamic	Difference between reference and measured PV voltage.
$\dot{e}^{(k)}$	Error derivative	Dynamic	Rate of change of tracking error.
$e_i^{(k)}$	Integral error	Dynamic	Accumulated tracking deviation.
$e_n^{(k)}$	Normalised error	$[-1, 1]$	Scaled tracking error for fuzzy/RL stability.
$\dot{e}_n^{(k)}$	Normalised derivative error	$[-1, 1]$	Scaled derivative error for transient analysis.
$e_i^{(k)}$	Normalised integral error	$[-1, 1]$	Scaled accumulated error signal.
$u_f^{(k)}$	Fuzzy output	Dynamic	Raw fuzzy-controller correction signal.
$u_{f,f}^{(k)}$	Filtered fuzzy output	Dynamic	Smoothed fuzzy-control action.
$s^{(k)}$	Sliding surface	Dynamic	Combined PID-type sliding variable.
K_f	Fuzzy gain	Adaptive	Controls fuzzy correction strength.
K_s	Sliding gain	Adaptive	Controls sliding-mode aggressiveness.
K_i	Integral gain	Adaptive	Controls integral compensation strength.
w_{far}	Distance weighting	$0 \rightarrow 1$	Indicates distance from MPP operating point.
ΔD	Duty increment	Dynamic	Duty-cycle correction increment.
α_d	Duty smoothing factor	$0.05 \rightarrow 0.95$	Low-pass filtering strength of duty command.
$Q(s, a)$	Q-table value	Adaptive	Reinforcement-learning state-action quality estimate.
α_{QL}	Learning rate	0.35	RL learning update speed.
γ_{QL}	Discount factor	0.95	Importance weighting of future rewards.
ϵ	Exploration factor	$0.25 \rightarrow 0.02$	Exploration-exploitation balancing parameter.
mK_f	Fuzzy gain multiplier	$0.35 \rightarrow 3.20$	Adaptive fuzzy gain scaling factor.
mK_s	Sliding gain multiplier	$0.45 \rightarrow 2.80$	Adaptive sliding-gain scaling factor.
mK_i	Integral gain multiplier	$0.70 \rightarrow 1.50$	Adaptive integral-gain scaling factor.
mD_{lim}	Duty increment multiplier	$0.35 \rightarrow 3.20$	Maximum allowable duty-cycle movement factor.

Table A2. Explanation of weighting coefficients, normalisation constants, adaptive smoothing laws, and reinforcement-learning tuning parameters used in the proposed controller.

Coefficient/ Parameter	Meaning	Typical Value/Range	Purpose and Physical Interpretation
0.70, 0.30 in V_{ref}	Reference smoothing coefficients	70% previous + 30% new	Low-pass filtering of MPP reference. Larger previous-value weight reduces abrupt reference jumps and suppresses oscillation during irradiance changes.
0.55, 0.45 in $V_{pv,f}$	Measurement filter coefficients	55% previous + 45% measured	Smooths PV voltage measurement while preserving adequate dynamic response.
0.45, 0.55 in $\dot{e}_f$	Derivative filter coefficients	Weighted smoothing	Reduces high-frequency noise amplification caused by derivative computation.
0.99 in e_i update	Integral memory coefficient	99% previous memory	Provides slow decay of integral accumulation while preventing uncontrolled integrator growth.
0.20 before de_n	Derivative weighting factor	Empirically tuned	Gives moderate influence to transient dynamics without making the controller overly sensitive to noise.
0.10 before e_n^i	Integral weighting factor	Empirically tuned	Prevents excessive integral dominance while still improving steady-state accuracy.
2.6 in $\tanh(2.6s)$	Sliding sensitivity factor	Fixed nonlinear slope	Controls steepness of the hyperbolic tangent transition. Larger value increases switching aggressiveness and disturbance rejection.
14.0 in reward	Error penalty weight	Strong penalty	Makes RL prioritise rapid reduction of tracking error.
1.5 in reward	Error-derivative penalty	Moderate penalty	Penalises excessive transient oscillation and rapid voltage variation.
0.3 in reward	Integral-error penalty	Small penalty	Prevents long-term accumulated steady-state offset.
60.0 in reward	Duty-cycle activity penalty	Strong penalty	Suppresses excessive duty-cycle chattering and aggressive switching activity.
0.82 $\rightarrow$ 0.62 w_{far}	Adaptive smoothing law	Variable smoothing	Provides stronger filtering near MPP and faster response when far from MPP.
0.08 V_{oc} , 0.12 V_{oc}	Normalisation denominators	Adaptive scaling ranges	Normalise signals into $[-1, 1]$ while preserving numerical stability and consistent fuzzy inference operation.
1.1×10^{-3} , 3.5×10^{-3}	Fuzzy gain base/adaptive terms	Adaptive	Small baseline gain ensures stable operation; added weighted term increases aggressiveness during large tracking errors.
1.2×10^{-4} , 4.5×10^{-4}	Sliding gain base/adaptive terms	Adaptive	Sliding gain increases when operating point moves far from MPP to improve robustness.
2.0×10^{-4} , 2.8×10^{-3}	Duty-limit coefficients	Adaptive bound	Restricts excessive duty-cycle jumps while still allowing fast transient correction.