ARTICLE | Article in Press

Fuzzy logic based hybrid load balancing for solar–grid supply with tiered load shedding: a simulation study

Jimmy Nabende Wanzala · Michael Byamukama

Received: 20 July 2026 / Accepted: 13 October 2026
© The Author(s) 2026

Abstract

Hybrid installations that combine a solar inverter and battery with the utility grid are increasingly common in Uganda, yet they are typically managed by threshold-based automatic transfer switches (ATS) that ignore solar availability, grid quality, and load criticality. This paper presents the design, modelling, and *simulation-based* evaluation of a multi-input fuzzy logic based hybrid load balancing controller. The framework fuses four inputs (solar output, battery state of charge (SOC), grid status, and load demand) through a Mamdani-type fuzzy inference system to produce two coordinated outputs: a source-priority signal and a tiered load-shedding signal spanning three load classes (critical, flexible, and non-essential). The study is entirely simulation-based. The system was implemented in Python at one-minute resolution over a synthetic but representative 24-hour scenario featuring two grid outages, a brownout, and cloud-induced solar variability, and was benchmarked against a conventional threshold-based transfer switch, an ATS with critical-load priority, and a crisp rule-based tiered controller; it was further tested on seven stress cases and a 100-run Monte Carlo evaluation. In the headline scenario the fuzzy controller sustained 100% availability of critical load against 91.9% for the baseline, which suffered 67 minutes of complete blackout; raised solar utilisation from 47.1% to 100%; cut grid energy from 13.43 kWh to 7.95 kWh (40.8%); and reduced source-switching events from four to two. These gains came from deliberately curtailing all non-essential load and 4.6% of flexible load, so they are not a pure efficiency gain: total unserved energy rose from 1.84 to 3.18 kWh, although a criticality-weighted reliability index still favoured the fuzzy controller (96.4% versus 92.6%). Broader testing confirmed the solar-utilisation and grid-energy advantages over the threshold baseline, but showed that an ATS with simple critical-load priority protected critical load at least as well under stress, identifying rule-base tuning as the main need. The results support a simulation-based proof of concept. A real-time embedded demonstration, in which the controller and plant model ran on a Raspberry Pi 5 with an external touch display using emulated inputs, reproduced the simulated behaviour; testing with live sensors and relays and field validation remain future work.

Keywords Fuzzy logic · Hybrid energy systems · Solar photovoltaic · Load shedding · Energy management · Mamdani inference

This Article in Press is shared early to give you faster access to new research. It is citable and carries a permanent DOI. The final edited version will replace it automatically.

https://doi.org/10.1038/s41598-026-76485-x

Published online: 21 October 2026

scientific reports

Introduction

Background

Small solar photovoltaic installations paired with battery storage and a utility grid connection have become a standard response to unreliable grid supply in East Africa. In Uganda, distribution-level interruptions and scheduled load shedding remain frequent, and rooftop solar adoption continues to grow as households and small institutions seek to reduce their dependence on the national grid. As solar and grid sources become more commonly combined at the same premises, the manner in which the two are coordinated becomes an important determinant of both system reliability and cost. The national distribution network was operated by Umeme Limited (UMEME) until March 2025 and since April 2025 by the Uganda Electricity Distribution Company Limited (UEDCL).

In most installations, coordination between the solar inverter and the grid is delegated to an automatic transfer switch (ATS) or the inverter's built-in changeover logic, which acts on fixed thresholds: typically, the inverter carries the load until battery voltage crosses a set point, at which time the system transfers to the grid, or the entire supply is lost if the grid is absent. This threshold-based behaviour is simple to implement but takes no account of the surrounding operating context.

Related work

Fuzzy logic control (FLC) has been extensively applied to power and energy systems where measurements are imprecise, system behaviour is non-linear, and expert-style reasoning is more tractable than a full analytical model[1,2]. Reviews of fuzzy applications in renewable energy systems report widespread use across solar, wind, micro-grid, and hybrid configurations[3–5]. For grid-connected PV-battery systems specifically, fuzzy energy management system (EMS) design has been used to regulate battery storage and grid-tie power flow[6], to supervise stand-alone wind/PV/hydrogen/battery plants[7], to schedule storage against electricity prices[8], and to manage DC-link power balance in hybrid industrial supplies[9]. Prior work by our group has demonstrated fuzzy load balancing across feeders in electric power distribution[10], fuzzy decision systems for safety-critical applications[11], fuzzy irrigation control[12], fuzzy-PI converter control for hybrid energy storage[13], and multi-input switching architectures[14].

On the load side, computational-intelligence load-shedding schemes have been reviewed extensively[15], with fuzzy-analytic hierarchy ranking of loads[16], neural under-frequency shedding[17], multi-criteria shedding for industrial systems[18], smart-load-based under-frequency schemes[19], and shedding strategies for isolated multi-microgrids[20]. These works establish that *which* load is shed matters as much as *how much*; however, they target utility- or microgrid-scale frequency events rather than premises-scale source arbitration. In the East African context, frequent outages impose large backup costs and emissions[21], and the cost of achieving high reliability from decentralised solar alone is substantial[22], motivating controllers that squeeze more reliability from modest storage[23].

Complementary studies address the converter-level and grid-support layers that sit beneath the energy-management layer considered here. Sridhar *et al.*[24] design a modular neural-network PI-controller ensemble with Karush–Kuhn–Tucker conditions for grid-connected PV systems under ground-fault conditions[25]; presents an optimally tuned super-twisting sliding-mode controller for active and reactive power control in a grid-interfaced PV system[26]; improves the grid-current quality of a grid-connected PV inverter using an adaptive hysteresis-band current controller; and[27] proposes a real-time coordinated hydro governor with dual STATCOM for low-frequency oscillation damping, tuned by a hybrid mean-DE and TVAPSO algorithm. These works operate on dynamic time scales (milliseconds to seconds) and target fault response, power quality, or oscillation damping of a grid-connected converter or network. The controller proposed here instead acts as a minute-scale supervisory layer that decides which source supplies the premises and which load tiers are permitted when sources are constrained; none of the four studies addresses premises-level source arbitration or tiered load shedding. The two

layers are therefore complementary: the fuzzy supervisor issues source and load-tier commands, while converter-level controllers of the kinds above would regulate inverter current, power quality, and fault response beneath it. The metaheuristic tuning used in[27] also offers a template for the systematic rule-base optimisation identified as future work in Section IV-D.

However, existing hybrid solar-grid installations in practice typically retain simple ATS logic rather than any form of intelligent arbitration. Threshold logic has three structural weaknesses that motivate the present work. First, it is blind to context: the decision to draw down the battery at mid-day under strong sunshine and the same decision in the evening before an anticipated outage are treated identically. Second, it treats supply as all-or-nothing: when stored energy is exhausted during an outage, critical loads such as refrigeration, lighting, and communications fail together with decorative or deferrable loads. Third, abrupt threshold crossings produce unnecessary transfer events and, in marginal conditions, oscillation.

Problem statement

Hybrid solar-grid installations are limited by rigid, threshold-based transfer switching that treats the grid as either fully available or fully absent, and that does not distinguish between load categories when a shortfall occurs. This results in inefficient use of available solar energy, unnecessary reliance on an unstable grid, and no structured way to protect critical loads when both sources are constrained simultaneously. There is therefore a need for an intelligent controller that continuously evaluates solar output, battery state of charge, and grid stability together, and produces a graduated response rather than a binary switching decision.

Contributions of this work

This study's primary novelty lies in coupling source-selection arbitration with an explicit three-tier load-shedding policy within a single fuzzy inference system, together with a reproducible simulation methodology reflecting Ugandan grid conditions. The key contributions are:

1. *Design of a coupled source-and-tier fuzzy controller*: a four-input, two-output Mamdani fuzzy logic system that jointly decides source priority (solar/battery versus grid) and load-tier permission (critical, flexible, non-essential), rather than treating these as independent problems.
2. *A reserve-conservation rule*: the rule base explicitly protects battery reserve during stable-grid periods in anticipation of outages, a behaviour not present in conventional ATS logic.
3. A *reproducible 24-hour simulation benchmark*: a one-minute-resolution energy-balance simulation featuring synthetic but representative Ugandan outage patterns (including extended evening load shedding), used to quantitatively compare the proposed controller against a conventional threshold-based baseline across service continuity, solar utilisation, grid dependence, and switching behaviour.
4. *Broader benchmarking and testing*: two additional benchmarks (an ATS with critical-load priority and a crisp rule-based tiered controller), a criticality-weighted reliability metric, seven stress cases, and a 100-run Monte Carlo evaluation, so that the benefit of fuzzy arbitration can be separated from that of tiered shedding.

Materials and methods

Design of the fuzzy logic based hybrid load balancing system circuit

Figure 1 shows a system built around four sensing channels: solar inverter current and voltage (ACS712 and ZMPT101B sensors), utility grid voltage (ZMPT101B), battery state of charge (INA219 monitor), and aggregate load current (ACS712). Conditioned signals are digitised by the analogue-to-digital converter (ADC) of an

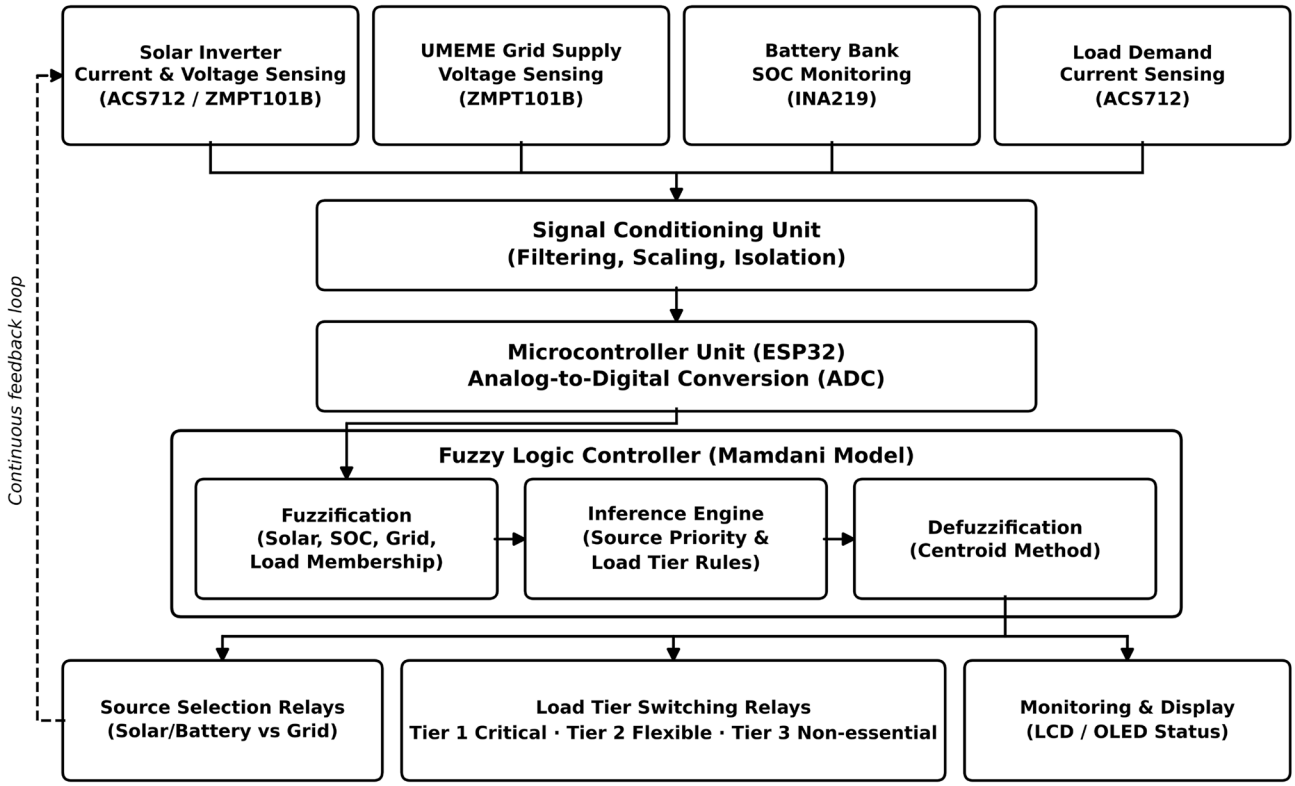


Fig. 1 Block diagram of the fuzzy logic based hybrid load balancing system (redrawn at higher resolution). Sensing channels: solar inverter current and voltage (ACS712 Hall-effect current sensor and ZMPT101B voltage transformer), which give the solar input S; utility grid RMS voltage (ZMPT101B), which is classified into the grid-status input G (Absent, Unstable, Stable); battery voltage and current (INA219 monitor), from which the state of charge B is estimated; and aggregate load current (ACS712), which gives the demand input L. After signal conditioning, the ESP32 digitises the four channels and runs the Mamdani controller (fuzzification, inference, centroid defuzzification). Relay groups: the source-selection relays connect the premises to the solar/battery subsystem or to the grid according to the source priority P^*; the three load-tier switching relays connect or shed the Tier 1 (critical), Tier 2 (flexible), and Tier 3 (non-essential) load groups according to the tier permission T^*. The display unit shows the live status, and the dashed line marks the continuous feedback loop through the plant.

ESP32 (Espressif Systems) microcontroller, which executes the fuzzy inference engine. Two actuation groups realise the controller's decisions: source-selection relays that route supply from either the solar/battery subsystem or the grid, and three load-tier relays that connect or shed the critical, flexible, and non-essential load groups. An LCD/OLED unit displays real-time status, and the control loop operates continuously.

Loads are grouped into three tiers. Tier 1 (critical) comprises lighting, refrigeration, and network equipment; it is served first from whichever source has capacity and is shed only as a last resort. Tier 2 (flexible) comprises deferrable loads such as the water pump, water heater, and device charging; these run opportunistically when solar and battery conditions are favourable. Tier 3 (non-essential) comprises decorative and secondary loads permitted only when solar, battery, and grid are all in a favourable state.

Fuzzy controller design

All inputs are normalised to [0, 1]. Solar output, battery SOC, and load demand each carry three triangular membership functions (Low, Medium, High); grid status carries Absent, Unstable, and Stable, derived from measured grid voltage quality. The first output, source priority $P \in [0,1]$, expresses preference between grid supply ($P \to 0$) and solar/battery supply ($P \to 1$), with an intermediate Blend term. The second output, tier permission $T \in [0,3]$, carries the terms Tier 1 only, Tiers 1–2, and All tiers. The membership functions are shown in Fig. 2. Triangular

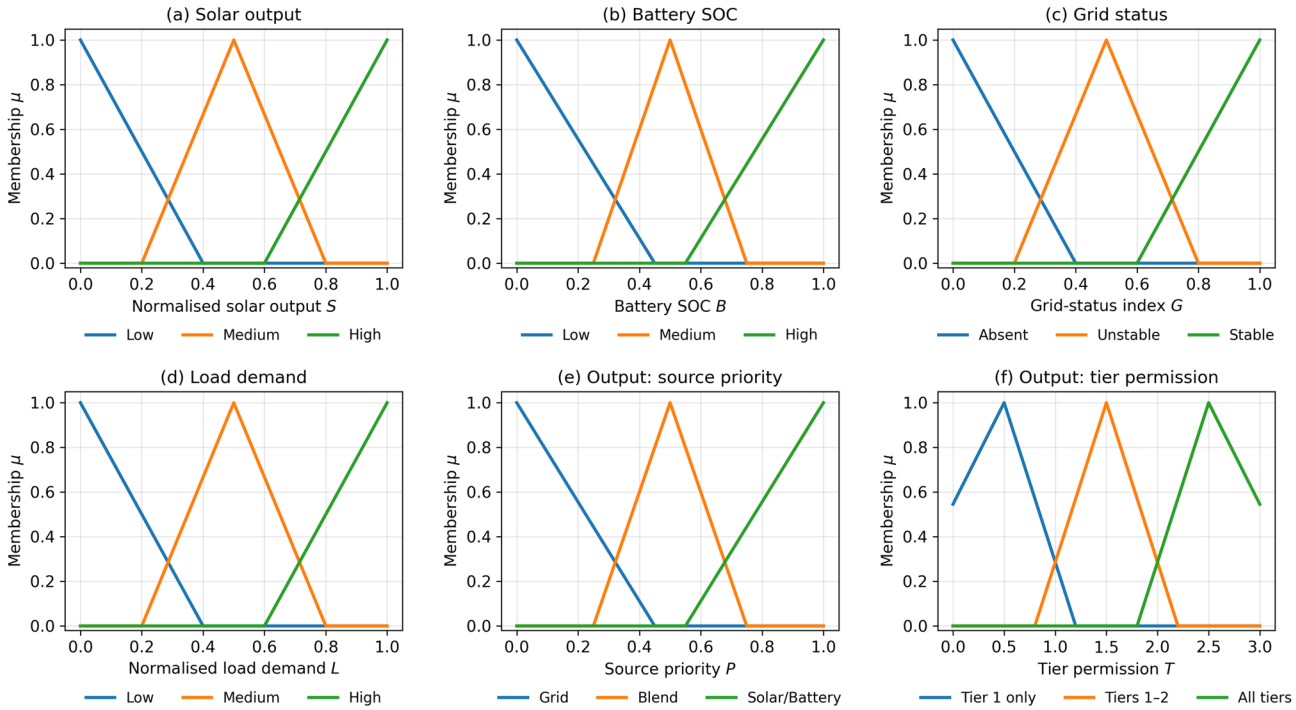


Fig. 2 Membership functions of the four inputs (**a**–**d**) and the two outputs (**e**–**f**). The exact numerical breakpoints are given in Table 1.

Table 1 Membership-function parameters (a, b, c) of Eq. (1) for all inputs and outputs (normalised universes).

Variable (universe)	Term 1	Term 2	Term 3
Solar output S [0, 1]	Low (−0.4, 0, 0.4)	Medium (0.2, 0.5, 0.8)	High (0.6, 1.0, 1.4)
Battery SOC B [0, 1]	Low (−0.4, 0, 0.45)	Medium (0.25, 0.5, 0.75)	High (0.55, 1.0, 1.4)
Grid status G [0, 1]	Absent (−0.4, 0, 0.4)	Unstable (0.2, 0.5, 0.8)	Stable (0.6, 1.0, 1.4)
Load demand L [0, 1]	Low (−0.4, 0, 0.4)	Medium (0.2, 0.5, 0.8)	High (0.6, 1.0, 1.4)
Source priority P [0, 1]	Grid (−0.4, 0, 0.45)	Blend (0.25, 0.5, 0.75)	Solar/Battery (0.55, 1.0, 1.4)
Tier permission T [0, 3]	Tier 1 only (−0.6, 0.5, 1.2)	Tiers 1–2 (0.8, 1.5, 2.2)	All tiers (1.8, 2.5, 3.6)

functions were chosen for their negligible computational cost on the target ESP32 platform. Each membership function is defined by

$$\mu(x;a,b,c) = \max\Big(\min\Big(\frac{x-a}{b-a}, \frac{c-x}{c-b}\Big), 0\Big) \tag{1}$$

where a, b, and c denote the left foot, peak, and right foot of the triangle, respectively. The numerical parameters of every membership function are listed in Table 1.

Input normalisation and grid classification. The crisp inputs are formed as

$$\begin{aligned} S &= \min\Big(\frac{P_{\text{sol}}}{P_{\text{sol,max}}}, 1\Big), \qquad B = \text{SOC}, \\ L &= \min\Big(\frac{P_{\text{d1}} + P_{\text{d2}} + P_{\text{d3}}}{L_{\text{max}}}, 1\Big), \end{aligned} \tag{2}$$

where $P_{\text{sol,max}} = 1.5$ kW is the peak solar output, $\text{SOC} \in [0,1]$ is the battery state of charge, $P_{\text{d}i}$ is the demand of Tier i, and $L_{\text{max}} = 2.0$ kW is the rated maximum aggregate demand of the three tiers. The grid-status input $G \in [0,1]$ is a grid-quality index. In the simulation it takes the value 1.0 for stable supply, 0.5 for the brownout, and 0.0 during an outage; the supply is treated as available to the load when $G \geq 0.4$, and grid charging of the battery is

permitted only when $G \geq 0.9$. For the hardware prototype, G is derived from the RMS grid voltage V_g measured by the ZMPT101B sensor about the 240 V nominal supply: Stable ($G = 1$) for $216 \leq V_g \leq 264$ V ($\pm 10\%$), Unstable ($G = 0.5$) for $180 \leq V_g < 216$ V or $264 < V_g \leq 276$ V, and Absent ($G = 0$) for $V_g < 180$ V or $V_g > 276$ V. The initial battery SOC is 60% for both controllers. All 18 rules carry unit weight. Conflicting rule activations are not resolved by priority ordering: every activated rule clips its consequent set, the clipped sets are combined by the max operator in Eq. (4), and the centroid of Eq. (5) yields a single crisp output. The output universes are discretised with 201 points on [0, 1] for P and 301 points on [0, 3] for T.

The rule base contains 18 rules constructed from operational reasoning; a representative excerpt is given in Table 2, and the complete rule base together with the controller and plant implementation code is provided in the Appendix (Sections VII-A to VII-C). Two design decisions merit emphasis. First, a *reserve-conservation* rule (grid Stable AND battery not High ⇒ prefer Grid) causes the controller to protect stored energy during stable-grid hours in anticipation of outages; because solar generation is always consumed first regardless of the deficit source, this rule costs no solar utilisation. Second, tier permission is decoupled from source priority, allowing the controller to remain on solar/battery during an outage while progressively shedding Tiers 3 and 2 as SOC declines. Inference uses the Mamdani min–max scheme[2]. For the k-th rule with antecedent memberships evaluated at the crisp inputs S (solar), B (SOC), G (grid), and L (load), the firing strength is

$$w_k = \min\left(\mu_{A_k}(S),\, \mu_{B_k}(B),\, \mu_{C_k}(G),\, \mu_{D_k}(L)\right) \tag{3}$$

(for rules with fewer than four antecedents the missing terms are omitted from the minimum) and the aggregated output membership over each output universe u is

$$\mu_{\text{agg}}(u) = \max_k \; \min\left(w_k,\, \mu_k(u)\right). \tag{4}$$

where $\mu_k(u)$ is the consequent membership function of rule k. Both outputs are defuzzified by the centroid method,

$$u^* = \frac{\int u\, \mu_{\text{agg}}(u)\, du}{\int \mu_{\text{agg}}(u)\, du}, \tag{5}$$

evaluated numerically on a discretised universe. The Python implementation in Appendix B follows these equations exactly. The crisp tier permission T^* is quantised as

$$N_{\text{tiers}} = \begin{cases} 3, & T^* \geq 2.0 \\ 2, & 1.0 \leq T^* < 2.0 \\ 1, & T^* < 1.0 \end{cases} \tag{6}$$

and a hysteresis law is applied to the source relay state $\sigma \in \{\text{grid}, \text{solar}\}$ to suppress chattering:

Table 2 Table for fuzzy logic rules based on solar output, battery SOC, and grid status (representative excerpt, 10 of 18 rules).

Solar	SOC	Grid	Consequent
High	Med/High	Any	Src: Solar/Battery
Any	High	Any	Src: Solar/Battery
Any	Any	Absent	Src: Sol./Batt. (forced)
Any	Not High	Stable	Src: Grid (conserve)
Medium	Medium	Any	Src: Blend
High	High	Any	Tiers: All
Any	Medium	Stable	Tiers: All
Any	Low	Stable	Tiers: 1–2
Any	Medium	Absent	Tiers: 1–2
Any	Low	Abs./Unst.	Tiers: 1 only

$$\sigma(t) = \begin{cases} \text{solar}, & P^*(t) > 0.55 \\ \text{grid}, & P^*(t) < 0.45 \\ \sigma(t^-), & \text{otherwise.} \end{cases} \tag{7}$$

Simulation and testing of the hybrid load balancing system

The controller was implemented in Python (NumPy) together with a quasi-static energy-balance model of the plant, executed over a 24-hour horizon at one-minute resolution. The solar profile follows a raised-sine day curve peaking at 1.5 kW with 5% stochastic variation and a cloud event reducing output to 30% between 13:00 and 14:30. The grid profile includes a morning outage (09:30–11:00), a brownout (15:00–15:30), and an extended evening load-shedding event (18:30–22:30) representative of Ugandan distribution practice; this outage profile is a synthetic, representative scenario constructed by the authors and is not derived from measured utility data. The battery is modelled with 4.8 kWh capacity, 90% round-trip efficiency, 0.6 kW maximum charge and 1.2 kW maximum discharge, and a 10% SOC floor. Demand profiles for the three tiers total 20.96 kWh/day against 9.40 kWh of available solar energy. Table 3 summarises the parameters. The controller and plant code are listed in Appendices B and C, and the complete code is available in the public repository given in the Declarations.

At each time step of duration Δt, solar generation is dispatched to the permitted load first,

$$P_{\text{sol,used}} = \min(P_{\text{sol}}, P_{\text{dem}}), \qquad P_{\text{def}} = P_{\text{dem}} - P_{\text{sol,used}} \tag{8}$$

where P_{dem} is the aggregate demand of the tiers currently permitted by Eq. (6), and the deficit P_{def} is allocated to the battery or the grid according to the source priority of Eq. (7), subject to the discharge limit $P_{\text{dis,max}}$ (1.2 kW) and the SOC floor (10%). The battery state of charge evolves as

$$\text{SOC}(t+\Delta t) = \text{SOC}(t) + \frac{\eta\, P_{\text{ch}}\, \Delta t}{E_{\text{cap}}} - \frac{P_{\text{dis}}\, \Delta t}{\eta\, E_{\text{cap}}} \tag{9}$$

clipped to $[0, 1]$, where E_{cap} is the battery capacity, η the one-way efficiency, and $P_{\text{ch}} = \min(P_{\text{surplus}}, P_{\text{ch,max}}, (1 - \text{SOC})E_{\text{cap}}/\Delta t)$ limits charging by available surplus, converter rating ($P_{\text{ch,max}} = 0.6$ kW), and remaining headroom. In the dispatch layer, the battery may additionally be charged from the grid at up to 0.4 kW when $P^* < 0.55$, $\text{SOC} < 0.8$, and $G \geq 0.9$. The two headline metrics are computed as

$$A_1 = \frac{\int P_{\text{served},1}\, dt}{\int P_{\text{dem},1}\, dt} \times 100\% \tag{10}$$

$$U_{\text{sol}} = \frac{\int \left(P_{\text{sol,used}} + P_{\text{ch,sol}}\right) dt}{\int P_{\text{sol}}\, dt} \times 100\% \tag{11}$$

Table 3 Simulation parameters.

Parameter	Value
Horizon/resolution	24 h at 1-minute steps
Scenario data	Synthetic, representative (not measured)
Initial SOC	60% (both controllers)
Peak solar output	1.5 kW (cloud dip to 30%, 13:00–14:30)
Grid outages	09:30–11:00 and 18:30–22:30; brownout 15:00–15:30
Battery capacity/eff.	4.8 kWh/90% round-trip
Charge/discharge limits	0.6 kW/1.2 kW; SOC floor 10%
Tier-1 demand (critical)	0.30 kW base; 0.45 kW evenings (8.03 kWh/day)
Tier-2 demand (flexible)	Pump/heater/charging windows (10.22 kWh/day)
Tier-3 demand (non-ess.)	0.60 kW, 18:30–23:00 (2.71 kWh/day)
Baseline inverter cutoff	SOC 20% (all-or-nothing supply)

where A_1 is Tier-1 availability and U_{sol} is solar utilisation, with the integrals evaluated over the full 24-hour horizon; $P_{served,1}$ and $P_{dem,1}$ are the served and demanded Tier-1 power, and $P_{ch,sol}$ is the solar power used for battery charging.

The baseline against which the controller was compared reproduces a conventional ATS-managed hybrid installation: when the grid is present, all load is served from the grid while solar charges the battery; when the grid is absent, the inverter serves the entire (untiered) load from solar and battery until SOC reaches the 20% inverter cutoff, after which supply is lost entirely until the grid returns. Both controllers were evaluated on identical input profiles using: Tier-1 availability (energy served divided by energy demanded), unserved energy per tier, complete-blackout and degraded-supply durations, solar utilisation (solar energy used directly or stored, divided by solar energy available), grid energy drawn, and the number of source-switching events.

Additional benchmarks, criticality-weighted metric, and test scenarios

To separate the contribution of fuzzy arbitration from that of tiered shedding and reserve conservation, two further benchmarks were simulated on identical inputs. *B2, ATS with critical-load priority*: the grid serves all load when present and solar only charges the battery; during an outage the inverter serves tiers in priority order, shedding Tier 3 below 50% SOC and Tier 2 below 30% SOC, and Tier 1 is served down to the 10% SOC floor; there is no reserve conservation. *B3, crisp rule-based tiered controller*: the same decision structure as the fuzzy controller (prefer solar/battery when the grid is absent or SOC is high, prefer the grid when it is stable and SOC is not high, and grade tier permission by SOC and grid state) implemented with hard thresholds (all tiers at SOC $\geq$ 70%, Tiers 1–2 at 35%–70%, Tier 1 only below 35% when the grid is absent or unstable; Tier 3 withheld below 35% SOC when the grid is stable) and a relay hysteresis band of 70–80% SOC. The thresholds of B2 and B3 are handset and have not been optimised.

Because total unserved energy treats all loads equally, a criticality-weighted unserved energy and a priority-reliability index are defined as

$$\begin{aligned} \mathrm{CWUE} &= \sum_{i=1}^{3} w_i E_i^{\mathrm{uns}}, \\ \mathrm{PRI} &= \left(1 - \frac{\mathrm{CWUE}}{\sum_{i=1}^{3} w_i E_i^{\mathrm{dem}}}\right) \times 100\%, \end{aligned} \tag{12}$$

where E_i^{uns} and E_i^{dem} are the unserved and demanded energy of Tier i and the weights are $(w_1, w_2, w_3) = (10, 3, 1)$, chosen to reflect the strict priority order of the tiers. A sensitivity check with (5, 2, 1) is reported with the results.

Beyond the headline scenario, seven stress cases were simulated: a low-solar day (50% peak output), a high-load day (all tiers +30%), a longer evening outage (17:00–23:00), six repeated 30-minute outages, an evening outage with 20% initial SOC, a prolonged cloud event (20% output, 13:00–17:00) overlapping a 12:30–16:30 outage, and a critical-heavy demand mix (Tier 1 ×1.8, Tier 2 ×0.6). A Monte Carlo evaluation of 100 runs randomised the peak irradiance (40–100% of 1.5 kW), cloud timing, depth, and duration, an optional morning outage (probability 0.8, 0.5–2.5 h), the evening outage start (17:00–20:00) and duration (1.5–6 h), the brownout (probability 0.5), the initial SOC (20–100%), and the demand of each tier (Tier 1 ×0.8–1.3; Tiers 2 and 3 ×0.7–1.3), with a fixed seed. The same random scenario was applied to all four controllers.

Results

Time-domain behaviour

Figure 3 presents the 24-hour trajectories generated by the simulation. During the morning outage (09:30–11:00) both controllers sustain supply, but the fuzzy controller immediately restricts permission to Tiers 1–2, limiting battery draw. Through the stable midday period, the reserve-conservation rule holds the fuzzy system on grid-assisted operation while solar serves load directly and recharges the battery, ending the afternoon near full SOC.

When the four-hour evening load-shedding event begins at 18:30, the fuzzy controller sheds Tier 3 immediately and subsequently defers Tier-2 load, stretching the 4.8 kWh battery across the entire outage: Tier-1 supply remains unbroken and SOC bottoms at 11.5%, just above the floor. The baseline, by contrast, feeds all tiers indiscriminately from the outset of the outage; its discharge-power ceiling degrades supply immediately, since the 1.35 kW aggregate demand exceeds the 1.2 kW inverter limit, and at 21:23 the battery reaches the 20% cutoff, producing a complete blackout, including critical loads, for the final 67 minutes of the outage.

Aggregate comparison

Table 4 and Fig. 4 summarise the aggregate outcomes over the full 24-hour simulation. The fuzzy controller achieved 100% Tier-1 availability against 91.9% for the baseline (0.65 kWh of unserved critical energy), utilised all 9.40 kWh of available solar energy in the simulated 24-hour scenario against 47.1% for the baseline, whose battery saturates mid-morning while load remains on the grid, and drew 40.8% less grid energy (7.95 kWh versus 13.43 kWh). Source switching was smoother (2 events versus 4), reflecting both the hysteresis band and the graded nature of the fuzzy output.

The cost of these gains is explicit and by design: the controller curtailed all 2.71 kWh of non-essential Tier-3 demand and 4.6% of Tier-2 demand during constrained periods, whereas the baseline served 68.2% of Tier 3 before collapsing. Total unserved energy is therefore higher for the fuzzy controller (3.18 kWh versus 1.84 kWh), but it is concentrated entirely in the lowest-value tier; the baseline's smaller total conceals a critical-load blackout. The reduction in grid energy is therefore partly a service sacrifice and not a pure efficiency gain: 2.71 kWh of Tier-3 and 0.47 kWh of Tier-2 demand went unserved. The criticality-weighted metrics of Eq. (12) reflect the control objective better than total unserved energy: CWUE is 4.12 for the fuzzy controller and 8.37 for the baseline, giving PRI values of 96.4% and 92.6% (94.2% and 92.4% with weights (5, 2, 1)).

Additional benchmarks

Table 5 compares the four controllers on the nominal scenario. These runs used the controller and plant code of Appendices B and C, extended with the two benchmarks and a scenario generator. Their nominal-case results agree with Table 4 on Tier-1 availability, blackout duration, minimum SOC, and switching events, and differ by less than 2.2 percentage points in solar utilisation and by at most 0.27 kWh in grid energy. All three tier-aware controllers keep Tier-1 availability at 100%, so tiered shedding alone removes the baseline's critical-load blackout. The ATS with critical-load priority (B2) attains the lowest weighted unserved energy and the highest PRI (97.8%) because it keeps serving Tier 2 and part of Tier 3 from the grid, but it leaves solar energy unused (45.0% utilisation) and draws 13.57 kWh from the grid. The fuzzy and crisp rule-based controllers use essentially all of the solar energy (98.9%) and reduce grid energy by 39–41% relative to B2 (8.22 and 8.02 versus 13.57 kWh), at the cost of withholding almost all Tier-3 demand. In this nominal case the fuzzy and crisp controllers are nearly equivalent in service metrics; the fuzzy controller needs fewer source-switching events (2 versus 8).

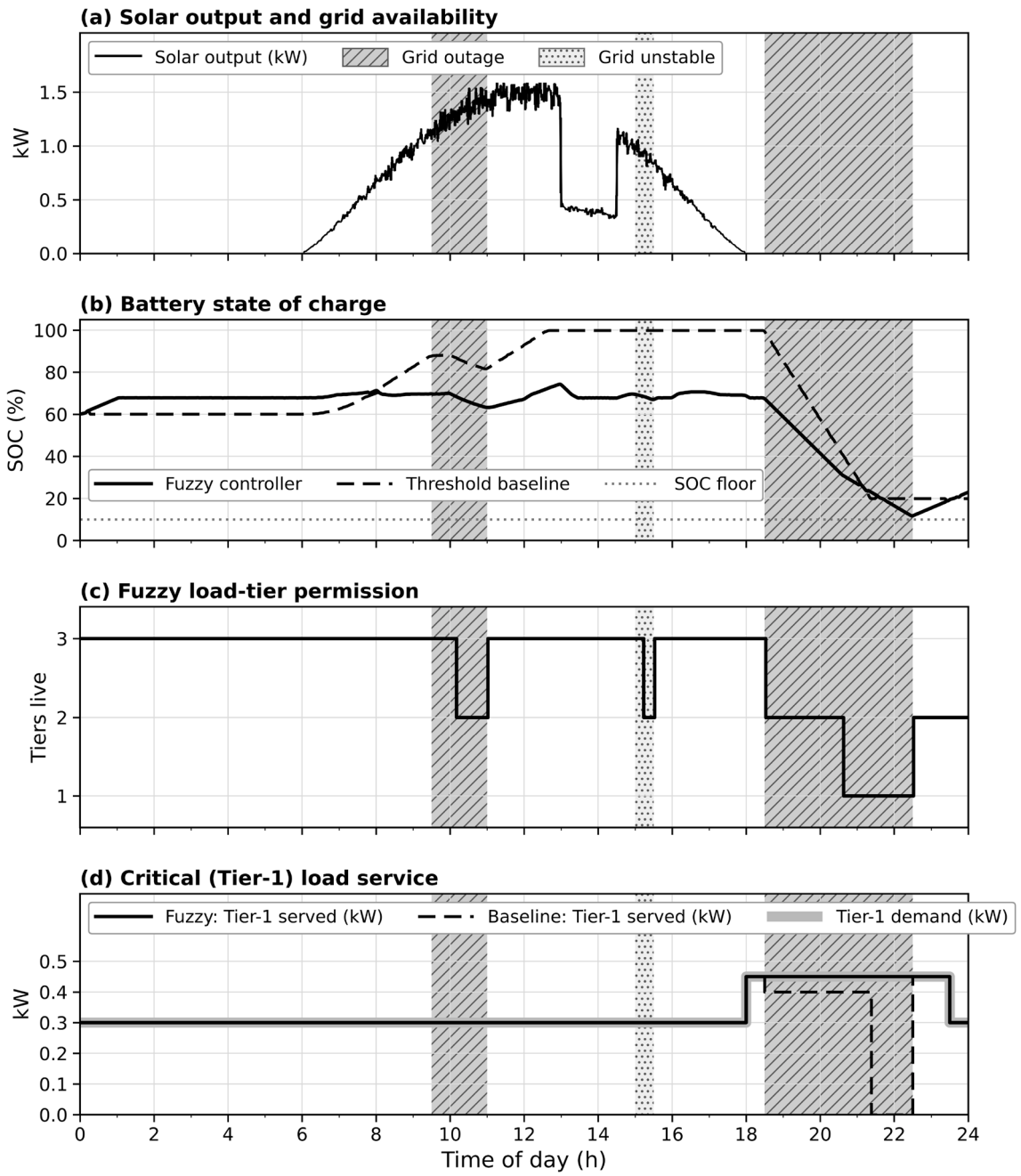


Fig. 3 24-hour simulation: (**a**) solar output and grid availability; (**b**) battery SOC; (**c**) fuzzy tier permission; (**d**) Tier-1 service, showing the baseline blackout after 21:23. Hatched bands mark grid outages (09:30–11:00 and 18:30–22:30) and the dotted band the unstable-grid (brownout) period (15:00–15:30); the bands are repeated in every panel so that each event can be traced through the battery, tier, and Tier-1 responses. Solid lines: fuzzy controller; dashed lines: threshold baseline; grey band in (**d**): Tier-1 demand.

Stress cases

Table 6 reports Tier-1 availability and PRI for the seven stress cases. The fuzzy controller outperformed the threshold ATS on both indicators in all seven cases and kept 100% Tier-1 availability in the low-solar, repeated-outage, and low-initial-SOC cases. It did not, however, protect Tier-1 load as well as B2 in any case in which the

Table 4 Aggregate 24-hour results for the fuzzy controller and the threshold baseline. Tier values are rounded, so they may not sum exactly to the totals; weights (10, 3, 1) are used for the weighted rows.

Metric	Fuzzy	Baseline
Tier-1 availability (%)	100.0	91.9
Tier-1 unserved energy (kWh)	0.00	0.65
Tier-2 unserved energy (kWh)	0.47	0.34
Tier-3 unserved energy (kWh)	2.71	0.86
Total unserved energy (kWh)	3.18	1.84
Crit.-weighted unserved energy	4.12	8.37
Priority-reliability index (%)	96.4	92.6
Complete blackout (min)	0	67
Degraded Tier-1 supply (min)	0	240
Solar utilisation (%)	100.0	47.1
Grid energy drawn (kWh)	7.95	13.43
Tier-2 demand served (%)	95.4	96.7
Tier-3 demand served (%)	0.0	68.2
Source switching events	2	4
Minimum battery SOC (%)	11.5	19.9

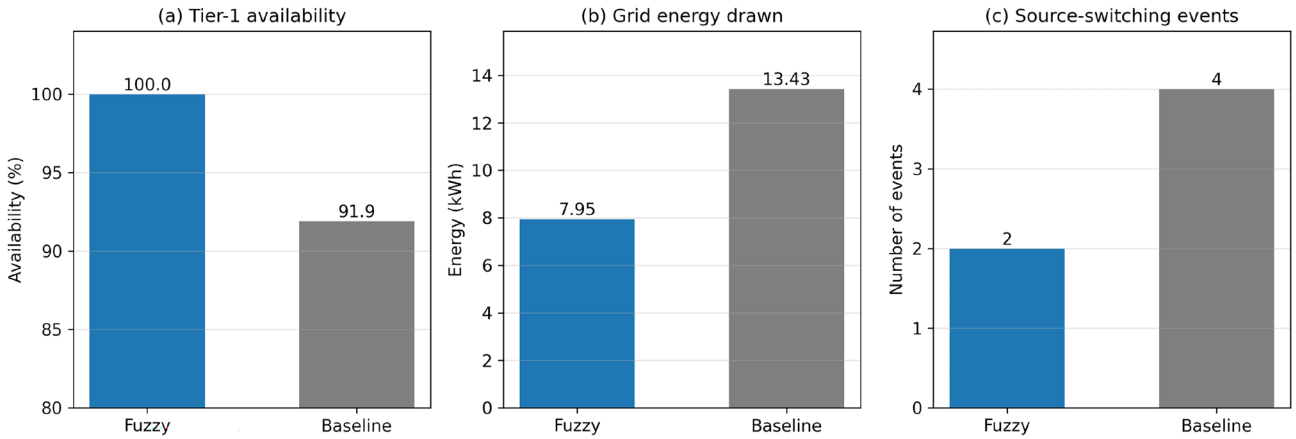


Fig. 4 Key metric comparison between the fuzzy controller and the threshold baseline, shown in separate panels with their own units: (**a**) Tier-1 availability in %, (**b**) grid energy drawn in kWh, (**c**) number of source-switching events.

Table 5 Nominal-scenario comparison of the fuzzy controller, the threshold ATS baseline, the ATS with critical-load priority (B2), and the crisp rule-based tiered controller (B3). CWUE and PRI use weights (10, 3, 1).

Metric	Fuzzy	Threshold ATS	B2: ATS + priority	B3: Crisp tiered
Tier-1 availability (%)	100.0	91.9	100.0	100.0
Tier-2 served (%)	94.5	95.8	97.0	94.2
Tier-3 served (%)	0.0	68.1	41.1	0.3
Total unserved energy (kWh)	3.26	1.94	1.90	3.28
Crit.-weighted unserved energy (weighted kWh)	4.39	8.62	2.52	4.46
Priority-reliability index PRI (%)	96.1	92.4	97.8	96.1
Complete blackout (min)	0	67	0	0
Solar utilisation (%)	98.9	45.0	45.0	98.9
Grid energy drawn (kWh)	8.22	13.57	13.57	8.02
Source-switching events	2	4	4	8
Minimum SOC (%)	11.5	19.9	19.0	14.3

Table 6 Stress cases: Tier-1 availability A_1 (%) and priority-reliability index PRI (%) for F = fuzzy, T = threshold ATS, B2 = ATS with critical-load priority, B3 = crisp rule-based. The last column gives solar utilisation (%) of F/T.

Case	A_1(%)				PRI (%)				U_{sol}F/T
	F	T	B2	B3	F	T	B2	B3	
Low-solar day (50% peak)	100.0	91.9	100.0	100.0	96.1	92.3	97.7	96.2	100/94
High-load day (+30%)	95.4	88.6	100.0	97.2	92.5	89.3	97.1	93.9	100/53
Long evening outage (17:00–23:00)	87.4	82.2	95.4	90.1	84.8	83.3	92.0	86.9	99/46
Repeated short outages (6 x 30 min)	100.0	99.4	100.0	100.0	99.5	99.4	99.9	99.6	99/47
Evening outage, low initial SOC (20%)	100.0	91.9	100.0	100.0	96.1	92.4	97.8	96.1	99/47
Cloud during outage (13:00–17:00 at 20%)	93.7	82.3	93.9	96.9	90.4	83.2	91.0	92.8	99/44
Critical-heavy tiers (T1 x1.8, T2 x0.6)	92.9	89.8	98.5	94.5	91.4	89.8	97.1	92.8	99/42

Table 7 Monte Carlo results (100 randomised scenarios; mean ± standard deviation).

Metric	Fuzzy	Threshold ATS	B2: ATS + priority	B3: Crisp tiered
Tier-1 availability (%)	97.8±3.2	92.7±5.1	99.5±1.2	99.0±2.4
Priority-reliability index PRI (%)	95.3±3.3	93.3±4.7	97.8±1.9	96.3±2.7
Complete blackout (min)	26±36	64±55	5±15	12±27
Solar utilisation (%)	99.8±1.4	50.8±22.4	50.8±22.4	99.6±1.7
Grid energy drawn (kWh)	11.35±3.03	15.18±2.26	15.18±2.26	11.53±3.08
Tier-3 served (%)	17.4±19.4	76.1±18.2	54.1±23.2	22.8±23.6
Source-switching events	3.0±1.3	3.4±0.9	3.4±0.9	11.6±3.0
Runs with full Tier-1 service (%)	50	3	84	68
Runs with any Tier-1 blackout (%)	50	73	16	31

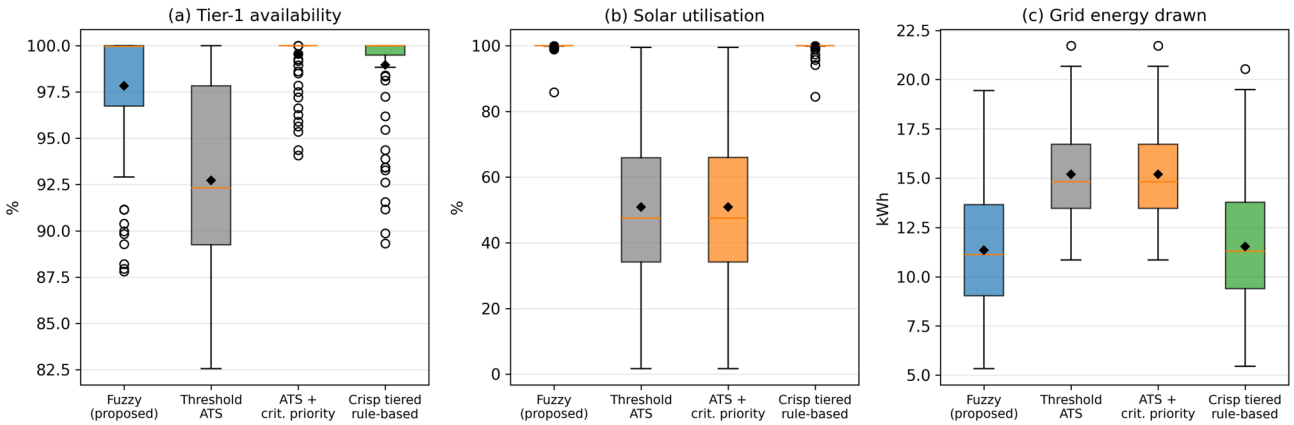


Fig. 5 Monte Carlo distributions over 100 randomised scenarios for the four controllers: (**a**) Tier-1 availability, (**b**) solar utilisation, (**c**) grid energy drawn. Boxes show the interquartile range, the line the median, and the diamond the mean.

battery was stressed (longer outage 87.4% versus 95.4%, cloud during outage 93.7% versus 93.9%, high load 95.4% versus 100%, critical-heavy mix 92.9% versus 98.5%), and the crisp controller B3 matched or exceeded the fuzzy controller in Tier-1 availability in every case. The fuzzy controller's PRI was lower than that of B2 in all seven cases. The fuzzy and crisp controllers retained the solar-utilisation advantage (98–100% versus 42–94% for the ATS-based controllers).

Monte Carlo evaluation

Table 7 and Fig. 5 summarise the 100-run evaluation. Averaged over the randomised scenarios, the fuzzy controller improved on the threshold ATS in Tier-1 availability (97.8% versus 92.7%), complete-blackout time (26

versus 64 min), solar utilisation (99.8% versus 50.8%), and grid energy (11.35 versus 15.18 kWh), and it had a higher PRI in 71% of the runs. It ran without any Tier-1 blackout in 50% of runs, compared with 27% for the threshold ATS, 84% for B2, and 69% for B3. B2 and B3 achieved higher mean Tier-1 availability than the fuzzy controller (99.5% and 99.0%), and the fuzzy controller exceeded B2 in PRI in only 5% of runs and B3 in 4%. The advantages of the fuzzy controller are thus confined to solar utilisation and grid-energy reduction (shared with B3) and to smoother switching relative to B3 (3.0 versus 11.6 events); with the present hand-tuned rule base it does not offer a critical-load-protection advantage over a simple priority rule. This finding motivates the systematic rule-base tuning discussed in Section IV.

Real-time embedded demonstration

To check that the controller runs in real time outside the offline simulation, the fuzzy inference engine, the plant model, and the threshold baseline were executed on a Raspberry Pi 5 single-board computer attached to an external touch display (Fig. 6). The display renders the 24-hour trajectories live together with a status line (time, solar power, load, SOC of both controllers, grid state, the crisp outputs P^* and T^*, the selected source, and the number of live tiers). The solar, grid, and demand inputs were emulated from the profiles of Section II-C, and the 24-hour day was replayed in compressed time with a pause control. In the set-up of Fig. 6 the battery feeds the inverter, which is connected to the Raspberry Pi; in this demonstration, however, all controller inputs (solar, grid, demand, and battery SOC) came from the emulated profiles and plant model of Section II-C rather than from measurements on these units. Fig. 7f in Appendix D shows the display at 22:31, one minute after the evening outage ends: the fuzzy controller has returned to grid supply ($P^* = 0.157$) with two tiers live ($T^* = 1.077$), its SOC is 11.7%, and Tier-1 supply has been continuous, whereas the baseline (SOC 19.9%) shows the blackout that began at about 21:23. These values match the offline results (minimum SOC 11.5% and 19.9%, Table 4). Snapshots at further instants of the day are given in Appendix D. The demonstration confirms real-time execution of the controller logic; it does not use live sensors or switch relays, so sensor noise, relay transfer times, and inverter transients are still not tested.

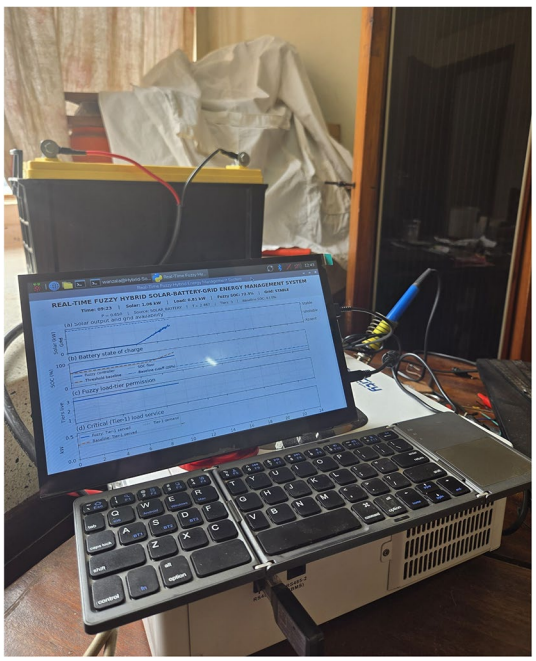


Fig. 6 Real-time embedded demonstration set-up: Raspberry Pi 5 with an external touch display running the fuzzy controller, plant model, and threshold baseline with emulated inputs. The battery feeds the inverter, which is connected to the Raspberry Pi.

Table 8 Comparison of the proposed system with existing fuzzy energy-management approaches.

Work	Context	Grid arbitr.	Battery mgmt.	Tiered shedding	Reserve conserv.
Alankrita et al.[6]	Grid-connected PV-battery	✓	✓	–	–
García et al.[7]	Stand-alone WT/PV/H_2/battery	–	✓	–	–
Mbende et al.[9]	Industrial PV-hydro DC link	–	✓	–	–
Le et al.[16]	Utility-scale load shedding	–	–	✓	–
Atim et al.[10]	Distribution feeder balancing	–	–	–	–
Athari and Ardehali[8]	On-grid hybrid RES, price-driven storage	✓	✓	–	–
Refs.[24–27]	Converter-level/grid-support control	–	–	–	–
Proposed system	Premises solar–utility hybrid	✓	✓	✓	✓

Discussion

Performance of the proposed fuzzy controller in hybrid load balancing

The simulation results demonstrate that the proposed fuzzy controller substantially outperforms the conventional threshold-based baseline in the dimension that matters most for a hybrid solar-grid installation: continuity of supply to critical load during an outage. The reserve-conservation rule is likely to have contributed to this outcome (no ablation of the rule was performed), since it caused the controller to enter the evening outage with a nearly full battery rather than one already partially discharged by daytime grid-assisted operation. The decoupling of tier permission from source priority allowed the controller to make two distinct decisions simultaneously, namely, which source to draw from and how much load to serve, which a single-threshold ATS design cannot express.

The improvement in solar utilisation (100% versus 47.1%) arises primarily because the baseline's battery saturates while the grid is present and serving load, leaving no headroom to absorb additional solar generation, whereas the fuzzy controller's reserve-conservation logic keeps the battery below full charge during stable-grid hours specifically so that it can continue absorbing solar energy. The reduction in grid energy consumption (40.8%) follows directly from this improved solar utilisation and from the tiered shedding strategy, which reduces the total energy that must be supplied from any source during constrained periods. Both results depend on the assumed solar and load profiles, the battery size, the grid behaviour, and the reserve-conservation strategy, and should not be read as a guarantee of 100% solar utilisation in general. The additional benchmarks show that the Tier-1 protection is attributable mainly to tiered shedding rather than to fuzzy logic as such, whereas the solar-utilisation and grid-energy gains stem from the reserve-conservation logic that both the fuzzy and the crisp rule-based controllers implement.

Table 8 situates the proposed controller against representative fuzzy energy-management systems from the literature. Existing grid-connected fuzzy EMS designs regulate battery and grid-tie power flow but treat the connected load as a single indivisible block[6,9]; supervisory fuzzy EMS work for stand-alone hybrid plants manages multiple sources but has no utility grid to arbitrate against[7]; fuzzy load-shedding schemes rank and shed loads but operate at utility scale without premises-level source selection[16]; and prior feeder-balancing work by our group balances load across distribution feeders rather than between source types[10]. To the best of our knowledge, the joint treatment of source arbitration, reserve conservation, and tiered premises-level load shedding within a single fuzzy inference system is the distinguishing feature of the present work. The comparators were chosen to represent the closest published fuzzy energy-management work on grid-connected, stand-alone, and industrial hybrid systems ([6–9]), fuzzy load shedding[16], and fuzzy load balancing[10]. The converter-level and grid-support controllers of[24–27] are listed for contrast: they act at a different layer of the system and are complementary to, rather than competitors of, the supervisory controller proposed here.

Trade-offs and practical implications

The comparison in Table 4 shows that the fuzzy controller's gains are not free: all Tier-3 (non-essential) demand and a small fraction of Tier-2 demand were curtailed during constrained periods, an outcome that the baseline avoided by instead risking (and ultimately suffering) a complete blackout of critical load. This is the intended behaviour of a tiered energy-management policy, and it reframes the comparison from a single aggregate "unserved energy" number toward a more meaningful question of *which* loads are served under stress. A user deploying this system should therefore configure the tier assignment deliberately, since the practical value of the controller depends on critical loads being correctly identified as Tier 1.

Demonstrated results versus future work

Demonstrated by simulation: in the synthetic scenarios studied, the fuzzy controller removed the baseline's critical-load blackout, raised solar utilisation, reduced grid energy, and reduced source switching relative to the threshold ATS. *Demonstrated on a Raspberry Pi 5 test bench:* real-time execution of the controller with emulated inputs, reproducing the simulated trajectories (Section III-F, Appendix D). *Not demonstrated:* superiority over simple priority-based or crisp rule-based controllers in critical-load protection; performance on measured outage and irradiance data; operation with live sensors and relays on the ESP32 prototype, relay transfer times, and sensor noise; and field reliability and user acceptance. These remain future work.

Limitations

The study is simulation-based: sensor noise, relay transfer times, inverter transients, and battery ageing are not modelled, and the demand and outage profiles, while representative, are synthetic. The rule base and membership functions were hand-tuned rather than optimised; systematic tuning, for example using a genetic algorithm or particle swarm optimisation, may yield further gains, following GA-based fuzzy(-PID) tuning demonstrated in related work[28–30]. The headline results correspond to a single synthetic 24-hour scenario; the stress cases and the 100-run Monte Carlo evaluation of Section III extend this but remain synthetic and have not been checked against measured outage or irradiance records. The criticality weights of Eq. (12) are a modelling choice, and the benchmarks B2 and B3 use hand-set thresholds. The real-time demonstration used emulated inputs; hardware validation with live sensors and relays on the ESP32-based prototype specified in Section II is the subject of ongoing work.

Conclusion

A Mamdani fuzzy controller coupling source selection with three-tier load shedding was designed for solar-grid hybrid systems and evaluated in simulation against a conventional threshold-based transfer switch, an ATS with critical-load priority, and a crisp rule-based tiered controller. In a representative 24-hour Ugandan supply scenario the fuzzy controller showed promising simulated improvements over the threshold baseline: uninterrupted critical-load service through a four-hour evening outage that drove the baseline into a 67-minute blackout, full solar utilisation in that scenario (100% versus 47.1%), a 40.8% reduction in grid energy, and fewer source-switching events. These improvements were obtained by deliberately curtailing non-essential and some flexible load. Stress cases and a 100-run Monte Carlo evaluation confirmed the solar-utilisation and grid-energy advantages but showed that, with the present hand-tuned rule base, a simple critical-load-priority rule protected critical load at least as well. The evidence therefore supports a simulation-based proof of concept and not field-validated performance. The low-computation design (triangular membership functions, centroid defuzzification) is suited

to low-cost embedded platforms. The controller has been executed in real time on a Raspberry Pi 5 with emulated inputs, but not yet with live sensors and relays on the ESP32 prototype.

Future work should focus on the practical implementation and further enhancement of the proposed system through several key directions:

- A low-cost hardware prototype should be developed using an ESP32 or similar embedded platform, integrating the specified sensors and implementing the fuzzy controller in real-time firmware.
- Extensive experimental validation should be carried out through controlled laboratory testing and field deployment in Ugandan homes and institutions, to evaluate long-term reliability and user acceptance under real operating conditions.
- The rule base and membership functions should be tuned using an optimisation method such as a Genetic Algorithm[28,29], to further reduce unnecessary Tier-2/3 curtailment while preserving critical-load continuity.
- Day-ahead solar and outage forecasts could be incorporated as additional fuzzy inputs to further improve reserve-conservation decisions.
- Measured outage and irradiance records should replace the synthetic profiles, and the rule base should be re-evaluated against the priority-based benchmarks after tuning.
- The supervisory controller could be combined with converter-level controllers of the types studied in[24–26] in a two-layer architecture, and metaheuristic tuning such as the hybrid DE/TVAPSO approach of[27] could be explored for the rule base.

Appendix

Complete fuzzy rule base

The complete 18-rule base of the controller is listed below. Rules R1–R8 drive the source-priority output P; rules R9–R18 drive the tier-permission output T. AND denotes the min operator (Eq. (3)); OR denotes the max operator. All rules carry unit weight.

R1. IF Solar is High AND SOC is (Medium OR High) THEN Source is Solar/Battery
R2. IF SOC is High THEN Source is Solar/Battery
R3. IF Grid is Absent THEN Source is Solar/Battery (forced)
R4. IF Solar is Low AND SOC is Low AND Grid is Stable THEN Source is Grid
R5. IF SOC is (Low OR Medium) AND Grid is Stable THEN Source is Grid (reserve conservation)
R6. IF Solar is Medium AND SOC is Medium THEN Source is Blend
R7. IF Grid is Unstable AND SOC is Medium THEN Source is Blend
R8. IF Solar is High AND SOC is Low AND Grid is Stable THEN Source is Blend
R9. IF Solar is High AND SOC is High THEN Tiers is All
R10. IF Solar is High AND SOC is Medium THEN Tiers is All
R11. IF SOC is High THEN Tiers is All
R12. IF Grid is Stable AND SOC is Medium THEN Tiers is All
R13. IF Grid is Stable AND SOC is Low THEN Tiers is 1–2
R14. IF Grid is Unstable AND SOC is Medium THEN Tiers is 1–2
R15. IF Grid is Absent AND SOC is Medium THEN Tiers is 1–2
R16. IF Grid is Absent AND SOC is Low THEN Tiers is Tier 1 only
R17. IF Grid is Unstable AND SOC is Low THEN Tiers is Tier 1 only
R18. IF Solar is Low AND SOC is Low AND Load is High THEN Tiers is Tier 1 only

Fuzzy controller implementation Code

The Python (NumPy) implementation of the membership functions (Eq. (1)), Mamdani inference (Eqs. (3)–(4)), and centroid defuzzification (Eq. (5)) used to generate all results in Section III:

```
def tri(x, a, b, c):
    x = np.asarray(x, dtype=float)
    left = (x - a) / (b - a) if b != a else np.ones_like(x)
    right = (c - x) / (c - b) if c != b else np.ones_like(x)
    return np.clip(np.minimum(left, right), 0, 1)

# Input MFs (all normalized 0..1)
def solar_low(x):  return tri(x, -0.4, 0.0, 0.4)
def solar_med(x):  return tri(x, 0.2, 0.5, 0.8)
def solar_high(x): return tri(x, 0.6, 1.0, 1.4)

def soc_low(x):  return tri(x, -0.4, 0.0, 0.45)
def soc_med(x):  return tri(x, 0.25, 0.5, 0.75)
def soc_high(x): return tri(x, 0.55, 1.0, 1.4)

def grid_absent(x):   return tri(x, -0.4, 0.0, 0.4)
def grid_unstable(x): return tri(x, 0.2, 0.5, 0.8)
def grid_stable(x):   return tri(x, 0.6, 1.0, 1.4)

def load_low(x):  return tri(x, -0.4, 0.0, 0.4)
def load_med(x):  return tri(x, 0.2, 0.5, 0.8)
def load_high(x): return tri(x, 0.6, 1.0, 1.4)

# Output universes
P_UNIV = np.linspace(0, 1, 201)      # source priority: 0=grid
    ↪ ... 1=solar/battery
T_UNIV = np.linspace(0, 3, 301)      # tier permission: ~0.5
    ↪ T1only, ~1.5 T1+2, ~2.5 all

def p_grid(u):  return tri(u, -0.4, 0.0, 0.45)
def p_blend(u): return tri(u, 0.25, 0.5, 0.75)
def p_solar(u): return tri(u, 0.55, 1.0, 1.4)

def t_t1(u):  return tri(u, -0.6, 0.5, 1.2)
def t_t12(u): return tri(u, 0.8, 1.5, 2.2)
def t_all(u): return tri(u, 1.8, 2.5, 3.6)

def centroid(univ, agg):
    s = agg.sum()
    return float((univ * agg).sum() / s) if s > 1e-9 else float(
    ↪ univ.mean())

def fuzzy_controller(S, B, G, L):
    sL, sM, sH = solar_low(S), solar_med(S), solar_high(S)
    bL, bM, bH = soc_low(B), soc_med(B), soc_high(B)
    gA, gU, gS = grid_absent(G), grid_unstable(G), grid_stable(G
    ↪ )
    lL, lM, lH = load_low(L), load_med(L), load_high(L)

    # ---- Source-priority rules (antecedent strength, output MF
    ↪ ) ----
    p_rules = [
        (min(sH, max(bM, bH)), p_solar),      # R1 high solar &
    ↪ decent battery -> solar
        (bH,                   p_solar),      # R2 battery high
    ↪  -> solar/battery
        (gA,                   p_solar),      # R3 grid absent
    ↪ -> forced solar/battery
        (min(bL, gS, sL),      p_grid),       # R4 low solar &
    ↪ low battery & stable grid -> grid
        (min(max(bL, bM), gS), p_grid),       # R5 grid stable
    ↪ & battery not high -> conserve reserve, use grid
        (min(sM, bM),          p_blend),      # R6 medium solar
    ↪  & medium battery -> blend
        (min(gU, bM),          p_blend),      # R7 unstable
    ↪ grid & medium battery -> blend
        (min(sH, bL, gS),      p_blend),      # R8 high solar
    ↪ but low battery, stable grid -> blend
    ]
    agg_p = np.zeros_like(P_UNIV)
    for w, mf in p_rules:
        if w > 0:
            agg_p = np.maximum(agg_p, np.minimum(w, mf(P_UNIV)))
    P = centroid(P_UNIV, agg_p)

    # ---- Tier-permission rules ----
    t_rules = [
        (min(sH, bH),          t_all),        # R9 abundant
    ↪ solar+battery -> all tiers
        (min(sH, bM),          t_all),        # R10
        (bH,                   t_all),        # R11 battery
    ↪ high -> all tiers
        (min(gS, bM),          t_all),        # R12 stable grid
    ↪  & medium battery -> all
        (min(gS, bL),          t_t12),        # R13 stable grid
    ↪  but low battery -> defer tier 3
        (min(gU, bM),          t_t12),        # R14 unstable
    ↪ grid, medium battery -> tiers 1-2
        (min(gA, bM),          t_t12),        # R15 outage with
    ↪  medium battery -> tiers 1-2
        (min(gA, bL),          t_t1),         # R16 outage with
    ↪  low battery -> critical only
        (min(gU, bL),          t_t1),         # R17 unstable
    ↪ grid & low battery -> critical only
        (min(sL, bL, lH),      t_t1),         # R18 no solar,
    ↪ low battery, heavy demand -> critical only
    ]
    agg_t = np.zeros_like(T_UNIV)
    for w, mf in t_rules:
        if w > 0:
            agg_t = np.maximum(agg_t, np.minimum(w, mf(T_UNIV)))
    T = centroid(T_UNIV, agg_t)
    return P, T
```

Plant simulation and baseline code

The per-minute dispatch loop implementing the power balance (Eq. (8)) and battery dynamics (Eq. (9)) for the proposed controller, followed by the conventional threshold-based ATS baseline:

```
def run_fuzzy():
    soc = SOC0
    hist = dict(soc=[], P=[], T=[], served1=[], served2=[],
     ↪ served3=[],
                grid_kW=[], solar_used=[], tiers=[])
    for k in range(N):
        S = min(solar[k] / 1.5, 1.0)
        G = grid[k]
        Ld = (tier1[k] + tier2[k] + tier3[k]) / L_MAX
        P, T = fuzzy_controller(S, soc, G, min(Ld, 1.0))
        # crisp tier gate
        if T >= 2.0:    allowed = 3
        elif T >= 1.0: allowed = 2
        else:           allowed = 1
        d1, d2, d3 = tier1[k], tier2[k], tier3[k]
        demand = d1 + (d2 if allowed >= 2 else 0) + (d3 if
     ↪ allowed >= 3 else 0)

        sol_used = min(solar[k], demand)
        deficit = demand - sol_used
        g_kW = 0.0; batt_out = 0.0
        grid_ok = G >= 0.4
        if P >= 0.5:
            can = min(DIS_MAX, max(0.0, (soc - SOC_FLOOR) * CAP
     ↪ / DT))
            batt_out = min(deficit, can)
            deficit -= batt_out
            if deficit > 0 and grid_ok:
                g_kW = deficit; deficit = 0.0
        else:
            if grid_ok:
                g_kW = deficit; deficit = 0.0
            else:
                can = min(DIS_MAX, max(0.0, (soc - SOC_FLOOR) *
     ↪ CAP / DT))
                batt_out = min(deficit, can)
                deficit -= batt_out
        unserved = deficit
        # charging
        surplus = max(0.0, solar[k] - sol_used)
        ch = min(surplus, CH_MAX, (1.0 - soc) * CAP / DT)
        grid_ch = 0.0
        if P < 0.55 and soc < 0.8 and G >= 0.9:
            grid_ch = min(0.4, CH_MAX - ch, (1.0 - soc) * CAP /
     ↪ DT - ch)
            grid_ch = max(0.0, grid_ch)
        soc = soc + (ch + grid_ch) * EFF * DT / CAP - batt_out *
     ↪  DT / (EFF * CAP)
        soc = float(np.clip(soc, 0, 1))
        g_kW += grid_ch

        # record served per tier (proportional shortfall to
     ↪ lowest priority served set)
        s3 = d3 if (allowed >= 3 and unserved <= 1e-9) else (max
     ↪ (0.0, d3 - unserved) if allowed >= 3 else 0.0)
        rem_unserved = max(0.0, unserved - (d3 if allowed >= 3
     ↪ else 0.0) + s3) if allowed >= 3 else unserved
        s2 = d2 if (allowed >= 2 and rem_unserved <= 1e-9) else
     ↪ (max(0.0, d2 - rem_unserved) if allowed >= 2 else 0.0)
        rem2 = max(0.0, rem_unserved - (d2 if allowed >= 2 else
     ↪ 0.0) + s2) if allowed >= 2 else rem_unserved
        s1 = max(0.0, d1 - rem2)

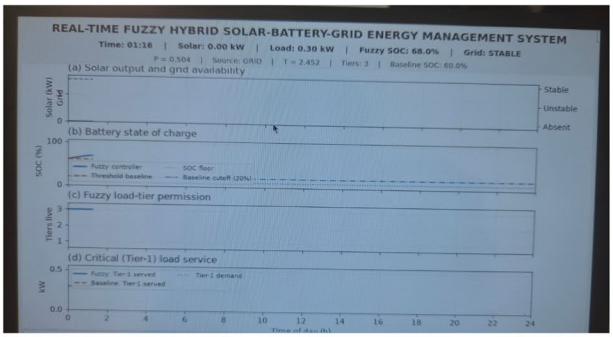
        hist['soc'].append(soc); hist['P'].append(P); hist['T'].
     ↪ append(T)
        hist['served1'].append(s1); hist['served2'].append(s2);
     ↪ hist['served3'].append(s3)
        hist['grid_kW'].append(g_kW); hist['solar_used'].append(
     ↪ sol_used + ch)
        hist['tiers'].append(allowed)
    return {k: np.array(v) for k, v in hist.items()}

def run_baseline():
    """Conventional ATS: grid serves everything when present; on
     ↪  outage inverter serves ALL
      tiers from battery+solar until SOC<0.20 cutoff -> total
     ↪ blackout. Solar only charges battery."""
    soc = SOC0
    hist = dict(soc=[], served1=[], served2=[], served3=[],
     ↪ grid_kW=[], solar_used=[], on_grid=[])
    inverter_lock = False
    for k in range(N):
        d1, d2, d3 = tier1[k], tier2[k], tier3[k]
        demand = d1 + d2 + d3
        grid_ok = grid[k] >= 0.4
        g_kW = 0.0; sol_used = 0.0; s1 = s2 = s3 = 0.0
        if grid_ok:
            inverter_lock = False
            g_kW = demand; s1, s2, s3 = d1, d2, d3
            ch = min(solar[k], CH_MAX, (1.0 - soc) * CAP / DT)
            soc = float(np.clip(soc + ch * EFF * DT / CAP, 0, 1)
     ↪ )
            sol_used = ch
        else:
            if soc <= 0.20: inverter_lock = True
            if not inverter_lock:
                sol_direct = min(solar[k], demand)
                deficit = demand - sol_direct
                can = min(DIS_MAX, max(0.0, (soc - SOC_FLOOR) *
     ↪ CAP / DT))
                batt_out = min(deficit, can)
                served = sol_direct + batt_out
                frac = served / demand if demand > 0 else 1.0
                s1, s2, s3 = d1 * frac, d2 * frac, d3 * frac
                surplus = max(0.0, solar[k] - sol_direct)
                ch = min(surplus, CH_MAX, (1.0 - soc) * CAP / DT
     ↪ )
                soc = float(np.clip(soc + ch * EFF * DT / CAP -
     ↪ batt_out * DT / (EFF * CAP), 0, 1))
                sol_used = sol_direct + ch
            else:
                ch = min(solar[k], CH_MAX, (1.0 - soc) * CAP /
     ↪ DT)
                soc = float(np.clip(soc + ch * EFF * DT / CAP,
     ↪ 0, 1))
                sol_used = ch
        hist['soc'].append(soc); hist['served1'].append(s1);
     ↪ hist['served2'].append(s2)
        hist['served3'].append(s3); hist['grid_kW'].append(g_kW)
        hist['solar_used'].append(sol_used); hist['on_grid'].
     ↪ append(grid_ok)
    return {k: np.array(v) for k, v in hist.items()}
```

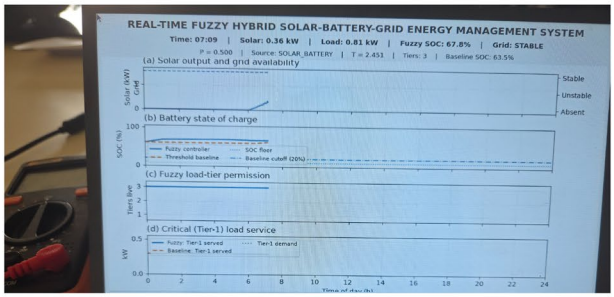
Note: the scenario profiles (solar, grid, and tier demand) and constants used by this code are those of Section II-C and Table 3.

Real-time demonstration snapshots

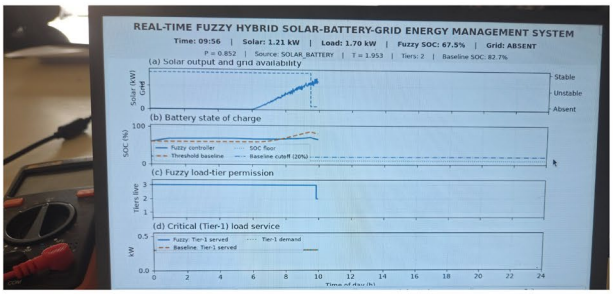
Figure 7 shows the Raspberry Pi 5 display at seven instants of the replayed day. The photographs are reproduced unedited. (a) 01:16: grid stable, all three tiers live, fuzzy SOC rising from the initial 60% by grid charging. (b) 07:09: solar generation begins; source priority at the hysteresis midpoint ($P^* = 0.500$). (c) 09:56: morning outage; source switched to solar/battery and Tier 3 shed ($T^* = 1.953$). (d) 18:32: start of the evening outage; Tier 3 shed while the fuzzy SOC is 67.0% and the baseline SOC 98.6%. (e) 21:34: fuzzy controller on Tier 1 only ($T^* = 0.510$, SOC 21.0%) while the baseline is in blackout. (f) 22:31: grid restored; fuzzy controller back on grid supply with two tiers live (SOC 11.7%), baseline SOC 19.9% after its blackout. (g) End of the run: the complete 24-hour trajectories.



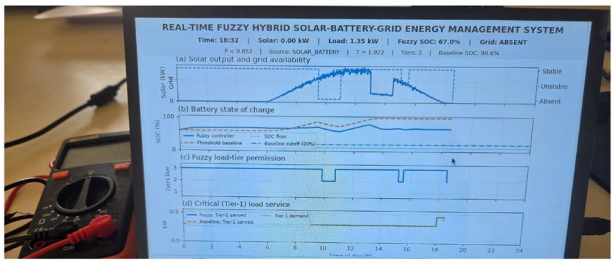
(a) 01:16



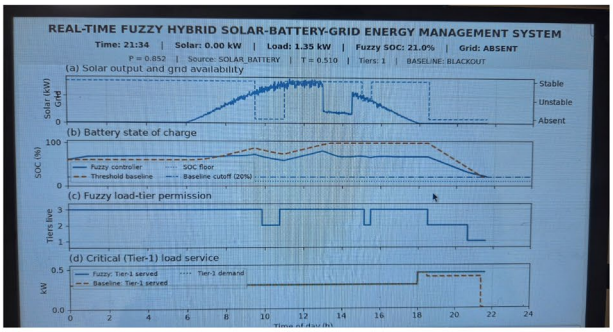
(b) 07:09



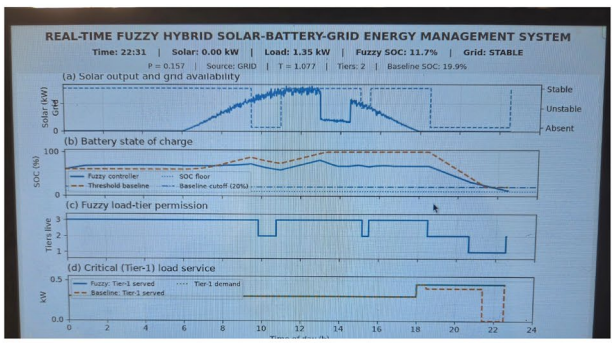
(c) 09:56



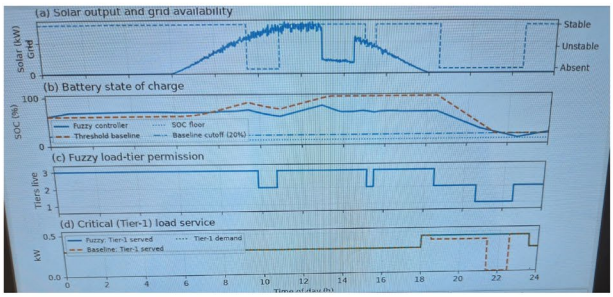
(d) 18:32



(e) 21:34



(f) 22:31



(g) End of run (24 h)

Fig. 7 Real-time embedded demonstration: Raspberry Pi 5 touch display at seven instants of the replayed 24-hour scenario.

Acknowledgements We thank Mbarara University of Science and Technology through Faculty of Science for awarding us with the research fund.

Author contributions J.N.W. came up with the concept and designed the system; M.B. reviewed the system connections and manuscript.

Funding Funding was attained from Mbarara University of Science and Technology through Faculty of Science research

funds.

Data availability The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request. All scenario data are synthetic. The controller and plant code are listed in Appendices B and C, and the complete simulation code (controllers, benchmarks, stress cases, Monte Carlo evaluation, and figure scripts) is openly available at https://github.com/wanzalajn/fuzzy-hybrid-load-balancing.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Zadeh, L. A. Fuzzy sets. *Information and Control* **8**(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X (1965).
2. Mamdani, E. H. & Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* **7**(1), 1–13. https://doi.org/10.1016/S0020-7373(75)80002-2 (1975).
3. Suganthi, L., Iniyan, S. & Samuel, A. A. Applications of fuzzy logic in renewable energy systems–a review. *Renewable and Sustainable Energy Reviews* **48**, 585–607. https://doi.org/10.1016/j.rser.2015.04.037 (2015).
4. Olatomiwa, L., Mekhilef, S., Ismail, M. S. & Moghavvemi, M. Energy management strategies in hybrid renewable energy systems: A review. *Renewable and Sustainable Energy Reviews* **62**, 821–835. https://doi.org/10.1016/j.rser.2016.05.040 (2016).
5. Vivas, F. J., De las Heras, A., Segura, F. & Andújar, J. M. A review of energy management strategies for renewable hybrid energy systems with hydrogen backup. *Renew. Sustain. Energy Rev.* **82**, 126–155. https://doi.org/10.1016/j.rser.2017.09.014 (2018).
6. Alankrita, A. et al. Fuzzy logic based energy management for grid connected hybrid PV system. *Energy Reports* **8**, 751–758. https://doi.org/10.1016/j.egyr.2022.05.217 (2022).
7. García, P., Torreglosa, J. P., Fernández, L. M. & Jurado, F. Optimal energy management system for stand-alone wind turbine/photovoltaic/hydrogen/battery hybrid system with supervisory control based on fuzzy logic. *International Journal of Hydrogen Energy* **38**(33), 14146–14158. https://doi.org/10.1016/j.ijhydene.2013.08.106 (2013).
8. Athari, M. H. & Ardehali, M. M. Operational performance of energy storage as function of electricity prices for on-grid hybrid renewable energy system by optimized fuzzy logic controller. *Renewable Energy* **85**, 890–902. https://doi.org/10.1016/j.renene.2015.07.055 (2016).
9. Mbende, E. T. et al. A simple fuzzy logic based DC link energy management system for hybrid industrial power supply. *Energy Reports* **10**, 3619–3628. https://doi.org/10.1016/j.egyr.2023.10.047 (2023).
10. Atim, M.R., Katrini Anguma, S. & Wanzala Nabende, J. Classification of Fuzzy Logic Linguistic Terms for Accurate Load Balancing in Electric Power Distribution System, in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. 1–6. https://doi.org/10.1109/ICECET52533.2021.9698588 (2021).
11. Wanzala, J. N., Atim, M. R. & Obungoloch, J. Design of Fuzzy Logic-Based ARDS Berlin Definition for Ventilator Adjustments to Ensure Lung Protection. *International Journal of Fuzzy Systems* **25**, 1935–1951. https://doi.org/10.1007/s40815-023-01489-y (2023).
12. Kibazo, P., Nabende, W. J. & Atim, M. R. Design and simulation of an intelligent irrigation system using fuzzy logic. *Discover Electronics* **2**, 5. https://doi.org/10.1007/s44291-025-00045-2 (2025).
13. Akatwijuka, B., Wanzala, J. N. & Atim, M. R. Design and simulation of a fuzzy-PI based bidirectional converter for HESS applications. *Discover Applied Sciences* https://doi.org/10.1007/s42452-026-08637-7 (2026).
14. Wanzala, J. N. & Atim, M. R. Design and simulation of a smart master switch system based on multi-input XOR logic gate. *Discover Electronics* **1**, 23. https://doi.org/10.1007/s44291-024-00028-9 (2024).

15. Laghari, J. A., Mokhlis, H., Bakar, A. H. A. & Mohamad, H. Application of computational intelligence techniques for load shedding in power systems: A review. *Energy Conversion and Management* **75**, 130–140. https://doi.org/10.1016/j.enconman.2013.06.010 (2013).
16. Le, T. N., Quyen, H. A. & Nguyen, N. A. Application of fuzzy-analytic hierarchy process algorithm and fuzzy load profile for load shedding in power systems. *International Journal of Electrical Power & Energy Systems* **77**, 178–184. https://doi.org/10.1016/j.ijepes.2015.11.044 (2016).
17. Hooshmand, R. & Moazzami, M. Optimal design of adaptive under frequency load shedding using artificial neural networks in isolated power system. *International Journal of Electrical Power & Energy Systems* **42**(1), 220–228. https://doi.org/10.1016/j.ijepes.2012.04.021 (2012).
18. Goh, H. H., Kok, B. C., Yeo, H. T., Lee, S. W. & Mohd Zin, A. A. Combination of TOPSIS and AHP in load shedding scheme for large pulp mill electrical system. *Int. J. Electr. Power Energy Syst.* **47**, 198–204. https://doi.org/10.1016/j.ijepes.2012.10.059 (2013).
19. Darbandsari, A. & Amraee, T. Under frequency load shedding for low inertia grids utilizing smart loads. *Int. J. Electr. Power Energy Syst.* **135**, art. 107506. https://doi.org/10.1016/j.ijepes.2021.107506 (2022).
20. Kalajahi, S. M., Seyedi, H. & Zare, K. Under-frequency load shedding in isolated multi-microgrids. *Sustain. Energy Grids Netw.* **27**, 100494. https://doi.org/10.1016/j.segan.2021.100494 (2021).
21. Farquharson, D., Jaramillo, P. & Samaras, C. Sustainability implications of electricity outages in sub-Saharan Africa. *Nature Sustainability* **1**, 589–597. https://doi.org/10.1038/s41893-018-0151-8 (2018).
22. Lee, J. T. & Callaway, D. S. The cost of reliability in decentralized solar power systems in sub-Saharan Africa. *Nature Energy* **3**(11), 960–968. https://doi.org/10.1038/s41560-018-0240-y (2018).
23. Dhundhara, S., Verma, Y. P. & Williams, A. Techno-economic analysis of the lithium-ion and lead-acid battery in microgrid systems. *Energy Conversion and Management* **177**, 122–142. https://doi.org/10.1016/j.enconman.2018.09.030 (2018).
24. Sridhar, B. et al. Design and analysis of modular neural network-based PI-controller ensemble with Karush-Kuhn-Tucker conditions for grid-connected photovoltaic systems under ground fault conditions. *IEEE Access* **13**, 67196–67219. https://doi.org/10.1109/ACCESS.2025.3559139 (2025).
25. Mohapatra, B. et al. A novel optimally tuned super twisting sliding mode controller for active and reactive power control in grid-interfaced photovoltaic system. *IET Energy Systems Integration* https://doi.org/10.1049/esi2.12117 (2023).
26. Ravi Sankar, R. S., Venkatesh, A. & Kollipara, D. Adaptive hysteresis band current control of grid connected PV inverter. *International Journal of Electrical and Computer Engineering (IJECE)* **11**(4), 2856–2863. https://doi.org/10.11591/ijece.v11i4.pp2856-2863 (2021).
27. Satapathy, S., Nahak, N. & Sharma, R. A real time coordinated hydro governor with dual STATCOM for low frequency oscillation damping using hybrid mean DE & TVAPSO. *IEEE Access* **13**, 82065–82083. https://doi.org/10.1109/ACCESS.2025.3567088 (2025).
28. Awamu, A., Wanzala, J. N., Atim, M. R. & Dumba, C. Simulation of a scalable corridor waning radiance system using fuzzy PID and genetic algorithm. *Discover Applied Sciences* **7**, art. 895. https://doi.org/10.1007/s42452-025-06837-1 (2025).
29. Khan, M. W., Choudhry, M. A., Zeeshan, M. & Ali, A. Adaptive fuzzy multivariable controller design based on genetic algorithm for an air handling unit. *Energy* **81**, 477–488. https://doi.org/10.1016/j.energy.2014.12.061 (2015).
30. Ambroziak, A. & Chojecki, A. The PID controller optimisation module using fuzzy self-tuning PSO for air handling unit in continuous operation. *Eng. Appl. Artif. Intell.* **117**, 105485. https://doi.org/10.1016/j.engappai.2022.105485 (2023).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Jimmy Nabende Wanzala[1] · **Michael Byamukama**[2]

✉ Jimmy Nabende Wanzala
wanzalajimmy@gmail.com

[1] Physics Department, Mbarara University of Science and Technology, Mbarara, Uganda

[2] Mathematics Department, Mbarara University of Science and Technology, Mbarara, Uganda