Lafia Journal of Scientific & Industrial Research (LJSIR), Vol. 5(1), 2027
p-ISSN: 3026 – 9288 e-ISSN: 3027 – 1800 pages: 1–16
https://lafiascijournals.org.ng/index.php/ljsir/index ⇒ Published by the
Faculty of Physical Sciences, Federal University of Lafia, Nasarawa State, Nigeria

Occupancy-Gated Fuzzy Logic Control for Energy-Efficient Air-Conditioning in Context-Aware Smart Homes: A Simulation Study

Chima K. Igumbor*, Ikenna A. Onyegbadue and Fred A. Izilein
Department of Computer Engineering, Igbinedion University, Okada, Edo State, Nigeria

Abstract

Air-conditioning is the largest single electrical load in many homes, and the on/off thermostats that still govern most units run at full power, cycle the compressor often, and keep cooling empty rooms. This study designs and evaluates a context-aware fuzzy logic controller that reads two signals, room temperature and occupancy, and sets air-conditioner power. A single room was modelled as a first-order thermal plant and controlled by a Mamdani fuzzy inference system. An initial design, V1, treated occupancy as a soft influence and overcooled empty rooms. A corrected design, V2, writes occupancy into the rule base as a gate with an explicit off state, so an empty room drives cooling to zero. The controllers were re-implemented in Python and compared against an on/off thermostat and an occupancy-aware thermostat under a fixed 180-minute test, a 24 h day, and 300 randomised scenarios, with switching counted for every controller by one threshold-crossing metric. With the room occupied throughout the fixed test, V2 stayed in the comfort band 99.2 % of the time on 1.05 kWh against 1.13 kWh for the thermostat, without switching. On realistic days V2 cut mean daily energy by 18.5 %, but occupied comfort fell to 88.7 % on average against 97.1 % for the thermostat, a trade-off of about eight points, and V2 still restarted the compressor about five times a day against 129 for the thermostat. Most of the saving came from not cooling an empty room. An occupancy-aware thermostat captured most of it without any fuzzy inference.

Keywords: Air-conditioning control; Context-aware smart home; Energy efficiency; Fuzzy logic control; Occupancy-based control

Article History

Submitted
September 30, 2026

Revised
November 14, 2026

First Published Online
November 21, 2026

Correspondences
C. K. Igumbor ✉
kesniel.igumbor@iuokada.edu.ng

doi.org/10.62050/ljsir2027.v5n1.959

Introduction

Buildings account for a large share of global electricity use, and inside a typical dwelling the heating, ventilation and air-conditioning (HVAC) plant is the heaviest consumer [1, 2]. Work on building energy efficiency has concentrated on the systems that condition indoor air, because the savings are large and because comfort and air quality depend on how those systems run [3–5]. The pressure is sharper where electricity is scarce or expensive, as in many tropical and developing regions, where cooling demand is climbing while the grid stays stretched. Reducing and reshaping demand inside the home is the move households can make directly, which is why demand-side management has become central to smart-grid planning [6, 7].

The home is where much of that demand is decided, and it is where sensing and connectivity have advanced fastest. Low-cost sensors, wireless links and edge processors let a house measure its own state and act on it [3, 8–11]. A controller that knows the context of a room, who is in it and what they are doing, can make decisions a blind thermostat cannot. Recognising human presence and activity from sensor streams has grown into a research field of its own, covering ambient and wearable sensing and, increasingly, deep learning models whose inferences still have to be trusted and explained [12–14]. Whether households accept such systems is a separate question, turning on perceived usefulness, trust and environmental attitude [15, 16].

Above this sensing layer sits the home energy management system, the software that schedules and controls loads. Reviews of the field map a wide design space, from rule-based scheduling to optimisation and learning, and they agree that the controller, not the hardware, decides how much energy is saved [17–21]. Data-driven methods have been added on top, using machine learning to model and predict consumption [22–24]. These systems set the context a single-appliance controller has to fit into.

The controllers themselves fall into a few families. Model predictive control uses a plant model to plan ahead and has been applied to HVAC and to standalone energy systems, with strong results when a good model is available [25–28]. Reinforcement learning drops the need for an explicit model and learns a policy from interaction, and has been tested on thermal comfort, whole-building HVAC and demand response [29–33]. Related work optimises comfort and consumption together [34], folds in renewable generation and storage [35, 36], and links the building to a digital twin for planning and maintenance [37, 38]. These methods are powerful. They are also demanding, needing training data and computation that a small residential unit may

not have. Fuzzy logic control sits in a more practical middle ground. It encodes expert reasoning as linguistic rules and handles nonlinear, uncertain plants without an exact model. Mamdani systems have balanced indoor climate from temperature and humidity [39], replaced classical proportional-integral-derivative (PID) loops in real HVAC equipment with measured gains in quality and energy [40], been combined with a PID core for microclimate control [41], and driven low-cost home automation with on-board forecasting [42]. The plant such controllers act on is often a reduced-order thermal model of the room [43]. Comfort and air quality stay the targets any of these designs must respect [2, 30, 44, 45].

Two gaps run through this work. Fuzzy controllers are usually compared against a PID loop or a hardware prototype rather than against the plain on/off thermostat that still governs most installed room air-conditioners, so the gain over what people actually use is rarely quantified [40]. And occupancy is often listed as an input but allowed to nudge the output softly instead of overriding it, which leaves the controller cooling empty rooms. Switching a unit off when a room is empty is itself a standard setback strategy, so the contribution here is narrow and specific. It writes that override into the fuzzy rule base as an explicit off state, and it measures what the override saves against the thermostat people own and against an occupancy-aware thermostat that applies the same setback without fuzzy inference. The study designs, simulates and evaluates this controller. The objectives are to define the two contextual inputs and a first-order thermal model of the room; to build a Mamdani fuzzy inference system for air-conditioner power; to correct a design in which occupancy failed to suppress idle cooling; to re-implement the controllers in Python and check them against the original MATLAB and Simulink work from which the design is taken; and to compare energy, comfort, mean room temperature and actuator switching under a fixed test, a 24-hour day and a large set of randomised scenarios.

Materials and Methods

System overview

The controlled system is a single residential room with one air-conditioning unit. Two signals describe the context of the room. Room temperature is the regulated variable, and occupancy is a binary presence signal, a 0 or 1 that in a deployed system would come from a motion or presence sensor. Both feed one fuzzy inference system whose crisp output is the commanded air-conditioner power. Occupancy enters the rule base as a gate, so an empty room drives the command to the off state before it reaches the unit. The same plant is also driven by two reference controllers, a conventional on/off thermostat and an occupancy-aware thermostat, so the contribution of fuzzy inference can be separated from the benefit of occupancy sensing alone. Fig. 1 shows the scheme.

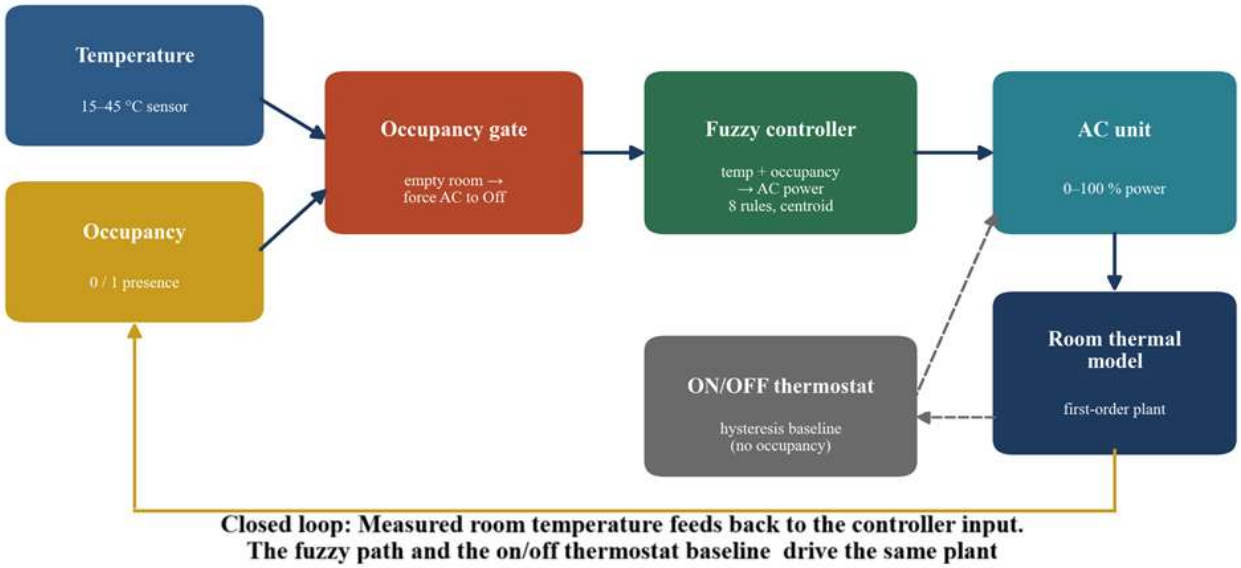


Figure 1: Occupancy-gated fuzzy control scheme for the room air-conditioner

Room temperature and occupancy feed a Mamdani fuzzy inference system through an occupancy gate to the air-conditioning unit and the first-order room plant. The measured temperature feeds back to the controller input. An on/off thermostat baseline, with and without an occupancy override, drives the same plant for comparison.

Room thermal model

The room is represented by a first-order lumped thermal model, the reduced-order form widely used for control design of a single space [43]. Room temperature rises with heat gained from the warmer outdoor air and from occupants, and falls with the cooling the air-conditioner applies. Occupancy multiplies the internal gain, so an

empty room has no occupant heat. The balance is given by Eq. (1).

$$\frac{d\theta}{dt} = \frac{(\theta_a - \theta)}{\tau} + g \cdot n - c \cdot u \qquad (1)$$

Here θ is room temperature, θ_a is outdoor temperature, τ is the thermal time constant, g is the combined occupant and solar heat-gain rate, n is occupancy (0 or 1), c is the cooling slope at full power, and u is the applied command on a 0 to 1 scale. The values used below, τ = 60 min, g = 0.04 °C/min and c = 0.42 °C/min, place the room in a fast, lightly built regime typical of a small tropical space with limited thermal mass. They were chosen so that the open-loop warming and the full-power cooling times are of the order of tens of minutes. They are not fitted to a measured room. A later sensitivity analysis sweeps τ, g, c and the set-point around these values, and the occupancy sweep in a later section shows how strongly the results depend on how long the room stays occupied. The model is deliberately simple. It omits humidity and the latent cooling load, solar gain through the envelope, and infiltration, and it collapses wall and air into a single node, so the absolute numbers are indicative and a two-node or measured model would sharpen them. Reference [43] is a heating study, drawn on here only for the first-order form, not for its climate. The comfort band is the set-point of 26 °C plus or minus 1 °C, that is 25 to 27 °C.

Fuzzy controller design

The controller is a Mamdani fuzzy inference system with two inputs and one output. An initial version, V1, described temperature with three sets, cold, comfortable and hot, and output with three sets, low, medium and high. Occupancy in V1 only softened the command toward low when the room was empty. Static testing exposed a flaw. Because the lowest available output was a nonzero low level, an empty room still received cooling. The corrected version, V2, adds a warm temperature set and an explicit off output set, and it rewrites the rules so that every unoccupied condition map to off. Table 1 lists the membership functions of V2 and Table 2 gives its rule base. V1 is given in full in Table 3, so the flaw and its correction can be reproduced; V1 uses the same two occupancy sets as V2, the occupied rules map cold, comfortable and hot to low, medium and high, and every unoccupied rule map to low. Both controllers recover the crisp command by centroid defuzzification. Fig. 2 plots the V2 membership functions and Fig. 3 shows the control surface with slices at the two occupancy values.

Table 1: Membership functions of the corrected controller (V2)

Variable	Fuzzy set	Type	Parameters
Temperature (°C)	Cold	Trapezoid	15, 15, 21, 24
	Comfortable	Triangle	22, 24.5, 27
	Warm	Triangle	26, 29, 32
	Hot	Trapezoid	30, 35, 45, 45
Occupancy	Unoccupied	Trapezoid	0, 0, 0.25, 0.5
	Occupied	Trapezoid	0.5, 0.75, 1, 1
AC power (%)	Off	Triangle	0, 0, 10
	Low	Triangle	5, 20, 35
	Medium	Triangle	30, 50, 70
	High	Trapezoid	60, 85, 100, 100

Table 2: Rule base of the corrected controller (V2)

Occupancy	Cold	Comfortable	Warm	Hot
Unoccupied	Off	Off	Off	Off
Occupied	Off	Low	Medium	High

Occupancy acts as a complete gate: every unoccupied cell map to Off.

Table 3: Full definition of the initial controller (V1)

Variable	Fuzzy set	Type	Parameters
Temperature (°C)	Cold	Trapezoid	15, 15, 20, 27
	Comfortable	Triangle	22, 26, 32
	Hot	Trapezoid	28, 38, 45, 45
Occupancy	Unoccupied	Trapezoid	0, 0, 0.25, 0.5
	Occupied	Trapezoid	0.5, 0.75, 1, 1
AC power (%)	Low	Triangle	0, 10, 40
	Medium	Triangle	30, 50, 70
	High	Triangle	60, 90, 100

The occupied rules map cold, comfortable and hot to low, medium and high; every unoccupied rule map to low, which is the soft influence that leaves idle cooling in an empty room.

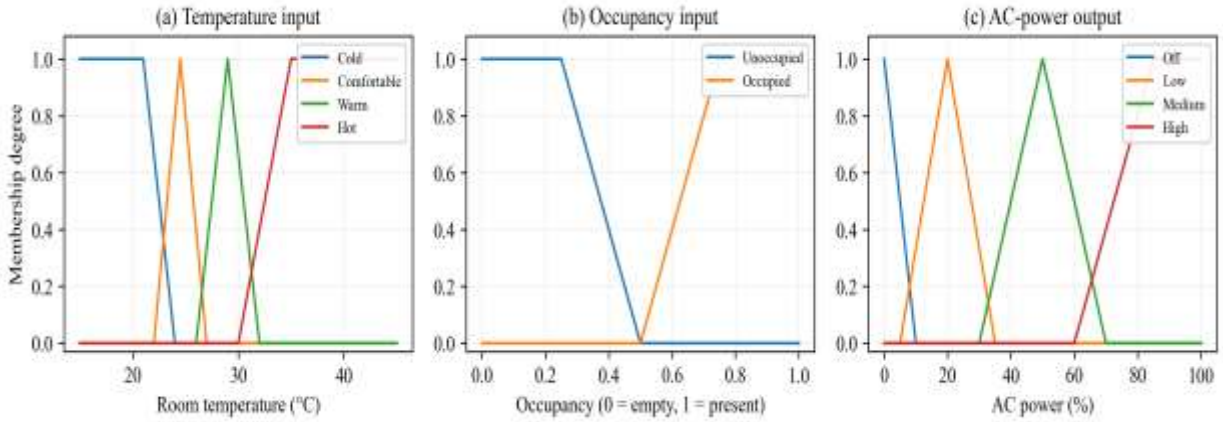


Figure 2: Membership functions of the V2 controller [(a) Temperature input, (b) occupancy input, (c) air-conditioner power output]

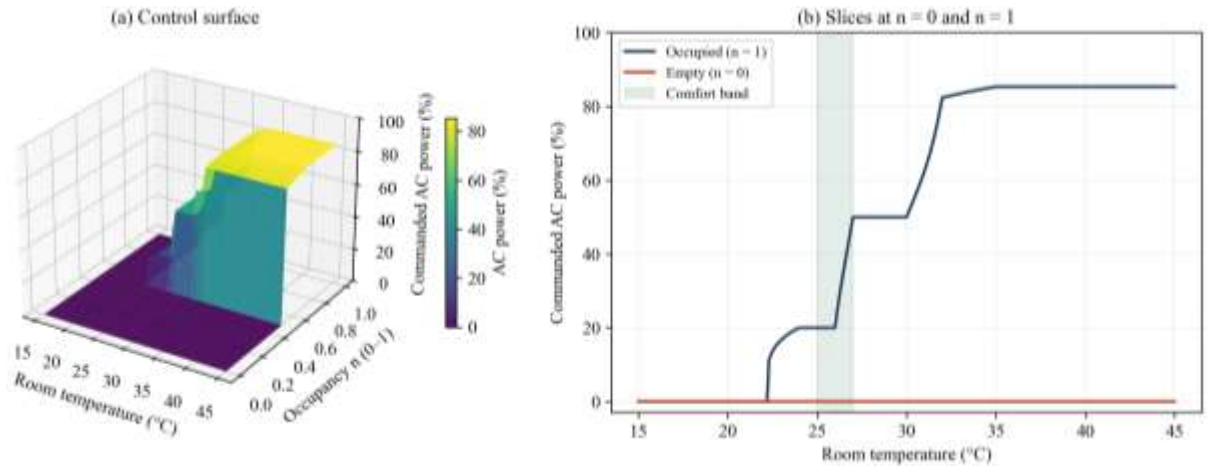


Figure 3: V2 control surface [(a) Commanded power against temperature and occupancy, with the vertical axis and colour bar in percent of rated power. (b) Slices at n = 0 (empty) and n = 1 (occupied); the occupied curve saturates near 85% and the empty curve is held at zero by the gate]

The crisp command is recovered from the aggregated output set by centroid defuzzification, Eq. (2),

$$u* = \frac{\sum_i \mu(z_i)\cdot z_i}{\sum_i \mu(z_i)} \qquad (2)$$

where z_i are samples of the output universe and $\mu(z_i)$ is the aggregated membership. Two design points matter for what follows. First, the Off set is a triangle on 0, 0, 10, whose centroid is about 3.3 %, so centroid defuzzification alone would leave a small trickle in an empty room. A working threshold issues any raw command below 10 % of rated power as a true zero, and the genuine shutdown comes from the rule base and this threshold together, not from the rule base alone. Second, the High set cannot be reached fully by centroid defuzzification, so the commanded power saturates near 85 % even at 38 °C. Full power is never commanded, which slows pull-down after a long absence and feeds into the comfort cost reported in the dynamic-day results. The Comfortable set peaks at 24.5 °C rather than at the 26 °C set-point, a bias inherited from the dissertation design that pushes the occupied command low near the set-point; one effect, visible in the mean-temperature columns of Tables 7 and 8, is that the occupied room settles toward the upper edge of the band rather than at 26 °C. Re-centring the sets on the set-point is left for future work. Occupancy is only ever 0 or 1 in every scenario, so the controller never evaluates the n = 0.5 point at which the two occupancy sets both fall to zero; with a binary input the fuzzy gate reduces to an if-else switch on presence, and it is described here as such rather than as a continuous fuzzy variable.

Baseline controllers, energy metric and test scenarios

The first baseline is a conventional on/off thermostat with hysteresis. It switches the unit to full power when the room reaches the top of the comfort band and off when it reaches the bottom, and it has no knowledge of occupancy, which is how most installed units behave. The hysteresis spans the whole comfort band and no minimum off-time is modelled, so the switch counts for the thermostat are an upper bound; a unit with short-cycle protection would switch less often. The second baseline adds an occupancy override to the same thermostat, forcing the unit off whenever the room is empty. This occupancy-aware thermostat applies the same setback as the fuzzy gate without any fuzzy inference, so comparing it with V2 separates the benefit of occupancy sensing from the benefit of fuzzy logic. Electrical energy is the applied power integrated over the run, Eq. (3),

$$E = \int_0^{t_f} u(t)\cdot P_r\, dt \qquad (3)$$

with P_r the electrical rating of the unit and t_f the run duration. P_r scales all controllers by the same factor and does not change their ranking. It was set so that the thermostat draws about 1.13 kWh over the fixed test, the value reported in the dissertation, so the close agreement on thermostat energy in the fixed-test and dynamic-day results is a calibration, not an independent validation. Power is modelled as proportional to the command, which ignores part-load efficiency, the minimum compressor frequency and start-up transients, so the energy figures are indicative and a part-load efficiency curve would refine them. Comfort is the fraction of occupied time inside the band, since comfort matters only when someone is present. Switching is counted once, for every controller, as the number of times the applied command crosses a minimum-power threshold of 5 % of rated power. For the thermostat this equals the on/off count; for the fuzzy controller, whose command is continuous, it counts each fall to zero and each restart. The switching comparison assumes an inverter, variable-speed unit for the fuzzy controller, whereas the thermostat is a fixed-speed unit. Run time, the fraction of the run with the command above the threshold, and mean power are reported alongside the switch count.

Three scenarios were used. The fixed test holds outdoor temperature at 32 °C for 180 min with the room occupied throughout, which isolates steady behaviour. The dynamic test runs a 24-h day in which outdoor temperature follows a diurnal swing between about 28 °C at night and 36 °C in mid-afternoon, with the room empty during a long morning outing and a shorter evening one. The randomised study repeats the day 300 times with the outing times and durations, the ambient peak and the phase drawn at random, to test whether the findings hold beyond one hand-picked day. Table 4 gives the full scenario generator so the study can be reproduced. All controllers were re-implemented from the same rule base in Python with NumPy and integrated at a 0.25-min step. A self-contained reference implementation is listed in Appendix A, and the full scripts and generated result set are available from the authors on request (see the Data and Code Availability statement).

Table 4: Scenario generator for the 300-run randomised study

Parameter	Distribution or value
Outings per day	Uniform integer, 2 to 4
Outing start time	Uniform, 06:00 to 21:00
Outing duration	Uniform, 0.75 to 3.0 h
Diurnal mean outdoor temperature	Uniform, 29.0 to 31.5 °C
Afternoon peak outdoor temperature	Uniform, 33.0 to 37.0 °C
Peak phase (hour of maximum)	Uniform, 07:00 to 10:00
Set-point and comfort band	26 °C; 25 to 27 °C
Integration step	0.25 min
Days simulated	300
Random seed	20260930

Results and Discussion

Results are given in two layers. The static and closed-loop behaviour of the Python controllers is reported next to the completed MATLAB and Simulink work from which the design is taken, so the two implementations can be checked against each other. The randomised study then extends the evaluation beyond a single day. That earlier work is an unpublished M.Eng. study by the first author, not a separately citable source; its configuration is summarised in Table 5, and its model and result files are available from the authors on request. What is new here is the independent Python re-implementation, the occupancy-aware thermostat baseline, the single switching metric applied to every controller, the mean-temperature reporting, and the 300-run randomised study with paired comfort statistics. Because the dissertation is the source of the reference columns in Tables 7 and 8, its agreement with the Python model is reported as a cross-check, not as validation against independent measurement.

Table 5: Configuration of the reference MATLAB and Simulink dissertation

Item	Setting in the dissertation
Platform	MATLAB and Simulink
Plant	First-order room thermal model
Inputs	Room temperature and binary occupancy
Controllers	On/off thermostat, Fuzzy V1, Fuzzy V2
Fixed test	Outdoor 32 °C, 180 min, room occupied
Dynamic test	Single 24-hour day
Reported energy, fixed (kWh)	1.13 / 1.73 / 1.11 (ON/OFF / V1 / V2)
Reported energy, 24-h (kWh)	7.96 / 12.19 / 8.25 (ON/OFF / V1 / V2)

Table 6: Commanded air-conditioner power against temperature for an occupied room, initial (V1) and corrected (V2) controllers

Temperature(°C)	V1 command (%)	V2 command (%)
24	34.9	20.0
25	39.4	20.0
26	43.9	20.0
27	50.0	50.0
28	50.0	50.0
30	62.0	50.0
32	81.8	82.6
35	82.9	85.6
38	83.3	85.6

Static response and the overcooling flaw

Table 6 lists the commanded power of both controllers against temperature for an occupied room. The two agree at high temperature, where cooling should be strong, and they part near the set-point. Around 26 °C V1 already calls for more than 40 % power while V2 holds at its low level of 20 %, and that difference is what drives V1 to overcool once the loop is closed. The empty-room response is the sharper contrast. In the dissertation V1 kept cooling an empty room at roughly 30 to 33 % of full power across the comfortable range. The Python V1 also overcools, but at a lower mean of about 17 %, so the two V1 implementations are not identical. V1 is unpublished, so the difference cannot be traced in detail, and the Python V1 is treated here as a qualitative reproduction of the flaw rather than an exact one. V2 gates the same conditions to zero. Fig. 4 plots both cases.

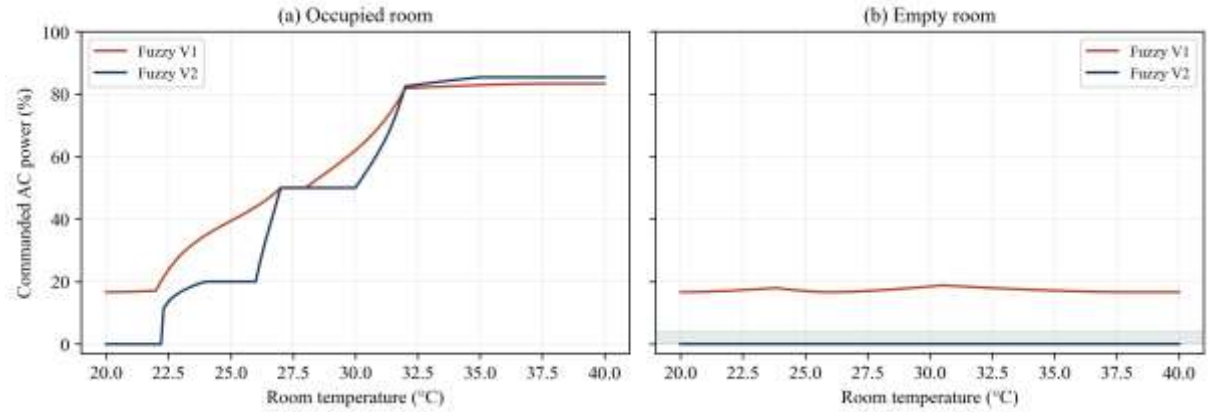


Figure 4: Static command against temperature [(a) An occupied room and (b) an empty room. V1 keeps cooling the empty room; V2 holds it at zero]

Fixed-condition test

Table 7 gives the fixed-test outcome. With the room occupied throughout, the occupancy-aware thermostat never triggers its override, so it matches the plain thermostat exactly; the two differ only once absences appear, in the dynamic-day and occupancy-sweep results. The thermostat energy matches the dissertation to the reported precision (1.13 kWh), which is expected because the rating was calibrated to that value, so the match is by construction rather than validation. For the fuzzy controllers the implementations differ more: V1 computes 1.28 kWh against the 1.73 kWh reported, about 26 % lower, while V2 computes 1.05 kWh against 1.11 kWh, about 5 % lower. V2 held the room in band 99.2 % of the time with no switching and the lowest energy. The mean-temperature column shows why part of the saving appears: V2 runs at a mean of 26.4 °C, above the 26 °C set-point and higher than the thermostat at 26.0 °C, so some of the energy difference is the room sitting slightly warmer. Fig. 5 shows the traces: the thermostat saws between full power and off, V1 pulls the room below the band, and V2 rises once and settles.

Table 7: Fixed-condition results over 180 minutes, room occupied throughout

Controller	Energy ref (kWh)	Energy computed (kWh)	Comfort (%)	Mean T (°C)	Switches	Run time (%)
ON/OFF	1.13	1.13	97.4	26.0	16	35.4
ON/OFF + gate	n/a	1.13	97.4	26.0	16	35.4
Fuzzy V1	1.73	1.28	34.3	25.1	0	100
Fuzzy V2	1.11	1.05	99.2	26.4	0	100

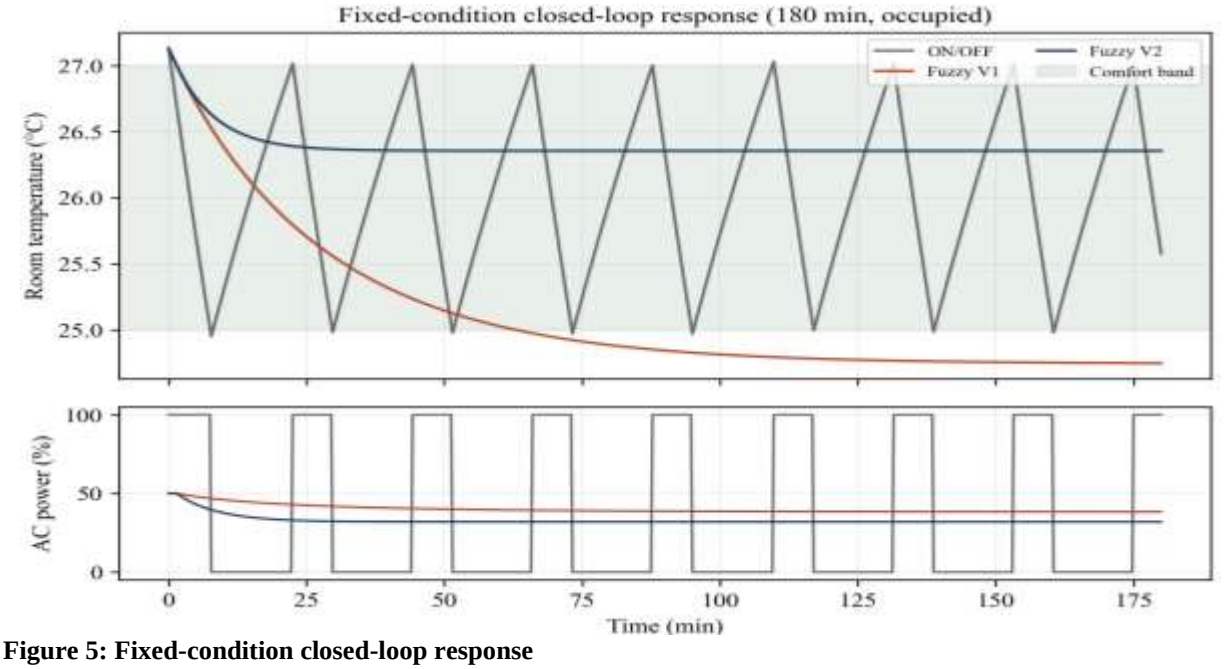


Figure 5: Fixed-condition closed-loop response

Table 8: Twenty-four-hour dynamic-day results

Controller	Energy ref (kWh)	Energy computed (kWh)	Saving (%)	Comfort (%)	Mean T (°C)	Switches
ON/OFF	7.96	7.96	0.0	97.1	26.0	119
ON/OFF + gate	n/a	6.46	18.8	93.1	27.5	85
Fuzzy V1	12.19	8.46	−6.2	30.9	25.6	0
Fuzzy V2	8.25	6.18	22.4	86.4	27.8	4

Reference energy is from the dissertation; other values are computed in Python with the common 5 %-threshold switching metric

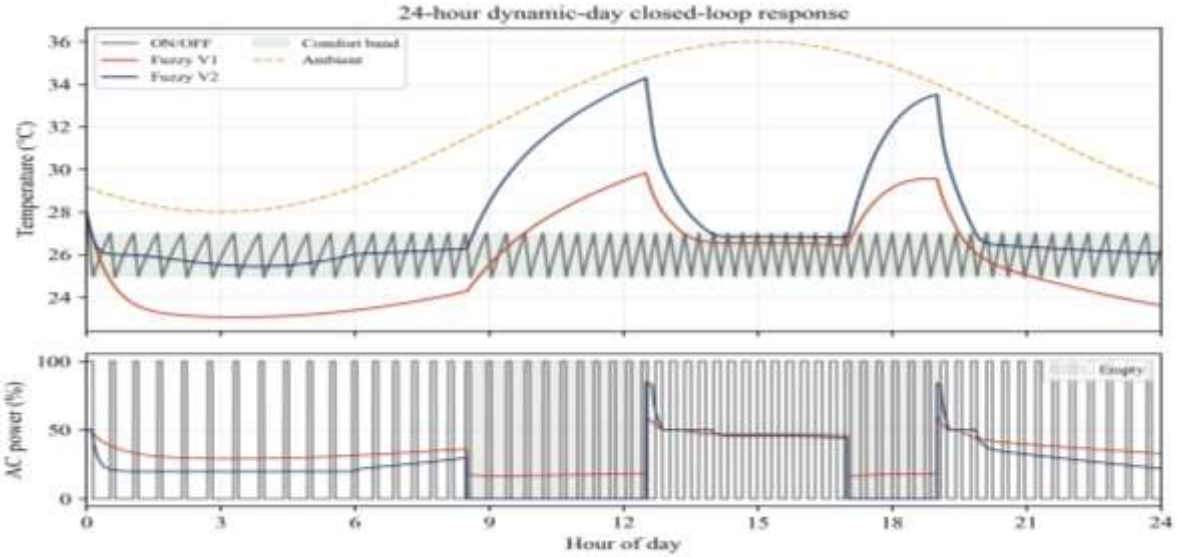


Figure 6: Twenty-four-hour closed-loop response
During the shaded empty periods the thermostat keeps cycling while V2 drops its power to zero, then recovers the room when the occupant returns

Twenty-four-hour dynamic day

The day exposes the controllers to a moving ambient and to real absences. Table 8 reports the outcome. The thermostat again matches the reference closely, 7.96 kWh computed against 7.96 kWh reported, and it switched the compressor 119 times. V2 used 6.18 kWh, 22.4 % below the thermostat. Most of that saving is not from fuzzy inference. The occupancy-aware thermostat, which only adds the same off-when-empty override, already saves 18.8 % on this day, so the fuzzy controller adds about 3.6 percentage points beyond what occupancy sensing alone delivers. On the common metric V2 is not switch-free on this day: it falls to zero and restarts at each outing, which counts as four threshold crossings, against 119 for the thermostat and 85 for the occupancy-aware thermostat. The energy saving carries a comfort cost. Occupied comfort for V2 falls to 86.4 %, against 97.1 % for the thermostat and 93.1 % for the occupancy-aware thermostat, because V2 recovers the room slowly after a long absence, held back by the capped output noted earlier. V2 also runs warmer, at a mean of 27.8 °C over the day. Fig. 6 shows the traces: during the shaded empty periods the thermostat keeps cycling to hold a band no occupant is there to feel, while V2 drops to zero and lets the room drift, then pulls it back on return.

The sign of the 24-h energy result differs from the dissertation, which found V2 using a few percent more energy than the thermostat on its own day (8.25 against 7.96 kWh), whereas the Python model here finds V2 using less. Two things drive the difference. The occupancy pattern is not the same, since the dissertation day and its exact outing schedule are not available to re-run, and the V1 and V2 implementations are not identical, as an earlier comparison showed. The sweep in the next section places both results on one curve: at high occupancy the two controllers sit close together, and the small saving or small excess depends on the exact occupancy pattern of the day. Reproducing the dissertation day exactly in Python is needed to close this fully and is left for future work.

Sensitivity and randomised scenarios

Figure 7 sweeps the daily occupied fraction and reports V2 against the thermostat, with the thermostat comfort included for reference in Table 9. The energy saving falls steadily as the home is occupied for more of the day, from about 45 % when the room is empty most of the time to about 4 % when it is almost always occupied. Comfort moves the other way, rising toward full comfort as absences shrink and recovery gaps disappear, while the thermostat holds near 97 % throughout. At full occupancy V2 still saves about 4 %,

a small saving rather than the small excess the dissertation day showed, which is the sign difference discussed above seen from the other end of the curve.

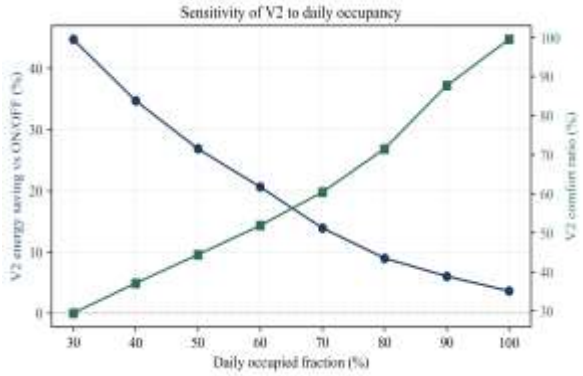


Figure 7: Sensitivity of V2 to the daily occupied fraction
Energy saving against the thermostat falls as occupancy rises, while V2 comfort improves

Table 9: Occupancy sweep

Occupied fraction (%)	V2 saving (%)	V2 comfort (%)	Thermostat comfort (%)
30	44.7	29.4	96.7
40	34.7	37.1	96.6
50	26.9	44.4	97.1
60	20.6	51.9	96.9
70	13.9	60.4	96.9
80	9.0	71.5	97.2
90	6.0	87.7	97.3
100	3.6	99.5	97.2

V2 energy saving and comfort against the daily occupied fraction, with the thermostat comfort for comparison

Table 10: Summary of the 300-run randomised study, with the common switching metric and paired comfort statistics

Metric	Value
Runs	300
Mean V2 energy saving vs ON/OFF (%)	18.5 (95 % CI ±0.6)
Std. dev. of V2 saving (%)	5.3
5th to 95th percentile of V2 saving (%)	10.6 to 27.2
Mean ON/OFF + gate saving vs ON/OFF (%)	12.9
Mean occupied comfort, V2 (%)	88.7
Worst-day occupied comfort, V2 (%)	68.9
Mean occupied comfort, ON/OFF (%)	97.1
Mean occupied comfort, ON/OFF + gate (%)	93.6
Paired comfort drop, ON/OFF − V2 (points)	8.3 (95 % CI ±0.5)
ON/OFF switches per day (mean, max)	129, 139
ON/OFF + gate switches per day (mean)	102
Fuzzy V2 switches per day (mean, range)	5 (2 to 8)

The randomised study puts numbers on the two headline claims across 300 days. The thermostat switched the compressor a mean of 129 times a day and up to 139, the occupancy-aware thermostat a mean of 102, and V2 a mean of about 5 (between 2 and 8). V2 cuts switching by more than 95 % against the thermostat, but it does not remove it: the unit still restarts from zero on every return, so start-up transients are reduced in number, not eliminated, and were not modelled here. The mean daily energy saving was 18.5 % with a standard deviation of 5.3 and 95 % confidence interval near 0.6 %, and the middle 90 % of days fell between 10.6 and 27.2 %; the occupancy-aware thermostat alone captured 12.9 % of the 18.5 %. Occupied comfort for V2 averaged 88.7 % and fell to 68.9 % on the worst day, against 97.1 % for the thermostat, a paired drop of 8.3 points with a 95 % confidence interval near 0.5 points. Table 10 collects these figures and Fig. 8 shows the two distributions.

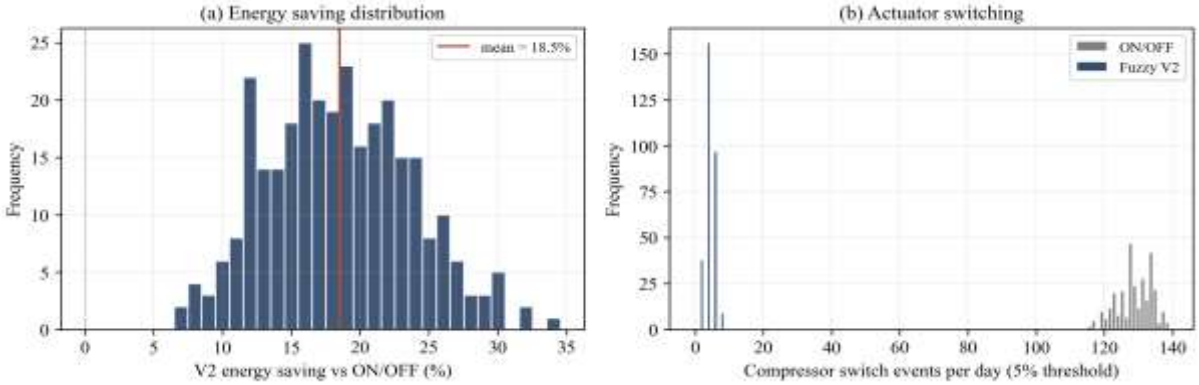


Figure 8: Randomised 300-day study [(a) Distribution of V2 energy saving against the thermostat. (b) Daily compressor switching for the thermostat]

The results qualitatively match reports that fuzzy control smooths actuation relative to a classical loop [40], and they add a comparison that is usually missing: against the on/off thermostat people actually own and against an occupancy-aware thermostat. The larger point is about how context is used. A controller that merely receives an occupancy signal, as V1 did, can still waste energy, because a soft rule lets a nonzero command survive into an empty room. Writing occupancy into the rule base as a gate, with an explicit off state to land on, is what turns the signal into a saving. The occupancy-aware thermostat shows that most of the saving is the occupancy override itself, with fuzzy inference adding a few points and a smoother command. Whether this lesson carries to the learning and predictive controllers reviewed earlier [27, 29, 31]

is a reasonable conjecture, since those too depend on how presence enters the reward or cost function, but it was not tested here and is offered as a hypothesis, not a result.

On the demand side, a room controller that shuts cooling off in empty rooms and restarts the compressor far less often reduces both the energy drawn and the switching stress on the supply. The study uses no site-specific climate, tariff or occupancy data, so these are general simulation findings rather than a claim about any one grid, and a field study with local data would be needed to put numbers on the benefit in a specific setting.

The study has limits, several of which the stress tests probe directly before listing what remains. The occupancy input is a clean binary, whereas real presence sensing is noisy and sometimes wrong. Because the gate is a hard override, a false empty reading would shut cooling off while someone is present, and a delayed detection would cost comfort, so tolerance of sensor error is the main practical risk rather than a distant concern; a later test injects sensor dropouts and shows that a minimum-run safeguard restores comfort [12–14]. The sensitivity of the headline numbers to τ, g, c and the set-point is reported in the parameter sweep, a two-node plant and a part-load efficiency curve are examined in the stronger-plant tests, and stronger baselines, including a tuned PID loop, appear in the baseline comparison [26, 43]. What those tests do not remove are the deeper gaps. The plant is still not fitted to a measured room; fixed-speed and inverter units are not separated, though the switching argument assumes an inverter; and ventilation and air quality are not modelled, so comfort is judged on temperature alone [2, 44, 45]. A hardware trial on a real room, with a test of household acceptance, would close the gap between simulation and use [16, 48].

Sensitivity, Stronger Baselines and Stress Tests

The single day and the 300-run study fix one plant and one family of controllers. This section varies both, to find where the two headline claims, an energy saving from the occupancy gate and a large cut in switching, hold and where they weaken. Every run uses the 24-h dynamic day of the earlier analysis unless stated otherwise, with the same 5 % threshold switching metric, so the numbers here line up directly with Table 8.

Parameter sensitivity

The two gaps left open in the model description are closed here. Each plant and set-point parameter was swept around the base case of τ = 60 min, g = 0.04 °C/min, c = 0.42 °C/min and a 26 °C set-point, holding the others fixed. Fig. 9 plots the result and Table 11 gives the spans. The energy saving of V2 against the thermostat stayed positive at every value tried: from 35.6 % down to 11.6 % as τ rose from 40 to 90 min, between 19.6 and 23.0 % across the occupant gain, from 33.3 % down to 14.3 % across the cooling slope, and from 31.6 % down to 10.5 % as the set-point rose from 25 to 27 °C. The direction is steady. A faster room, a weaker cooler or a lower set-point leaves more idle cooling for the gate to remove, so the saving grows. Comfort is less orderly, peaking near the base case and falling at the extremes where the capped output cannot keep pace. No swept value erased the saving, which is what the sweep was meant to check.

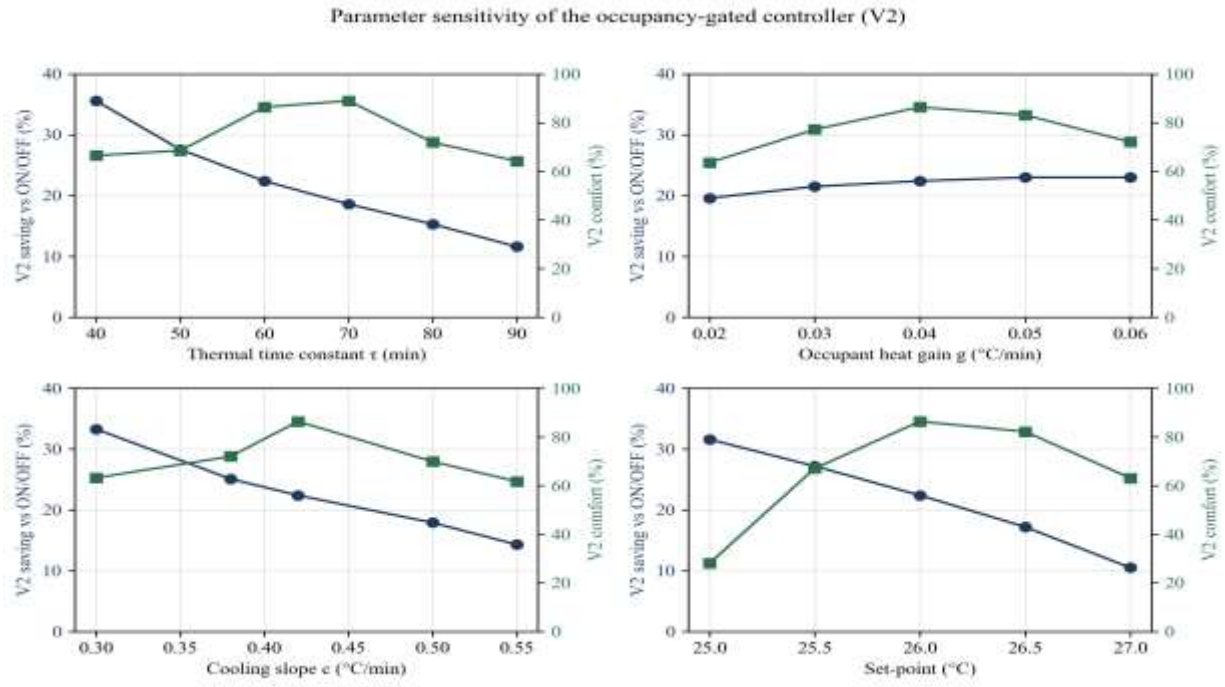


Figure 9: Parameter sensitivity of V2 on the 24-h day

Each panel sweeps one parameter around the base case; the navy curve is energy saving against the thermostat and the green curve is V2 occupied comfort.

Table 11: Parameter sweep on the dynamic day

Parameter	Range swept	V2 saving span (%)	V2 comfort span (%)
Time constant τ (min)	40 to 90	35.6 to 11.6	64 to 89
Occupant gain g (°C/min)	0.02 to 0.06	19.6 to 23.0	64 to 86
Cooling slope c (°C/min)	0.30 to 0.55	33.3 to 14.3	62 to 86
Set-point (°C)	25 to 27	31.6 to 10.5	28 to 86

The base case (τ = 60, g = 0.04, c = 0.42, set-point 26 °C) gives 22.4 % saving at 86.4 % comfort. The saving stays positive across every value

Table 12: Stronger baselines on the dynamic day

Controller	Energy (kWh)	Saving (%)	Comfort (%)	Mean T (°C)	Switches
ON/OFF	7.96	0.0	97.1	26.0	119
ON/OFF + 5-min min off-time	7.96	0.0	97.1	26.0	119
PID, occupancy-blind	7.98	−0.3	99.7	26.0	0
PID + occupancy gate	6.55	17.8	95.6	27.4	4
Fuzzy V2	6.18	22.4	86.4	27.8	4

The occupancy gate, added to a plain PID loop, captures most of V2's saving at higher comfort, which places the energy result on the gate rather than on the fuzzy inference

Stronger baselines and a near-fully-occupied day

The on/off thermostat is a weak reference, so two stronger ones were added and run on the same day (Table 12). A five-minute minimum off-time on the thermostat changed nothing, still 119 switches and 7.96 kWh, because the wide comfort band already keeps the unit off longer than five minutes between cycles, so short-cycle protection has nothing to catch here. A proportional-integral-derivative loop tuned to the set-point but blind to occupancy held comfort at 99.7 % yet saved nothing (−0.3 %), since it too cools the empty room. Adding the same occupancy gate to that PID loop recovered 17.8 % of the energy at 95.6 % comfort, higher comfort than V2's 86.4 %. The reading is direct. The gate, not the fuzzy core, carries the energy result, and a gated PID holds the room more comfortably than the gated fuzzy controller because it has the integral action the fuzzy rule base lacks (noted earlier). A separate near-fully-occupied day, with the room empty only briefly, shrank V2's saving to 5.8 % at 94.1 % comfort, close to the thermostat. This is the regime the earlier M.Eng. day sat in, and it explains why that day showed V2 using about as much energy as the thermostat rather than clearly less: when the room is almost never empty, the gate has almost nothing to switch off, and the sign of the small remaining difference depends on the exact schedule.

Decomposing the design: the gate and the off-set

Two ablations separate the parts of V2 (Table 13). Removing the gate, so occupancy no longer forces the command to zero, lifted comfort to 99.4 % but cut the saving to 2.9 %: without the gate the fuzzy controller is little more than a smoother thermostat. Running V1, whose soft rule leaves idle cooling in an empty room, used 6.2 % more energy than the thermostat at 30.9 % comfort. Adding only an explicit Off output set to V1, without rebuilding its rules, recovered a 5.3 % saving. The off-state matters, then, but it is not sufficient on its own; the gate and the off state together are what produce the full result.

Table 13: Ablation on the dynamic day

Configuration	Energy (kWh)	Saving (%)	Comfort (%)	Mean T (°C)	Switches
Fuzzy V2 (full)	6.18	22.4	86.4	27.8	4
V2 without gate	7.73	2.9	99.4	26.3	0
Fuzzy V1	8.46	−6.2	30.9	25.6	0
V1 + Off set only	7.54	5.3	27.6	26.5	4

The gate supplies most of the saving; the Offset alone helps but does not substitute for it

Membership re-centring, the off-threshold and pre-cooling

Three design changes flagged in the design discussion were tried on the day (Table 14). Re-centring the Comfortable set on the 26 °C set-point while leaving the near-set-point rule at Low pushed the mean room temperature to 28.3 °C and cut comfort to 42.5 % while lifting the saving to 29.1 %: moving the set without changing the rule simply lets the room run warmer. Re-centring and also mapping the comfortable rule to Medium brought the mean back to 26.5 °C but held comfort at only 26.7 %, because the controller then limit-cycles around the set-point with no integral action to settle it, the same gap the gated PID exposed in the baseline comparison. Removing the 10 % off-threshold, so the raw centroid command is issued directly, changed little (19.9 % saving, 87.0 % comfort), which confirms that the shutdown comes mostly from the rule base, not the threshold. Adding a 30-minute pre-cooling pulse before the scheduled return lifted comfort to 94.7 % for a few points of saving (17.9 %), the most promising single remedy for the recovery lag behind V2's comfort cost.

Table 14: Design variants on the dynamic day

Variant	Energy (kWh)	Saving (%)	Comfort (%)	Mean T (°C)
V2 (baseline)	6.18	22.4	86.4	27.8
Re-centred, comfortable rule Low	5.65	29.1	42.5	28.3
Re-centred, comfortable rule Medium	7.53	5.4	26.7	26.5
No 10% off-threshold	6.37	19.9	87.0	27.6
30-min pre-cooling	6.54	17.9	94.7	27.4

Re-centring alone trades comfort for saving; pre-cooling is the cleanest comfort recovery

Table 15: Plant and efficiency stress tests

Scenario and controller	Energy (kWh)	Saving (%)	Comfort (%)	Switches
Part-load COP, ON/OFF	11.94	0.0	n/a	n/a
Part-load COP, ON/OFF + gate	9.70	18.8	n/a	n/a
Part-load COP, Fuzzy V2	6.44	46.1	n/a	n/a
Two-node plant, ON/OFF	7.75	0.0	97.7	97
Two-node plant, ON/OFF + gate	6.79	12.4	92.8	69
Two-node plant, Fuzzy V2	6.41	17.3	73.7	4
Near-occupied day, ON/OFF	8.52	0.0	97.1	123
Near-occupied day, Fuzzy V2	8.02	5.8	94.1	2

The energy and switching rankings hold under a part-load efficiency curve and a two-node plant; the saving only nearly closes on a near-fully-occupied day

Plant and efficiency stress tests

Two harder plant models and one efficiency model test whether the ranking survives a model the first-order linear plant leaves out (Table 15). Replacing the constant cooling slope with a part-load coefficient of performance, which penalises the thermostat's full-power cycling, widened V2's saving to 46.1 %, since the gate avoids exactly the full-power runs a part-load curve punishes. A two-node plant that separates a wall mass from the room air slowed recovery and cut V2 comfort to 73.7 %, but the saving held at 17.3 % and the switching gap was untouched, 4 against 97. The near-occupied day of the stronger-baseline tests is the one regime where the saving nearly closes. Across all of these the energy ranking and the switching ranking stayed the same, so neither headline claim is an artefact of the first-order plant or the linear power model.

Occupancy-sensor noise and a minimum-run safeguard

The gate has so far been driven by a perfect occupancy signal. To test the practical risk named in the reliability discussion, occupancy false negatives, the sensor reporting an occupied room as empty, were injected at rising rates (Fig. 10, Table 16). With the hard gate, V2 occupied comfort fell from 86.0 % at no noise to 69.5 % at a 10 % false-negative rate, as momentary dropouts cut cooling while someone was present. A minimum-run safeguard that holds the unit on for ten minutes after each genuine detection removed the effect: comfort stayed at 86.0 % at every noise level tried, because the hold rides through brief dropouts. The addition is small and it closes the failure mode most likely to matter once the clean binary of the simulation is replaced by a real presence sensor [12, 14].

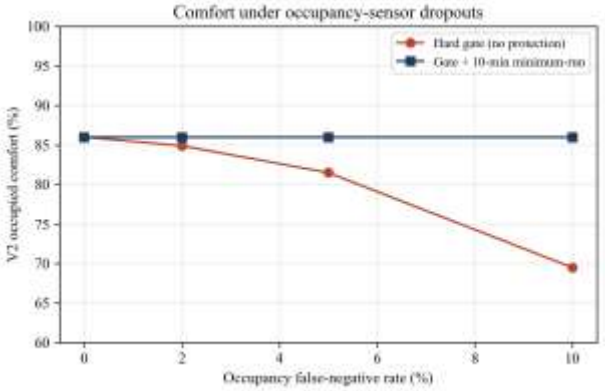


Figure 10: V2 occupied comfort under occupancy-sensor false negatives

The hard gate loses comfort as dropouts rise; a ten-minute minimum-run hold keeps comfort flat across all tested noise levels.

Table 16: Occupancy-sensor noise

False-negative rate (%)	Comfort, hard gate (%)	Comfort, gate + 10-min min-run (%)
0	86.0	86.0
2	84.9	86.0
5	81.5	86.0
10	69.5	86.0

A minimum-run hold restores the comfort the hard gate loses to sensor dropouts

Conclusion

A context-aware fuzzy controller for a room air-conditioner was designed, corrected and evaluated against the on/off thermostat that governs most installed units and against an occupancy-aware thermostat. The corrected controller reads temperature and occupancy and enforces presence as a gate, so an empty room

receives no cooling and an occupied room is regulated smoothly. Re-implemented in Python and run across a fixed test, a full day and 300 random days, it cut mean daily energy by 18.5 % against the thermostat and reduced compressor switching by more than 95 %, from about 129 restarts a day to about five. The saving tracks how long the home stands empty, from a few percent in a house occupied all day to more than 40 % in one left empty for long stretches. It comes at a cost in occupied comfort, which fell to 88.7 % on average and to 68.9 % on the worst day, against 97.1 % for the thermostat, and most of the energy saving is the occupancy override rather than the fuzzy inference, since an occupancy-aware thermostat, and equally a plain PID loop with the same gate, captured most of it at higher comfort than the fuzzy controller. In this plant and these scenarios, a hard occupancy override saved more energy than the soft V1 rule, which points to writing presence into the rules as an override rather than supplying it as a soft input. The saving stayed positive as the thermal time constant, the heat gain, the cooling slope and the set-point were each swept, and it held under a part-load efficiency curve and a two-node plant, so neither headline claim depends on the simplest model. The one real-world failure mode, a missed occupancy reading that shuts cooling off while someone is present, was removed by a ten-minute minimum-run hold. Fitting the plant to a measured room, separating fixed-speed from inverter units, and a hardware trial are the clear routes forward.

Data and Code Availability

A self-contained reference implementation of the controllers, the plant and the evaluation metrics is listed in Appendix A, enough to reproduce the static response and the closed-loop runs. The full scripts that generate every table and figure, the computed result set, and the model and result files of the earlier M.Eng. study are available from the corresponding author on request.

Conflict of interest: The authors declare no conflict of interest.

References

[1] Asim, N., Badiei, M., Mohammad, M., Razali, H., Rajabi, A., Chin Haw, L. & Jameelah Ghazali, M. (2022). Sustainability of heating, ventilation and air-conditioning (HVAC) systems in buildings—An overview. *International Journal of Environmental Research and Public Health*, 19(2), 1016. https://doi.org/10.3390/ijerph19021016

[2] Jia, L. R., Han, J., Chen, X., Li, Q. Y., Lee, C. C. & Fung, Y. H. (2021). Interaction between thermal comfort, indoor air quality and ventilation energy consumption of educational buildings: A comprehensive review. *Buildings*, 11(12), 591. https://doi.org/10.3390/buildings11120591

[3] Metallidou, C. K., Psannis, K. E. & Egyptiadou, E. A. (2020). Energy efficiency in smart buildings: IoT approaches. *IEEE Access*, 8, 63679-63699. https://doi.org/10.1109/access.2020.2984461

[4] Farzaneh, H., Malehmirchegini, L., Bejan, A., Afolabi, T., Mulumba, A. & Daka, P. P. (2021). Artificial intelligence evolution in smart buildings for energy efficiency. *Applied Sciences*, 11(2), 763. https://doi.org/10.3390/app11020763

[5] Nabavi, S. A., Motlagh, N. H., Zaidan, M. A., Aslani, A. & Zakeri, B. (2021). Deep learning in energy modeling: Application in smart buildings with distributed energy generation. *IEEE Access*, 9, 125439-125461. https://doi.org/10.1109/access.2021.3110960

[6] Saleem, M., Shakir, M., Usman, M., Bajwa, M., Shabbir, N., Shams Ghahfarokhi, P. & Daniel, K. (2023). Integrating smart energy management system with internet of things and cloud computing for efficient demand side management in smart grids. *Energies*, 16(12), 4835. https://doi.org/10.3390/en16124835

[7] Saleem, M. U., Usman, M. R., Usman, M. A. & Politis, C. (2022). Design, deployment and performance evaluation of an iot based smart energy management system for demand side management in smart grid. *IEEE Access*, 10, 15261-15278. https://doi.org/10.1109/access.2022.3147484

[8] Ullah, A., Anwar, S. M., Li, J., Nadeem, L., Mahmood, T., Rehman, A. & Saba, T. (2023). Smart cities: the role of Internet of Things and machine learning in realizing a data-centric smart environment. *Complex & Intelligent Systems*, 10(1), 1607-1637. https://doi.org/10.1007/s40747-023-01175-4

[9] Sepasgozar, S., Karimi, R., Farahzadi, L., Moezzi, F., Shirowzhan, S., M. Ebrahimzadeh, S., Hui, F. & Aye, L. (2020). A systematic content review of artificial intelligence and the internet of things applications in smart home. *Applied Sciences*, 10(9), 3074. https://doi.org/10.3390/app10093074

[10] Poyyamozhi, M., Murugesan, B., Rajamanickam, N., Shorfuzzaman, M. & Aboelmagd, Y. (2024). IoT—A promising solution to energy management in smart buildings: A systematic review, applications, barriers, and future scope. *Buildings*, 14(11), 3446. https://doi.org/10.3390/buildings14113446

[11] Yar, H., Imran, A. S., Khan, Z. A., Sajjad, M., & Kastrati, Z. (2021). Towards smart home automation using IoT-enabled edge-computing paradigm. *Sensors*, 21(14), 4932. https://doi.org/10.3390/s21144932

[12] Bouchabou, D., Nguyen, S. M., Lohr, C., LeDuc, B. & Kanellos, I. (2021). A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies,

challenges, and opportunities with deep learning. *Sensors*, 21(18), 6037. https://doi.org/10.3390/s21186037

[13] Das, D., Nishimura, Y., Vivek, R. P., Takeda, N., Fish, S. T., Plötz, T. & Chernova, S. (2023). Explainable Activity Recognition for Smart Home Systems. *ACM Transactions on Interactive Intelligent Systems*, 13(2), 1-39. https://doi.org/10.1145/3561533

[14] Candanedo, L. M. & Feldheim, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. *Energy and Buildings*, 112, 28-39. https://doi.org/10.1016/j.enbuild.2015.11.071

[15] Marikyan, D., Papagiannidis, S. & Alamanos, E. (2020). Cognitive dissonance in technology adoption: A study of smart home users. *Information Systems Frontiers*, 25(3), 1101-1123. https://doi.org/10.1007/s10796-020-10042-3

[16] Ferreira, L., Oliveira, T. & Neves, C. (2023). Consumer's intention to use and recommend smart home technologies: The role of environmental awareness. *Energy*, 263, 125814. https://doi.org/10.1016/j.energy.2022.125814

[17] Zafar, U., Bayhan, S. & Sanfilippo, A. (2020). Home energy management system concepts, configurations, and technologies for the smart grid. *IEEE Access*, 8, 119271-119286. https://doi.org/10.1109/access.2020.3005244

[18] Leitao, J., Gil, P., Ribeiro, B. & Cardoso, A. (2020). A survey on home energy management. *IEEE Access*, 8, 5699-5722. https://doi.org/10.1109/access.2019.2963502

[19] Han, B., Zahraoui, Y., Mubin, M., Mekhilef, S., Seyedmahmoudian, M., & Stojcevski, A. (2023). Home energy management systems: a review of the concept, architecture, and scheduling strategies. *IEEE Access*, 11, 19999-20025. https://doi.org/10.1109/access.2023.3248502

[20] Mischos, S., Dalagdi, E. & Vrakas, D. (2023). Intelligent energy management systems: A review. *Artificial Intelligence Review*, 56(10), 11635-11674. https://doi.org/10.1007/s10462-023-10441-3

[21] Mishra, P. & Singh, G. (2023). Energy management systems in sustainable smart cities based on the internet of energy: A technical review. *Energies*, 16(19), 6903. https://doi.org/10.3390/en16196903

[22] Machorro-Cano, I., Alor-Hernández, G., Paredes-Valverde, M. A., Rodríguez-Mazahua, L., Sánchez-Cervantes, J. L. & Olmedo-Aguirre, J. O. (2020). HEMS-IoT: A big data and machine learning-based smart home system for energy saving. *Energies*, 13(5), 1097. https://doi.org/10.3390/en13051097

[23] Alzoubi, A. (2022). Machine learning for intelligent energy consumption in smart homes. *International Journal of Computations, Information and Manufacturing* (IJCIM), 2(1). https://doi.org/10.54489/ijcim.v2i1.75

[24] Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., Bensaali, F. & Amira, A. (2022). AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives. *Artificial Intelligence Review*, 56(6), 4929-5021. https://doi.org/10.1007/s10462-022-10286-2

[25] Kathirgamanathan, A., De Rosa, M., Mangina, E. & Finn, D. P. (2021). Data-driven predictive control for unlocking building energy flexibility: A review. *Renewable and Sustainable Energy Reviews*, 135, 110120. https://doi.org/10.1016/j.rser.2020.110120

[26] Carli, R., Cavone, G., Ben Othman, S. & Dotoli, M. (2020). IoT based architecture for model predictive control of HVAC systems in smart buildings. *Sensors*, 20(3), 781. https://doi.org/10.3390/s20030781

[27] Arroyo, J., Manna, C., Spiessens, F. & Helsen, L. (2022). Reinforced model predictive control (RL-MPC) for building energy management. *Applied Energy*, 309, 118346. https://doi.org/10.1016/j.apenergy.2021.118346

[28] Rehman, S., Habib, H. U. R., Wang, S., Buker, M. S., Alhems, L. M. & Al Garni, H. Z. (2020). Optimal design and model predictive control of standalone HRES: A real case study for residential demand side management. *IEEE Access*, 8, 29767-29814. https://doi.org/10.1109/access.2020.2972302

[29] Xu, X., Jia, Y., Xu, Y., Xu, Z., Chai, S. & Lai, C. S. (2020). A multi-agent reinforcement learning-based data-driven method for home energy management. *IEEE Transactions on Smart Grid*, 11(4), 3201-3211. https://doi.org/10.1109/tsg.2020.2971427

[30] Gao, G., Li, J. & Wen, Y. (2020). DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet of Things Journal*, 7(9), 8472-8484. https://doi.org/10.1109/jiot.2020.2992117

[31] Azuatalam, D., Lee, W. L., de Nijs, F. & Liebman, A. (2020). Reinforcement learning for whole-building HVAC control and demand response. *Energy and AI*, 2, 100020. https://doi.org/10.1016/j.egyai.2020.100020

[32] Biemann, M., Scheller, F., Liu, X. & Huang, L. (2021). Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control. *Applied Energy*, 298, 117164. https://doi.org/10.1016/j.apenergy.2021.117164

[33] Alfaverh, F., Denai, M. & Sun, Y. (2020). Demand response strategy based on reinforcement learning and fuzzy reasoning for home energy management. *IEEE Access*, 8, 39310-39321. https://doi.org/10.1109/access.2020.2974286

[34] Hosamo, H. H., Tingstveit, M. S., Nielsen, H. K., Svennevig, P. R. & Svidt, K. (2022). Multiobjective optimization of building energy consumption and thermal comfort based on integrated BIM framework with machine learning-NSGA II. *Energy and Buildings*, 277, 112479. https://doi.org/10.1016/j.enbuild.2022.112479

[35] Dinh, H. T., Yun, J., Kim, D. M., Lee, K. H. & Kim, D. (2020). A home energy management system with renewable energy and energy storage utilizing main grid and electricity selling. *IEEE Access*, 8, 49436-49450. https://doi.org/10.1109/access.2020.2979189

[36] Dinh, H. T. & Kim, D. (2021). An optimal energy-saving home energy management supporting user comfort and electricity selling with different prices. *IEEE Access*, 9, 9235-9249. https://doi.org/10.1109/access.2021.3050757

[37] Agostinelli, S., Cumo, F., Guidi, G. & Tomazzoli, C. (2021). Cyber-physical systems improving building energy management: Digital twin and artificial intelligence. *Energies*, 14(8), 2338. https://doi.org/10.3390/en14082338

[38] Coupry, C., Noblecourt, S., Richard, P., Baudry, D. & Bigaud, D. (2021). BIM-based digital twin and XR devices to improve maintenance procedures in smart buildings: A literature review. *Applied Sciences*, 11(15), 6810. https://doi.org/10.3390/app11156810

[39] Soberano, P. J. A., Robles, C. L. & Gatdula, M. M. V. (2025). Design and validation of a mamdani-type fuzzy inference system for dynamic indoor climate balancing. *International Journal of Research and Scientific Innovation*, 12(11), 427-434. https://doi.org/10.51244/ijrsi.2025.12110039

[40] Chojecki, A., Ambroziak, A. & Borkowski, P. (2023). Fuzzy controllers instead of classical PIDs in HVAC equipment: Dusting off a well-known technology and today's implementation for better energy efficiency and user comfort. *Energies*, 16(7), 2967. https://doi.org/10.3390/en16072967

[41] Katayev, N., Zhakish, A., Kulmyrzayev, N., Abuova, A., Toxanova, S., Ostayeva, A. & Dossanova, G. (2024). An intelligent fuzzy-PID controller for supporting comfort microclimate in smart homes. *International Journal of Advanced Computer Science and Applications*, 15(2), 520-530. https://doi.org/10.14569/ijacsa.2024.0150255

[42] Ozgonenel, O. & Gözüoğlu, A. (2021). Fuzzy logic based smart home automation and forecasting electric energy consumption. *Balkan Journal of Electrical and Computer Engineering*, 9(4), 365-370. https://doi.org/10.17694/bajece.928537

[43] Bastida, H., Ugalde-Loo, C. E., Abeysekera, M., Qadrdan, M. & Wu, J. (2019). Thermal dynamic modelling and temperature controller design for a house. *Energy Procedia*, 158, 2800-2805. https://doi.org/10.1016/j.egypro.2019.02.041

[44] Saini, J., Dutta, M. & Marques, G. (2020). Indoor air quality monitoring systems based on internet of things: A systematic review. *International Journal of Environmental Research and Public Health*, 17(14), 4942. https://doi.org/10.3390/ijerph17144942

[45] Zhang, H. & Srinivasan, R. (2020). A systematic review of air quality sensors, guidelines, and measurement studies for indoor air quality management. *Sustainability*, 12(21), 9045. https://doi.org/10.3390/su12219045

[46] The MathWorks, Inc. (2024). Thermal Model of a House. MATLAB & Simulink Documentation. https://www.mathworks.com/help/simulink/slref/thermal-model-of-a-house.html [Accessed 15 September 2025].

[47] The MathWorks, Inc. (2024). Fuzzy logic control for a house heating system. MATLAB & Simulink Documentation. https://www.mathworks.com/help/fuzzy/fuzzy-logic-controller-for-a-house-heating-system.html [Accessed 15 September 2025].

[48] Singh, R., Gehlot, A., Kuchhal, P., Choudhury, S., Akram, S. V., Priyadarshi, N. & Khan, B. (2023). Internet of things enabled intelligent automation for smart home with the integration of PSO algorithm and PID controller. *Journal of Electrical and Computer Engineering*, 1-25. https://doi.org/10.1155/2023/9611321

Appendix A: Reference implementation

Listing A1. Controllers, plant, baselines and evaluation metrics

This self-contained Python listing implements the two fuzzy controllers, the on/off and occupancy-gated thermostat baselines, the first-order room plant of Eq. (1) and the energy, comfort and switching metrics of the methodology. It depends only on NumPy. Running it reproduces the static commands of Table 6 and the closed-loop behaviour of the controllers; the full scripts that generate every table and figure, including the randomised study and the stress tests of the extended analysis, follow the same structure and are available from the corresponding author on request.

```
# Occupancy-gated fuzzy air-conditioning control: reference implementation.
# Reproduces the controllers (Fuzzy V1, Fuzzy V2), the on/off and occupancy-gated
# thermostat baselines, the first-order room plant and the evaluation metrics.
# Python 3 with NumPy.  Author implementation; see the paper for parameter sources.
import numpy as np


# ---- membership function primitives (coincident knots give vertical edges) ---
def tri(x, a, b, c):
    if x <= a or x >= c:
        return 0.0
    if x <= b:
        return (x - a) / (b - a) if b > a else 1.0
    return (c - x) / (c - b) if c > b else 1.0

def trap(x, a, b, c, d):
    if x < a or x > d:
        return 0.0
    if x < b:
        return (x - a) / (b - a) if b > a else 1.0
    if x <= c:
        return 1.0
    return (d - x) / (d - c) if d > c else 1.0

# ---- fuzzy inference system (Mamdani, centroid defuzzification) --------------
class FIS:
    def __init__(self, temp_sets, occ_sets, out_sets, rules):
        self.temp, self.occ, self.out, self.rules = temp_sets, occ_sets, out_sets, rules
        self.z = np.linspace(0, 100, 501)          # output universe, % of rated power

    def _mu(self, sets, x):
        return {k: f(x) for k, f in sets.items()}

    def command(self, temperature, occupancy):
        mt = self._mu(self.temp, temperature)
        mo = self._mu(self.occ, occupancy)
        agg = np.zeros_like(self.z)
        for (t_lbl, o_lbl, out_lbl) in self.rules:
            w = min(mt[t_lbl], mo[o_lbl])          # AND = min
            if w > 0:
                agg = np.maximum(agg, np.minimum(w, self.out[out_lbl](self.z)))
        s = agg.sum()
        return 0.0 if s == 0 else float((self.z * agg).sum() / s)

# ---- controller definitions (Table 1 = V2, Table 3 = V1) ---------------------
OCC_SETS = {"unocc": lambda x: trap(x, 0, 0, 0.25, 0.5),
            "occ":   lambda x: trap(x, 0.5, 0.75, 1, 1)}

def make_v2():
    temp = {"cold": lambda x: trap(x, 15, 15, 21, 24),
            "comf": lambda x: tri(x, 22, 24.5, 27),
            "warm": lambda x: tri(x, 26, 29, 32),
            "hot":  lambda x: trap(x, 30, 35, 45, 45)}
    out = {"off":  lambda z: np.array([tri(v, 0, 0, 10) for v in z]),
           "low":  lambda z: np.array([tri(v, 5, 20, 35) for v in z]),
           "med":  lambda z: np.array([tri(v, 30, 50, 70) for v in z]),
           "high": lambda z: np.array([trap(v, 60, 85, 100, 100) for v in z])}
    rules = [("cold", "occ", "off"), ("comf", "occ", "low"),
             ("warm", "occ", "med"), ("hot", "occ", "high"),
             ("cold", "unocc", "off"), ("comf", "unocc", "off"),
             ("warm", "unocc", "off"), ("hot", "unocc", "off")]
    return FIS(temp, OCC_SETS, out, rules)
```

```
def make_v1():
    temp = {"cold": lambda x: trap(x, 15, 15, 20, 27),
            "comf": lambda x: tri(x, 22, 26, 32),
            "hot":  lambda x: trap(x, 28, 38, 45, 45)}
    out = {"low":  lambda z: np.array([tri(v, 0, 10, 40) for v in z]),
           "med":  lambda z: np.array([tri(v, 30, 50, 70) for v in z]),
           "high": lambda z: np.array([tri(v, 60, 90, 100) for v in z])}
    rules = [("cold", "occ", "low"), ("comf", "occ", "med"), ("hot", "occ", "high"),
             ("cold", "unocc", "low"), ("comf", "unocc", "low"), ("hot", "unocc", "low")]
    return FIS(temp, OCC_SETS, out, rules)

# ---- plant, baselines and simulation ----------------------------------------
TAU, G, C = 60.0, 0.04, 0.42          # min, deg C/min, deg C/min
SET, BAND = 26.0, (25.0, 27.0)
P_RATED, DT, THR = 1.064, 0.25, 0.05  # kW, min, 5% switch threshold

def step_plant(theta, amb, occ, u):
    dtheta = (amb - theta) / TAU + G * occ - C * u
    return theta + dtheta * DT

def thermostat(theta, state, occ, gated):
    if gated and occ < 0.5:
        return 0.0, 0
    if theta >= BAND[1]:
        return 1.0, 1
    if theta <= BAND[0]:
        return 0.0, 0
    return float(state), state

def simulate(ctrl, minutes, amb_fn, occ_fn, theta0=28.0):
    n = int(minutes / DT)
    theta, u_prev, state = theta0, 0.0, 0
    E = sw = 0.0
    occ_time = in_band = 0
    for k in range(n):
        t = k * DT
        amb, occ = amb_fn(t), occ_fn(t)
        if ctrl in ("onoff", "gated"):
            u, state = thermostat(theta, state, occ, ctrl == "gated")
        else:                                  # fuzzy V1 / V2
            raw = ctrl.command(theta, occ)
            u = 0.0 if raw < 10.0 else raw / 100.0
        if (u_prev <= THR) != (u <= THR):
            sw += 1
        E += u * P_RATED * (DT / 60.0)         # kWh
        if occ >= 0.5:
            occ_time += 1
            if BAND[0] <= theta <= BAND[1]:
                in_band += 1
        theta = step_plant(theta, amb, occ, u)
        u_prev = u
    comfort = 100.0 * in_band / max(occ_time, 1)
    return dict(energy=round(E, 3), comfort=round(comfort, 1), switches=int(sw))

if __name__ == "__main__":
    amb = lambda t: 32.0                       # fixed-condition test, occupied
    occ = lambda t: 1.0
    for name, c in [("ON/OFF", "onoff"), ("V1", make_v1()), ("V2", make_v2())]:
        print(name, simulate(c, 180, amb, occ))
```

Citing this Article

Igumbor, C. K., Onyegbadue, I. A., & Izilein, F. A. (2026). Occupancy-gated fuzzy logic control for energy-efficient air-conditioning in context-aware smart homes: A simulation study. *Lafia Journal of Scientific and Industrial Research*, *5*(1), 190–205. https://doi.org/10.62050/ljsir2027.v5n1.959