arXiv:2608.09453v1 [cs.LG] 10 Aug 2026

Learning to Modulate, Not to Cycle: Soft Actor–Critic Recovers Inverter-Style Heat-Pump Control

Faizan Ahmed, Aniket Dixit, and James Brusey

Centre for Computational Science and Mathematical Modelling,
Coventry University, United Kingdom
ahmedf84@uni.coventry.ac.uk

Abstract. On–off cycling is the main cause of compressor wear in residential heat pumps, yet reinforcement learning (RL) controllers for buildings typically optimise only energy cost and thermal comfort, ignoring how much the learned policy cycles. We add a levelised compressor-wear term to the control reward and study how the resulting behaviour depends on the RL algorithm. Training Soft Actor–Critic (SAC) and Proximal Policy Optimisation (PPO) on an identical Markov decision process for the BOPTEST `bestest_hydronic_heat_pump` case, we find that SAC learns a continuous modulation policy that keeps the compressor permanently engaged—the operating principle of an inverter-driven heat pump—achieving zero start-ups per day, whereas PPO collapses to bang–bang control that cycles more than the baseline. On the BOPTEST emulator the SAC policy cuts thermal discomfort by up to 90.7 % for an 11.5 % cost increase, while eliminating all baseline cycling.

Keywords: heat pumps, compressor cycling, reinforcement learning, Soft Actor–Critic, BOPTEST

1 Introduction

Heat pumps are central to the decarbonisation of residential heating, but their delivered efficiency and service life depend strongly on *how* they are operated, not only on *how much* energy they consume. Single-stage units satisfy a heating load by switching the compressor fully on and off; every start incurs a transient during which refrigerant migration, oil return and pressure equalisation depress the coefficient of performance (COP), and the accumulated number of starts is a primary driver of mechanical wear and eventual failure [1, 2, 3]. The standardised part-load degradation coefficient C_d used in rating procedures encodes precisely this cycling penalty [4]. Inverter-driven (variable-capacity) heat pumps were introduced specifically to suppress cycling: by modulating compressor speed they track the load continuously, run for longer at lower capacity, and thereby raise seasonal efficiency while reducing the number of damaging start-ups [1, 2, 3].

A controller for such equipment should therefore be judged on three axes simultaneously: thermal comfort, operating (electricity) cost, and *control smoothness*, i.e. how much avoidable cycling it commands. Reinforcement learning is

a popular model-free route to building HVAC control [5, 6], with supervised benchmarking through the open BOPTEST framework [7]. Yet most RL control studies reward only a weighted sum of comfort violation and energy cost; the cycling behaviour of the learned policy is rarely measured and almost never optimised so that a policy can earn an excellent cost–comfort score while chattering the compressor in a way that real hardware could not sustain.

This paper closes that gap for hydronic heat-pump control. We make three contributions.

1. *A smoothing-aware reward* that augments the comfort-plus-cost objective with a physically grounded, monetised compressor-wear term, levelising the compressor cost over its rated number of starts (Section 3.2)—making switching a first-class objective without an explicit minimum-runtime constraint.
2. *An algorithm-class effect on smoothness:* holding the environment, reward and observations fixed, the RL algorithm determines the *qualitative* form of the control signal. SAC learns a continuous-modulation policy that never switches off (zero starts/day, inverter-style operation), whereas PPO learns a bang–bang policy that cycles *more* than the uncontrolled baseline.
3. *A validated comfort–cost trade-off:* transferred to the BOPTEST emulator for this test case, the smooth SAC policy beats the built-in baseline on thermal discomfort by up to 90.7 % for a modest cost increase, while removing all baseline cycling.

Our focus is the reward design, the resulting control behaviour, and its BOPTEST validation.

2 Related Work

RL for building and HVAC control: Data-driven building control has been surveyed extensively [5, 6], with deep RL built on actor–critic foundations for continuous control [8, 9]. Model-predictive control remains the strong supervised baseline [10], and hybrid RL–MPC schemes have also been explored [11]. BOPTEST [7] standardises the comparison through a common emulator/KPI interface; we adopt its `bestest_hydronic_heat_pump` case and KPI definitions.

Cycling and capacity control of heat pumps: The penalty of on–off operation is well established in the refrigeration literature: Madani et al. [1, 2] and Karlsson and Fahlén [3] quantifies how variable-capacity (inverter) control reduces start-ups and raises seasonal performance, and rating standards encode the part-load cycling loss through the degradation coefficient [4]. These results motivate treating the *number of starts* as a control objective. Finally, that unconstrained RL produces high-frequency, oscillatory actions is a known failure mode addressed by explicit smoothness regularisers [12]; we show that for heat-pump cycling the *algorithm class*—entropy-regularised off-policy learning—can itself supply the smoothness bias.

3 Problem Formulation

We formulate single-zone heat-pump control as a discrete-time Markov decision process (MDP) with a control step of $\Delta t = 900\,\mathrm{s}$ (15 min), matching the BOPTEST test case. Episodes span 14 days (1344 steps).

3.1 Action, Observation, and Heat-Pump Operating Principle

The action is the scalar heat-pump command $u_{\mathrm{HP}} \in [0, 1]$, the normalised compressor capacity, where 0 is fully off and 1 is rated capacity. The observation is a Case-D–aligned vector (zone temperature, ambient temperature, direct-normal irradiance, internal gains, the comfort set-points, the current price, and a sine/cosine time-of-day encoding) followed by an 8-step look-ahead forecast of the exogenous signals; two duty features (previous on/off state and normalised dwell time) are appended when the wear term is active.

The action semantics tie the learning problem to the hardware. A single-stage (on–off) unit admits only the endpoints $u_{\mathrm{HP}} \in \{0, 1\}$ and meets partial loads by duty-cycling between them—each alternation a start. An inverter unit admits the whole interval and meets a partial load by sustaining an intermediate capacity, eliminating the alternation [3, 1, 2]. Our action space exposes the full interval, so the agent is *free* to behave as an on–off thermostat (the endpoints) or as an inverter (the interior); the emerged behaviour is selected by the learning algorithm, and that selection is the object of study.

3.2 Reward

The per-step reward is the negative weighted sum of three integrated costs over Δt:

$$r_t = -\big(w_{\mathrm{disc}}\, J_t^{\mathrm{disc}} + w_{\mathrm{cost}}\, J_t^{\mathrm{cost}} + w_{\mathrm{wear}}\, J_t^{\mathrm{wear}} \big). \tag{1}$$

The thermal-discomfort term integrates the violation of the time-varying comfort band, $J_t^{\mathrm{disc}} = \big(\max(T_t^{\mathrm{lo}} - T_t^{\mathrm{zone}}, 0) + \max(T_t^{\mathrm{zone}} - T_t^{\mathrm{hi}}, 0) \big)\, \Delta t_h$, in kelvin-hours; the electricity-cost term is $J_t^{\mathrm{cost}} = P_t^{\mathrm{tot}}\, \Delta t_h\, \pi_t / 1000$ in euros, with π_t the (highly dynamic) tariff. The novel compressor-wear term monetises switching. With a binary duty $d_t = \mathbb{1}[u_{\mathrm{HP},t} > \varepsilon]$ ($\varepsilon = 10^{-3}$), a *start* is a rising edge $\max(d_t - d_{t-1}, 0)$, and each start is charged a levelised cost

$$J_t^{\mathrm{wear}} = c_{\mathrm{cyc}} \cdot \max(d_t - d_{t-1}, 0), \qquad c_{\mathrm{cyc}} = 0.0133\,€, \tag{2}$$

obtained by amortising a representative compressor replacement ($\sim$2000 €) over its rated $\sim$150 000 starts. A short-cycle surcharge penalises starts separated by fewer than four steps. We deliberately express wear in the same currency (euros) as the energy term so that the weights w have a transparent economic meaning; in all experiments below $w_{\mathrm{disc}} = w_{\mathrm{cost}} = w_{\mathrm{wear}} = 1$.

A key property of (2) is that a controller can drive J^{wear} to zero in two ways: by never running the compressor, or by running it *continuously* so that no

rising edge ever occurs. The first is excluded by the discomfort term (the zone would freeze); the second is exactly the inverter-modulation regime. Whether an RL agent discovers the second, benign optimum turns out to depend on the algorithm.

4 Method

Training environment (surrogate plant): The agents are not trained on the BOPTEST emulator directly but on a fast, data-driven surrogate of the plant identified offline using Sparse Identification of Nonlinear Dynamics (SINDy) [13]. The surrogate advances the zone thermal state one control step at a time, mapping the current state, the heat-pump command u_{HP} and the exogenous inputs to the predicted next-step state; the agent's observation at each step is formed directly from this predicted state, so the full 5×10^6-step training budget is collected at a small fraction of the cost of stepping the high-fidelity emulator. A companion map supplies the heat-pump electrical power entering the cost and wear terms of (1). We adopt this surrogate as-is from our earlier work [14] and regard its identification as outside the present scope; the resulting policies are transferred to the genuine BOPTEST emulator for all evaluation reported below.

We train two model-free agents with Stable-Baselines3 [15]. **SAC** [8] is an off-policy, maximum-entropy actor–critic whose stochastic policy, entropy bonus and replay buffer reward exploration of the *interior* of the action interval, biasing it away from the saturated rails. **PPO** [9] is an on-policy clipped policy-gradient method for which the cheapest way to raise the average duty is to shift probability mass onto the $u_{\mathrm{HP}} = 1$ and 0 rails—a tendency towards bang–bang. Both agents share network sizes, discount ($\gamma = 0.99$), learning rate (3×10^{-4}), observation space and the reward of (1); only the algorithm differs, and each is trained for 5×10^6 steps before out-of-sample evaluation on the genuine BOPTEST emulator.

Evaluation protocol: We evaluate on the two BOPTEST scenarios—the *peak* and *typical* heating periods, each 14 days under the highly dynamic tariff—against the test case's built-in baseline controller. We report (i) BOPTEST closed-loop KPIs; (ii) cycling metrics from the binary duty signal (switches and starts per day, and the wear of (2)); and (iii) statistics of the commanded signal. The baseline duty is inferred from its power trace ($P_{\mathrm{HeaPum}} > 50\,\mathrm{W} \Rightarrow$ on).

5 Results

5.1 SAC Eliminates Cycling; PPO Worsens It

Table 1 reports the cycling behaviour. SAC records *zero* switches and *zero* starts on both scenarios, eliminating the 1.07 to 1.50 daily starts of the baseline. PPO does the opposite, cycling 2 to 4 times more often than the baseline and turning the wear term *negative*. The same reward, optimised by two algorithms, yields opposite cycling outcomes.

Table 1. Compressor cycling and monetised wear over the 14-day scenarios. Lower is better. "Starts/day" is the number of off→on transitions; wear follows Eq. (2); TV is the mean step-to-step change of the command.

Scenario	Controller	Switch/d	Start/d	Wear (€/d)	TV
Peak	Baseline	2.14	1.07	0.0143	0.021
	SAC	**0.00**	**0.00**	**0.0000**	0.046
	PPO	8.51	4.29	0.0570	0.096
Typical	Baseline	3.00	1.50	0.0200	0.018
	SAC	**0.00**	**0.00**	**0.0000**	0.025
	PPO	5.29	2.64	0.0352	0.055

Table 2. BOPTEST closed-loop KPIs (14-day scenarios). t_{dis}: thermal discomfort (K h); cost (€); energy: source energy (kW h); emis: emissions (kg CO_2). Lower is better for discomfort, cost and emissions.

Scenario	Controller	t_{dis}	cost	energy	emis
Peak	Baseline	8.382	0.909	3.478	0.581
	SAC	**0.777**	1.013	3.860	0.645
	PPO	6.884	0.631	2.436	0.407
Typical	Baseline	9.446	0.413	1.773	0.296
	SAC	**6.322**	0.631	2.721	0.454
	PPO	9.088	0.255	1.107	0.185

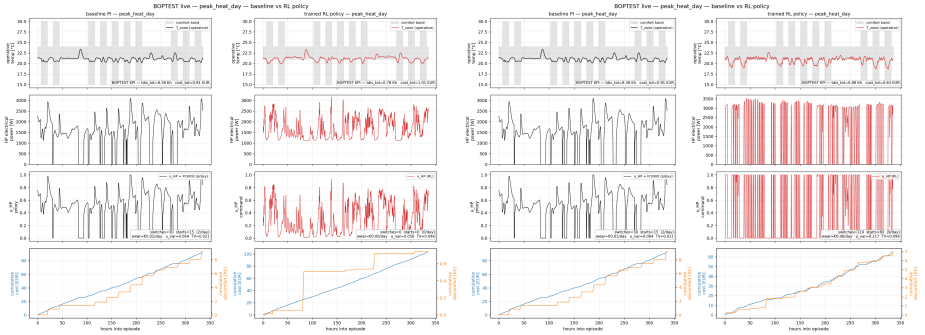
5.2 The Control Signal is Continuous, not Gated

The commanded traces explain *how* SAC reaches zero starts. SAC's u_{HP} varies substantially (non-zero TV in Table 1) yet *never* drops to the off state: $u_{\mathrm{HP}} \in [0.007, 0.93]$ with 0.0 % of steps below the duty threshold on the peak day, and $[0.001, 0.85]$ with again 0.0 % off on the typical day—the compressor turns on once and then modulates continuously, the signature of inverter operation. PPO instead spends 60.6 % (peak) and 82.3 % (typical) of steps fully off and saturates at the $u_{\mathrm{HP}} = 1$ rail—a bang–bang controller. This is the central empirical finding of the paper, and Fig. 1 shows it directly for both the peak and typical days.

5.3 Beating Discomfort on BOPTEST

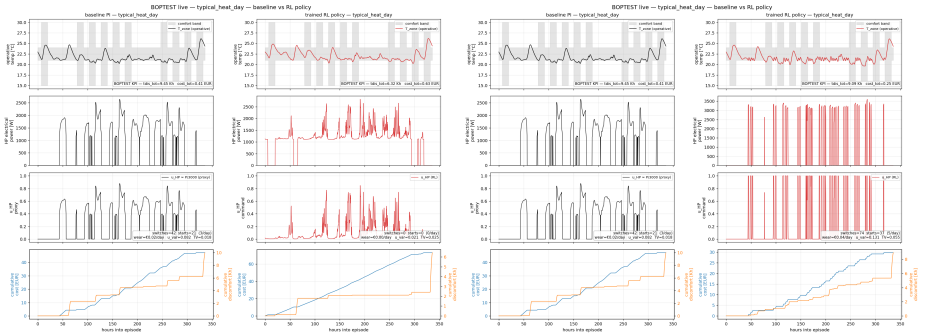
On the peak day (Table 2) SAC cuts thermal discomfort from 8.38 to 0.78 K h (a 90.7 % reduction) for an 11.5 % increase in operating cost, and reduces it by 33.1 % on the typical day. The smooth policy thus *purchases* comfort with a small, bounded cost premium while removing all cycling. PPO occupies the opposite corner: it lowers cost (it heats less) but barely improves comfort (17.9 % and 3.8 %) and cycles heavily.

The energy column of Table 2 locates SAC's cost premium: it draws 11.0 % more source energy than the baseline, almost exactly its 11.5 % cost increase,



(a) SAC, peak day — smooth modulation, 0 starts

(b) PPO, peak day — 4.29 starts/day



(c) SAC, typical day — smooth modulation, 0 starts

(d) PPO, typical day — 2.64 starts/day

Fig. 1. Same reward, opposite controllers. Closed-loop traces on BOPTEST for the peak (top) and typical (bottom) heating days; each panel shows zone temperature against its comfort band (upper) and the commanded capacity u_{HP} (lower). SAC (a, c) modulates smoothly and never lets the compressor restart—zero starts/day with comfort intact, learned inverter operation even on the milder day. PPO (b, d) slams u_{HP} between off and on, firing 4.29 and 2.64 starts/day, more than the baseline.

so the premium buys *capacity*, not tariff mistiming—the modulating compressor simply delivers more heat. PPO's apparently low cost and emissions are instead an artefact of under-delivery: it leaves the zone cold and cycles to do so.

6 Discussion

Why SAC recovers inverter operation? The wear term (2) has two zero-cost optima—never on, or always on—and the discomfort term rules out the first. Reaching the second requires holding u_{HP} strictly inside $(\varepsilon, 1]$ for the whole episode while tracking a time-varying load. SAC's maximum-entropy objective values a broad, stochastic action distribution and penalises collapse onto the deterministic rails, while its off-policy replay smooths the critic; interior, continuously-modulated actions thus become the path of least resistance, and the agent converges to the benign always-on optimum. PPO, lacking a comparable entropy floor and updating on-policy, finds it easier to raise the average duty by mixing fully-on and fully-off actions, landing in the bang–bang basin. SAC's policy is an inverter controller, learned rather than engineered [3, 1, 2], while PPO's is a thermostat. That the zero-start result holds across two scenarios differing by an order of magnitude in mean capacity ($\overline{u} = 0.287$ peak versus 0.098 typical) indicates that continuous modulation is a robust attractor of the SAC objective, not an artefact of one operating point.

Trade-off and practical implications: The algorithms map onto opposite ends of the Pareto picture: SAC is comfort-seeking (a small, bounded cost premium for near-zero discomfort and zero wear) and PPO cost-seeking (under-heating to save energy); for a heat pump SAC's corner is the desirable one, since comfort is the primary obligation and eliminating cycling protects the most expensive component. Because the wear term is levelised in the same currency as operating cost, the two are directly comparable: over the 14-day peak scenario SAC avoids all 15 baseline starts, worth 0.20 € of levelised wear at the assumed 0.0133 € per start, which exceeds the 0.10 € (11.5 %) operating premium—so the premium is recovered roughly twofold (break-even at 0.007 € per start), and the 21 avoided starts on the typical scenario cover their premium likewise ($\approx 1.3\times$, break-even 0.010 € per start). Field controllers normally suppress cycling with hand-tuned timers and hysteresis; our result shows equivalent smoothness can arise endogenously by pricing each start, with w_{wear} as the single knob along the smoothness–cost frontier.

Limitations: The duty threshold $\varepsilon = 10^{-3}$ is permissive: a controller need only hold a near-zero capacity to register as "on," lowering the bar for the always-on optimum; a threshold at a real inverter's minimum stable modulation (10 % to 20 %) would be a stricter test. The per-start cost is small in absolute terms and serves as a *lifetime* proxy rather than an operating expense. Finally, training uses an uncalibrated surrogate plant, so we rely on the out-of-sample BOPTEST KPIs (Table 2) for cost–comfort claims and on the measured duty signal (Table 1, Fig. 1) for smoothness claims.

7 Conclusion

We added a monetised compressor-wear term to a heat-pump control reward and showed that the RL algorithm then determines the qualitative form of the learned control law. SAC discovered a continuous-modulation policy that eliminates cycling, reproducing the operating principle of an inverter-driven heat pump, while PPO collapsed to a bang–bang policy that cycled more than the baseline; on BOPTEST the SAC policy cut thermal discomfort by up to 90.7 % for a modest, bounded cost increase. For equipment whose lifetime is governed by switching, entropy-regularised off-policy learning is thus structurally aligned with the smoothness the hardware requires. Future work will sweep the wear weight to trace the switches-versus-(cost + discomfort) Pareto frontier and benchmark against an MPC baseline on BOPTEST.

References

[1] Hatef Madani, Joachim Claesson, and Per Lundqvist. "Capacity control in ground source heat pump systems: Part I: modeling and simulation". In: *International Journal of Refrigeration* 34.6 (2011), pp. 1338–1347. ISSN: 0140-7007. DOI: https://doi.org/10.1016/j.ijrefrig.2011.05.007. URL: https://www.sciencedirect.com/science/article/pii/S0140700711001137 (cit. on pp. 1–3, 7).

[2] Hatef Madani, Joachim Claesson, and Per Lundqvist. "Capacity control in ground source heat pump systems part II: Comparative analysis between on/off controlled and variable capacity systems". In: *International Journal of Refrigeration* 34.8 (2011), pp. 1934–1942. ISSN: 0140-7007. DOI: https://doi.org/10.1016/j.ijrefrig.2011.05.012. URL: https://www.sciencedirect.com/science/article/pii/S0140700711001186 (cit. on pp. 1–3, 7).

[3] Fredrik Karlsson and Per Fahlén. "Capacity-controlled ground source heat pumps in hydronic heating systems". In: *International Journal of Refrigeration* 30.2 (2007), pp. 221–229. ISSN: 0140-7007. DOI: https://doi.org/10.1016/j.ijrefrig.2006.08.008. URL: https://www.sciencedirect.com/science/article/pii/S0140700706001654 (cit. on pp. 1–3, 7).

[4] AHRI Standard. "Performance rating of unitary air-conditioning & air-source heat pump equipment". In: *AHRI Standard* 210 (2008), p. 240 (cit. on pp. 1, 2).

[5] Zhe Wang and Tianzhen Hong. "Reinforcement learning for building controls: The opportunities and challenges". In: *Applied Energy* 269 (2020), p. 115036. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2020.115036. URL: https://www.sciencedirect.com/science/article/pii/S0306261920305481 (cit. on p. 2).

[6] José R. Vázquez-Canteli and Zoltán Nagy. "Reinforcement learning for demand response: A review of algorithms and modeling techniques". In: *Applied Energy* 235 (2019), pp. 1072–1089. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0306261918317082 (cit. on p. 2).

[7] David Blum et al. “Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings”. In: *Journal of Building Performance Simulation* 14.5 (2021), pp. 586–610. DOI: 10.1080/19401493.2021.1986574. eprint: https://doi.org/10.1080/19401493.2021.1986574. URL: https://doi.org/10.1080/19401493.2021.1986574 (cit. on p. 2).

[8] Tuomas Haarnoja et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html (cit. on pp. 2, 4).

[9] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG]. URL: https://arxiv.org/abs/1707.06347 (cit. on pp. 2, 4).

[10] Ján Drgoňa et al. “All you need to know about model predictive control for buildings”. In: *Annual Reviews in Control* 50 (2020), pp. 190–232. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2020.09.001. URL: https://www.sciencedirect.com/science/article/pii/S1367578820300584 (cit. on p. 2).

[11] Javier Arroyo et al. “Reinforced model predictive control (RL-MPC) for building energy management”. In: *Applied Energy* 309 (2022), p. 118346. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.118346. URL: https://www.sciencedirect.com/science/article/pii/S0306261921015932 (cit. on p. 2).

[12] Siddharth Mysore et al. “Regularizing Action Policies for Smooth Control with Reinforcement Learning”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 1810–1816. DOI: 10.1109/ICRA48506.2021.9561138 (cit. on p. 2).

[13] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. “Discovering governing equations from data: Sparse identification of nonlinear dynamical systems”. In: *arXiv preprint arXiv:1509.03580* (2015) (cit. on p. 4).

[14] Aniket Dixit et al. *Learning from Less: SINDy Surrogates in RL*. 2025. arXiv: 2504.18113 [cs.LG]. URL: https://arxiv.org/abs/2504.18113 (cit. on p. 4).

[15] Antonin Raffin et al. “Stable-Baselines3: Reliable Reinforcement Learning Implementations”. In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html (cit. on p. 4).