A Simulation Study on Energy Optimization in Building Control with Reinforcement Learning
*

Peter Bolt[1], Volker Ziebart[1], Christian Jaeger[1], Nicolas Schmid[1], Thilo Stadelmann[1,2], and Rudolf M. Füchslin[1,2]

[1] Zurich University of Applied Sciences Winterthur, Switzerland
[2] European Centre for Living Technology (ECLT), Ca' Bottacin, Venice, Italy

Abstract. We propose and evaluate a deep reinforcement learning control paradigm for building energy systems. In comparison to other advanced control techniques, namely Model Predictive Control, the reinforcement learning paradigm avoids the costs and uncertainties associated with the requirement for a control-oriented model. We apply a mixed agent for the Proximal Policy Optimization algorithm, similar to the algorithm proposed in [7] as well as a non-discounted finite horizon optimization scheme.
We investigate the capabilities of the proposed reinforcement learning controller regarding energy efficiency, comparing it against the most widely used rule-based control paradigm as a baseline controller. We verify our proposed paradigm in a simulation study with building models implemented in Dymola.

Keywords: smart building · building control · reinforcement learning.

1 Introduction

The transition from fossil fuels to renewable energy sources entails increased complexity in energy management, due to varying availability and the difficulties of storing electrical and thermal energy. Building energy systems (BES), representing an important share of overall energy consumption, offer significant potential in reducing the carbon footprint, provided that an optimized energy management adapted to both availability and usage patterns can be realized.

Prior work has established that advanced control techniques such as Model Predictive Control (MPC) can yield substantial energy savings in BES [3, 5, 12]. However, despite intensive research, especially in MPC for BES, most buildings still rely on simple rule-based controllers (RBC). A possible reason for this is the high computational complexity associated with online MPC, along with the need for an accurate, control-oriented model to describe the physical system in a way amenable to real-time optimization. For many applications, this restricts the model to being essentially linear. Creating this control-oriented model is

* Funded by Innosuisse 31326.1 IP-EE.

often the most time-consuming part of MPC design and can result in a compromise between prediction accuracy and computational complexity that leads to suboptimal controller performance.

This work is motivated by the idea that Deep Machine Learning (ML) holds promise in solving the energy management problem. ML, as a data-driven technique, has gained traction as a powerful paradigm across various application domains, including building control [11, 15]. Particularly in an engineering context [1], ML can complement the model-driven methodologies. We explore the potential advantages of a control system based on deep reinforcement learning (Reinforcement Learning Control, RLC) in the context of BES. We investigate various RLC variants and compare them to conventional controllers. Notably, the investigated RLC schemes do not necessitate a control-oriented model, potentially reducing development time and enhancing controller performance. However, they do require a numerical model of the BES to gather training data—often spanning several simulated years.

We apply a mixed agent for the Proximal Policy Optimization algorithm (PPO) [9], capable of handling the mixed continuous and discrete action space inherent to our problem (see Section 3.2) and a finite horizon optimization scheme to replace the traditional infinite horizon discounting scheme (see Section 3.3). This change is driven by the observation that the discounting scheme made delaying the import of energy appear cheaper to the controller, whereas in reality, it led to increased costs due to energy import becoming an urgent need at an ill-suited moment.

2 Building energy system (BES)

The investigated system is a simplified version of the one used by the authors in [3], which is based on a real application scenario. It integrates a small-scale, grid-connected photovoltaic collector (PV), one heat pump (HP) charging one thermal storage and supplying energy to a low-temperature surface heating. The consumer is a single-family house modeled as a second-order Lumped Parameter Model, consisting of two resistors and two capacitors. Fig. 1a shows the schematic of the installation which integrates

- Ice storage (IceSt) 84 m^3
- Room heating storage (RhSt) 500 liter
- Photovoltaic (PV) collector, 30 m^2, south-facing
- Brine to water HP with nominal heat output 3.93 kW at B0/W35 with a Coefficient of Performance (COP) of 4.65

The heat pump has a variable speed compressor, allowing it to adjust its operating electrical power (power modulation). Ice storage is a low-temperature storage with a very long time constant and a high capacity. The room heating storage is a high-temperature storage with low capacity.

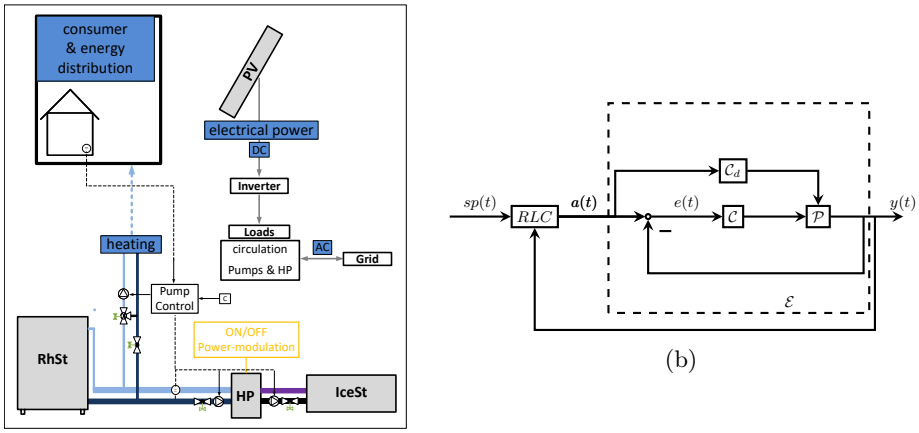


Fig. 1: **a)** Scheme of the HVAC system simulated with Dymola. Heating storage tank (RhSt), heat pump (HP), ice storage (IceSt), and photovoltaics (PV). **b)** Hierarchical control architecture. $\mathcal{P}$: Plant to be controlled, $\mathcal{C}$:continuous (PI) controllers, $\mathcal{C}_d$: discrete HP controller of the inner loop, and $\mathcal{E}$: Environment, constituting the inner closed-loop system. The RL module establishes setpoints for room temperature and HP electrical power, and handles the on/off operation of the HP.

3 Control Architecture

We employ the hierarchical architecture depicted in Fig. 1b. Proportional-Integral (PI) controllers are used in the inner loop to achieve system stabilization for the fast dynamics. The outer loop is a reference governor (RG) [2] for the inner loop. The RG's objective is to minimize the energy consumption of the inner loop while adhering to setpoints and system limitations.

3.1 The reinforcement learning controller

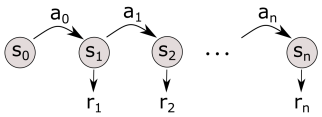


Fig. 2: Markov Decision Process with states s_t, actions a_t and rewards r_t

We formulate the control problem as a Markov Decision Process as depicted in Fig. 2 within the environment $\mathcal{E}$ [13]. The control episode starts a state s_0.

The controller then successively chooses actions a_t from a set of possible actions $\mathcal{A}$ according to the control policy $\pi(a, s)$ which is the probability of choosing the action a in the state s. Due to the applied actions and the dynamics of the environment, the states s_t will evolve over time. The performance of the controller is measured by the reward r_t which is a function of s_t and the action a_t.

In most RL applications the return R_t is defined as the discounted sum of all future rewards in the control episode of length n

$$R_t = \sum_{l=0}^{n-t} \gamma^l r_{t+l} \tag{1}$$

where $\gamma \in (0, 1]$ is the discount factor, which must be tuned during the training process. A value of $\gamma < 1$ means that rewards count less, the further they lie in the future, reflecting the fact that increased temporal distance introduces more uncertainties. Consequently, smaller yet more assured rewards in the near future are preferred over potentially larger but riskier rewards further from the present.

The return can be used to define a quality measure of the chosen actions, namely the Action-Value Function (Q-function):

$$Q^{\pi,\gamma}(s_t, a_t) := \mathbb{E}[R_{t+1} \mid s_t, a_t] \tag{2}$$

The Q-function represents the expected return when starting from state s_t, taking action a_t, following policy π thereafter, and using the discount factor γ. Similarly, the State-Value Function (V-function) is defined as the expected return when starting from state s_t, following policy π thereafter.

$$V^{\pi,\gamma}(s_t) := \mathbb{E}[R_{t+1} \mid s_t] \tag{3}$$

The function $A(s_t, a_t)^{\pi,\gamma}$ is called advantage and specifies how much better or worse the value of the action a_t is compared to the π-weighed average of all possible actions in the state s_t. It is given by

$$A(s_t, a_t)^{\pi,\gamma} = Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t) \tag{4}$$

In our research, we predominantly utilized the Proximal Policy Optimization (PPO) algorithm as outlined in [9]. PPO is a state-of-the-art reinforcement learning algorithm that combines policy gradient techniques with a clever clipping mechanism. This clipping approach is employed to mitigate the risk of excessively aggressive policy updates that could lead to instability in the learning process. Another advantage of PPO is its ability to engage with multiple environments in parallel, significantly expediting the training process. As described in [10], we approximately maximize the following objective during PPO training:

$$L = \hat{\mathbb{E}}[L_A - c_V L_V + c_S L_S] \tag{5}$$

Here, c_V and c_S are non-negative weighting factors, and L_S contributes to entropy loss for enhanced exploration. Despite the potential benefits of promoting exploration through the entropy term (e.g. [10]), our experimentation did not reveal any notable improvements. Therefore, we set the weighting factor c_S to zero. L_V corresponds to the aggregation of squared deviations between the learned value function V_θ—represented by the neural network's parameters θ—and the estimated returns $\hat{V} = R_{t+1}$ derived from interactions with the environments.

The first and most important term is given by

$$\hat{\mathbb{E}}[L_A] = \sum_{k=1}^{m} \sum_{t=0}^{n} A_{\theta_0}(s_t^k, a_t^k) \frac{\pi_\theta(a_t^k, s_t^k)}{\pi_{\theta_0}(a_t^k, s_t^k)} \tag{6}$$

where θ_0 denotes the parameter set of the neural network that was used when interacting with the environment and generating the data set $\{s_t^k, a_t^k, r_t^k\}_{k=1,\ldots,m}$ for the m parallel episodes. When optimizing the loss (equation 6), we employ the following estimate $\hat{A}$ for A [9]:

$$\hat{A}(s_t, a_t) = r_t + \gamma\, V_{\theta_0}(s_{t+1}) - V_{\theta_0}(s_t) \tag{7}$$

The optimization of L_A can be understood as follows: For $\theta = \theta_0$, the quotient $\lambda = \pi_\theta / \pi_{\theta_0}$ is equal to 1. Starting from $\theta = \theta_0$, the parameters of the neural network are optimized such that λ will be increased for positive advantages (actions with better-than-average action-value) and decreased for actions with negative advantages. Therefore, actions with positive advantages will have a higher probability of being chosen in the next iteration.

In our application, the environment comprises the sub-controllers $\mathcal{C}$ and the Building Energy System $\mathcal{P}$ (see Fig. 1b). The possible actions applied every hour are as follows:

(i) Heat pump on/off
(ii) Heat pump power modulation $60 - 100\%$
(iii) Indoor temperature setpoint $20 - 23°\mathrm{C}$

From a BES perspective, if necessary, we can easily convert (ii) and (iii) into discrete actions, provided that the discretization has a fine enough granularity. Concerning (i), the operation of the HP can be approached either discretely by switching the HP on or off every hour or continuously by turning it on or off for a duration between 0s and 3600s.

The state information comprises the current supply and return flow temperatures, the flow rates of the heating circuit, the room temperature of the building, as well as the room heating storage temperature and the power produced by the photovoltaic collector. Additionally, the state includes a 12-hour forecast of outdoor temperature and solar irradiation, updated every hour.

The reward consists of two terms: (i) a deviation of the room temperature from the tolerance band $20-23°\mathrm{C}$ results in a linearly increasing negative reward (a penalty), and (ii) an energy penalty proportional to the energy drawn from the grid. The use of electricity from the PV collector incurs no penalties.

3.2 Mixed Agent

In practice, even actions that are naturally continuous, such as temperature set points, are often discretized and handled by a discrete agent. However, if high resolution is required, the number of discretized actions can become large and it's uncertain whether the same performance can be achieved as with a continuous agent. The mixed Agent is analogous to the one proposed in [7] for the Maximum a Posteriori Policy Optimization (MPO) agent.

In the case of **continuous action** spaces, the Gaussian policy is commonly used. Here, actions $a_c = \{a_{c,i}\}$ are drawn from a parameterized uncorrelated normal distribution represented by the equation:

$$\pi_c(a_c \mid s) = \prod_i \frac{e^{-\frac{(a_{c,i} - \mu_{\theta,i}(s, a_{c,i}))^2}{2\sigma^2_{\theta,i}}}}{\sqrt{2\pi}\sigma_{\theta,i}} \tag{8}$$

The functions $\mu_{\theta,i}(s,a)$ and $\sigma_{\theta,i}(s,a)$ are approximated by a neural network.

For **discrete action** spaces with actions a_d , the softmax policy is used. It is defined by the equation:

$$\pi_d(a_d \mid s) = \frac{e^{\phi_\theta(s,a_d)}}{\sum_{a'_d \in \mathcal{A}} e^{\phi_\theta(s,a'_d)}} \tag{9}$$

Here, the function $\phi_\theta(s, a_d)$ is represented by a neural network, and $\pi_d(a_d \mid s)$ calculates the probability of taking the action a_d given state s. The softmax policy ensures that the probabilities of all possible actions sum up to 1.

For the **mixed agent** with actions $a = \{a_c, a_d\}$ we used the combined distribution given by the product

$$\pi(a, s) = \pi_c(a_c \mid s)\ \pi_d(a_d \mid s) \tag{10}$$

The functions $\mu_{\theta,i}(s,a)$, $\sigma_{\theta,i}(s,a)$, and $\phi_\theta(s,a_d)$ as well as $V_\theta(s)$ are all approximated by the same neural network with parameters θ, which are found by optimizing the loss given by equation 5.

Exploration in the case of a continuous action space is accomplished by sampling actions according to the latest version of the policy network μ_θ and then adding Gaussian noise. This results in the action $a_c = \mu_\theta + \mathcal{N}(0, \sigma_\theta)$, which is subsequently applied within the environment.

For the scenario of a discrete action space, we generate samples from the categorical distribution $\pi_d(a_d \mid s)$. In this case, the most recent iteration of the policy network ϕ_θ is perturbed according to: $a_d = \text{argmax}(\phi_\theta - \log(-\log(u)))$, with $u \sim \text{Uniform}(0, 1)$, and the resulting action is applied to the environment.

3.3 Finite horizon optimal control problem

Discounting future rewards (see equation 1) suffers from three disadvantages in our application.

The first one arises from the fact that future energy draws from the grid are discounted which means that the HP seems to need less energy the more the action "HP on" lies in the future. This is briefly illustrated by the following example: Let us assume that we have a sunny winter afternoon and in order to maintain the required room temperature we need to run the HP for one hour in the coming 6 hours. Currently, the PV system provides about 20% of the power the HP needs and therefore 80% must be taken from the grid. Thus, it seems to make sense to immediately switch on the HP and make use of the free PV power. However, if $\gamma = 0.95$ the return, which is always negative in our application, is higher if the power-up of the HP is delayed for the maximum of 5 hours because $\gamma^5 = 0.77$. But running the HP after sunset would certainly cost more grid energy.

This underweighting of future grid energy draws can be reduced by choosing a discount factor very close to one. But then the second disadvantage becomes apparent. Due to the very slow decay of the weighing factor γ^l if $\gamma \approx 1$ rewards that will be generated far in the future with almost no causality to the current action will be taken into account. Furthermore, the long optimization horizon increases the complexity exponentially, resulting in longer comutation time and higher variance in training performance.

The authors of [14] discuss how different values of γ would influence the behavior of an RL controller for a heat pump. They compare results with $\gamma = 0.25$ and $\gamma = 0.75$ (both strongly discounted) and observe the following: (i) higher values assign greater importance to achieve long-term rewards and accordingly lead to more frequent operation of the heat pump when the outdoor temperature is high (higher coefficient of performance (COP)). Pre-heating during periods with a high COP consumes more energy at the current time step while saving energy for upcoming time steps. If γ is small, the discounted future savings could not justify the current costs.

Because of these shortcomings we introduced a finite non-discounted optimization window for PPO which favors achieving long-term rewards over a reasonable time horizon without increasing the complexity of the optimization problem too much. The return is given by

$$R_t = \sum_{l=0}^{N_h} r_{t+l} + V_F \tag{11}$$

The prediction horizon N_h replaces the discounting factor γ. V_F accounts for the return value of the final state, reached after N_h steps. We obtained our results by simply setting $V_F = 0$. This can be justified by the fact that the planning horizon is sufficiently long in comparison with the time constants of the system that the final cost term is negligible for the determination of a_t. Additional work may be needed to fully understand the behaviour with respect to this term. A reasonable value for N_h can be estimated based on the time constants present in the system. In our case, the time constants of two thermal heat storages, RhSt and building capacity, are relevant. Equations 2 to 10 remain applicable without modification to this new return formulation.

In the following chapter, we present and compare results for various configurations of the PPO algorithm using both discounted and finite non-discounted reward windows. In addition, we also present results we obtained using the Soft Actor-Critic (SAC) [6] algorithm to find an optimal control strategy. Table 1 lists the investigated configurations.

#	algorithm	action 1	action 2	action 3	γ	N_h
1	$\text{PPO}_{d:\gamma=0.95}$	on/off	$\{60, 65, ..., 100\}\%$	$\{20, 21, 22, 23\}°\text{C}$	0.95	-
2	$\text{PPO}_{d:\gamma=0.99}$	on/off	$\{60, 65, ..., 100\}\%$	$\{20, 21, 22, 23\}°\text{C}$	0.99	-
3	$\text{PPO}_{d:N_h=12}$	on/off	$\{60, 65, ..., 100\}\%$	$\{20, 21, 22, 23\}°\text{C}$	-	12
4	$\text{PPO}_{m:\gamma=0.95}$	on/off	$[60, 100]\%$	$[20, 23]°\text{C}$	0.95	-
5	$\text{PPO}_{m:\gamma=0.99}$	on/off	$[60, 100]\%$	$[20, 23]°\text{C}$	0.99	-
6	$\text{PPO}_{m:N_h=12}$	on/off	$[60, 100]\%$	$[20, 23]°\text{C}$	-	12
7	SAC	rt. var.	$[60, 100]\%$	$[20, 23]°\text{C}$		

Table 1: Evaluated parameter sets and discrete or continuous actions of PPO and SAC. Actions: (1) Heatpump turned on/off every hour or in case of the SAC the controller can choose a variable runtime ("var.rt.") between 0s and 3600s every hour, (2) Heatpump power modulation in percentage of maximum power, (3) Indoor temperature setpoint.

4 Simulation Results

The RLCs were trained in co-simulation with a BES modeled in Dymola [4]. For the implementation of the RLCs, we employed the OpenAI framework [8]. Our optimization approach for the RLCs aimed at achieving two primary goals: (i) minimizing electricity consumption from the grid and (ii) maintaining room temperature within the range of $20 - 23°\text{C}$. The range enables the RLC to effectively utilize the building mass as an additional heat storage mechanism (see Fig. 5b). Leveraging the building's mass roughly doubles the short-term storage capacity of the system. In the upper plot of Fig. 5b, it is evident that the RLC capitalizes on this by activating the heat pump during periods of available surplus PV electricity. Throughout the investigation, only a negligible number of small constraint violations were observed. To assess and compare the performance of the RLCs, we have included a baseline RBC and an improved RBC in our evaluation. The baseline RBC uses a heat pump with fixed operating power (no power modulation) and a constant setpoint for the room temperature of 20°C. Furthermore, the heat pump is turned on and off according to a constant threshold temperature in the RhSt. In the improved RBC, both the heatpump power and the room heating storage setpoint were decreased until

there was just enough energy left to cover the requirement. Moreover, a time-dependent threshold temperature in the RhSt was introduced, which increases if PV electricity is available. This adjustment results in significantly higher self-consumption of PV electricity. It is only possible with prior knowledge of the energy requirement and availability of PV-energy, therefore it demonstrates the quasi-upper-performance limit achievable with the used RBC structure. Fig. 3 shows, that this dramatically reduces the RBC's grid electricity consumption.

4.1 Performance Comparison and Training Speed

Fig. 3 illustrates the electricity export versus import from the grid. The controllers minimize electricity import, while export is neither optimized nor constrained. The RLCs demonstrate remarkable energy efficiency results. Among the top-performing RLCs, PPO_d with a prediction horizon of $N_h = 12$ hours consumed an average of $940\frac{kWh}{a}$ of energy from the electricity grid. A improved RBC, also utilizing the building as a heat storage, required $1280\frac{kWh}{a}$, while a simple RBC required $2060\frac{kWh}{a}$. Furthermore, PPO_d exported more electrical energy to the grid than the improved RBC. Overall, RLCs with non-discounted finite windows, specifically PPO_d and PPO_m, exhibit outstanding performance with significantly lower variance compared to RLCs with discounted infinite windows. All simulations were conducted under identical circumstances.

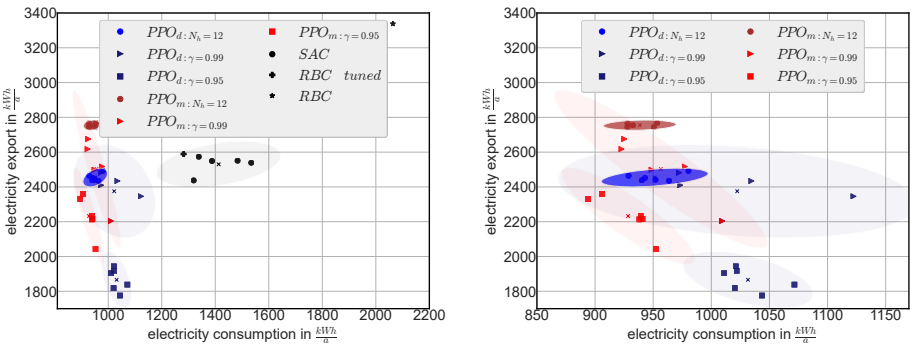
Table 2 shows the number of samples required to achieve $85\%, 90\%, 95\%$, and 99% of the maximum reward. The mixed agents achieve high rewards more quickly. For instance, to reach 90% of the maximum performance, $PPO_{m:N_h=12}$ requires 1.8×10^6 samples, while $PPO_{d:N_h=12}$ requires 3.7×10^6 samples.

PPO parameters / % of max reward	d:N_h=12	d:γ=0.99	d:γ=0.95	m:N_h=12	m:γ=0.99	m:γ=0.95
85%	2.0	3.1	2.0	1.0	1.8	1.6
90%	3.7	3.9	2.9	1.8	2.7	1.8
95%	4.3	4.9	3.1	3.7	3.3	5.1
99%	9.8	8.8	9.2	9.2	7.1	6.9
yearly electricity consumption kWh	951	1022	1032	939	957	928
mean ± std	±17	±55	±20	±11	±33	±121

Table 2: Number of samples in millions required to reach $85\%, 90\%, 95\%$, and 99% of the maximum reward of each agent for the first time.

4.2 Discounted versus Non-discounted Windows

We conducted an empirical investigation to examine the impact of different types of optimization windows. The proposed non-discounted window consistently matches or surpasses the discounted versions in all scenarios, resulting in



(a) (b)

Fig. 3: **a)** Energy efficiency comparison among different controllers optimized to minimize grid electricity consumption (kWh) while maintaining room temperature within $20 - 23°$C. The vertical axis: surplus energy produced by the PV system and supplied to the grid. Subscripts denote specific properties of the PPO Controller: PPO_d indicates discrete actions, PPO_m indicates mixed actions. The parameter γ represents the discount factor used in the optimization window, if a discounted window was employed. All simulations were run under identical (environmental) conditions. The variance over different runs is due to the stochastic nature of the sampling inherent to reinforcement learning. Fig. **b)** Zoomed-in View of Fig. 3a.

not only lower mean energy consumption but also reduced energy consumption variance. The outcomes are depicted in Fig. 3, where various RLCs with different windows were tested with identical environments.

Optimization becomes more challenging when there's a limited availability of surplus PV electricity. In Fig. 4, we reduced the PV area from $30m^2$ to $15m^2$. Notably, in this scenario, the non-discounted PPO_d outperforms the discounted $\text{PPO}_{d:\gamma=0.95}$ by approximately 13% in grid electricity consumption reduction. Concurrently, the exported electricity increases by over 30%.

5 Conclusion and Outlook

We have shown that an RLC has a great potential for saving primary energy. In contrast to MPC the RLC approach avoids the expenses and compromises associated with developing a control-oriented model.

The use of a mixed agent has significantly accelerated the learning process, achieving performance that is at least comparable to, if not superior to, a standard PPO agent. Nevertheless, the amount of data required remains impractical for online learning applications. In fact, we would need a reduction of several orders of magnitude to eliminate the dependency on building simulations.

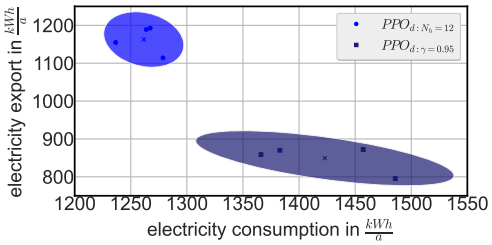
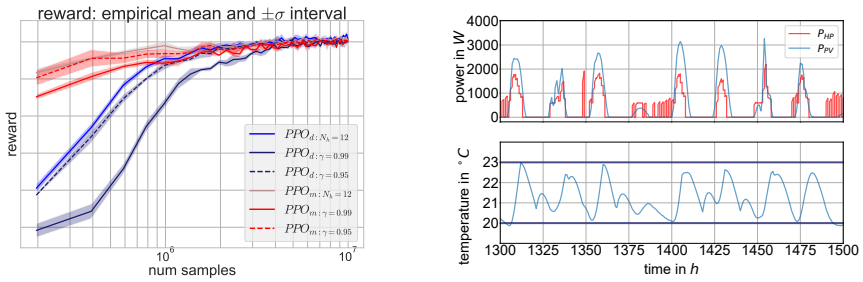


Fig. 4: Simulation with Reduced PV Area. The PV area is $15m^2$, compared to the original $30m^2$.



(a) (b)

Fig. 5: **a)** Evolution of reward during training. Solid lines: mean performance. The shaded regions: lower 10th and upper 90th percentiles over 6 trials. Note the significant variation in the initial phase. **b)** Top: Electricity production with PV and the actively modulated power consumption of the heat pump. Bottom: Room temperature within the target range of $20 - 23°C$.

While deep RL provides a comprehensive framework for deriving optimal control strategies directly from data, further progress is needed to effectively apply RL to real-world BES. Within the current range of RL methods, two intriguing paths for exploration emerge: One involves initiating the learning process with a RBC as the initial policy, rather than starting with a randomly initialized neural network. This approach would facilitate the learning process with a reasonable performance baseline. Alternatively, we could leverage pre-trained models and adapt them, either with or without additional training, for applications in BES.

References

1. Balali, Y., Chong, A., Busch, A., O'Keefe, S.: Energy modelling and control of building heating and cooling systems with data-driven and hybrid models—a review. Renewable and Sustainable Energy Reviews **183**, 113496 (2023)
2. Bemporad, A.: Reference governor for constrained nonlinear systems **43**(3), 415–419. https://doi.org/10.1109/9.661611

3. Bolt, P., Ziebart, V., Jaeger, C., Ritzmann, R., Meier, O., Füchslin, R.M.: Model predictive control for building automation. pp. 1330–1341. International Solar Energy Society. https://doi.org/10.21256/zhaw-3296, https://digitalcollection.zhaw.ch/handle/11475/16903
4. Fritzson, P.: Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley, Hoboken, NJ (2004)
5. Fux, S.F., Ashouri, A., Benz, M.J., Guzzella, L.: EKF based self-adaptive thermal model for a passive house **68, Part C**, 811–817. https://doi.org/10.1016/j.enbuild.2012.06.016, http://www.sciencedirect.com/science/article/pii/S0378778812003039
6. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor (Aug 2018). https://doi.org/10.48550/arXiv.1801.01290, http://arxiv.org/abs/1801.01290, arXiv:1801.01290 [cs, stat]
7. Neunert, M., Abdolmaleki, A., Wulfmeier, M., Lampe, T., Springenberg, T., Hafner, R., Romano, F., Buchli, J., Heess, N., Riedmiller, M.: Continuous-discrete reinforcement learning for hybrid control in robotics. In: Conference on Robot Learning. pp. 735–751. PMLR (2020)
8. OpenAI: OpenAI. https://openai.com/ (2003–2021), accessed: September 2021
9. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation http://arxiv.org/abs/1506.02438
10. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
11. Stadelmann, T., Tolkachev, V., Sick, B., Stampfli, J., Dürr, O.: Beyond imagenet: deep learning in industrial practice. Applied data science: lessons learned for the data-driven business pp. 205–232 (2019)
12. Sturzenegger, D.: Model predictive building climate control - steps towards practice
13. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. Adaptive computation and machine learning, MIT Press
14. Vázquez-Canteli, J., Kämpf, J., Nagy, Z.: Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted q-iteration **122**, 415–420. https://doi.org/10.1016/j.egypro.2017.07.429, https://www.sciencedirect.com/science/article/pii/S1876610217332629
15. Wang, Z., Hong, T.: Reinforcement learning for building controls: The opportunities and challenges. Applied Energy **269**, 115036 (2020)