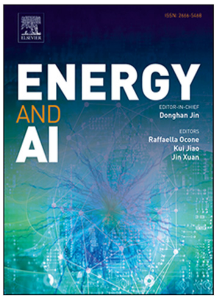


Combined peak reduction and self-consumption using proximal policy optimization

Thijs Peirelinck, Chris Hermans, Fred Spiessens, Geert Deconinck

PII: S2666-5468(23)00095-2
DOI: https://doi.org/10.1016/j.egyai.2023.100323
Reference: EGYAI 100323

To appear in: *Energy and AI*

Received date : 4 October 2022
Revised date : 6 June 2023
Accepted date : 19 November 2023

Combined Peak Reduction and Self-Consumption Using Proximal Policy Optimization

Thijs Peirelinck[a,b,c,*], Chris Hermans[b,c], Fred Spiessens[b,c], Geert Deconinck[a,c]

[a] *ESAT, KU Leuven, Kasteelpark Arenberg 10 bus 2445, 3001 Leuven, Belgium*
[b] *AMO, Flemish Institute for Technological Research (VITO), Boeretang 200, 2400 Mol, Belgium*
[c] *AMO, EnergyVille, Thor Park 8310, 3600 Genk, Belgium*

Abstract

Residential demand response programs aim to activate demand flexibility at the household level. In recent years, reinforcement learning (RL) has gained significant attention for these type of applications. A major challenge of RL algorithms is data efficiency. New RL algorithms, such as proximal policy optimisation (PPO), have tried to increase data efficiency. Additionally, combining RL with transfer learning has been proposed in an effort to mitigate this challenge. In this work, we further improve upon state-of-the-art transfer learning performance by incorporating demand response domain knowledge into the learning pipeline. We evaluate our approach on a demand response use case where peak shaving and self-consumption is incentivised by means of a capacity tariff. We show our adapted version of PPO, combined with transfer learning, reduces cost by 14.51% compared to a regular hysteresis controller and by 6.68% compared to traditional PPO.

1. Introduction

Renewable Energy Sources (RES) are reshaping the energy sector's landscape. Due to their decentralised and intermittent nature, market and tariff designs are challenged. In the Flemish region of Belgium the energy regulator (VREG) has recently announced a change of distribution fee design [24]. Previously, Flemish residential electricity distribution fees have been energy-based. The rise of residential photovoltaic (PV) installations and net-metering meant a reduction in income for the Distribution Network Operator (DNO). With the introduction of digital metering, the regulator takes the opportunity to introduce a capacity tariff, starting from 2023 [24]. The regulator motivates its decision by arguing that the DNO's main costs are capacity-based rather than energy-based [24]. As a result, a more cost-reflective price signal is provided to consumers. At the same time, by tying the distribution grid fees to power rather than energy consumption, the VREG encourages consumers, aiming to minizime cost, to reduce their peak power consumption. This causes an interesting application for Demand Response (DR), which we will consider in this work.

[*] Corresponding author

Preprint submitted to Elsevier

Thermostatically Controlled Loads (TCLs) are considered excellent appliances for DR, as they have an inherent energy buffer [13]. In other DR applications with TCLs, Reinforcement Learning (RL) has shown promising results. For instance, Ruelens *et al.* [16] showed that Fitted Q-Iteration (FQI) can reduce energy consumption cost of a heat pump by 19 %, compared to a default controller, in an energy arbitrage scenario. And Mbuwir *et al.* [8] apply the same algorithm for local optimisation in their transactive control framework. Their methodology manages to reduce grid congestion using flexibility available in the microgrid's heat pumps.

Additionally, multiple examples exist of successful Electric Water Heater (EWH) control with RL [16, 6, 3, 23]. For example, [16] shows RL can be used to find a open-loop heat pump thermostat schedules required to participate in the day-ahead market. Reducing peak power consumption can (in part) be accomplished by local PV self-consumption. Self-consumption is a DR application in itself, and earlier work has applied RL for maximising residential self-consumption. Soares *et al.* [20] use a model-based RL algorithm to control residential batteries and heat pumps. In their field-test, they achieve a 68 % average self-consumption rate. This means, on average, 68 % of heat pump energy is covered by local PV generation. In a similar field-test, using the same model-based RL approach, but only scheduling the EWH heat cycle, De Somer *et al.* [4] manage to increase PV self-consumption by 20 %.

The capacity tariff DR application, which is considered here, differs from Soares *et al.* [19, 20] and De Somer *et al.* [4] as the goal is not to maximise self-consumption. Rather, it is to minimise peak power consumption. In the past, we have already touched upon a capacity tariff scenario [14]. However, the capacity tariff treated here is different, and is as proposed by the Flemish regulator in Belgium, *i.e.*, VREG [24]. The DR use case of peak power reduction is of interest in other regions as well [1].

State-of-the-art RL has improved over time. Since its introduction, policy gradient methods [21] have gradually gained interest. Compared to value iteration methods, the policy gradient approach uses a function approximator that explicitly represents the policy [21]. To further improve the policy gradient, which is used to update the approximator of the policy, an estimate of the expected future reward can be used [7]. This is achieved by the introduction of a critic [7]. These actor-critic methods thus combine the advantages of value iteration and policy iteration [7]. Proximal Policy Iteration (PPO) [18] is a recently introduced family of actor-critic methods and is now widely used in RL research. It has been introduced in an effort to mitigate two of the main drawbacks of RL: (1) data (in)effiency and (2) the (dis)ability to perform well in a variety of domains [18]. Earlier research [12] has shown the benefits of model-free RL in residential DR applications. In short, due to the hetergenuous nature of the appliances and users, the problem quickly becomes intractable. Additionally, the two mentioned properties of PPO are of particular interest for DR. The first property (data effiency) is important because the end user prefers to have a well performing agent as soon as possible. The second property (perform well in a variety of domains) motivates the effort to apply PPO in the DR domain. Consequently, we have opted to use PPO in this work. Furthermore, because earlier work [12] has shown transfer learning can further improve data effiency of RL algorithms, we combine PPO with transfer learning.

Thus, in this work, PPO has been adapted for automated DR when a consumer with an EWH is billed, at least partly, based on quarter hourly peak power consumption. The RL controller's aim is to minimise final energy cost by turning the EWH on or off. Cost can be minimised by avoiding energy consumption when other (inflexible) loads are already using power or by self-consuming locally generated PV power. To achieve this goal, we use transfer learning to pre-train the agent based on readily available water consumption [5], electric power consumption [2] and production [15] data. This paper's main contribution is the presentation of a data-efficient model-free residental DR algorithm and learning pipeline. It does so by proposing a method to incorporate expert knowledge and transfer learning in the learning process, and by showing the benefits of a state-space design that aims to be independent of absolute environment parameters, such as PV system size. Additionally, this paper contributes to design of RL algorithms which can be applied in dynamically changing environments by showing that the proposed RL-agent adapts to the seasonality of the control problem.

We test our approach on data from real households, obtained from a field-test in the Netherlands [4]. Although the use case considers the Flemish tariff design, the method is more generally applicable. Moreover, reducing peak power demand is a widespread DR setting.

This paper has been divided in four sections. Section 2 gives a more detailed formulation of the capacity tariff design, formulates the Markov Decision Process (MDP) and lays out the RL algorithm and its modifications. The paper then goes on to Section 3, presenting the experiments and discussing their results. Finally, Section 4 concludes this work and gives future work directions.

2. Problem Formulation & Algorithm

The first part of this section elaborates on the capacity tariff, as designed by the Flemish electricity and gas regulator (VREG). The second part defines the Sequential Decision making Problem (SDP), and formulates it as an MDP. The final part presents the algorithm used to solve the MDP.

2.1. Tariff Design

A current Flemish residential electricity bill approximately consists of three parts; one part the energy cost [€/kWh], a second part the distribution costs [€/kWh] and a third part taxes and levies (only partly dependent on energy or power consumption). In the remainder of this work, we assume decoupling of the Flemish electricity bill in these three earlier mentioned tariff parts. Traditionally, residential consumers only have a Ferraris meter installed, which is limited to net metering of energy consumption. The introduction of residential PV installations meant the DNO saw its income reduce, as so-called *prosumers* have relatively low net energy consumption and, as mentioned, distribution costs are energy-based. To compensate for this loss, a *prosumer-tariff* was introduced. This tariff is charged based on the power inverter capacity of the PV installation [€/kW$_{\text{inverter}}$] [24].

With the introduction of digital metering comes the ability to measure electricity consumption and production separately, and have finer grained measurement points. Together with the observation that distribution grid investment cost is mainly tied to grid capacity (and not energy transported), the regulator opted for a capacity-based distribution fee, from 2023 onwards. This approximately results in the second part of the earlier mentioned electricity bill being capacity-based [€/kW] [24]. As such, a capacity fee provides a more cost-reflective price signal to end-consumers.

The peak power to calculate the bill is based on the quarter-hourly measurements of the digital meter. The capacity fee of a residential consumer will be calculated based on the running Mean Month Peak (MMP) of the past 12 months. It only takes into account net off-take, *i.e.*, there is no capacity fee based on grid injection. Thus, assuming P_t^m is the quarter-hourly power consumption time-series of the current month m in kW, *i.e.*, the digital meter output, and λ_P is the price per kW in euro, the capacity fee F of a residential consumer is calculated by Eq. (2).

$$\text{MMP} = \frac{\sum_{i=m-12}^{m} \max(2.5, \max(P_t^i))}{12} \tag{1}$$

$$F = \lambda_P \cdot \text{MMP} \tag{2}$$

Eq. (1) implies a minimal capacity fee based on a monthly peak power consumption of 2.5 kW, as in the tariff design [24]. The main aim of this work is to minimise the final energy bill, *i.e.*, including the parts related to energy consumption and taxes. The total energy bill is calculated by Eq. (3).

$$C = \lambda_E \cdot E + \lambda_P \cdot \text{MMP} + \lambda_{\text{tax}}^E \cdot E + \lambda_{\text{tax}} \tag{3}$$

with λ_E the energy price, E the total energy consumption of the considered year, λ_{tax}^E the taxes charged based on energy consumption and λ_{tax} the fixed taxes payable per year. Eq. (3) has been used to judge agent performance.

In RL, the reward function can be used to direct the agent to optimal parts of the solution space, based on expert knowledge. For example, here we know self-consumption of locally generated PV will be beneficial for both reducing energy consumption cost and reducing the MMP. Furthermore, λ_{tax} is independent of MMP and E. As a consequence, neither equation (2) nor equation (3) is used as the MDP's reward-function. The following part of this section formulates the control problem as MDP by presenting the state-space, action-space and reward-function.

2.2. Markov Decision Problem

The problem is formulated as a discrete-time MDP with time steps of length $\Delta t = 15$ minutes. The MDP consists of state-space $\mathcal{X}$, action-space $\mathcal{U}$, reward-function $r : (\mathcal{X}, \mathcal{U}) \rightarrow \mathbb{R}$ and state-transition probabilities $p(\cdot|x, u)$, which are unknown to the agent and can only be observed through interaction with the EWH model. The agent is unaware of these transition probabilities. The agent's goal is to learn a policy $\pi : \mathcal{X} \rightarrow \mathcal{U}$ which maximises cumulative reward.

With the setting as mentioned in the previous section, the main objective of this work is to reduce peak power consumption. Intuitively, reducing peak power consumption goes hand in hand with increasing self-consumption.

Our reward-function aims to formalise this intuition. Following the MDP framework, this objective is translated to reward-function (4).

$$r(x_t, u_t) = \begin{cases} \min(P^c - P_t^{net}, 0) + P_t^{sc} & P_t^{\mathrm{EWH}} \neq 0 \\ 0 & P_t^{\mathrm{EWH}} = 0 \end{cases} \tag{4}$$

Where P^c is 2.5 kW, P_t^{EWH} is the electrical power consumed by the EWH, P_t^{net} is the net power consumption and P_t^{sc} is the self-consumed EWH power, as defined by eq. (6), at quarter t. Eq. (4) is zero when the EWH is turned off. When it is turned on, the reward equals the EWH's power consumption that has been locally produced minus the the building's power consumption that exceeds the 2.5 kW threshold. As such, this reward function gives an incentive to turn the EWH on when local production and free capacity is available. Given P_t^D is the household's other inflexible electrical energy demand and P_t^{PV} is the electrical power produced by the PV installation, the net power consumption P_t^{net} and the EWH self-consumption P_t^{sc} are calculated by equations (5) and (6), respectively. Eq. (5) is the sum of all power consumption minus power production. Eq. (6) defines EWH self-consumption, which is the left of over power production, after inflexible load has been subtracted, consumed by the EWH.

$$P_t^{net} = P_t^{\mathrm{EWH}} + P_t^D - P_t^{PV} \tag{5}$$

$$P_t^{sc} = \min(\max(0, P_t^{PV} - P_t^D), P_t^{\mathrm{EWH}}) \tag{6}$$

An additional challenge considered here is the aim of reducing the need for extensive local PV and demand forecasts. On top of that, the learned control policy will be applied to different residential buildings and households. Therefore, to facilitate policy transfer, the state-space is designed to be independent of environment parameters. For instance, the learned policy should preferably be independent of inverter capacity, as different households will have different PV installations. Preference for policy independence on environment parameters can be illustrated by imagining a simple policy that turns the EWH on whenever PV power production is above 2 kW. When this policy would be transferred to a different household, this absolute number does not apply anymore. Moreover, even within one single household this number would have to change between seasons.

At time-step $t \in \{0, \ldots, 95\}$, state $x_t \in \mathcal{X}$ is defined by (7), with μ_t^T the mean temperature inside the buffer at time-step t, $\Delta_{T_b^t - T_{\min}}$ the difference between the sensor measurement and the minimal allowed water temperature, $t_{\cos}$ and $t_{\sin}$ the projection of the time-step on a circle [11]. Based on other work and to restore the Markov property we have opted to use a history of three time steps for the mean temperature [16, 9].

$$x_t = \{\mu_t^T, \mu_{t-1}^T, \ldots, \mu_{t-3}^T, \Delta_{T_t^b - T_{\min}}, F_{PV}^E, \frac{P_t^{net}}{F_{PV}^P}, t_{\cos}, t_{\sin}, t\} \tag{7}$$

There are two state features dependent on a forecast of the local PV production: F_{PV}^E and F_{PV}^P. They are defined by equations (8) and (9), respectively. F_{PV}^E is the forecast of the current day's energy production, scaled with a rough estimate of the maximally possible energy production given the inverter power P_{inv}. And, F_{PV}^{P} is a forecast of the peak power production of that same day. The agent has no (explicit) information on local power demand. The former forecast provide valuable information to determine if water can best be heated at night or not. When there is barely any production forecasted for the day, it is better to heat at night to avoid power peaks during the day. The latter forecast allows the agent to temporarily interrupt a heating cycle when current net power consumption peaks.

$$F_{PV}^{E} = \sum_{i=\lfloor t/96 \rfloor}^{\lfloor t/96 \rfloor + 96} \frac{E_i^{PV}}{96/2 \cdot \Delta t \cdot P_{\text{inv}}} \tag{8}$$

$$F_{PV}^{P} = \max_{i=\lfloor t/96 \rfloor}^{\lfloor t/96 \rfloor + 96} \left(\frac{E_i^{PV}}{\Delta t} \right) \tag{9}$$

This work considers a binary action-space. Every quarter t, the agent chooses an action $u_t \in \mathcal{U} = \{0, 1\}$, turning the EWH on or off. When temperature constraints would be violated, a backup controller H overrules the agent's action according to Eq. (10), with T_t^b the temperature at the backup sensor location (halfway up the buffer's height), $T_{\text{min}} = 45°C$ and $T_{\text{max}} = 55°C$.

$$u_{\text{phys}} = H(T_t^b, u_t) = \begin{cases} 1 & T_t^b \leq T_{\text{min}} \\ u_t & T_t^b > T_{\text{min}} \text{ and } T_t^b < T_{\text{max}} \\ 0 & T_t^b \geq T_{\text{max}} \end{cases} \tag{10}$$

A two-layer EWH model, similar to earlier work [11] is used as a virtual test-bed, *i.e.*, the agent interacts with the model but does state-transition probabilities are unknown. The model is based on the heat balance equations. A more detailed presentation is given in [11].

2.3. Algorithm

We use PPO [18] to obtain a stochastic policy, maximising expected total reward, given reward-function (4). It is an actor-critic algorithm. The family of actor-critic algorithms has been introduced in effort to combine the advantages of both policy-iteration and value-iteration algorithms. The actor *acts*, *i.e.*, it decides which action to take, and is trained based on a policy-iteration approach. The critic informs the actor about the state value and how it should update its policy in the right direction. The critic is trained with a value-iteration algorithm. Both actor and critic are parametrised using a (separate) Neural Network (NN), with parameters θ_{actor} and θ_{critic}, respectively.

At every iteration k, the actor's objective L_k^{actor} (11) and the critic's objective L_k^{critic} (12) are (approximately) minimised.

$$L_k^{\text{actor}} = \hat{\mathbb{E}}_k \Big[\min\Big(g_k(\theta_{\text{actor}}) \hat{A}_k, \text{clip}(g_k(\theta_{\text{actor}}), 1-\epsilon, 1+\epsilon) \hat{A}_k) \Big) \Big] \tag{11}$$

$$L_k^{\text{critic}} = \hat{\mathbb{E}}_k \left[\left(V_{\theta_k^{\text{critic}}}(x_k) - V_k^{\text{targ}} \right)^2 \right] \tag{12}$$

By the use of these update rules, an estimate of the value-function $V(x)$ and policy $\pi(x)$ is obtained by the critic and actor, respectively. In Eq. (11), $r_k(\theta)$ denotes the probability ratio and is defined by (13) [18]. $\hat{A}_k$ denotes an advantage estimate and is defined by (14) [17], with γ and λ hyper-parameters and δ_t^V given by (15). The clipping in update rule (11) avoids destructively large policy updates [18]. We use a clipping value of $\epsilon = 0.2$, and $\gamma = \lambda = 0.99$. Both the actor and critic use a learning rate of 0.001 and a batch size of 25. During training an entropy bonus is added to the actor's loss, scaled with a factor of 0.01. Finally, every simulated day one actor epoch and five critic epochs are performed.

$$g_k(\theta) = \frac{\pi_\theta(u_k|x_k)}{\pi_{\theta_{\text{old}}}(u_k|x_k)} \tag{13}$$

$$\hat{A}_k = \sum_{l=0}^{\infty}(\gamma\lambda)(\delta_{t+l}^V) \tag{14}$$

$$\delta_t^V = r_t + \gamma V(x_t + 1) - V(x_t) \tag{15}$$

To improve convergence speed and policy transfer capabilities the actor-NN is tailored to the task at hand, *i.e.*, expert knowledge is incorporated into the actor NN's design. The domain knowledge is included in such a way that it does not restrict the applicability to one household or one type of TCL. The actor's NN has been designed based on three main observations:

1. Time-steps close to each other have a similar policy.
2. If the current net consumption is close to the forecasted maximum PV output of the day, its quite likely beneficial to heat water.
3. The backup controller affects the policy and can be incorporated into the actor.

To incorporate the first observation and taking into account earlier experience with FQI, which has shown that explicitly taking into account time-dependence of the policy improves convergence speed [11], the actor is divided in 24 subnetworks, *i.e.*, one for every hour of the day. Each of these networks has 3 layers, of which the first is a feature extraction layer. The subnetwork architecture is shown in Fig. 1. While this architecture is implemented using one NN, inputs are only processed by a subset of the NN, depending on the current time step. Training this network can thus be performed exactly how any other fully connected NN is trained. The output of subnetwork $\mathcal{T} \in \{0, \ldots, 23\}$, with parameters $\theta_{\mathcal{T}}^{\text{actor}}$, equals the probability of choosing action $u_t = 1$.

The second observation has been implemented by means of the first layer's rightmost neuron (as depicted in fig. 1). A ReLu activation function ensures an increase in output probability when the ratio between the current net consumption and the forecasted maximum PV output of the day P_t^s increases. This relationship inverses during the night.

The final neuron of each subnetwork implements Eq. (10), assuring temperature stays within comfort bounds. As such, the third observation has been taken into account in the NN design. The actor thus uses only part of the observable state and, so does the critic. The full observable state is given in (7). The part used by the actor and critic is defined by

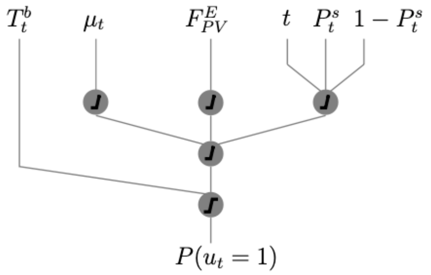


Figure 1: Actor sub-NN architecture, with $P_t^s = P_t^{PV}/F_{PV}^P$.

(16) and (17), respectively. The time-step is not omitted in x_t^{actor}, as it is explicitly used in the actor's policy to determine if it is better to turn the EWH on at night or not.

$$x_t^{\text{actor}} = \{T_t^b, \mu_t^T, F_{PV}^E, \frac{P_t^{net}}{F_{PV}^P}, t\} \tag{16}$$

$$x_t^{\text{critic}} = \{\mu_t^T, \mu_{t-1}^T, \dots, \mu_{t-3}^T, \Delta_{T_t^b - T_{\min}}, F_{PV}^E, \frac{P_t^{net}}{F_{PV}^P}, t_{\cos}, t_{\sin}\} \tag{17}$$

The critic's NN has a conventional fully-connected architecture, with two layers of 28 neurons. Apart from the neuron representing the backup controller, every neuron uses a ReLu activation function. All NNs have been implemented in PyTorch [10].

3. Simulations & Results

3.1. Experiment Set-up

In earlier work we have shown transfer learning increases performance for agents applied in a DR setting [11]. Hence, the task is separated in a pre-training and test phase. The pre-training phase only uses readily available data. Three data streams are needed. First, simulated PV production data is taken from the *ninja* tool developed by Pfenninger and Staffell [15]. Second, electrical load data is generated using *Strobe* [2]. Third, training phase simulations use Domestic Hot Water (DHW) consumption data from Edwards *et al.* [5]. These three sets contain data of one year. For pre-training, the agent interracts with the EWH model for a total 15 simulation-years and trains using the collected state-transition tuples.

After the initial pre-training phase, the obtained parameters θ_{actor} and θ_{critic} are used as initial values for the test phase. During the test-phase, which lasts one simulation-year, we use real residential PV, load and DHW data, from five houses, obtained from a field-test in the Netherlands [4, 22]. In this test-phase, NN parameters (θ_{actor} and θ_{critic}) are further fine-tuned using PPO. Data is available starting from the first of October. Each experiment has been conducted 10 times to account for variability in both phases. Table 1 gives some general metrics of the training- and test data. Clearly, a variety of households has been considered.

Table 1: General metrics of training- and test data, for 1 year.

	DHW cons. [l/day]	$\sum E^D$ [kWh]	$\sum E^{PV}$ [kWh]
Training	188.93	3781.55	3766.76
House 1	57.7	2929.57	7894.09
House 2	116.06	5966.85	8269.54
House 3	33.03	3627.04	7467.20
House 4	29.05	3761.71	8296.90
House 5	185.98	5335.82	7920.31

The presented approach has been compared with three other control approaches: Hysteresis Control (HC), Rule-Based Control (RBC) and a non-expert version of RL (PPO). The hystersis (or, bang-bang) controller assures user comfort and turns the EWH on or off according to Eq. (10). Like RL, RBC adds an additional layer on top of this hystersis controller. The implemented rule-based controller turns the EWH on for four hours, at a fixed start time t_{RBC}. While the choice of t_{RBC} may affect control performance, its optimal value is unknown beforehand. Therefore, all RBC simulations have been run four times, with $t_{\text{RBC}} \in \{10, 11, 12, 13\}$ hour. The non-expert version of RL also uses PPO as a training algorithm. However, the actor has a more traditional fully connected NN with two layers of 10 neurons each. This allows to confirm if the tailor made actor increases performance. Since comfort is not guaranteed through the final neuron anymore, its actions are subject to eq. (10), *i.e.*, it is overruled when needed to ensure comfort.

In the next section we show different result metrics, such as the final energy bill of the considered household, calculated by (3). The Flemish regulator has published capacity tariff values λ_P for different DNOs. We have chosen the average value $\lambda_P = 47.78$ €/kW.

3.2. Results

This part presents the results of the simulations. We start with a visualisation of the three main control approaches (HC, RBC, (expert) RL). Thereafter, we compare their performance differences in a more thorough manner. For the sake of simplicity, only expert RL has been considered at the start. It has been referred to as RL from now on. Only at the final detailed comparison of the main metrics, non-expert RL has been included.

Fig. 2 shows the mean and 95% confidence interval of the total reward captured each pre-training year. Clearly, over time the agent manages to improve its control policy.

Fig. 3 shows several test-phase days for the three considered control approaches. In each subfigure, the grey area depicts net uncontrollable load, *i.e.*, inflexible load P_t^D minus PV production P_t^{PV}. The EWH's power demand P_t^{EWH} for each control approach is depicted by different line-styles. As explained earlier, simulations have been conducted with several values for t_{RBC}. In all subfigures of Fig. 3, results of the t_{RBC} value which resulted in the best final RBC performance for the considered house has been shown.

Fig. 3a shows the three days immediately succeeding pre-training. At first sight, RBC seems to be a good initial control approach, consuming power when local PV production is high. This is, however, as expected, as the choice for RBC is the result of expert domain

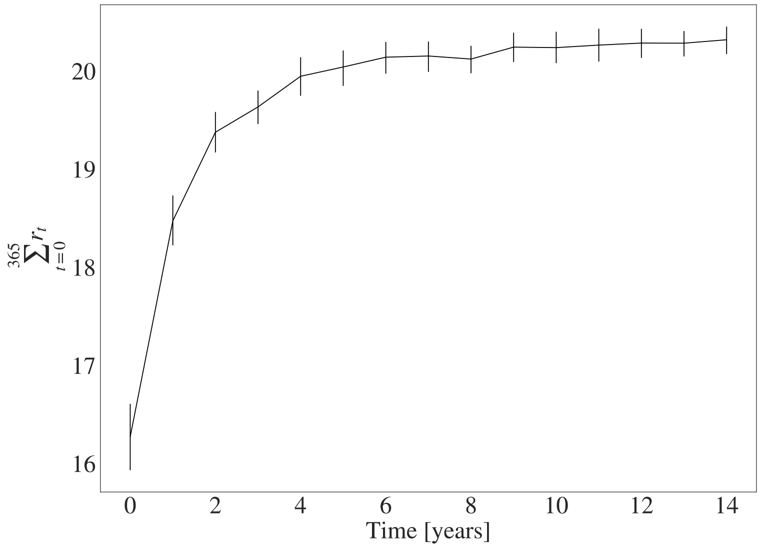


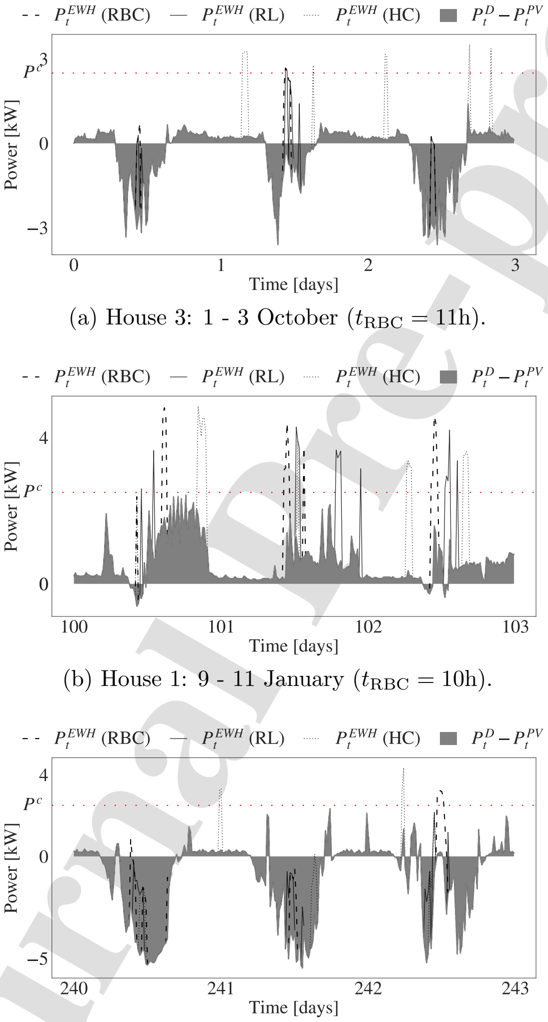
Figure 2: Total reward captured each pre-training year.

knowledge. These three shown days further suggest RL has resulted in similar behaviour as RBC. This confirms the observation that RBC is a rather good initial approach.

Fig. 3b shows three winter days, which illustrate how RL further improves upon RBC. The second and third day, the RL agent turns the EWH on at night, this avoids a heating cycle in the afternoon (during RBC hours), when PV production is low and consumption high. Clearly, HC performs worse than the other approaches, as it does not take into account net consumption whatsoever.

Finally, Fig. 3c presents three spring days. All have quite a lot of PV production and, therefore, clear negative consumption in the afternoon. However, PV production is intermittent and has certain drops during the day. While the RL agent has no forecast of PV production in the next quarter, it has current net consumption as an input. As a result, and contrary to RBC, it temporarily interrupts heating when PV output suddenly drops. This is illustrated by the third day (day 242) of Fig. 3c.

Fig. 4 gives a more extensive overview of the results. It presents an entire test-phase simulation year for house 4. The bottom graph shows each month's total PV energy production $\sum E^{PV}$, total inflexible load $\sum E^{D}$ and average DHW consumption. These measures are useful for interpretation of control performance. Second, the middle graph shows self-consumption ratio of each month. For RBC and RL it shows mean and standard deviation of all simulation runs. The self-consumption ratio is the share of total EWH power consumption which has been locally produced by the PV installation. Intuitively, it is clear that this has to be maximised. The graph shows RL performs slightly better than RBC in this regard, for all months. But, especially in months in which PV production is low, RL manages to capture more of the scarce local renewable energy for own consumption.



(a) House 3: 1 - 3 October ($t_{\text{RBC}} = 11$h).

(b) House 1: 9 - 11 January ($t_{\text{RBC}} = 10$h).

(c) House 2: 29 - 31 May ($t_{\text{RBC}} = 11$h).

Figure 3: Example days of three controllers and three houses, with best choice for t_{RBC}.

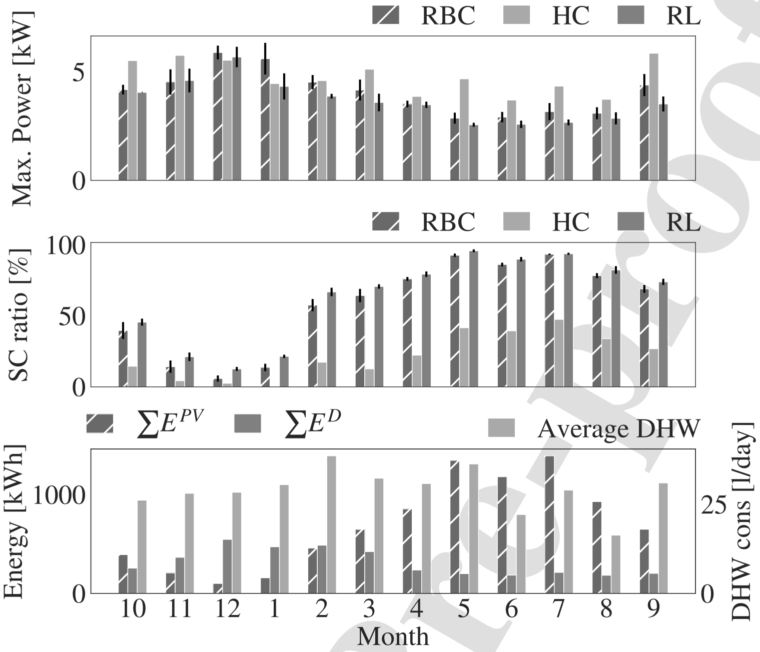


Figure 4: Test-year monthly performance metrics (house 4).

Finally, the top figure shows $P^{\max}$, *i.e.*, the month peak, for each month and for all three control approaches, with mean and standard deviation for RBC and RL. Remembering our final goal, $P^{\max}$ should be minimised. RL outperforms the other control approaches for all months, except November, in this household.

In the end, the main goal is reducing the final yearly energy bills of households in Flanders, prone to a capacity tariff. Fig. 5 shows all interesting metrics for the whole test-phase year and for all houses. The top figure shows the MMP, calculated by (1). (Expert) RL outperforms RBC, HC and non-expert RL for each household. More precisely, on average over all houses, RL reduces the MMP by 16.85 % compared to HC, and by 6.84 % compared to RBC. In absolute numbers, this is a reduction of 0.90 kW and 0.33 kW, respectively.

As in Fig. 4, the middle graph of Fig. 5 shows the EWH's self-consumption ratio. This figure shows that, for each household, RL manages to shift EWH power consumption better to quarters in which local energy production is available. Compared to HC and RBC, expert RL captures 192.34 % and 9.93 % more PV production, respectively. This means that, over the year on average, 495.21 kWh more EWH energy consumption is locally produced, compared to HC, or 57.24 kWh compared to RBC. Of further interest is that expert RL manages to greatly reduce variability of the final performance, compared to non-expert RL.

The bottom bar chart shows the final (electrical) energy bill of each household, defined by (3). (Expert) RL is 14.51 % cheaper than HC and 4.59 % cheaper than RBC. For these households and the considered capacity cost, this thus results in an average reduction in cost

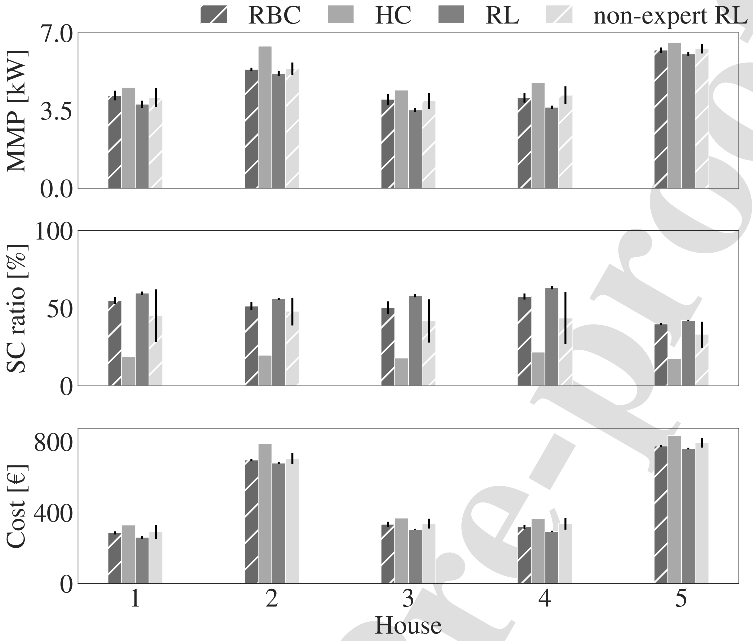


Figure 5: Final (test-phase) results for all 5 houses, with mean and 95% confidence interval over all 10 simulation runs.

of €78.07 and €22.13 compared to HC and RBC, respectively. Moreover, by incorporating domain knowledge into the RL algorithm, we have managed to reduce costs with 6.68 % or €32.96. Additionally, the performance's variability has decreased.

The training-phase is an important step in the design and implementation of this RL set-up for DR. RL is known to be data inefficient [18]. A pre-training-phase mitigates this drawbacks as data is less scarce in this phase. Fig. 3a has illustrated that, because of the pre-training-phase, the RL agent performs its task well immediately at the start of the test-phase. Fig. 6 aims to inspect if final pre-training performance affects test-phase performance. This figure shows that, although pre-training's final year mean reward varies between simulation runs, test-phase mean reward is always rather similar. This result suggests that, while it is important to pre-train the agent, it might not be necessary to somehow find the *best* pre-trained agent. The average Pearson correlation coefficient between the mean reward of the final year of pre-training and the mean test-phase reward is 0.23.

4. Conclusions & Future Work

We adapted state-of-art RL, based on PPO, to the DR setting and have applied it for EWH control in a capacity tariff use case. The considered setting is highly relevant in Belgium, as the (Flemish) regulator has decided to introduce a capacity tariff for residential consumers as of 2022. In this scenario, the goal was identified to be two-fold; reduce the

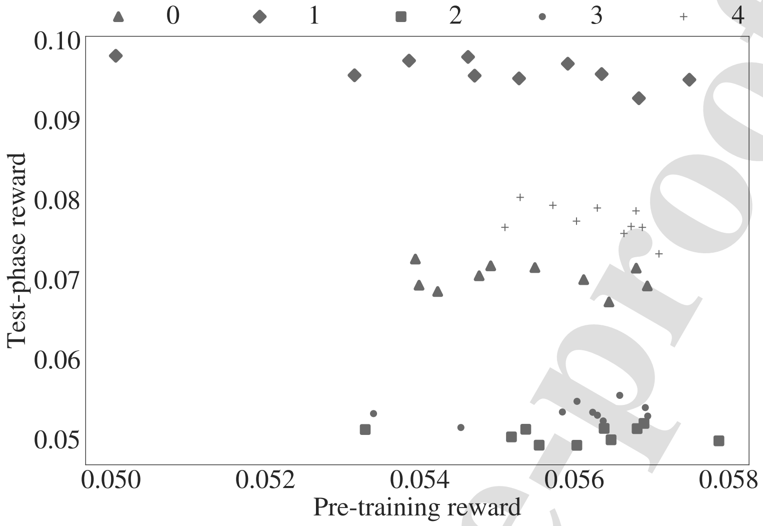


Figure 6: Scatter plot of mean pre-training-phase reward versus mean test-phase reward, for each of the five runs [0, 1, 2, 3, 4].

EWH's peak power consumption and increase self-consumption of local rooftop PV production. In our test-phase, we have used real-life data from five houses, all with different consumption patterns, but equipped with the same EWH. Beforehand, the agent had been trained with readily available data. The setting is particularly challenging as the agent has no forecast of residential load and limited knowledge on future PV production. Furthermore, we proposed an extension of PPO where domain knowledge is incorporated into the actor's design. The results indicate that, using this approach, above RBC performance is achieved.

In our experiments, we compared the RL controller with RBC and HC. The experiments showed, for all considered houses, expert RL outperforms both these two other control approaches. Self-consumption ratio was increased and final energy bill reduced. One should note that, while RBC performs relatively well, it is the result of expert knowledge and the best choice of t_{RBC} differs for each house. Hence, it would need to be tuned for each individual household. In contrary, once deployed, expert RL automatically adapts to its environment. Additionally, our experiments show a low correlation coefficient between pre-training and final mean reward, indicating an approach which is robust against pre-training succes.

Our experiments indicate that data which has been generated/collected for other puposes, such as district level simulations or feasability studies, can be used to improve RL performance. This has the potential to facilitate RL-based control uptake. Furthermore, while model-free learning has proven to be promising in DR settings, it has also shown some drawbacks. Our experiments show that by including expert knowledge into the learning pipeline it is possible to mitigate some of these drawbacks. This is in line with the current observations in physics-informed machine learning research.

We believe the introduction of a capacity tariff is an opportunity for Flemish residential consumers to adopt smart control approaches for EWHs (and other TCLs). Aggregators can potentially provide the necessary technology. Therefore, our future work is directed towards incorporating local control for reducing the MMP within an aggregator framework.

5. Acknowledgments

Thijs Peirelinck, Chris Hermans and Fred Spiessens acknowledge support from the Flemish Institute for Technological Research (VITO) in the preparation of this manuscript.

References

[1] Donald Azuatalam, Wee Lih Lee, Frits de Nijs, and Ariel Liebman. Reinforcement learning for whole-building HVAC control and demand response. *Energy and AI*, 2:100020, nov 2020.

[2] Ruben Baetens, Roel De Coninck, Filip Jorissen, Damien Picard, Lieve Helsen, and Dirk Saelens. Openideas-an open framework for integrated district energy simulations. In *Proceedings of Building Simulation 2015*, pages 347 – 354, 2015.

[3] Shahab Bahrami, Yu Christine Chen, and Vincent W.S. Wong. Deep Reinforcement Learning for Demand Response in Distribution Networks. *IEEE Transactions on Smart Grid*, 12(2):1496–1506, mar 2021.

[4] Oscar De Somer, Ana Soares, Tristan Kuijpers, Koen Vossen, Koen Vanthournout, and Fred Spiessens. Using Reinforcement Learning for Demand Response of Domestic Hot Water Buffers: a Real-Life Demonstration. In *2017 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–7, 2017.

[5] Skai Edwards, Ian Beausoleil-Morrison, and André Laperrière. Representative hot water draw profiles at high temporal resolution for simulating the performance of solar thermal systems. *Solar Energy*, 111:43–52, 2015.

[6] Hussain Kazmi, Fahad Mehmood, Stefan Lodeweyckx, and Johan Driesen. Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems. *Energy*, 144:159–168, 2018.

[7] Vijay R Konda and John N Tsitsiklis. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pages 1008–1014, 1999.

[8] Brida V. Mbuwir, Fred Spiessens, and Geert Deconinck. Distributed optimization for scheduling energy flows in community microgrids. *Electric Power Systems Research*, 187:106479, oct 2020.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. In *NIPS Deep Learning Workshop 2013*, 2013.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury Google, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf Xamla, Edward Yang, Zach Devito, Martin Raison Nabla, Alykhan Tejani, Sasank Chilamkurthy, Qure Ai, Benoit Steiner, Lu Fang Facebook, Junjie Bai Facebook, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, pages 8026–8037. Curran Associates, Inc., 2019.

[11] Thijs Peirelinck, Chris Hermans, Fred Spiessens, and Geert Deconinck. Domain Randomization for Demand Response of an Electric Water Heater. *IEEE Transactions on Smart Grid*, 12(2):1370–1379, may 2020.

[12] Thijs Peirelinck, Hussain Kazmi, Brida V Mbuwir, Chris Hermans, Fred Spiessens, Johan Suykens, and Geert Deconinck. Transfer learning in demand response: A review of algorithms for data-efficient modelling and control. *Energy and AI*, 7:100126, 2022.

[13] Thijs Peirelinck, Frederik Ruelens, and Geert Deconinck. Using reinforcement learning for optimizing heat pump control in a building model in Modelica. In *2018 IEEE International Energy Conference (ENERGYCON)*, pages 1–6. IEEE, 2018.

[14] Thijs Peirelinck, Fred Spiessens, Chris Hermans, and Geert Deconinck. Double Q-learning for Demand Response of an Electric Water Heater. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–5. IEEE, 2019.

[15] Stefan Pfenninger and Iain Staffell. Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, nov 2016.

[16] F Ruelens, B J Claessens, S Vandael, B De Schutter, R Babuška, and R Belmans. Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning. *IEEE Transactions on Smart Grid*, 8(5):2149–2159, sep 2017.

[17] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016*, jun 2016.

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. jul 2017.

[19] Ana Soares, Oscar De Somer, Dominic Ectors, Filip Aben, Jan Goyvaerts, Milo Broekmans, Fred Spiessens, Dennis Van Goch, and Koen Vanthournout. Distributed Optimization Algorithm for Residential Flexibility Activation-Results from a Field Test. *IEEE Transactions on Power Systems*, 34(5):4119–4127, 2019.

[20] Ana Soares, Davy Geysen, Fred Spiessens, Dominic Ectors, Oscar De Somer, and Koen Vanthournout. Using reinforcement learning for maximizing residential self-consumption – Results from a field test. *Energy and Buildings*, 207, 2020.

[21] Richard S Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems (NIPS 1999)*, volume 12, pages 1057–1063. MIT Press, 2000.

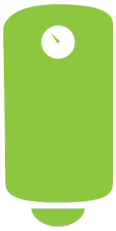
[22] Dennis van Goch, Marc Eulen, Stefan Lodeweyckx, and Chris Caerts. Rennovates, Flexibility Activated Zero Energy Districts, H2020. Technical report, 2017.

[23] José R. Vázquez-Canteli and Zoltán Nagy. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy*, 235:1072–1089, feb 2019.

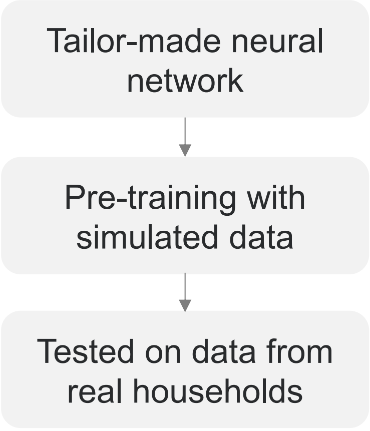
[24] Vlaamse Regulator Energie en Gas (VREG). Toekomst nettarieven – capaciteitstarief — VREG.

Combined Peak Reduction and Self-consumption Using Proximal Policy Optimisation

Expert knowledge and transfer learning can improve control performance of a reinforcement learning agent in a demand response setting



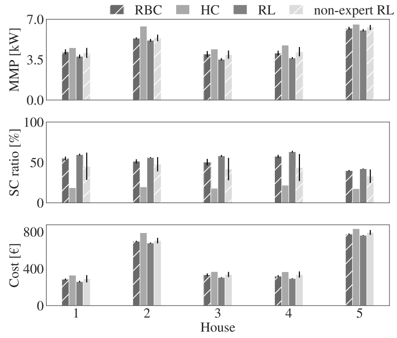
Peak shaving of electric boiler power consumption



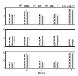
Tailor-made neural network

Pre-training with simulated data

Tested on data from real households



Proposed method outperforms other considered methods



EnergyVille
VITO | KU Leuven

Thijs Peirelinck, EnergyVille, KU Leuven

www.energyville.be

- Successful application of transfer learning in a demand response setting
- Reinforcement learning can be used to reduce peak power consumption of an electric water heater
- Proximal policy optimization with transfer learning can successfully reduce peak power consumption and increase solar self-consumption
- Policy shaping improves the proximal policy optimization performance in a demand response setting

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: