

Article

## Learning Agent for a Heat-Pump Thermostat with a Set-Back Strategy Using Model-Free Reinforcement Learning

Frederik Ruelens <sup>1,2,\*</sup>, Sandro Iacovella <sup>1,2</sup>, Bert J. Claessens <sup>2,3</sup> and Ronnie Belmans <sup>1,2</sup>

<sup>1</sup> Division ELECTA, Department of Electrical Engineering, Faculty of Engineering, KU Leuven, Kasteelpark Arenberg 10, Box 2445, Leuven 3001, Belgium;

E-Mails: Sandro.iacovella@esat.kuleuven.be (S.I.); ronnie.belmans@esat.kuleuven.be (R.B.)

<sup>2</sup> EnergyVille, Thor park 8300, Genk 3600, Belgium; E-Mail: Bert.claessens@vito.be

<sup>3</sup> Energy Department of VITO, Flemish Institute for Technological Research, Boeretang 200, Mol 2400, Belgium

\* Author to whom correspondence should be addressed; E-Mail: frederik.ruelens@esat.kuleuven.be; Tel.: +32-16-32-85-14.

Academic Editor: Neville R. Watson

Received: 2 June 2015 / Accepted: 29 June 2015 / Published: 6 August 2015

---

**Abstract:** The conventional control paradigm for a heat pump with a less efficient auxiliary heating element is to keep its temperature set point constant during the day. This constant temperature set point ensures that the heat pump operates in its more efficient heat-pump mode and minimizes the risk of activating the less efficient auxiliary heating element. As an alternative to a constant set-point strategy, this paper proposes a learning agent for a thermostat with a set-back strategy. This set-back strategy relaxes the set-point temperature during convenient moments, e.g., when the occupants are not at home. Finding an optimal set-back strategy requires solving a sequential decision-making process under uncertainty, which presents two challenges. The first challenge is that for most residential buildings, a description of the thermal characteristics of the building is unavailable and challenging to obtain. The second challenge is that the relevant information on the state, *i.e.*, the building envelope, cannot be measured by the learning agent. In order to overcome these two challenges, our paper proposes an auto-encoder coupled with a batch reinforcement learning technique. The proposed approach is validated for two building types with different thermal characteristics for heating in the winter and cooling in the summer. The simulation results indicate that the proposed learning agent can reduce the energy consumption by 4%–9% during 100 winter days and by 9%–11% during 80 summer days compared to the conventional constant set-point strategy.

**Keywords:** auto-encoder; batch reinforcement learning; heat pump; set-back thermostat

---

## 1. Introduction

Residential and commercial buildings use about 20%–40% of the global energy consumption [1]. Half of this energy is consumed by heating, ventilation and air conditioning (HVAC) systems. About two-thirds of these HVAC systems use fossil fuel sources, such as oil, coal and natural gas. Replacing this large share of fossil-fueled HVAC systems with more energy-efficient heat pumps can play an important role in reducing greenhouse gasses [2–4]. For instance, in [5], Bayer *et al.* report that replacing fossil fuel-based HVAC systems with electric heat pumps can help reduce greenhouse gasses in space heating by 30%–80% in different European countries. The cardinal factors that influence this reduction are the substituted fuel type, the energy efficiency of the heat pump and the electricity generation mix of the country.

This paper focuses on residential heat pumps equipped with an auxiliary heating element. This heating element can be a less efficient electric furnace or a gas- or oil-fired furnace. In its regular operation, a heat pump runs in its more energy-efficient heat-pump mode; however, when the temperature drops too low, both the heat pump and the auxiliary heating element are activated. Since most heat pumps are equipped with an electric auxiliary heating element, which can be four-times less efficient, the U.S. Department of Energy recommends to operate the thermostat with a constant target temperature during the day, even when the inhabitants are not at home [6].

As an alternative to the constant temperature set-point strategy, this paper presents a set-back method, in which the temperature set point is relaxed during convenient times, for example during the night or when the inhabitants are not at home. Such a set-back method can reduce the energy consumption compared to the constant set-point strategy under the condition that it can avoid the auxiliary heating to activate [7].

The remainder of this paper is organized as follows. Section 2 gives an overview of existing literature on heat-pump thermostats and their application to demand response. Section 3 addresses the challenges of developing a successful set-back strategy to reduce the energy consumption of a heat pump. Section 4 formulates the sequential decision-making problem of a thermostat agent as a stochastic Markov decision process. Section 5 proposes an approach based on an auto-encoder and fitted Q-iteration. The simulation results are given in Section 6, and finally, Section 7 summarizes the general conclusion of this work.

## 2. Literature Review

Driven by the potential of heat pumps to reduce greenhouse gasses, heat-pump thermostats have attracted attention from researchers [8,9] and commercial companies [10–14]. A popular control paradigm in the literature on the optimal control of heat-pump thermostats is a model-based approach. Within this paradigm, the first type of model-based controller uses a model predictive control approach [2,15]. At each decision step, the controller defines a control action by solving a fixed-horizon optimization problem, starting from the current time step and using a calibrated model of its environment.

For example, the authors of [9] use a mixed-integer quadratic programming solution to minimize the electricity cost and carbon output of a home heating system. However, the performance of these model-based approaches depends on the quality of the model and the availability of expert knowledge. Model-based approaches can achieve very good results within a reasonable learning period, but typically, they have to be tailored for their application, and they have difficulties with stochastic environments [16]. A second type of model-based controller formulates the control problem as a Markov decision problem and solves the corresponding problem using techniques from approximate dynamic programming [17,18]. For example, in [8], Urieli *et al.* use a linear regression model to fit the model of the building and then apply a tree-search algorithm for finding an intelligent set-back strategy for a heat-pump thermostat. Alternatively, in [19], Morel *et al.* propose an adaptive building controller that makes use of artificial neural networks and dynamic programming. Similarly, the authors of [20] propose a combined neuro-fuzzy model for dynamic and automatic regulation of indoor temperature. They use an artificial neural network to forecast the indoor temperature, which is then used as the input of a fuzzy logic control unit in order to manage energy consumption of the HVAC system. In addition, the authors of [21] report that an artificial neural network-based model can adapt to changing building background conditions, such as the building configuration, without the need for additional intervention by an expert.

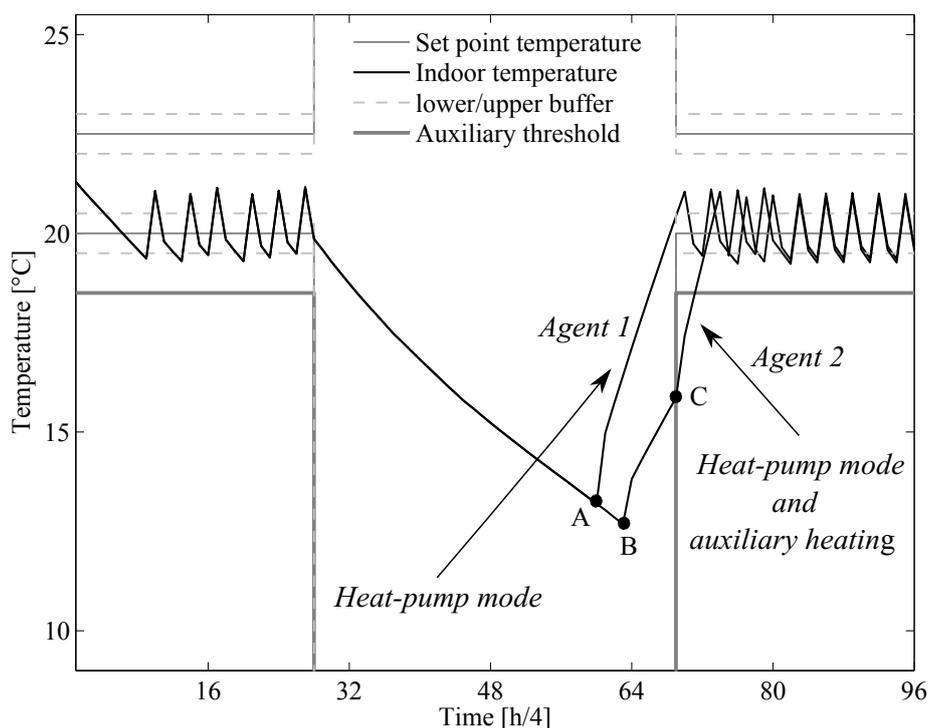
An alternative control paradigm makes use of model-free reinforcement learning techniques in order to avoid the system identification step of model-based controllers. For example, the authors of [22] propose a Q-learning approach to minimize the electricity cost of a thermal storage. In [23], Zheng *et al.* show how a Q-learning approach can be used for residential and commercial buildings by decomposing it over different device clusters. However, a main drawback of classic reinforcement learning algorithms, such as Q-learning and SARSA, is that they discard observations after each interaction with their environment. In contrast, batch reinforcement learning techniques do not require many interactions until convergence to obtain reasonable policies [24–26], since they store and reuse past observations. As a result, they have a shorter learning period, which makes them an attractive technique for real-world applications, such as a heat-pump thermostat. In both [27] and [28], the authors use a batch reinforcement learning technique, fitted Q-iteration, in combination with a market-based multi-agent system, in order to control a cluster of flexible devices, such as electric vehicles and electric water heaters.

This work contributes to the application of batch reinforcement learning to the problem of finding a successful set-back strategy for a heat-pump thermostat. This problem was previously addressed by the work of Urieli *et al.* in [8]. The main difference with their work is that our work proposes a model-free approach that can intrinsically capture the stochastic nature of the problem. The authors build on the existing literature on batch reinforcement learning, in particular fitted Q-iteration [24] and auto-encoders [29].

### 3. Problem Statement

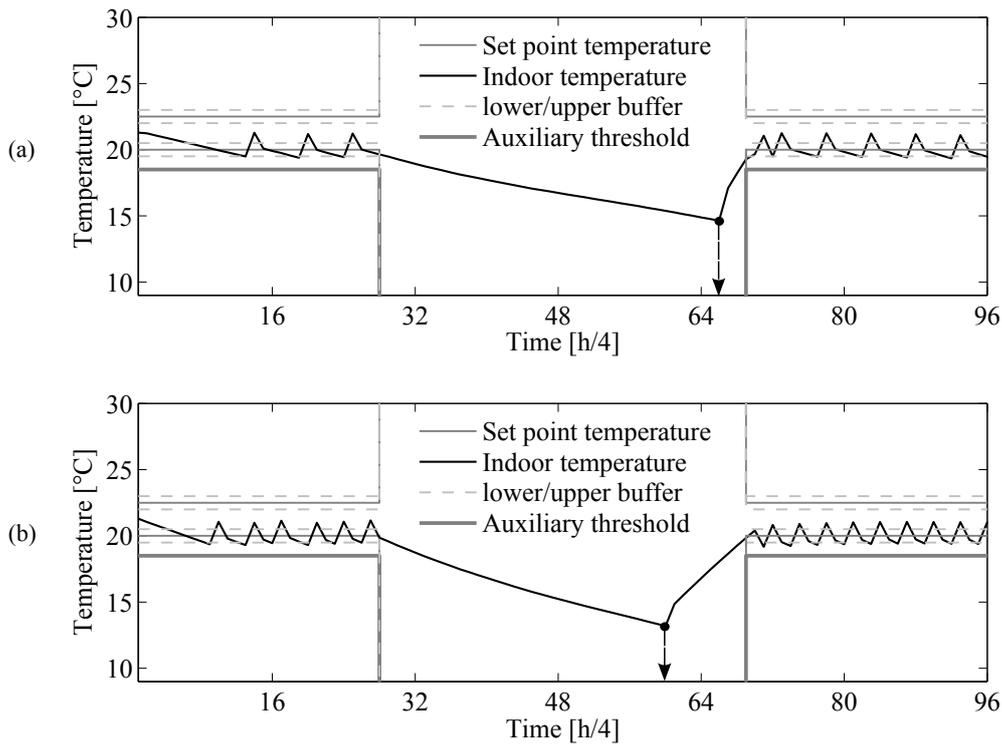
The main objective of this paper is to develop a model-free learning agent for a heat pump with an auxiliary heating element in order to overcome the following two challenges. The first challenge is that the auxiliary heating element activates when the indoor temperature reaches a predefined temperature threshold. The operation of the thermostat is given by Algorithm 2 and can be found in Appendix A.

More information in the temperature settings of the thermostat can be found in Table A2 of Appendix B. In order to illustrate the activation of the auxiliary heating element, two thermostat agents are depicted in Figure 1. Our set-back strategy relaxes the indoor temperature during working hours, *i.e.*, 7–17 h (Figure 1). It can be seen that the first agent correctly anticipates the comfort bounds at 17 h and begins to heat the building in normal heat-pump operating mode (Point A). The second agent postpones heating until Point B and triggers the electric auxiliary heating to switch on at Point C. As a result of the activation of the less efficient auxiliary heating element, the second agent consumes more energy than the recommended constant temperature set-point strategy.



**Figure 1.** Indoor temperature of two thermostat agents with a set-back strategy (7–17 h). Agent 1 operates in normal heat-pump mode, and Agent 2 activates the less efficient auxiliary heating.

A second important challenge when developing an intelligent set-back strategy is that the moment of activating the heat pump does not only depend on the weather conditions, but also on the thermal characteristics of the building. This challenge is illustrated by a second example, where a successful set-back strategy is depicted for two building types. Both building types have identical outside temperatures and inner disturbances. Figure 2a depicts the indoor temperature of a building with a high insulation level, whereas Figure 2b depicts the indoor temperature of a building with a low insulation level. It can be seen that the thermal characteristics of the building can have a significant impact on the operation of the thermostat agent. For instance, the set-back thermostat in Figure 2a can postpone its heating action until Quarter 68, while the set-back thermostat in Figure 2b needs to start heating around Quarter 60 in order to avoid the activation of the auxiliary heating.



**Figure 2.** Indoor temperature of two buildings with different thermal insulation. (a) Building with a high thermal insulation level; (b) building with a low thermal insulation level.

#### 4. Markov Decision Process

Motivated by the challenges presented in Section 3 and driven by recent advances in reinforcement learning [24,30,31], our paper introduces a model-free learning agent. In order to use reinforcement learning techniques, the sequential decision-making problem of a heat-pump thermostat with a set-back strategy is formulated as a stochastic Markov decision process [18,32].

At every decision step  $k$ , the thermostat agent chooses a control action  $u_k \in U \subset \mathbb{R}$ , and the state of its environment  $x_k \in X \subset \mathbb{R}^d$  evolves according to the transition function  $f$ :

$$x_{k+1} = f(x_k, u_k, w_k) \quad \forall k \in \{1, \dots, T - 1\}, \tag{1}$$

with  $w_k$  a realization of a random process drawn from a conditional probability distribution  $p_{w}(\cdot|x_k)$ . After the transition to the next state  $x_{k+1}$ , the agent receives an immediate cost  $c_k$  provided by:

$$c_k = \rho(x_k, u_k, w_k) \quad \forall k \in \{1, \dots, T\}, \tag{2}$$

where  $\rho$  is the cost function. The goal of the thermostat agent is to find a control policy  $h^* : X \rightarrow U$  that minimizes the expected  $T$ -stage return for any state in the state space. The expected  $T$ -stage return  $J_T^{h^*}$  starting from  $x_1$  and following  $h^*$  is defined as follows:

$$J_T^{h^*}(x_1) = \mathbf{E}_{w_k \sim p_w(\cdot|x_k)} \left[ \sum_{k=1}^T \rho(x_k, h^*(x_k), w_k) \right], \tag{3}$$

where  $\mathbf{E}$  denotes the expectation operator. A more convenient way to characterize the policy  $h^*$  is by using a state-action value function or Q-function:

$$Q^{h^*}(x, u) = \mathbf{E}_{w \sim p_W(\cdot|x)} \left[ \rho(x, u, w) + J_T^{h^*}(f(x, h(x), w)) \right]. \quad (4)$$

The Q-value is the cumulative return starting from state  $x$ , taking action  $u$  and following  $h^*$  thereafter. Starting from a Q-function for every state-action pair, the policy is calculated as follows:

$$h^*(x) \in \arg \min_{u \in U} Q^{h^*}(x, u), \quad (5)$$

where  $h^*$  satisfies the Bellman optimality equation [33]:

$$J^{h^*}(x) = \min_{u \in U} \mathbf{E}_{w \sim p_W(\cdot|x)} \left[ \rho(x, h(x), w) + J_T^{h^*}(f(x, h(x), w)) \right]. \quad (6)$$

The central idea behind batch reinforcement learning is to estimate the state-action value function  $Q^{h^*}$  based on a set of past observations (or batch) of the state, control action and reward. Note that this approach does not require a model of the environment  $f$  or the disturbances  $w$ . As a result, no system identification step is needed. The following five paragraphs give a tailored definition of the state, action, cost function and transition function of a heat-pump thermostat agent.

#### 4.1. Observable State

At each time step  $k$ , the thermostat agent can measure the following state information:

$$\mathbf{x}_k = (d, t, T_{\text{in},k}, T_{\text{out},k}, S_k), \quad (7)$$

where  $d \in \{1, \dots, 7\}$  represents the current day in the week and  $t \in \{1, \dots, 96\}$  the current quarter of the hour. The observable state information related to the physical state of the building is given by a measurement of the indoor temperature  $T_{\text{in},k}$ . The observable exogenous state information is defined by  $T_{\text{out},k}$  and  $S_k$ , which are the outdoor temperature and solar irradiance at time step  $k$ . Note that by including the measurements of  $T_{\text{out},k}$  and  $S_k$  at time step  $k$ , our approach captures a first-order correlation of these stochastic variables.

#### 4.2. Thermostat Function

In order to guarantee the comfort of the end-user, the heat pump is equipped with a thermostat mechanism (Algorithm 2). The thermostat logic maps the requested control action  $u_k$  taken in state  $\mathbf{x}_k$  to a physical control action  $u_k^{\text{ph}}$ :

$$u_k^{\text{ph}} = T(\mathbf{x}_k, u_k). \quad (8)$$

As such, the thermostat function  $T$  maps the requested control action to a physical quantity, which is required to calculate the cost value.

### 4.3. Augmented State

As previously stated, this paper assumes that the temperature of the building envelope cannot be measured. It is important to realize that the temperature of the building envelope contains essential information to accurately capture the transient response of the indoor air temperature. Moreover, the temperature of the building envelope represents information on the amount of thermal energy stored in the thermal mass of the building. A possible strategy is to represent the temperature of the building envelope by a handcrafted feature based on expert knowledge, which can be difficult to obtain for residential buildings. However, a more generic strategy is to include past observations of the state and action in the state variable [30,34]:

$$\mathbf{x}_{\text{aug},k} = (d, t, T_{\text{in},k}, \mathbf{z}_k, T_{\text{out},k}, S_k), \quad (9)$$

with:

$$\mathbf{z}_k = (T_{\text{in},k-1}, \dots, T_{\text{in},k-n}, u_{k-1}^{\text{ph}}, \dots, u_{k-n}^{\text{ph}}), \quad (10)$$

where  $n$  denotes the number of past observations of the indoor temperatures and physical actions. Note that the physical control actions have been included in the state, since they give an indication of the amount of energy added to the system. In the next section, a feature extraction technique is proposed to mitigate the “curse of dimensionality” [33] and to find a compact representation of the augmented state vector.

### 4.4. Transition Function

A detailed description of the transition function  $f$  that models the temperature dynamics of the building is given in Appendix B. This paper proposes a model-free approach and makes no assumption of the model type or its parameters.

### 4.5. Cost Function

The cost function  $\rho : X \times U \rightarrow \mathbb{R}$ , associated with a single transition, is given by:

$$c_k = u_k^{\text{ph}} \Delta t + \alpha_k, \quad (11)$$

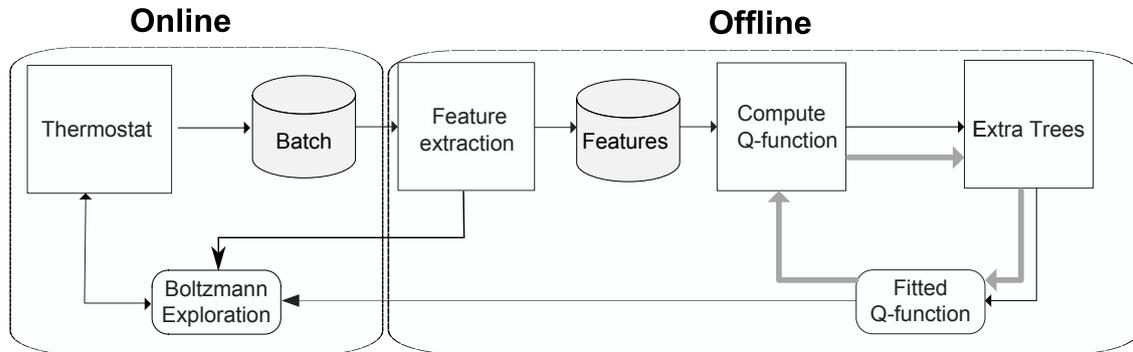
where the parameter  $\alpha_k$  represents a penalty for violating the comfort constraints and  $\Delta t$  represent the time interval of one control period. When the indoor air temperature  $T_{\text{in}}$  is lower than  $\underline{T}_s$  or higher than  $\bar{T}_s$ ,  $\alpha_k$  is set to  $10^5$  and otherwise zero.

## 5. Model-Free Batch Reinforcement Learning Approach

Given full knowledge of the transition function, it can be possible to find an optimal policy by solving the Bellman Equation (6) for every state-action pair using techniques from approximate dynamic programming [17,33]. This paper, however, applies a model-free batch reinforcement learning technique, where the sole information available to solve the problem is the one obtained from daily observations of the following one-step transitions:

$$\mathcal{F} = \{(\mathbf{x}_{\text{aug},l}, u_l, \mathbf{x}'_{\text{aug},l}, u_l^{\text{ph}})\}_{l=1}^{\#\mathcal{F}}, \quad (12)$$

where each tuple is made up of the augmented state  $\mathbf{x}_{\text{aug},l}$ , the control action  $u_l$ , physical control action  $u_l^{\text{ph}}$  and its successor state  $\mathbf{x}'_{\text{aug},l}$ . Figure 3 outlines the building blocks of the model-free batch reinforcement learning method, which consists of two interconnected loops.



**Figure 3.** Online and offline loop of our proposed approach, which consists of a feature extraction (offline), fitted Q-iteration (offline) and Boltzmann exploration (online).

### 5.1. Offline Loop

The offline loop contains a feature extraction technique and a batch reinforcement learning method.

#### 5.1.1. Feature Extraction

This paper proposes a feature extraction technique to find a low dimensional representation of the augmented state  $\mathbf{x}_{\text{aug},k} = (d, t, T_{\text{in},k}, \mathbf{z}_k, T_{\text{out},k}, S_k)$ , by reducing the dimensionality of the state information corresponding to past observations:

$$\mathbf{z}_{e,k} = \Phi(\mathbf{z}_k, W), \tag{13}$$

where  $\Phi : Z \rightarrow Z_e$  is a feature extraction function that maps  $\mathbf{z} \in Z \subset \mathbb{R}^p$  to the encoded state  $\mathbf{z}_e \in Z_e \subset \mathbb{R}^q$ , with  $q < p$  and where  $W$  contains the parameters corresponding to  $\Phi$ .

This work introduces a feature extraction technique based on an auto-encoder. An auto-encoder or auto-associative neural network is an artificial neural network, with the same number of input as output neurons and a smaller number of hidden feature neurons. These hidden feature neurons function as a bottleneck and can be seen as a reduced representation of  $\mathbf{z}_k$ . During training of the auto-encoder, the output data are set to be equal to the input data. The weights of the network are then trained to minimize the squared error between the inputs and its reconstruction. Different training methods to find  $W$  can be found in the literature [35–37]. However, comparing the performance of these training methods is out of the scope of this paper. This work uses a hierarchical training strategy that uses a conjugate gradient descent method [38].

The next paragraph explains how a popular batch reinforcement learning technique, *i.e.*, fitted Q-iteration, can be used, given:

$$\mathcal{F}_{\mathcal{R}} = \{(\hat{\mathbf{x}}_{\text{aug},l}, u_l, \hat{\mathbf{x}}_{\text{aug},l+1}, u_l^{\text{ph}})\}_{l=1}^{\#\mathcal{F}}, \tag{14}$$

with:

$$\hat{\mathbf{x}}_{\text{aug},l} = (d, t, T_{\text{in},l}, \Phi(\mathbf{z}_l, W), T_{\text{out},l}, S_l). \tag{15}$$

where  $\hat{\mathbf{x}}_{\text{aug},l}$  denotes the reduced augmented state, and  $\Phi(\mathbf{z}_l, W)$  denotes the encoded state information of the past observations  $\mathbf{z}_l$ .

### 5.1.2. Fitted Q-Iteration

Although other batch reinforcement learning techniques can be used, this work contributes to the application of fitted Q-iteration [24]. Fitted Q-iteration makes efficient use of gathered data and can be combined with different regression methods. In contrast to standard Q-learning [32], fitted Q-iteration computes the Q-function offline and makes use of the whole batch. An overview of the fitted Q-iteration algorithm is given in Algorithm 1. The algorithm iteratively builds a training set with all state-action pairs in  $\mathcal{F}_{\mathcal{R}}$  as the input. The target values consist of the corresponding cost values and the optimal Q-value, based on the approximation of the previous iteration, for the next state. This works uses an extremely randomized tree ensemble method [39] to find an approximation  $\widehat{Q}$  of the Q-function. The ensemble was set to 60 trees and a minimum of three samples for splitting a node. The number of samples selected at each node was set to the input dimension of the input space. More information on the regression method can be found in [39].

---

#### Algorithm 1 Fitted Q-iteration [24].

---

**Input:**  $\mathcal{F}_{\mathcal{R}} = \{(\hat{\mathbf{x}}_{\text{aug},l}, u_l, \hat{\mathbf{x}}_{\text{aug},l+1}, u_l^{\text{ph}})\}_{l=1}^{\#\mathcal{F}}$

1: Initialize  $\widehat{Q}_0$  to zero

2: **for**  $N = 1, \dots, T$  **do**

3:   **for**  $l = 1, \dots, \#\mathcal{F}$  **do**

4:      $c_l = \rho(\hat{\mathbf{x}}_{\text{aug},l}, u_l^{\text{ph}})$

5:      $Q_{N,l} \leftarrow c_l + \min_{u \in U} \widehat{Q}_{N-1,l}(\hat{\mathbf{x}}_{\text{aug},l+1}, u)$

6:   **end for**

7:   use regression to obtain  $\widehat{Q}_N$  from  $\{(\hat{\mathbf{x}}_{\text{aug},l}, u_l), Q_{N,l}\}, l = 1, \dots, \#\mathcal{F}\}$

8: **end for**

**Output:**  $Q^* = \widehat{Q}_N$

---

### 5.2. Online Loop

A Boltzmann exploration strategy [40] is used at each decision step to find the probability of selecting an action:

$$P(u|\hat{\mathbf{x}}_{\text{aug},k}) = \frac{e^{Q^*(\hat{\mathbf{x}}_{\text{aug},k},u)/\tau_d}}{\sum_{u' \in U} e^{Q^*(\hat{\mathbf{x}}_{\text{aug},k},u')/\tau_d}}, \quad (16)$$

where the parameter  $\tau_d$  controls the amount of exploration and  $Q^*$  is the Q-function obtained with Algorithm 1. The parameter  $\tau_d$  is decreased during the simulation following a harmonic sequence [17]:

$$\tau_d = \frac{1}{(d)^n}, \quad (17)$$

where  $d$  denotes the current day and  $n$  is set to 0.7. Note that, if  $\tau_d = 0$ , then the policy becomes greedy, and the best action is chosen.

## 6. Simulation Results

This section compares the performance of our learning agent to a default constant set-point strategy and a prescient set-back strategy.

### 6.1. Simulation Setup

The simulations use a second-order equivalent thermal parameter model to calculate the indoor air temperature and the envelope temperature of the building [41]. The parameters correspond to a building with a floor area of 200 m<sup>2</sup> and a window-to-floor ratio of 30%. The model equations and parameters are presented in Appendix B. The building is equipped with a heat pump to satisfy the heating or cooling demand of the inhabitants. The heat pump can change its power set point every 15 min with 10 discrete heating or cooling actions. This paper considers two comfort settings, *i.e.*, the default strategy and the set-back strategy. The default strategy has a constant temperature set point of 20.5 °C during the entire day. In contrast, the set-back strategy relaxes the set-point temperature from 7 to 17 h when the inhabitants are not present in the building.

The heat-pump thermostat is equipped with sensors to measure the outside temperature and solar irradiation, which are measurements obtained from a location in Belgium [42]. This work assumes that the heat-pump thermostat is provided with a forecast of the outside temperature and solar irradiation. However, internal heat gains caused by the inhabitants and electrical appliances cannot be measured or forecasted and are obtained from [43].

### 6.2. Learning Agent

The weights of the auto-encoder network are calculated at the beginning of each day and are used to calculate  $\mathcal{F}_R$ . The state information corresponding to the past observations consists of the previous 10 indoor temperatures and the previous 10 control actions. The number of hidden neurons of the auto-encoder network is set to six. Given the batch  $\mathcal{F}_R$ , the fitted Q-iteration algorithm constructs a Q-function for the next day (see Algorithm 1). This Q-function is then used online by the Boltzmann exploration strategy. Note that each simulation run begins with an empty batch of tuples and that observations of the previous day are daily added to  $\mathcal{F}_R$ .

### 6.3. Prescient Method

The prescient set-back strategy assumes that the model parameters are known, and it has prescient knowledge on the outside temperature, solar irradiation and internal heat gains. A detailed description of the prescient method can be found in Appendix C. The outcome of the prescient set-back strategy is used to evaluate the performance of the learning agent and can be seen as an absolute lower-bound.

### 6.4. Simulation Results

The experiments compare the energy consumption and temperature violations of the learning agent with a set-back strategy, conventional constant set-point strategy and prescient set-back strategy.

Note that the conventional constant temperature set point is the recommended strategy by the U.S. Department of Energy [6]. In order to examine the adaptability of the learning agent, an identical learning agent is applied to two building types, with a high and low thermal insulation level. The evaluation is repeated for 100 winter days (heating mode) and 80 summer days (cooling mode). Figure 4 depicts the cumulative energy consumption of the default controller, prescient controller and learning agent. As can be seen in Figure 4, the learning agent is able to reduce the total energy consumption compared to the default strategy for both building types. The simulation results indicate that the learning agent was able to reduce the energy consumption by 4%–9% during the winter and by 7%–11% during the summer. It should be noted, however, that the total energy consumption does not give a complete picture, as it does not consider the temperature violations. Remember that a comfort violation in the heating mode resulted in the activation of the less efficient auxiliary heating element. However, in the cooling mode, no auxiliary cooling is available. For this reason, Figure 5 shows the daily performance metric  $M_d$  and the daily deviation  $D_d$  between the temperature set point and the indoor temperature at 17 hour, which is the end of the set-back period. The daily deviation is calculated as follows:

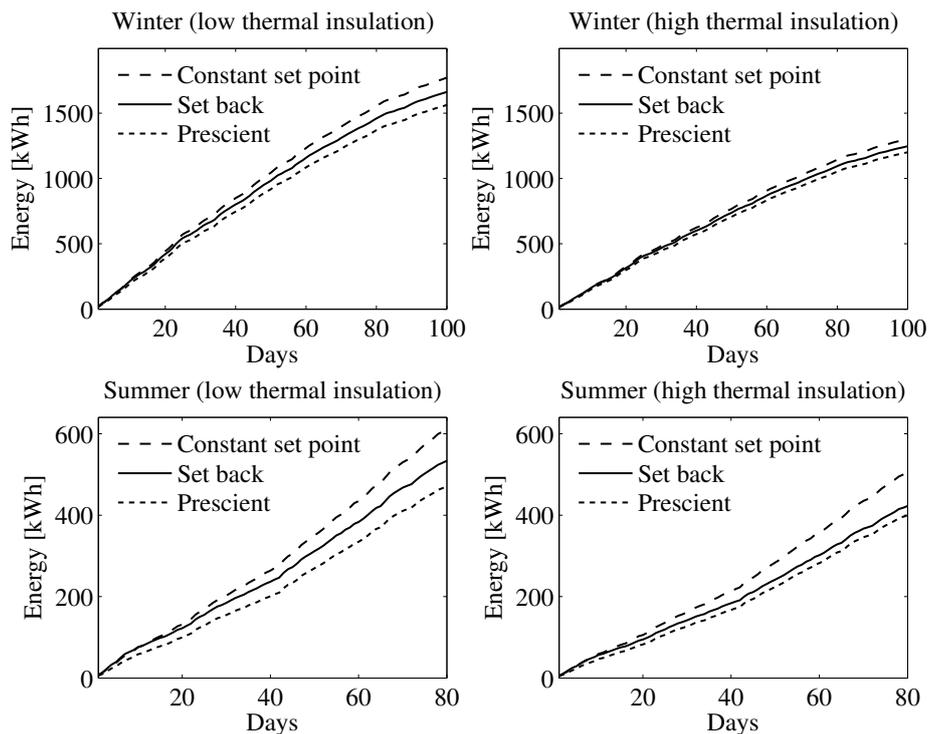
$$D_d = \max(T_{in,17} - \bar{T}_{s,17}, 0) + \max(\underline{T}_{s,17} - T_{in,17}, 0), \quad (18)$$

where  $T_{in,17}$  is the indoor temperature at 17 h, and where  $\underline{T}_{s,17}$  and  $\bar{T}_{s,17}$  are the minimum and maximum temperature set point at 17 h. The daily performance metric  $M_d$  is calculated as follows:

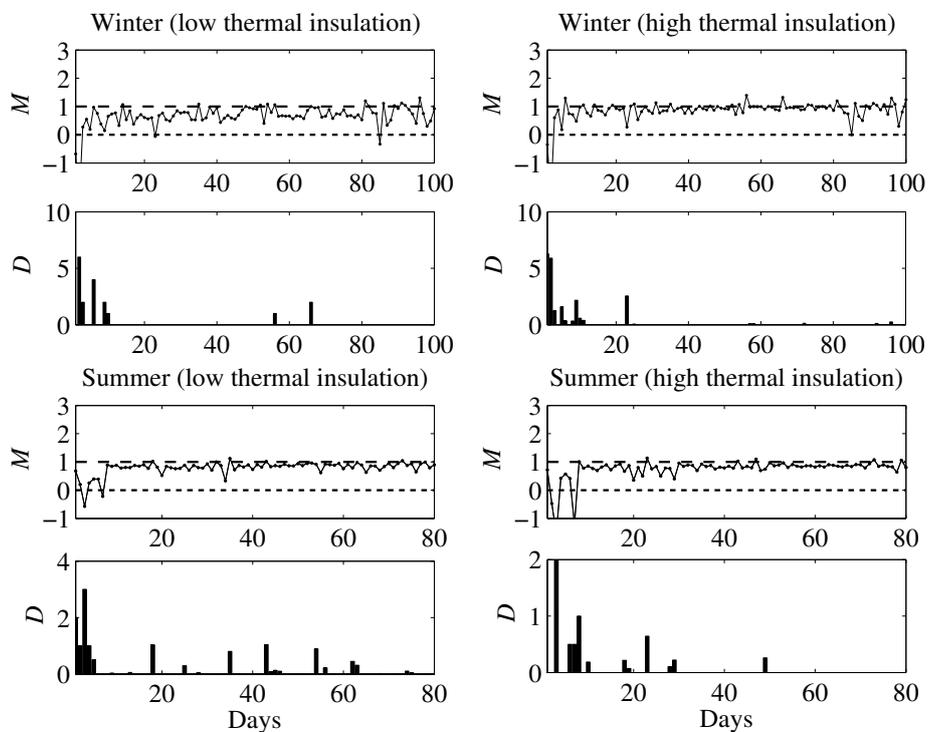
$$M_d = \frac{e_l - e_d}{e_p - e_d}, \quad (19)$$

where  $e_l$  denotes the daily energy consumption of the learning agent,  $e_d$  denotes the the daily energy consumption of the default strategy and  $e_p$  denotes the daily energy consumption of the prescient controller. As such, the metric  $M$  corresponds to zero if the learning agent obtains the same performance as the default strategy and corresponds to one if the learning agent obtains the same performance as the prescient controller. These figures show that the comfort violations decrease over the simulation horizon. At the same time, the performance metric  $M$  increases.

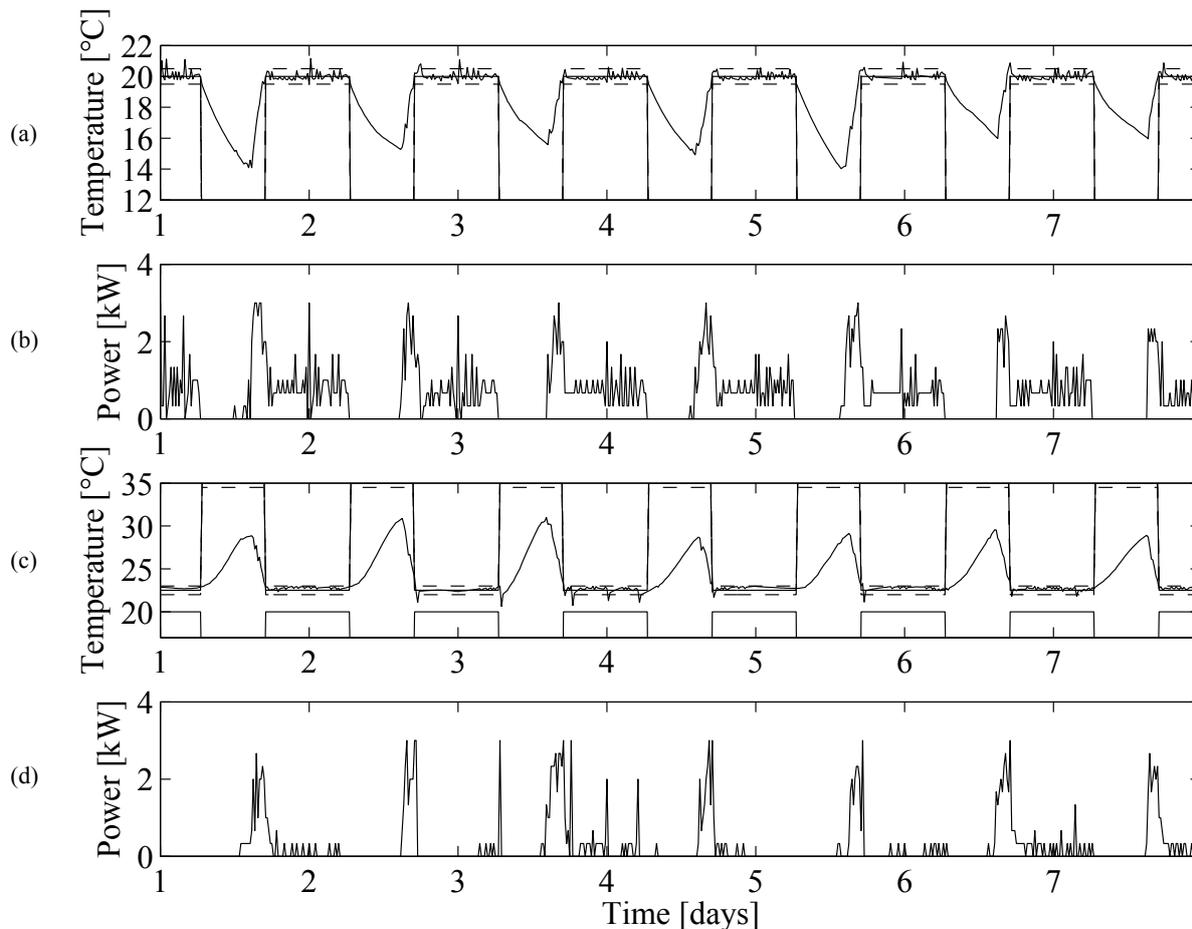
The results obtained with a mature controller (a batch size of 30 days) are depicted in Figure 6. Figure 6a and Figure 6b depict the indoor temperature and power consumption profile during seven winter days. Similarly, Figure 6c and Figure 6d depict the indoor temperature and power consumption during seven summer days. The simulation results indicate that the proposed learning agent can adapt itself to different building types and outside temperatures. In addition, the learning agent with a set-back strategy can reduce the energy consumption of a heat pump compared to the conventional constant temperature set-point strategy.



**Figure 4.** Cumulative energy plot of the default constant set-point strategy, our learning agent with a set-back strategy and a prescient set-back strategy.



**Figure 5.** The performance metric  $M$  and the temperature violations  $D$  for the learning agent with a set-back strategy.



**Figure 6.** Indoor temperature and power consumption of the learning agent with set-back strategy during seven winter (a,b) and summer days (c,d).

## 7. Conclusions and Future Work

This work addressed the challenge of developing a learning agent for a heat pump with a set-back strategy that saves energy compared to a constant temperature set-point strategy, which is recommended by the U.S. Department of Energy. To this end, this paper proposed an approach based on an auto-encoder and a popular model-free batch reinforcement learning technique, *i.e.*, fitted Q-iteration. The auto-encoder is used to reduce the dimensionality of the state vector, which contains past observations of the indoor temperatures and energy consumptions. The performance of the set-back strategy has been evaluated for heating in the winter and cooling in the summer for two building types with different thermal characteristics. An equivalent thermal parameter model has been used to simulate the temperature dynamics of the indoor air temperature and the temperature of the building envelope. During the winter period, the set-back strategy was able to reduce the energy consumption 4%–9% compared to the default strategy. During the summer period, the set-back strategy saved 7%–11% compared to the default strategy. The results indicated that the proposed learning agent can adapt itself to different building types and weather conditions. The proposed learning agent obtained these results without making assumptions on the model or its parameters. As a result, the learning agents can be applied to virtually any building type.

With this work, we intended to show that model-free batch reinforcement learning techniques can provide a valuable alternative to model-based controllers. In our future work, we plan to focus on including real-time electricity prices in the objective and on implementing the presented approach in a lab environment.

## Acknowledgments

Frederik Ruelens has a Ph.D. grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). KU Leuven is a partner of EnergyVille, Thor park, 3600 Genk, Belgium.

## Author Contributions

This paper is part of the doctoral research of Frederik Ruelens, supervised by Ronnie Belmans. All authors have been involved in the preparation of the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## Appendix A: Thermostat Logic

Algorithm 2 illustrates the working of the heat-pump thermostat. The lower and upper temperature set points are given by  $\underline{T}_s$  (20 °C) and  $\bar{T}_s$  (22.5 °C). When the indoor temperature  $T_{in}$  drops below  $\underline{T}_s - T_b^a$  (18.5 °C), the auxiliary heating element is activated in addition to the heat pump until the indoor temperature reaches  $\underline{T}_s + T_b$  (20.5 °C). The activation of the auxiliary heating is independent of the requested control action and can be seen as an overrule mechanism that guarantees the comfort of the end-user. If the indoor temperature  $T_{in}$  is between  $\underline{T}_s + T_b$  and  $\bar{T}_s$ , the thermostat controller follows the requested control action. When the indoor temperature  $T_{in}$  rises above  $\bar{T}_s$ , the cooling mode is activated until  $\bar{T}_s - T_b$  (19.5 °C). In cooling mode, no cooling element is used.

---

### Algorithm 2 Thermostat of a heat pump with auxiliary heating.

---

- 1: Measure  $T_{in}$
  - 2: Initialize  $\underline{T}_s, T_b, T_b^a, \bar{T}_s$
  - 3: **if**  $T_{in} < \underline{T}_s - T_b^a$   
 $u^{ph} = P_h + P_a$  until  $T_{in} \geq \underline{T}_s + T_b$
  - 4: **elseif**  $\underline{T}_s - T_b^a \leq T_{in} \leq \underline{T}_s + T_b$   
 $u^{ph} = P_h$
  - 5: **elseif**  $\underline{T}_s + T_b < T_{in} < \bar{T}_s$   
 $u^{ph} = u$
  - 6: **elseif**  $T_{in} \geq \bar{T}_s$   
 $u^{ph} = P_c$  until  $T_{in} \leq \bar{T}_s - T_b$
  - 7: **end if**
-

## Appendix B: Model Equations

In order to obtain system trajectories of the indoor air temperature, an equivalent thermal parameter model is used to calculate the indoor air temperature  $T_{in}$  and envelope temperature  $T_m$  of a residential building [41,44]:

$$\begin{aligned} \dot{T}_i &= \frac{1}{C_a}(T_m H_m - T_{in}(U_a + H_m) + Q_i + T_{out} U_a) \\ \dot{T}_m &= \frac{1}{C_m}(H_m(T_{in} - T_m) + Q_m), \end{aligned} \quad (A1)$$

where  $H_m$  is the thermal conductance of the building envelope,  $U_a$  is the thermal conductance between air and mass,  $C_a$  is the thermal mass of the air and  $C_m$  is the thermal mass of the building and its contents. The heat added to the interior air mass  $Q_i$  is given by a fraction  $\alpha$  of the internal heat gains  $Q_g$ , a fraction  $\beta$  of the solar heat gains  $Q_s$  and the heat gains generated by the heat pump  $Q_h$ . The heat added to the interior solid mass  $Q_m$  is given by the other fractions of  $Q_s$  and  $Q_g$ :

$$\begin{aligned} Q_i &= \alpha Q_g + \beta Q_s + Q_h \\ Q_m &= (1 - \alpha) Q_g + (1 - \beta) Q_s. \end{aligned} \quad (A2)$$

Tables A1 and A2 give the parameters used in the simulations. The outside temperature  $T_{out}$  and solar irradiance were obtained from a location in Belgium [42]. Although more detailed building models exist in the literature [45], the authors believe that the model used is accurate enough to illustrate the working of the proposed model-free approach and, at the same time, flexible enough in terms of parameters and computational speed.

**Table A1.** Lumped parameters.

Parameters	Low Thermal	High Thermal	Unit
$U_a$	1154	272	W/°C
$H_m$	6863	6863	W/°C
$C_a$	2.441	2.441	MJ/°C
$C_m$	9.896	9.896	MJ/°C

**Table A2.** Heat-pump parameters and thermostat settings.

Parameters	Value
$P_h / P_c$	2500 W
$P_a$	3000 W
$\underline{T}_s$	20 °C
$\bar{T}_s$	22.5 °C
$T_b$	0.5 °C
$T_b^a$	1.5 °C

## Appendix C: Prescient Method

The optimization problem of the prescient method is formulated as follows:

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^T u^{\text{ph}} \\ & \text{subject to} && f(x_k, u_k, w) = x_{k+1}. \\ & && u^{\text{ph}} = T(x_k, u_k), \end{aligned} \quad (\text{A3})$$

where the plant model  $f$  is defined by Equation (A1) and thermostat  $T$  by Algorithm 2. In contrast to our model-free approach, the prescient method knows the plant model and future disturbances. An optimal solution of this optimization problem was found by applying a mixed-integer linear programming solver using Gurobi [46] and YALMIP [47].

## References

1. U.S. Energy Information Administration (EIA). EIA Online Statistics. Available online: <http://www.ia.org/topics/electricity/> (accessed on 21 May 2015)
2. Moretti, E.; Bonamente, E.; Buratti, C.; Cotana, F. Development of Innovative Heating and Cooling Systems Using Renewable Energy Sources for Non-Residential Buildings. *Energies* **2013**, *6*, 5114–5129.
3. Luickx, P.J.; Peeters, L.F.; Helsen, L.M.; D’haeseleer, W.D. Influence of massive heat-pump introduction on the electricity-generation mix and the GHG effect: Belgian case study. *Int. J. Energy Res.* **2008**, *32*, 57–67.
4. Forsén, M.; Boeswarth, R.; Dubuisson, X.; Sandström, B. *Heat Pumps: Technology and Environmental Impact*; 2005. Available online: [http://ec.europa.eu/environment/ecolabel/about\\_ecolabel/reports/hp\\_tech\\_env\\_impact\\_aug2005.pdf](http://ec.europa.eu/environment/ecolabel/about_ecolabel/reports/hp_tech_env_impact_aug2005.pdf) (accessed on 31 July 2015).
5. Bayer, P.; Saner, D.; Bolay, S.; Rybach, L.; Blum, P. Greenhouse gas emission savings of ground source heat pump systems in Europe: A review. *Renew. Sustain. Energy Rev.* **2012**, *16*, 1256–1267.
6. U.S. Department of Energy. Limitations for Homes with Heat Pumps, Electric Resistance Heating, Steam Heat, and Radiant Floor Heating. Available online: <http://energy.gov/energysaver/articles/thermostats> (accessed on 21 March 2015).
7. Michaud, N.; Megdal, L.; Baillargeon, P.; Acocella, C. *Billing Analysis & Environment that “Re-Set” Savings for Programmable Thermostats in New Homes*; 2009. Available online: [http://www.anevaluation.com/pubs/090604\\_econoler\\_iepec\\_paper-lm.pdf](http://www.anevaluation.com/pubs/090604_econoler_iepec_paper-lm.pdf) (accessed on 31 July 2015).
8. Urieli, D.; Stone, P. A Learning Agent for Heat-pump Thermostat Control. In Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), Saint Paul, MN, USA, 6–10 May 2013; pp. 1093–1100.
9. Rogers, A.; Maleki, S.; Ghosh, S.; Nicholas R, J. Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. In Proceedings of the 2nd International Workshop on Agent Technology for Energy Systems (ATES), Taipei, Taiwan, 2–3 May 2011; pp. 71–78.

10. Nest Labs. The Nest Learning Thermostat. Available online: <https://nest.com/> (accessed on 21 March 2015).
11. Honeywell. Programmable Thermostats. Available online: <http://yourhome.honeywell.com/home/products/thermostats/> (accessed on 21 March 2015).
12. BuildingIQ. BuildingIQ, a Leading Energy Management Software Company. Available online: <https://www.buildingiq.com/> (accessed on 21 March 2015).
13. Neurobat. Neurobat Interior Climate Technologies. Available online: <http://www.neurobat.net/de/home/> (accessed on 21 March 2015).
14. Plugwise. Smart thermostat Anna. Available online: <http://www.whoisanna.com/> (accessed on 21 March 2015).
15. Treado, S.; Chen, Y. Saving Building Energy through Advanced Control Strategies. *Energies* **2013**, *6*, 4769–4785.
16. Cigler, J.; Gyalistras, D.; Široký, J.; Tiet, V.; Ferkl, L. Beyond theory: The challenge of implementing Model Predictive Control in buildings. In Proceedings of the 11th REHVA World Congress (CLIMA), Prague, Czech Republic, 16–19 June 2013.
17. Powell, W. *Approximate Dynamic Programming: Solving The Curses of Dimensionality*, 2nd ed.; Wiley-Blackwell: Hoboken, NJ, USA, 2011.
18. Bertsekas, D. *Dynamic Programming and Optimal Control*; Athena Scientific: Belmont, MA, USA, 1995.
19. Morel, N.; Bauer, M.; El-Khoury, M.; Krauss, J. Neurobat, a predictive and adaptive heating control system using artificial neural networks. *Int. J. Solar Energy* **2001**, *21*, 161–201.
20. Collotta, M.; Messineo, A.; Nicolosi, G.; Pau, G. A Dynamic Fuzzy Controller to Meet Thermal Comfort by Using Neural Network Forecasted Parameters as the Input. *Energies* **2014**, *7*, 4727–4756.
21. Moon, J.W.; Chang, J.D.; Kim, S. Determining adaptability performance of artificial neural network-based thermal control logics for envelope conditions in residential buildings. *Energies* **2013**, *6*, 3548–3570.
22. Henze, G.P.; Schoenmann, J. Evaluation of reinforcement learning control for thermal energy storage systems. *HVACR Res.* **2003**, *9*, 259–275.
23. Wen, Z.; O'Neill, D.; Maei, H. Optimal Demand Response Using Device-Based Reinforcement Learning. Available online: <http://web.stanford.edu/class/ee292k/reports/ZhengWen.pdf> (accessed on 27 July 2015).
24. Ernst, D.; Geurts, P.; Wehenkel, L. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.* **2005**, *6*, 503–556.
25. Ernst, D.; Glavic, M.; Capitanescu, F.; Wehenkel, L. Reinforcement learning versus model predictive control: A comparison on a power system problem. *IEEE Trans. Syst. Man Cybern. Syst.* **2009**, *39*, 517–529.
26. Lange, S.; Gabel, T.; Riedmiller, M. Batch Reinforcement Learning. In *Reinforcement Learning: State-of-the-Art*; Wiering, M., van Otterlo, M., Eds.; Springer: New York, NY, USA, 2012; pp. 45–73.

27. Claessens, B.; Vandael, S.; Ruelens, F.; de Craemer, K.; Beusen, B. Peak shaving of a heterogeneous cluster of residential flexibility carriers using reinforcement learning. In Proceedings of the 2nd IEEE Innovative Smart Grid Technologies Conference (ISGT Europe), Lyngby, Danish, 6–9 October 2013; pp. 1–5.
28. Ruelens, F.; Claessens, B.; Vandael, S.; Iacovella, S.; Vingerhoets, P.; Belmans, R. Demand Response of a Heterogeneous Cluster of Electric Water Heaters Using Batch Reinforcement learning. In Proceedings of the 18th IEEE Power System Computation Conference (PSCC), Wroclaw, Poland, 18–22 August 2014; pp. 1–8.
29. Lange, S.; Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. In Proceedings of the IEEE 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
30. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with deep reinforcement learning. Available online: <http://arxiv.org/pdf/1312.5602v1.pdf> (accessed on 27 July 2015).
31. Riedmiller, M.; Gabel, T.; Hafner, R.; Lange, S. Reinforcement learning for robot soccer. *Auton. Robots* **2009**, *27*, 55–73.
32. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
33. Bellman, R. *Dynamic Programming*; Dover Publications, Inc.: New York, NY, USA, 2003.
34. Bertsekas, D.; Tsitsiklis, J. *Neuro-Dynamic Programming*; Athena Scientific: Nashua, NH, USA, 1996.
35. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.
36. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the 1993 IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; pp. 586–591.
37. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bureau Stand.* **1952**, *49*, 409–436.
38. Scholz, M.; Vigário, R. *Nonlinear PCA: A New Hierarchical Approach*; ESANN: Bruges, Belgium, 2002; pp. 439–444.
39. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42.
40. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285.
41. Chassin, D.; Schneider, K.; Gerkenmeyer, C. GridLAB-D: An open-source power systems modeling and simulation environment. In Proceedings of the IEEE Transmission and Distribution Conference and Exposition, Chicago, IL, USA, 21–24 April 2008; pp. 1–5.
42. Crawley, D.B.; Lawrie, L.K.; Winkelmann, F.C.; Buhl, W.F.; Huang, Y.J.; Pedersen, C.O.; Strand, R.K.; Liesen, R.J.; Fisher, D.E.; Witte, M.J.; *et al.* EnergyPlus: Creating a new-generation building energy simulation program. *Energy Build.* **2001**, *33*, 319–331.

43. Dupont, B.; Vingerhoets, P.; Tant, P.; Vanthournout, K.; Cardinaels, W.; de Rybel, T.; Peeters, E.; Belmans, R. LINEAR breakthrough project: Large-scale implementation of smart grid technologies in distribution grids. In Proceedings of the 3rd IEEE PES Innovative Smart Grid Technologies Conference, (ISGT Europe), Berlin, Germany, 14–17 October 2012; pp. 1–8.
44. Sonderegger, R.C. Dynamic Models of House Heating Based on Equivalent Thermal Parameters. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1978.
45. Klein, S.A. *TRNSYS, a Transient System Simulation Program*; Solar Energy Laboratory, University of Wisconsin: Madison, WI, USA, 1979.
46. Gurobi Optimization. Gurobi optimizer reference manual. Available online: <http://www.gurobi.com> (accessed on 21 March 2015).
47. Löfberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proceedings of the IEEE 2004 International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, 2–4 September 2004; pp. 284–289.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).