Experimental analysis of data-driven control for a building heating system

G.T. Costanzo[a], S. Iacovella[b,d], F. Ruelens[b,d], T. Leurs[b], B.J. Claessens[c,d,*]

[a] *Danish Technical University, Department of Electrical Engineering, Frederiksborgvej 399, 4000 Roskilde, Denmark.*
[b] *Electrical Engineering of KU Leuven, Kasteelpark Arenberg 10, bus 2445, 3001 Leuven, Belgium.*
[c] *Flemish Institute for Technological Research (VITO), Boeretang 200, B-2400 Mol, Belgium.*
[d] *EnergyVille, Thor Park Poort Genk 8130, 3600 Genk, Belgium.*

Abstract

Driven by the opportunity to harvest the flexibility related to building climate control for demand response applications, this work presents a data-driven control approach building upon recent advancements in reinforcement learning. More specifically, model-assisted batch reinforcement learning is applied to the setting of building climate control subjected to a dynamic pricing. The underlying sequential decision making problem is cast on a Markov decision problem, after which the control algorithm is detailed. In this work, fitted Q-iteration is used to construct a policy from a batch of experimental tuples. In those regions of the state space where the experimental sample density is low, virtual support tuples are added using an artificial neural network. Finally, the resulting policy is shaped using domain knowledge. The control approach has been evaluated quantitatively using a simulation and qualitatively in a living lab. From the quantitative analysis it has been found that the control approach converges in approximately 20 days to obtain a control policy with a performance within 90% of the mathematical optimum. The experimental analysis confirms that within 10 to 20 days sensible policies are obtained that can be used for different outside temperature regimes.

Keywords:
Thermostatically controlled load, batch reinforcement learning, demand response, data-driven modelling, fitted Q-iteration

Nomenclature

ANN	Artificial Neural Network
BRL	Batch Reinforcement Learning
ELM	Extreme Learning Machine
FQI	Fitted Q-Iteration
HVAC	Heating, Ventilation and Air Conditioning
MABRL	Model-Assisted Batch Reinforcement Learning
MDP	Markov Decision Process
MF	Membership Function
MPC	Model Predictive Control
RL	Reinforcement Learning

1. Introduction

Perez *et al.* estimate that 20 to 40% of the global energy is consumed in buildings [1]. About half of this energy is used for HVAC [2]. As a consequence, control strategies for HVAC have received considerable academic attention in recent years. A popular class of control strategies is that of model-based strategies, such as Model Predictive Control (MPC) [3]. MPC for HVAC systems has been largely investigated in the recent literature [4, 5, 6, 7, 8], in both of its main aspects, modelling [9, 10], and control [11]. In MPC, at regular time intervals, a control action is selected by solving an optimization problem over a finite time horizon, which is typically a day for HVAC control. In MPC the impact of future disturbances, such as internal heating and meteorological conditions, is taken into account using forecasts. Predictive control allows using the load flexibility related to thermal storage, e.g. through the thermal inertia of the building or through direct heat storage [12].

This flexibility can be harvested to enable demand response and provide load control services, which value has been increasing together with the share of renewable energy in the production mix. Examples of services are peak shaving and valley filling for a distribution system operator [13], ancillary services towards a transmission system operator [14, 15] or energy arbitrage [16]. However, deploying MPC can be a challenging task. The most significant challenge is to derive an accurate model which, in the case of thermal control, has to include the thermal dynamics and the actuation model. In [17], Širokỳ *et al.* give a detailed report on implementation issues of MPC controllers for building heating systems.

In this context, completely data-driven approaches are deemed interesting, sacrificing performance for practicality. One possible embodiment uses data-driven model in combination with an optimization algorithm to obtain a control policy

[*] Corresponding author: tel.: +3214335910;
e-mail: { bert.claessens@vito.be}

[18]. Alternatively, it is possible to learn directly the control policy by estimating a state-action value function through interaction with the system. For example in [19], Reinforcement Learning (RL) , a model-free control approach is applied to building thermal storage. In RL, the policy is updated online, i.e. at each time step. In Batch Reinforcement Learning (BRL) , on the other hand, the policy is calculated offline using a batch of historical data. Even though (B)RL is getting more mature [20], as discussed in [21], combining techniques of RL with prior (domain) knowledge is a logical control paradigm. It is towards this direction that this paper is positioned, i.e. in applying BRL in combination with prior knowledge to the operation of a building climate control system for demand response applications.

The basis of our approach is BRL with Fitted Q-Iteration (FQI) [22, 23], where the learning of an optimal control policy is enhanced by *virtual* data coming from a model. For this reason, such approach is called Model-Assisted Batch Reinforcement Learning (MABRL) as discussed in [24].

In Section 2 an overview of the related literature is provided and the contribution of this work is explained. Following the approach presented in [25], in Section 3 the building thermal scheduling is formalised as a sequential decision making problem under uncertainty. In Section 4 MABRL is detailed, while Section 5 presents a quantitative and qualitative assessment of the performance of the controller. Finally, Section 6 outlines the conclusions and discusses future research directions.

2. Related Work

This section gives a non-exhaustive overview of related work regarding MPC and RL for building climate control, after which the main contributions of this work are explained.

2.1. Model Predictive Control

When considering building climate control, MPC has received considerable attention in the recent literature [6, 7, 8, 26]. The overview of practical issues related to the implementation of an MPC controller can be found in [27]. The key elements of an MPC comprise mathematical models describing the building dynamics, comfort requirements and exogenous information such as user behavior and outdoor temperature. This information is used to cast an optimization problem that is solved to define optimal control actions with respect to a defined objective function, subject to constraints provided by the model.

In typical embodiments of MPC one tries to formalize the problem as a mixed integer problem to allow using fast solvers with performance guarantees. Therefore, a Linear Time Invariant model (LTI) of the system under control is to be identified. If no domain knowledge is available, black-box identification techniques are used, such as subspace identification methods [28, 29]. Alternatively, gray-box models can be used, where the model structure is defined and the parameters are estimated using experimental data [9]. In the context of thermal modelling a number of studies use thermal circuits [30, 31, 32, 33, 34].

Advanced climate control allows, besides efficient use of energy and comfort management, integration within aggregation schemes to provide ancillary services and portfolio management in demand side management [35]. For example, in [36] the aggregated flexibility of a cluster of buildings is used to provide balancing services using an aggregate-and-dispatch approach.

An alternative for LTI modelling is to use non-linear data-driven models, such as artificial neural networks (ANNs) [18, 37], in combination with Dynamic Programming (DP) [38] to compute a control policy. This form of control can be seen as a form of RL [39].

2.2. Reinforcement Learning

As discussed in Section 1 RL is a model-free control technique whereby a control policy is learned from interactions with the environment. A well established reinforcement learning method is Q-learning [40] where the state-action value function, or Q-function, is learned. Compared to techniques provided in the previous section, RL mitigates the risk of model-bias [24] as a policy is built around the data. When considering Q-learning and its applications to demand response, mainly traditional Q-learning has been used [41, 42, 19]. More recently BRL [43, 44] in the form FQI [21] has been investigated. The main advantage of BRL is the practical learning time required for convergence (20-40 days in [43, 44]) which comes at the cost of an increased computational complexity. Although BRL can rival the performance of MPC techniques, as indicated in [21], the context of demand response allows to add *prior knowledge* to the optimal control problem that can result in faster convergence. A first approach uses prior knowledge by shaping the policy, obtained with FQI, by means of constrained regression [22]. A second approach is described by Lampe *et al.* in [24]. Here virtual data from a model is used together with experimental data to obtain an approximation of the Q-function (state-action value function).

Building upon [43, 22, 24], this work has the following contributions:

- BRL in the form of FQI [21] in combination with virtual trajectories [24] and policy shaping is applied to a HVAC system for a typical objective of dynamic pricing [45]. This effectively results in a data-driven solution for building climate control systems, combining state of the art BRL with domain knowledge;
- Quantitative and qualitative performance assessment of MABRL in a simulated and experimental environment, where the operation of an air conditioner is subject to dynamic energy pricing.

3. Problem Formulation

Before presenting the control approach in Section 4, this Section formulates the decision-making process as a Markov Decision Process (MDP) [38, 46]. An MDP is defined by its state space X, its action space U, and a transition function f:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \tag{1}$$

which describes the dynamics from $\mathbf{x}_k \in X$ to $\mathbf{x}_{k+1}$, under the control action $\mathbf{u}_k \in U$, and subject to a random process $\mathbf{w}_k \in W$, with probability distribution $p_w(\cdot, \mathbf{x}_k)$. The reward accompanying each state transition is r_k:

$$r_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}) = \rho(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{2}$$

which is here considered to a cost, as it accounts for the energy price. Therefore, the objective is to find a control policy $h : X \rightarrow U$ that minimises the T-stage cost starting from state $\mathbf{x}_1$, denoted by $J^h(\mathbf{x}_1)$:

$$J^h(\mathbf{x}_1) = \mathbb{E}\left(R^h(\mathbf{x}_1, \mathbf{w}_1, ..., \mathbf{w}_T)\right), \tag{3}$$

with:

$$R^h(\mathbf{x}_1, \mathbf{w}_1, ..., \mathbf{w}_T) = \sum_{k=1}^{T} \rho(\mathbf{x}_k, h(\mathbf{x}_k), \mathbf{w}_k). \tag{4}$$

It is worth remarking that an optimal control policy, here denoted by h^*, satisfies the Bellman optimality equation:

$$J^{h^*}(\mathbf{x}) = \min_{\mathbf{u}} \underset{\mathbf{w} \sim P_w(.|\mathbf{x})}{\mathbb{E}} \{\rho(\mathbf{x}, \mathbf{u}, \mathbf{w}) + J^{h^*}(f(\mathbf{x}, \mathbf{u}, \mathbf{w}))\} \tag{5}$$

Typical techniques to find policies in an MDP framework are value iteration, policy iteration, and policy search [22]. As mentioned earlier, in this work MABRL (related to value iteration) is considered.

3.1. State description

Following the approach presented by Ruelens *et al.* in [25], it is assumed that the state space X consists of: time-dependent state information X_t, controllable state information X_{phys}, and exogenous (uncontrollable) state information X_{ex}:

$$X = X_t \times X_{phys} \times X_{ex}. \tag{6}$$

In the following, each component of the state space is detailed.

3.1.1. Timing

The time-dependent information component X_t contains information related to timing. In this implementation the quarter hour during the day has been used:

$$X_t = \{1, \ldots, 96\}, \tag{7}$$

in order to identify behavioral daily patterns. Extending this with, e.g. the day of the week, can be done at little extra cost. However, this extension is outside the scope of this work.

3.1.2. Physical representation

The controllable state information $x_{phys,k}$ consists of the indoor air temperature, T_k:

$$x_{phys,k} = T_k \mid \underline{T}_k < T_k < \overline{T}_k \tag{8}$$

where $\underline{T}_k$ and $\overline{T}_k$ denote the lower and upper bound set by the end consumer.

3.1.3. Exogenous Information

The exogenous (uncontrollable) information $\mathbf{x}_{ex,k}$ is considered to have an impact on $x_{phys,k}$, but it is invariant for control actions $\mathbf{u}_k$. In this study the exogenous state information consists of the outside temperature, $\mathbf{T}_o$, and the solar radiance, $\mathbf{S}$:

$$\mathbf{x}_{ex,k} = (T_{o,k}, S_k)\,. \tag{9}$$

In this work it assumed that a forecast of the outside temperature and the solar radiance is available when constructing the policy h, as will be detailed in Section 4.2 ($\hat{\cdot}$ is used to denotes a forecast).

3.1.4. Control action

In this work the control action is a binary value indicating if the HVAC system should switch ON or OFF:

$$u_k \in \{0, 1\}\,. \tag{10}$$

The control action of the previous control event u_{k-1} is also added to the state information, as it is relevant for the dynamics of the HVAC system. In fact, its value will be used to avoid too frequent switching as discussed in Section 3.2. As a result, the final state vector is defined as:

$$\mathbf{x}_k = (x_{t,k}, T_k, T_{o,k}, S_k, u_{k-1})\,. \tag{11}$$

As this state vector only contains part of the actual state of the system, a common approach to enrich the state vector is to add previous state tuples [47]. This, however, results in an increased state dimension that could be reduced by means of feature extraction, e.g. non-linear principal component analysis [48]. In this work however, the state vector is defined according to (11).

3.2. Backup controller and physical realisation

The HVAC system is assumed to be equipped with a backup controller, which acts as a filter to the control actions resulting from the policy h. The function $B : X \times U \longrightarrow U$ maps the requested control action u_k taken in state x_k to a physical control action u_k^{phys}:

$$u_k^{phys} = B(\mathbf{x}_k, u_k, \theta)\,, \tag{12}$$

with θ containing system specific information. In this case, θ contains $\underline{T}_k$ and $\overline{T}_k$, and $B(\cdot)$ is defined as:

$$B(\mathbf{x}_k, u_k, \theta) = \begin{cases} 1 & \text{if} \quad T_k \leq \underline{T}_k \\ 1 & \text{if} \quad T_k \leq \overline{T}_k \ \wedge \ u_{k-1} = 1 \\ u_k & \text{if} \quad T_k \leq \overline{T}_k \ \wedge \ u_{k-1} = 0 \\ 0 & \text{if} \quad T_k > \overline{T}_k \end{cases}, \tag{13}$$

3.3. Reward model

As discussed in the introduction, different applications can be considered to harvest the flexibility related to climate control of buildings. In dynamic pricing or energy arbitrage [16], i.e. responding to an external price vector λ, the reward function is defined as:

$$\rho\left(\mathbf{x}_k, u_k^{phys}, \lambda_k\right) = -P\Delta t \lambda_k u_k^{phys}\,, \tag{14}$$

with P is the average power consumption of the air conditioner during the time interval Δt.

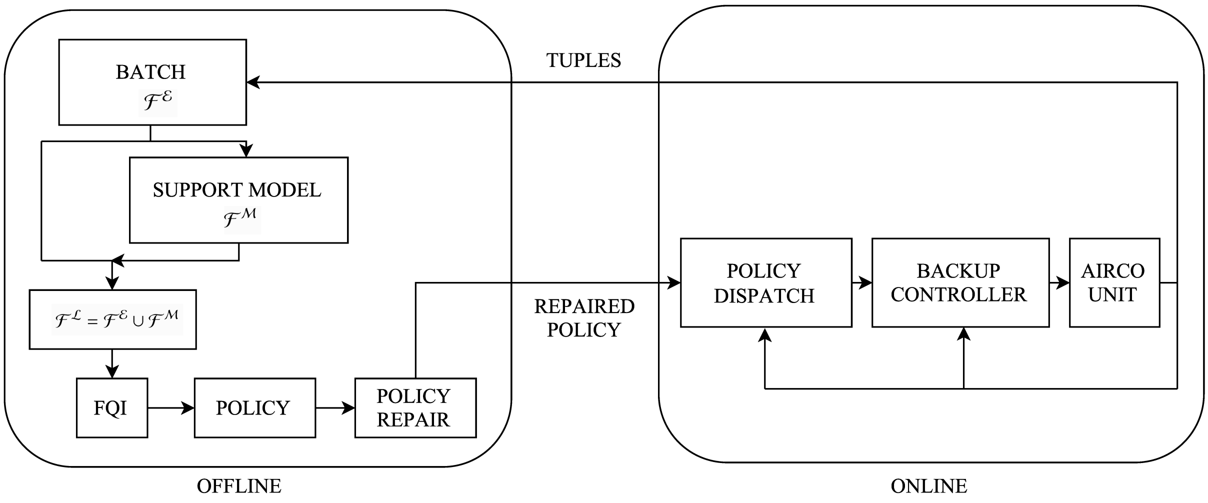
TUPLES
BATCH
$\mathcal{F}^{\mathcal{E}}$
SUPPORT MODEL
$\mathcal{F}^{\mathcal{M}}$
$\mathcal{F}^{\mathcal{L}} = \mathcal{F}^{\mathcal{E}} \cup \mathcal{F}^{\mathcal{M}}$
FQI
POLICY
POLICY REPAIR
REPAIRED POLICY
POLICY DISPATCH
BACKUP CONTROLLER
AIRCO UNIT
OFFLINE
ONLINE

Figure 1: Overview of the information flow in this implementation of MABRL using FQI and policy shaping.

4. Model-Assisted Batch Reinforcement Learning

As discussed in Section 2.2, the control policy h^* is obtained using MABRL in combination with policy shaping, as illustrated in Figure 1. To this end, FQI is used to obtain an approximation $\widehat{Q}^*$ of the state-action value function Q^* from a batch of four tuples $\mathcal{F}^{\mathcal{L}}$, as detailed in [49]:

$$\mathcal{F}^{\mathcal{L}} = \left\{(\mathbf{x}_l, u_l, r_l, \mathbf{x}'_l),\ l = 1, ..., \#\mathcal{F}^{\mathcal{L}}\right\}, \tag{15}$$

where $\mathbf{x}'_l$ denotes the successive state in time to $\mathbf{x}_l$.

As illustrated in Figure 1, $\mathcal{F}^{\mathcal{L}}$ is a combination of experimentally observed tuples $\mathcal{F}^{\mathcal{E}}$ and virtual tuples generated by a model $\mathcal{F}^{\mathcal{M}}$:

$$\mathcal{F}^{\mathcal{L}} = \mathcal{F}^{\mathcal{E}} \cup \mathcal{F}^{\mathcal{M}}\ . \tag{16}$$

From the resulting $\widehat{Q}^*(x, u)$ a control action u_k can be obtained following

$$u_k \in \arg\min_u \widehat{Q}^*(\mathbf{x}_k, u). \tag{17}$$

The following subsections detail the algorithms behind each block in Figure 1.

4.1. The support model: Artificial Neural Network

Following the approach presented in [24], an ANN is used to represent a support model. In this work, single-layer, single-output Extreme Learning Machines (ELMs) are trained to predict the change of internal temperature, ΔT. These are used as they allow for fast training of the weights of the network at the expense of reduced regression performance. The latter is partially mitigated by combining multiple ELMs in an ensemble [50].

The output of an ELM with p input neurons, n hidden neurons and one output node can be formulated as:

$$y(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g(\mathbf{w}_i \cdot \mathbf{x},\ b_i) = G(\mathbf{x})\boldsymbol{\beta}\ , \tag{18}$$

where $\mathbf{x} \in \mathbb{R}^p$ is the input vector, $\mathbf{w}_i \in \mathbb{R}^p \sim i.i.d.\, U(-1, 1)$ is the weight vector connecting the input nodes with the i-th hidden node, $b_i \in \mathbb{R} \sim U(0, 1)$ is the bias of the i-th hidden node, and $\beta_i \in \mathbb{R}$ is the output weight of i-th hidden node. $\boldsymbol{\beta} = [\beta_1 \dots \beta_n]^T$, $\boldsymbol{\beta} \in \mathbb{R}^{n\times1}$, is the vector of the output weights between the hidden neurons and the output node. $G(\mathbf{x}) = [g(\mathbf{w}_1 \cdot \mathbf{x},\ b_1) \dots g(\mathbf{w}_n \cdot \mathbf{x},\ b_n)] \in \mathbb{R}^{1\times n}$ is the output vector of the hidden layer, where the nodes activation function g is a sigmoid. As the parameters of the hidden nodes (bias and input weight) are randomly generated, training an ELM corresponds to determining the output weight matrix $\boldsymbol{\beta}$ based on the least-squares solution of $\mathbf{G}\boldsymbol{\beta} = \mathbf{Y}$:

$$\boldsymbol{\beta} = \left(\frac{1}{C} + \mathbf{G}^T\mathbf{G}\right)^{-1} \mathbf{G}^T\mathbf{Y}\ . \tag{19}$$

In Eq. 19 the term $\mathbf{Y} \in \mathbb{R}^{mx1}$ contains the target values from the train set: $\mathbf{Y} = [y_1, y_2, \dots, y_m]^T$, where m is the size of the train set; $\mathbf{G} \in \mathbb{R}^{m\times n}$ is the network output matrix, constructed as:

$$\mathbf{G} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1,\ b_1) & ... & g(\mathbf{w}_i \cdot \mathbf{x}_1,\ b_i) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_j,\ b_1) & \cdots & g(\mathbf{w}_i \cdot \mathbf{x}_j,\ b_i) \end{bmatrix}, \quad \begin{matrix} i \in \{1, \dots, n\} \\ j \in \{1, \dots, m\} \end{matrix} \tag{20}$$

while the regularisation term $C = 100$ is used to enhance the robustness and the generalisation of the solution [51].

This study uses an ensemble of 40 single-output ELMs, the ensemble model output $y_{ens}(\mathbf{x})$ is given by a weighted average of the L individual ELM outputs:

$$y_{ANN}(\mathbf{x}) = \frac{1}{L}\sum_{k=1}^{L} y_i(\mathbf{x})\ , \tag{21}$$

The input vector at time k is $\mathbf{x}_k = (x_{t,k}, T_k, T_{o,k}, S_k, u_{k-1})$, and the output is $y_k = \Delta T_k = T_{k+1} - T_k$. A detailed description of ELMs can be found in [52]. As the training process is fast, finding the appropriate number of hidden nodes is done using

Algorithm 1 Model-assisted Fitted Q-iteration using a forecast of the exogenous data

```
Input: $\mathcal{F}^{\mathcal{E}} = \left\{\mathbf{x}_l, u_l, \mathbf{x}'_l, u_l^{phys}\right\}_{l=1}^{\#\mathcal{F}^{\mathcal{E}}}$ , $\hat{X}_{ex} = \{\hat{\mathbf{x}}_{\mathrm{ex},k}\}_{k=1}^{T}$ ,
    $y_{ANN}$, $r$, $n, C\ \lambda$, $\theta$
 1: $\mathcal{F}^{\mathcal{M}}$ = generateTuples($\mathcal{F}^{\mathcal{E}}, y_{ANN}, r, n, C, \theta$)
 2: $\mathcal{F}^{\mathcal{L}} = \mathcal{F}^{\mathcal{E}} \cup \mathcal{F}^{\mathcal{M}}$
 3: let $\widehat{Q}_0$ be zero everywhere on $X \times U$
 4: for $N = 1, \ldots, T$ do
 5:     for $l = 1, \ldots, \#\mathcal{F}^{\mathcal{L}}$ do
 6:         $r_l \leftarrow \rho\left(\mathbf{x}_l, u_l^{phys}, \lambda\right)$
 7:         $\hat{\mathbf{x}}'_l \leftarrow \left(x'_{t,l}, T'_l, \hat{T}'_{o,l}, \hat{S}'_l, u'_{l-1}\right)$
 8:         $Q_{N,l} \leftarrow r_l + \min_{u \in U} \widehat{Q}_{N-1}(\hat{\mathbf{x}}'_l, u)$
 9:     end for
10:     use regression to obtain $\widehat{Q}_N$ from
        $\mathcal{T}_{\mathrm{reg}} = \left\{((\mathbf{x}_l, u_l), Q_{N,l}), l = 1, \ldots, \#\mathcal{F}^{\mathcal{L}}\right\}$
11: end for
12: return $\widehat{Q}^* = \widehat{Q}_N$
```

cross-validation. In the experimental trials, further detailed in Sec. 5.3, the ELMs regularization term is fixed to $C = 100$, and the ensemble of ELMs consists of 40 networks.

4.2. Fitted Q-Iteration

A popular BRL technique that found its way into several practical implementations is FQI [49]. Typically, BRL techniques construct policies based on a batch of tuples. However, since in this context the reward function is known a priori and the resulting actions of the backup controller can be measured, Algorithm 1 uses tuples of the form $\left(\mathbf{x}_l, u_l, \mathbf{x}'_l, u_l^{phys}\right)$. Algorithm 1 shows how FQI [49] can be used in a demand response application when a forecast of the exogenous data is available. Here $\hat{\mathbf{x}}'_l$ denotes the successor state to $\mathbf{x}_l$. In Algorithm 1, the observed external temperature and solar radiance in $\mathbf{x}'_{ex,l}$ are replaced by their forecasted value $\hat{\mathbf{x}}'_{ex,l}$ (line 7 in Algorithm 1). As such, $\widehat{Q}^*$ becomes biased towards the provided forecast. Before constructing $\widehat{Q}^*$ a set $\mathcal{F}^{\mathcal{M}}$, containing at most n virtual tuples is created with random state-action pairs. However, a randomly generated tuple is only accepted to $\mathcal{F}^{\mathcal{M}}$ if the nearest experimental tuple in $\mathcal{F}^{\mathcal{E}}$ falls outside a predefined radius r, following a distance metric Δ.[1] In order to keep Algorithm 2 tractable in the event of a dense set $\mathcal{F}^{\mathcal{E}}$, a computational budget H is added.

The distance metric is defined as: $\Delta\left((\mathbf{x}, \mathbf{x}'), (u, u')\right) = ||\mathbf{x} - \mathbf{x}'|| + ||u - u'||$, where $||\cdot||$ is the Euclidean norm. Algorithm 2 uses the artificial network y_{ANN} to generate the virtual tuples as in (21)

Similarly as in [49], Algorithm 1 uses an ensemble of extremely randomized trees [53] as regression algorithm to estimate the Q-function.

In principle, other regression algorithms, such as artificial neural networks or support vector machines can be applied.

Algorithm 2 generateTuples($\mathcal{F}^{\mathcal{E}}, y_{ANN}, r, n, \mathrm{H}, \theta$)

```
Input: $\mathcal{F}^{\mathcal{E}}$, $y_{ANN}$, $r$, $n$, $C, \theta$
$\mathcal{F}^{\mathcal{M}} = \{\emptyset\}$
$N \leftarrow 0$
while $\#\mathcal{F}^{\mathcal{M}} < n$ and $N < \mathrm{H}$ do
    generate random state action sample $\{\mathbf{x}_k, u_k\}$
    $d = \min_{(\mathbf{x}, u) \in \mathcal{F}^{\mathcal{E}}} \{||\mathbf{x} - \mathbf{x}_k|| + ||u - u_k||\}$
    if $d > r$ then
        $u_k^{phys} = B(\mathbf{x}_k, u_k, \theta)$
        $T'_{l,k} = y_{ANN}(\mathbf{x}_k, u_k^{phys}) + T_{l,k}$
        $\mathcal{F}^{\mathcal{M}} = \mathcal{F}^{\mathcal{M}} \cup \left\{(\mathbf{x}_k, u_k, \mathbf{x}'_k, u_k^{phys})\right\}$
    end if
    $N \leftarrow N + 1$
end while
return $\mathcal{F}^{\mathcal{M}}$
```

4.3. Policy Shaping

This section shows how to shape a policy h^* by using triangular Membership Functions (MFs) [22] and expert knowledge to enforce monotonicity in the policy.

The centers of the triangular MFs are located on an equidistant grid with N_g MFs along each dimension of the state space. This partitioning leads to ($N_g^{|X_{phys}|}$) state-dependent MFs for each action.

The parameter vector θ_g^* that approximates the original policy can be found by solving the following least-squares problem:

$$\theta_g^* \in \arg\min_{\theta_g} \sum_{l=1}^{\#\mathcal{F}^{\mathcal{L}}} \left([F(\theta_g)](x_l) - h(x_l)\right)^2, \quad \text{s.t. expert knowledge} \tag{22}$$

where F denotes an approximation mapping of a weighted linear combination of triangular MFs and $[F(\theta_g)](x)$ denotes the policy $F(\theta_g)$ evaluated at state x. A more detailed description of how these triangular MFs are defined can be found in [22].

4.4. Policy dispatch

The policy dispatch block is in charge of operating the air conditioner according to the policy coming from the MABRL controller. The air conditioners controller selects each action in any encountered state with nonzero probability (exploration), or dispatches the optimal control action from the policy by exploiting the acquired knowledge (exploitation) [40]. A common technique to balance the exploration-exploitation in RL is ε-greedy exploration:

$$u_k = \begin{cases} u \sim \text{Bernoulli}\,(0.5) & \text{if } \gamma \leq \varepsilon_j \\ [F(\theta_g^*)](\mathbf{x}_k) & \text{if } \gamma > \varepsilon_j \end{cases}, \tag{23}$$

where $\gamma \sim U\,(0, 1)$.

The following section presents the performance assessment of MABRL in simulated and real environment, where the exploration factor ε_j is designed to reduce by half every four days:

$$\varepsilon_j = \frac{\varepsilon_0 \varsigma}{\varsigma + j - 1}\ . \tag{24}$$

[1] Note that this radius can also be defined based upon local inter-tuple distances. This is however to be explored in future work.

In Eq. 24 the decay factor ς is 4, the initial probability ε_0 is 0.4, and the day index is $j \in \{1, 2, 3, \ldots\}$.

5. Performance assessment

This section provides a qualitative and quantitative performance assessment of the controller discussed in Section 4.2. Section 5.2 presents a quantitative assessment in a setting where the online part in Figure 1 is simulated via an equivalent thermal parameter model [54] of the air conditioner and the building (further detailed in Appendix A). Section 5.3 provides a qualitative analysis of MABRL performance in a real climate control application.

5.1. Policy representation

In the following paragraphs, a graphical representation of the control policy is used to study the impact of integrating forecasts, applying policy shaping and adding virtual tuples. In order to make a two-dimensional visualization, each control policy is depicted for the forecasted outside temperature and solar radiation. As such, each two-dimensional mapping in Figure 4 maps the current quarter in the day and indoor temperature to a binary control action. Depending on the value of the binary control action, switching the HVAC system ON is represented by black areas and switching the HVAC system OFF by white areas. Note, the original control policy is denoted by $h(x)$ and the shaped policy is denoted by $[F(\theta_g^x)](x)$. Especially for the experimental results, analyzing the policy is relevant as it gives offline insight in what the algorithm has learned and to evaluate the contribution of different features in the control algorithm without having to copy exactly the same measurement conditions. For example evaluating the effects of policy shaping, adding virtual tuples and the sensitivity of the policy to different forecasts of external conditions.

5.2. Simulated environment

In order to test the convergence and the performance of the MABRL controller, a benchmark is required. As the experimental setup is a living lab, exact external conditions cannot be reproduced from day to day. Thus, the model described in Appendix A has been used to simulate the air conditioner and the lab room. An optimal solution to thermal scheduling obtained using MPC is taken as benchmark and is depicted in Figure 2. This benchmark solution is explained in more detail in Appendix B.

The simulations have been performed using different exogenous information and price profiles from day to day [55]. In Figure 2 the top graph shows the cumulated cost of the different controllers: BRL, MABRL, Optimal (MPC), and default thermostatic control.

These results indicate that indeed the control approach as presented in Section 4 is able to find near optimal control policies in a learning time of approximately 20 days, after which the performance relative to a mathematical optimum is stable. However, adding virtual tuples, as illustrated in the procedure in Section 4.2, has limited contribution, with a slight economic advantage of the MABRL approach over the BRL. The following subsection presents the policy computation on the basis of experimental data, together with its experimental validation in a living lab.

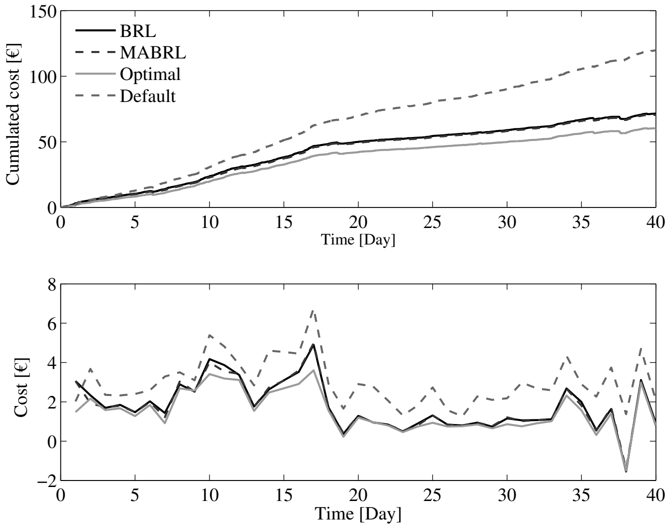

Figure 2: Cost performance of the different controllers: BRL (no virtual tuples), MABRL, Optimal (MPC), and Default (hysteresis). Top plot: cumulative electricity cost. Bottom plot: daily economic performance.

5.3. Experimental environment

The setup consists of two air conditioners (Fig. 3), one temperature sensor to measure the room temperature in one point, one pyranometer to measure the solar radiation on the roof, one temperature sensor to measure the external air temperature, and one power meter to measure the air conditioners power consumption. Data points are collected every 5 minutes.

External forecasts are provided three times a day with a granularity of 15 minutes and a prediction horizon of 8 hours. The policy is recomputed three times a day, as new forecasts arrive, and dispatched on 5 minutes basis.

5.3.1. Ability to integrate forecasts

A first analysis focuses on the ability of the control approach to effectively take into account the forecast of exogenous information. To this end, Figure 4 shows policies organized in a matrix for different forecasts of the external temperature, where each column corresponds to the same forecasted external temperature and where the numbers of experimental tuples are increased from top to bottom. No virtual tuples have been added to the batch. Starting from the top row, policies are computed using experimental batches of: two days, eight days, and sixteen days. Note that, recalling Eq. 13, once the heating system has been switched on it continues heating until the temperature upper bound is reached in order to avoid frequent switch events. When considering the first row, with only two days worth of data in the batch, the policies obtained are near invariant for the temperature forecasts used. This is attributed to the fact that the variation in the outside temperatures observed is limited. The

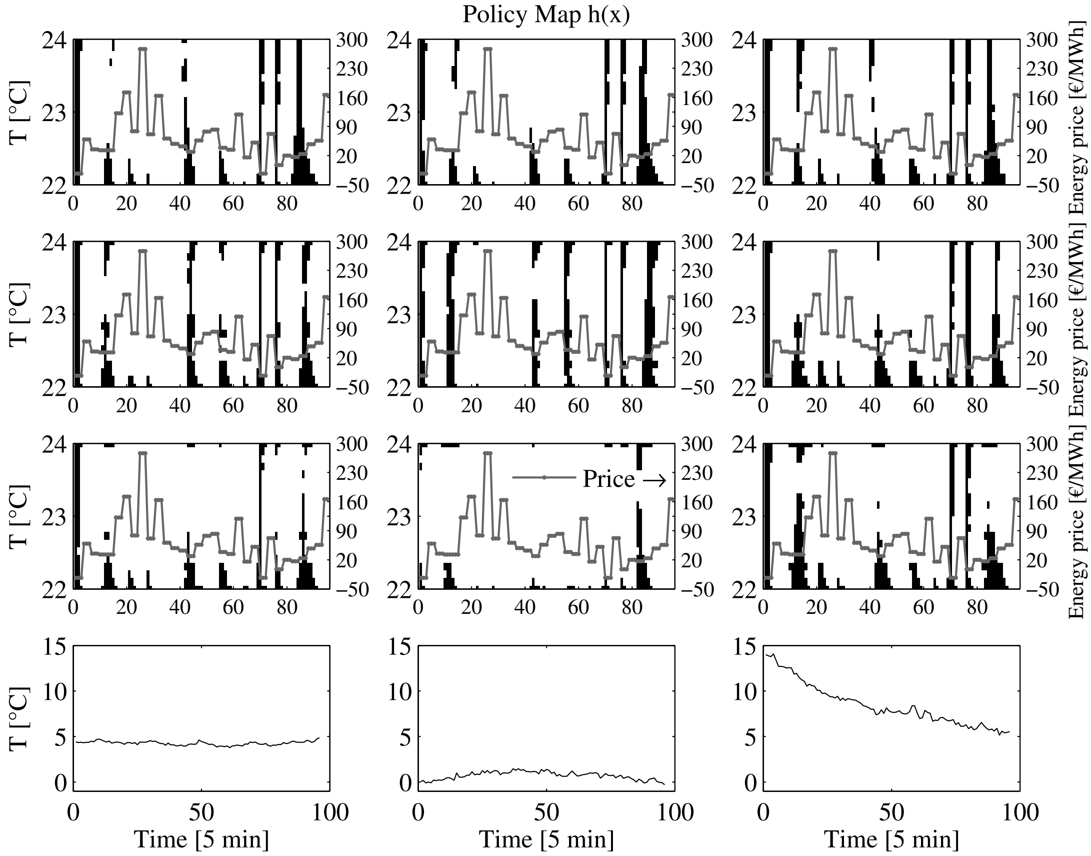

Figure 4: Policy projections obtained from experimental data for different forecast of the outside temperature. The forecasted outside temperature is the same for all policies in each column and is depicted in the lower row. The different policies in each row have been constructed with different amounts of experimental data. From top to bottom, batches containing experimental tuples from 2, 8 and 16 days have been used. The lowest row indicates the forecasts of outside temperature for which the policies have been calculated.

policies depicted in the second row and certainly in the third row show significantly more dependency to the forecasted outside temperatures as the batch of observed tuples contains more variation in terms of outside temperature. Considering the policies depicted in the third row, it can be observed that the policy corresponding to the lower forecasted outside temperature (second column) is less responsive to low energy prices. Whilst the policy obtained for a high forecasted outside temperature (third column) is more responsive. This is inferred from the policy advising the HVAC system to switch ON only for a small region in the state space in the second column, compared to the third column. This is meaningful as at a low outside temperature, energy stored in the room (through an increased temperature) is lost faster, preventing the HVAC system from avoiding high energy prices.

5.3.2. *Policy shaping*

Figure 5 shows the effect of shaping the policy as discussed in Section 4.3. Here triangular membership functions [22] have been used with the constraint that the policy needs to be strictly decreasing with increasing indoor temperature. This is a direct consequence from the physical understanding that a room at a higher temperature is subjected to higher losses to the environment. The results depicted in Figure 5 show the effect of the policy shaping. In the upper row, the *original* ($h(x)$) policies are depicted, whilst the second row shows the shaped policy ($F[\theta_g^*](x)$). It can be observed that the shaping results in *gaps* in the policy being filled. These gaps are attributed to a non-uniform sample density in the batch and the random nature of the extra-trees regression algorithm [49].

5.3.3. *Effect of adding virtual tuples*

Figure 6 shows the impact of virtual tuples on the convergence of the policy. The policies depicted in Figure 6 are computed using different proportions of experimental data and virtual tuples. The left graph in Figure 6 shows the computed policy with 2 days of experimental tuples and no virtual tuples, which is called *early policy*.

The right graph shows the policy computed using a large set of experimental data, without virtual tuples (20 days), called *regime policy*, this policy is considered to be the best possible policy that can be calculated using the data. The middle graph

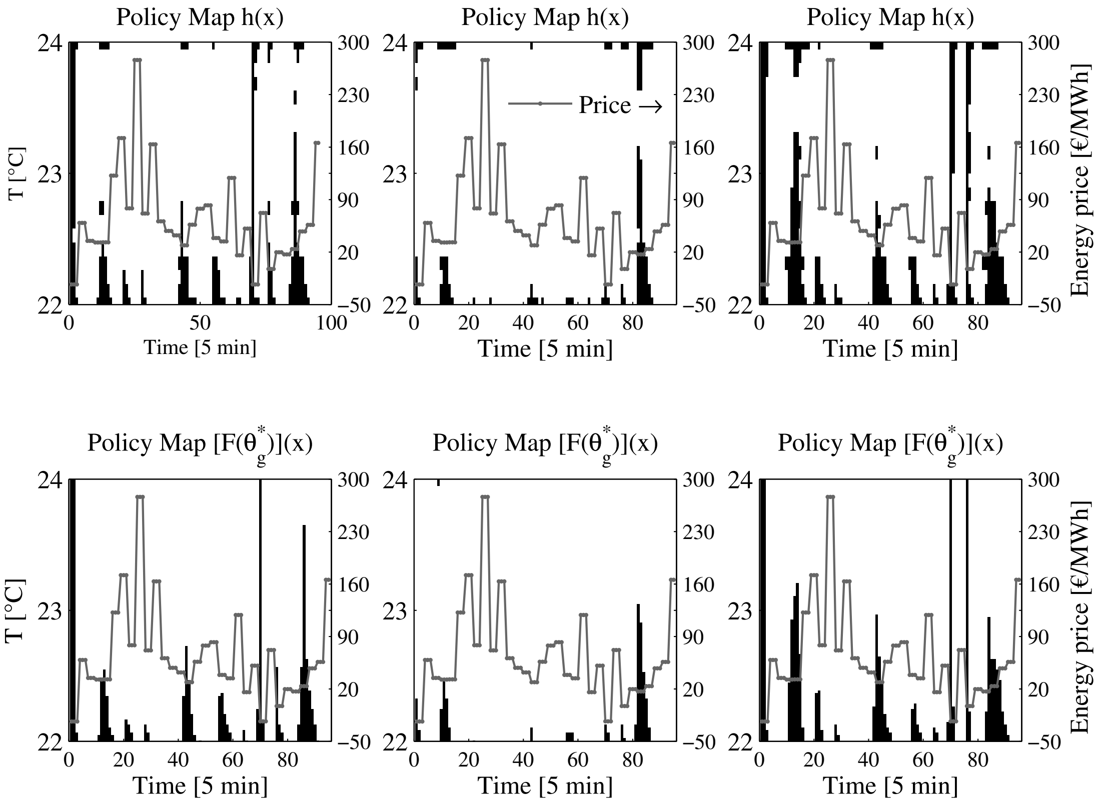

Figure 5: Illustrated effect of policy shaping. Original form (row n. 3 from Figure 3), in the upper row, versus shaped policies, in the lower row.

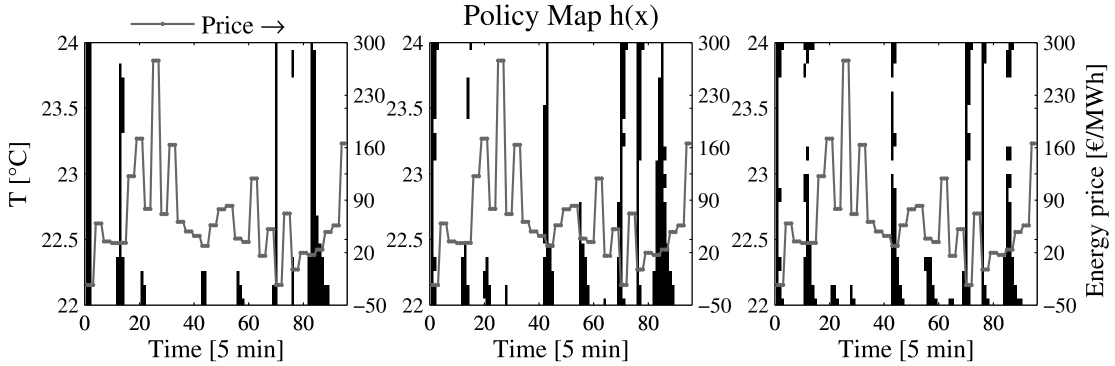

Figure 6: Computation of the closed-loop policies for different shares of virtual tuples over experimental data. The left graph shows the *early policy*, obtained from 2 days of experimental tuples, while the right graph shows the *regime policy*, obtained from 20 days of experimental tuples. The middle graph shows the *model-assisted policy* obtained using 2 days of experimental and virtual tuples.

shows the computed policy using 2 days of experimental tuples and 2 days of virtual tuples from the support model, which is trained using the 2 days of experimental data. This latter is called *model-assisted policy*. One can recognize that the model-assisted policy is more similar to the regime policy than the early policy (94% overlap with the regime policy for the model-assisted policy compared to 90% overlap for the early policy). For example the model-assisted policy around time steps 27, 42 and 78 resembles more the regime policy, this is attributed to the virtual tuples.

5.3.4. Power profiles

Finally, a more direct indication of the performance of the approach presented in this work is illustrated in Figures 7 and 8. These figures show the results of implementing the control approach after 12 and 16 days. The outside temperature during the experiment is depicted in the top graph. The middle graph shows the actual power consumption of the HVAC system and

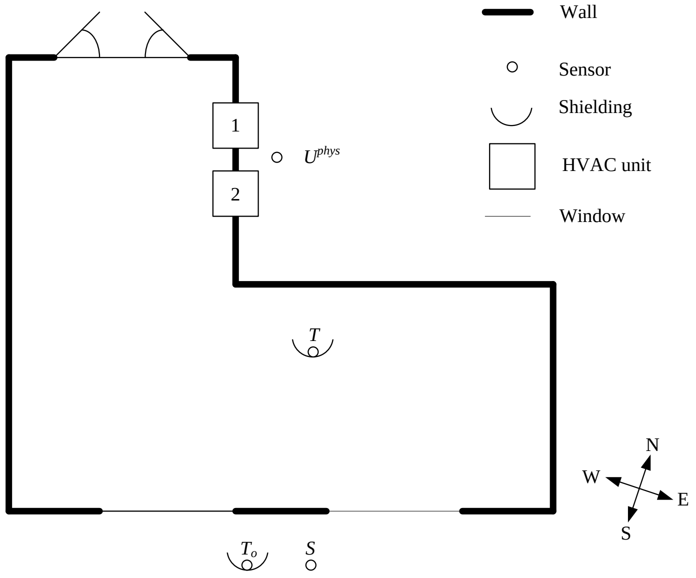

Figure 3: Artist impression of the experimental setup. Depicted are the two HVAC units and the sensors. Dimensions: 7.9m (L), 7.8m (W), 5m (H).

the energy price. The bottom graph depicts the internal temperature and the control policy. From both figures, it can be observed that the HVAC typically switches on at the beginning of a low price period, largely avoiding subsequent high prices, and this for two distinct outside temperature regimes. This demonstrates the efficacy of the model-free control method presented in this work for two different outside temperature regimes.

6. Conclusions

In this work, model-assisted batch reinforcement learning has been deployed to harvest the flexibility related to climate control. The results from a quantitative analysis using a simulated environment showed that a performance within 90% of a mathematical optimum is obtained within approximately 20 days. These results have been confirmed qualitatively after deploying the proposed control approach to a living lab. After collecting data for approximately 16 days the control approach was able to generalize policies for different forecasts of the outside temperature. A policy shaping method has been used to shape the policies by enforcing that the policies need to be strictly decreasing with an increasing indoor temperature. Furthermore, qualitative results indicate that adding virtual tuples from a support model can improve the quality of the control policies when the number of experimental tuples in the batch is small. However, adding virtual tuples resulted only in a limited increase of the performance.

Future work will be aimed towards evaluating the presented algorithm for different objectives related to demand response. Furthermore, other types of models, such as gray-box models, will be used to add virtual tuples.

Table A.1: ETP model parameters

Parameter	Value
R_a	$110\ °C/kW$
C_a	$2.5E+06\ kWh/°C$
R_m	$2000\ °C/kW$
C_m	$1.2E+07\ kWh/°C$
A_s	0.5
A_c	1

Acknowledgements

This work has been done under the FP-7 program Resilient. The authors would like to thank Peter Vrancx, Damien Ernst and Raphael Fonteneau for valuable discussions. The authors acknowledge the financial support of iPower, a project within the Danish Strategic Platform for Innovation and Research within Intelligent Electricity (www.ipower-net.dk).

Appendix A. ETP model

The data batch of the simulated experiments is provided by an Equivalent Thermal Parameter model (ETP) [54] that is fitted on experimental data from the living lab. The heat flow for the ETP model of a residential heating/cooling system is defined as follows:

$$\begin{aligned} \dot{T}_a &= \frac{1}{C_a}\left[\frac{1}{R_a}(T_o - T_a) + \frac{1}{R_m}(T_m - T_a) + A_s Q_s + A_c Q_{AC}\right] \\ \dot{T}_m &= \frac{1}{C_m}\left[\frac{1}{R_m}(T_a - T_m) + (1 - A_s)Q_s\right] \end{aligned}, \tag{A.1}$$

where C_m equals the thermal mass of the building envelope, T_o is the outside air temperature, T_a is the inside air temperature and T_m is the envelope temperature. R_m is the resistance between the inner air and the envelope, while C_a represents the thermal mass of the air. The heat flux into the interior air mass is given by a fraction A_s of the solar heat gain Q_s, a fraction A_c of the heat gains of the air conditioners Q_{AC}. The heat flux to the building envelope is given by the thermal exchange with the inner air and by a fraction A_s of the solar heat gain.

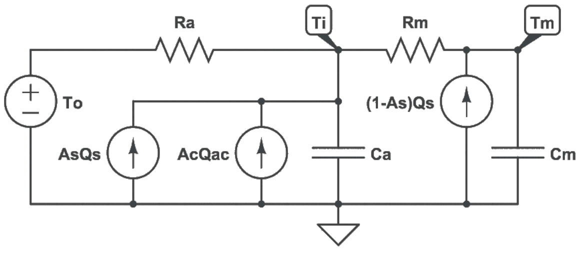

Figure A.9: Sketch of the equivalent thermal parameter model.

Appendix B. Benchmark

An optimal solution of the considered HVAC control problem can be found using a mixed-integer linear programming

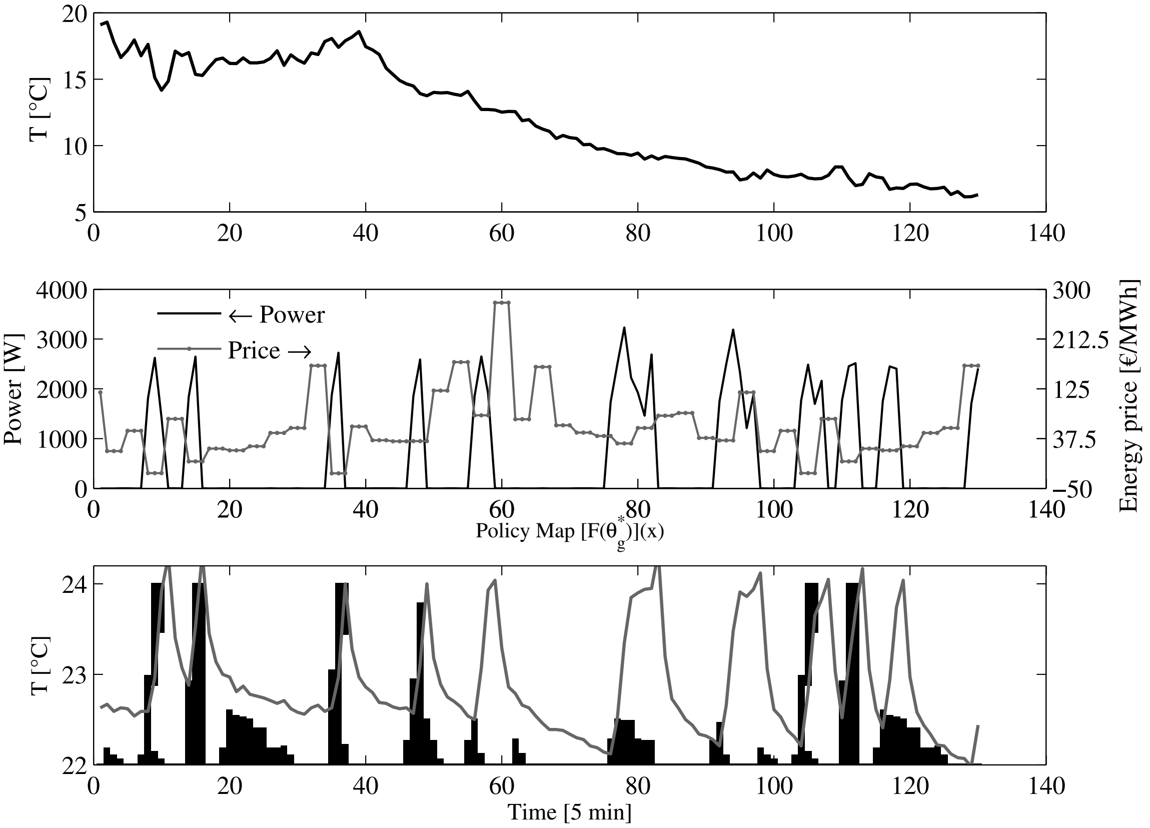

Figure 7: Experimental results 1. Top plot: outside temperature. Middle plot: consumed power by the HVAC and electricity price. Bottom plot: control policy obtained with MABRL and the resulting indoor air temperature.

solver. The objective of the HVAC controller is to minimize its electricity cost using known prices $\boldsymbol{\lambda} \in \mathbb{R}^T$:

$$\min_{\mathbf{u}^{phys}} \sum_{t=1}^{T} P \lambda_t u_t^{phys} \Delta t \tag{B.1}$$

subject to:

$$\begin{aligned} x_{t+1} &= f(x_k, u_k, w_k) \\ u_t^{phys} &= B(x_k, u_k, \theta), \end{aligned}$$

where f is the equivalent thermal parameter model defined by (A.1) and B is the backup controller defined by (13). Notice that the optimal controller knows the equivalent thermal parameter model, the settings of the backup controller and the future disturbances. An optimal solution of this optimization problem was found by applying a mixed-integer linear programming solver using Gurobi [56].

References

[1] L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy and buildings 40 (3) (2008) 394–398.

[2] F. Oldewurtel, D. Sturzenegger, M. Morari, Importance of occupancy information for building climate control, Applied Energy 101 (2013) 521–532.

[3] E. F. Camacho, C. B. Alba, Model predictive control, Springer Science & Business Media, 2013.

[4] R. Halvgaard, N. Poulsen, H. Madsen, J. Jorgensen, Economic model predictive control for building climate control in a smart grid, in: IEEE PES Innovative Smart Grid Technologies (ISGT), 2012, pp. 1–6.

[5] F. Oldewurtel, D. Sturzenegger, G. Andersson, M. Morari, R. S. Smith, Towards a standardized building assessment for demand response, in: IEEE 52nd Annual Conference on Decision and Control (CDC), IEEE, 2013, pp. 7083–7088.

[6] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, K. Wirth, Energy efficient building climate control using stochastic model predictive control and weather predictions, in: American control conference (ACC), 2010, IEEE, 2010, pp. 5100–5105.

[7] J. Ma, J. Qin, T. Salsbury, P. Xu, Demand reduction in building energy systems based on economic model predictive control, Chemical Engineering Science 67 (1) (2012) 92–100.

[8] A. Afram, F. Janabi-Sharifi, Theory and applications of HVAC control systems–a review of model predictive control (MPC), Building and Environment 72 (2014) 343–355.

[9] P. Bacher, H. Madsen, Identifying suitable models for the heat dynamics of buildings, Energy and Buildings 43 (7) (2011) 1511 – 1522.

[10] S. F. Fux, A. Ashouri, M. J. Benz, L. Guzzella, EKF based self-adaptive thermal model for a passive house, Energy and Buildings 68 (2014) 811–817.

[11] M. Morari, C. E. Garcia, D. M. Prett, Model predictive control: theory and practice, in: Workshop on Model Based Process Control, 2014, pp. 1–12.

[12] K. Vanthournout, R. D'hulst, D. Geysen, G. Jacobs, A smart domestic hot water buffer, IEEE Transactions on Smart Grid 3 (4) (2012) 2121–2127.

[13] C. Zhang, D. Yi, N. C. Nordentoft, P. Pinson, J. Østergaard, Flech: A danish market solution for DSO congestion management through der flexibility services, Journal of Modern Power Systems and Clean Energy 2 (2)

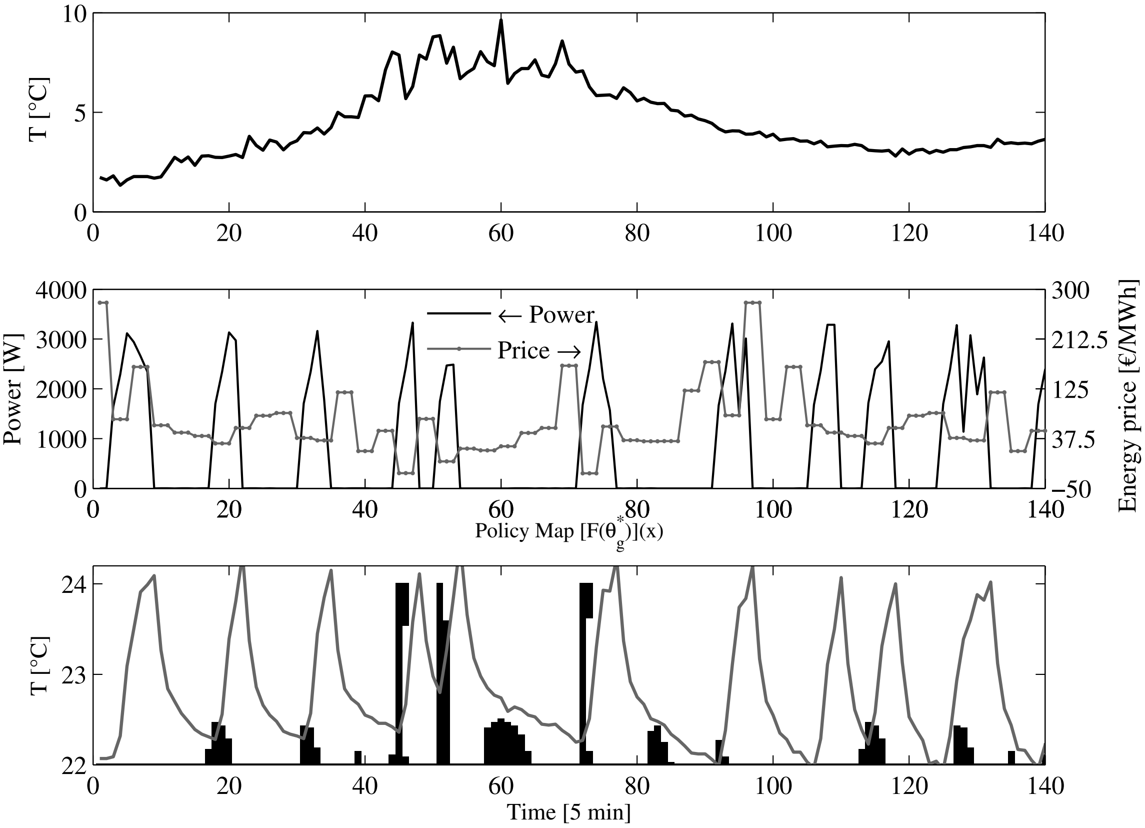

Figure 8: Experimental results 2. Top plot: outside temperature. Middle plot: consumed power by the HVAC and electricity price. Bottom plot: control policy obtained with MABRL and the resulting indoor air temperature.

(2014) 126–133.

[14] M. D. Galus, S. Koch, G. Andersson, Provision of load frequency control by phevs, controllable loads, and a cogeneration unit, IEEE Transactions on Industrial Electronics 58 (10) (2011) 4568–4582.

[15] S. Vandael, T. Holvoet, G. Deconinck, S. Kamboj, W. Kempton, A comparison of two GIV mechanisms for providing ancillary services at the university of delaware, in: IEEE International Conference on Smart Grid Communications (SmartGridComm), 2013, pp. 211–216.

[16] J. Mathieu, M. Kamgarpour, J. Lygeros, G. Andersson, D. Callaway, Arbitraging intraday wholesale energy market prices with aggregations of thermostatic loads, IEEE Transactions on Power Systems 30 (2) (2015) 763–772.

[17] J. Široký, F. Oldewurtel, J. Cigler, S. Prívara, Experimental analysis of model predictive control for an energy efficient building heating system, Applied Energy 88 (9) (2011) 3079–3087.

[18] N. Morel, M. Bauer, M. El-Khoury, J. Krauss, Neurobat, a predictive and adaptive heating control system using artificial neural networks, International Journal of Solar Energy 21 (2-3) (2001) 161–201.

[19] S. Liu, G. P. Henze, Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. theoretical foundation, Energy and Buildings 38 (2) (2006) 142 – 147.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[21] D. Ernst, M. Glavic, F. Capitanescu, L. Wehenkel, Reinforcement learning versus model predictive control: a comparison on a power system problem, Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics 39 (2) (2009) 517–529.

[22] L. Busoniu, R. Babuska, B. De Schutter, D. Ernst, Reinforcement learning and dynamic programming using function approximators, Vol. 39, CRC press, 2010.

[23] R. Fonteneau, S. A. Murphy, L. Wehenkel, D. Ernst, Batch mode reinforcement learning based on the synthesis of artificial trajectories, Annals of operations research 208 (1) (2013) 383–416.

[24] T. Lampe, M. Riedmiller, Approximate model-assisted neural fitted Q-iteration, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 2698–2704.

[25] F. Ruelens, B. Claessens, S. Vandael, B. De Schutter, R. Babuska, R. Belmans, Residential demand response applications using batch reinforcement learning, arXiv preprint arXiv:1504.02125.

[26] E. Atam, L. Helsen, A convex approach to a class of non-convex building HVAC control problems: Illustration by two case studies, Energy and Buildings 93 (0) (2015) 269 – 281.

[27] J. Cigler, D. Gyalistras, J. Široký, V. Tiet, L. Ferkl, Beyond theory: the challenge of implementing model predictive control in buildings, in: Proceedings of 11th Rehva World Congress, Clima, 2013.

[28] L. Ljung, System identification, Springer, 1998.

[29] P. Van Overschee, B. De Moor, A unifying theorem for three subspace system identification algorithms, Automatica 31 (12) (1995) 1853–1864.

[30] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, M. Morari, Use of model predictive control and weather forecasts for energy efficient building climate control, Energy and Buildings 45 (2012) 15–27.

[31] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, P. Haves, Model predictive control for the operation of building cooling systems, IEEE Transactions on Control Systems Technology 20 (3) (2012) 796–803.

[32] D. Bondy, J. Parvizi, Modeling, identification and control for heat dynamics of buildings using robust economic model predictive control, Master's thesis, Masters thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark (2012).

[33] F. Sossan, X. Han, H. Bindner, Dynamic behaviour of a population of controlled-by-price demand side resources, in: PES General Meeting— Conference & Exposition, 2014 IEEE, IEEE, 2014, pp. 1–5.

[34] G. Costanzo, F. Sossan, M. Marinelli, P. Bacher, H. Madsen, Grey-box

modeling for system identification of household refrigerators: A step toward smart appliances, in: 2013 4th International Youth Conference on Energy (IYCE), 2013, pp. 1–5. doi:10.1109/IYCE.2013.6604197.

[35] A. M. Kosek, G. T. Costanzo, H. W. Bindner, O. Gehrke, An overview of demand side management control schemes for buildings in smart grids, in: Smart Energy Grid Engineering (SEGE), 2013 IEEE International Conference on, IEEE, 2013, pp. 1–9.

[36] S. Koch, M. Zima, G. Andersson, Active coordination of thermal household appliances for load management purposes, in: IFAC Symposium on Power Plants and Power Systems Control, Citeseer, 2009.

[37] A. H. Neto, F. A. S. Fiorelli, Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption, Energy and Buildings 40 (12) (2008) 2169–2176.

[38] R. Bellman, Dynamic Programming, Dover Publications, Incorporated., NY, 2003.

[39] M. Deisenroth, C. E. Rasmussen, J. Peters, Model-based reinforcement learning with continuous states and actions.

[40] A. G. Barto, Reinforcement learning: An introduction, MIT press, 1998.

[41] Z. Wen, D. O Neill, H. Maei, Optimal demand response using device-based reinforcement learning, IEEE Transactions on Smart Grid PP (99) (2015) 1–1. doi:10.1109/TSG.2015.2396993.

[42] M. Gonzalez Vaya, L. Rosello, G. Andersson, Optimal bidding of plug-in electric vehicles in a market-based control setup, in: Power Systems Computation Conference (PSCC), 2014, 2014, pp. 1–8. doi:10.1109/PSCC.2014.7038108.

[43] F. Ruelens, B. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets, R. Belmans, Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning, in: Power Systems Computation Conference (PSCC), 2014, pp. 1–7. doi:10.1109/PSCC.2014.7038106.

[44] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, G. Deconinck, Reinforcement learning of heuristic ev fleet charging in a day-ahead electricity market, IEEE Transactions on Smart Grid PP (99) (2015) 1–1. doi:10.1109/TSG.2015.2393059.

[45] A. Faruqui, S. Sergici, Household response to dynamic pricing of electricity: a survey of 15 experiments, Journal of Regulatory Economics 38 (2) (2010) 193–225.

[46] D. P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, Belmont, MA, US, 1995.

[47] D. P. Bertsekas, J. N. Tsitsiklis, Neuro-dynamic programming (optimization and neural computation series, 3), Athena Scientific 7 (1996) 15–23.

[48] M. Scholz, M. Fraunholz, J. Selbig, Nonlinear principal component analysis: neural network models and applications, in: Principal manifolds for data visualization and dimension reduction, Springer, 2008, pp. 44–67.

[49] D. Ernst, P. Geurts, L. Wehenkel, Tree-based batch mode reinforcement learning (2005) 503–556.

[50] Z.-H. Zhou, Ensemble methods: foundations and algorithms, CRC Press, 2012.

[51] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.

[52] E. Cambria, G.-B. Huang, L. L. C. Kasun, H. Zhou, C.-M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, K. Li, et al., Extreme learning machines, IEEE Intelligent Systems 28 (6) (2013) 30–59.

[53] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine learning 63 (1) (2006) 3–42.

[54] W. Zhang, K. Kalsi, J. Fuller, M. Elizondo, D. Chassin, Aggregate model for heterogeneous thermostatically controlled loads with demand response, in: IEEE Power and Energy Society General Meeting, 2012, pp. 1–8.

[55] Elia - Belgian electricity transmission system operator, Grid data, http://www.elia.be/en/grid-data/data-download, [Online: accessed November 27, 2015].

[56] Gurobi Optimization, Inc. Gurobi optimizer reference manual, www.gurobi.com, [Online: accessed 11-July-2015].