# Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning

Frederik Ruelens, Bert J. Claessens, Stijn Vandael, Bart De Schutter, *Senior Member, IEEE*, Robert Babuška, and Ronnie Belmans, *Fellow, IEEE*

***Abstract*—Driven by recent advances in batch Reinforcement Learning (RL), this paper contributes to the application of batch RL to demand response. In contrast to conventional model-based approaches, batch RL techniques do not require a system identification step, making them more suitable for a large-scale implementation. This paper extends fitted Q-iteration, a standard batch RL technique, to the situation when a forecast of the exogenous data is provided. In general, batch RL techniques do not rely on expert knowledge about the system dynamics or the solution. However, if some expert knowledge is provided, it can be incorporated by using the proposed policy adjustment method. Finally, we tackle the challenge of finding an open-loop schedule required to participate in the day-ahead market. We propose a model-free Monte Carlo method that uses a metric based on the state-action value function or Q-function and we illustrate this method by finding the day-ahead schedule of a heat-pump thermostat. Our experiments show that batch RL techniques provide a valuable alternative to model-based controllers and that they can be used to construct both closed-loop and open-loop policies.**

***Index Terms*—Batch reinforcement learning, demand response, electric water heater, fitted Q-iteration, heat pump.**

## I. Introduction

THE INCREASING share of renewable energy sources introduces the need for flexibility on the demand side of the electricity system [1]. A prominent example of loads that offer flexibility at the residential level are thermostatically controlled loads, such as heat pumps, air conditioning units, and electric water heaters. These loads represent about 20% of the total electricity consumption at the residential level in the United States [2]. The market share of these loads is expected to increase as a result of the electrification of heating and cooling [2], making them an interesting domain for demand response [1], [3]–[5]. Demand response programs offer demand flexibility by motivating end users to adapt their consumption profile in response to changing electricity prices or other grid signals. The forecast uncertainty of renewable energy sources [6], combined with their limited controllability, have made demand response the topic of an extensive number of research projects [1], [7], [8] and scientific papers [3], [5], [9]–[11]. The traditional control paradigm defines the demand response problem as a model-based control problem [3], [7], [9], requiring a model of the demand response application, an optimizer, and a forecasting technique. A critical step in setting up a model-based controller includes selecting accurate models and estimating the model parameters. This step becomes more challenging considering the heterogeneity of the end users and their different patterns of behavior. As a result, different end users are expected to have different model parameters and even different models. As such, a large-scale implementation of model-based controllers requires a stable and robust approach that is able to identify the appropriate model and the corresponding model parameters. A detailed report of the implementation issues of a model predictive control strategy applied to the heating system of a building can be found in [12]. Moreover, the authors of [3] and [13] demonstrate a successful implementation of a model predictive control approach at an aggregated level to control a heterogeneous cluster of thermostatically controlled loads.

In contrast, Reinforcement Learning (RL) [14], [15] is a model-free technique that requires no system identification step and no a priori knowledge. Recent developments in the field of reinforcement learning show that RL techniques can either replace or supplement model-based techniques [16]. A number of recent papers provide examples of how a popular RL method, Q-learning [14], can be used for demand response [4], [10], [17], [18]. For example in [10], O'Neill *et al.* propose an automated energy management system based on Q-learning that learns how to make optimal decisions for the consumers. In [17], Henze and Schoenmann investigate the potential of Q-learning for the operation of commercial cold stores and in [4], Kara *et al.* use Q-learning to control a cluster of thermostatically controlled loads. Similarly, in [19] Liang *et al.* propose a Q-learning approach to minimize the electricity cost of the flexible demand and the disutility of the user. Furthermore, inspired by [20], Lee and Powell propose a bias-corrected form of Q-learning to operate battery charging in the presence of volatile prices [18].

Manuscript received April 3, 2015; revised July 2, 2015 and September 29, 2015; accepted December 7, 2015. Date of publication February 8, 2016; date of current version August 21, 2017. This work was supported by the Flemish Institute for the Promotion of Scientific and Technological Research in Industry (IWT). Paper no. TSG-00382-2015.

F. Ruelens, S. Vandael, and R. Belmans are with the Department of Electrical Engineering, Katholieke Universiteit Leuven/EnergyVille, Leuven 3000, Belgium (e-mail: frederik.ruelens@esat.kuleuven.be).

B. J. Claessens is with the Energy Department, Flemish Institute for Technological Research, Mol 2400, Belgium.

B. De Schutter and R. Babuška are with the Delft Center for Systems and Control, Delft University of Technology, Delft 2600 AA, The Netherlands.

Digital Object Identifier 10.1109/TSG.2016.2517211

1949-3053 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

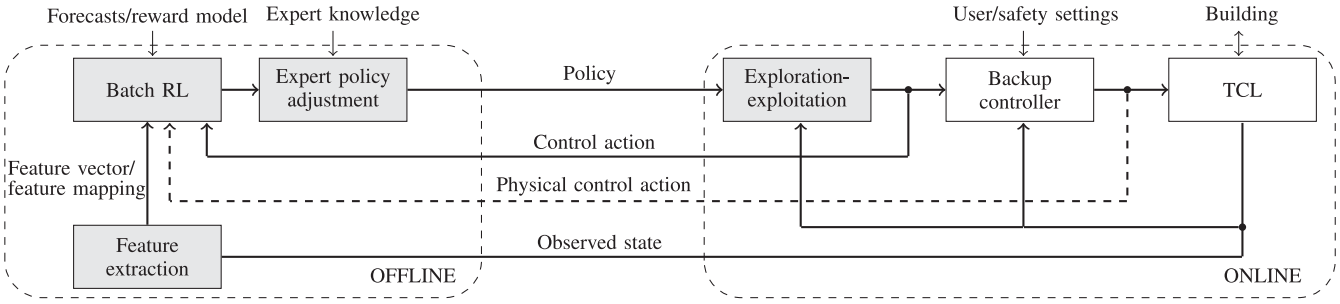


Fig. 1. Building blocks of a model-free Reinforcement Learning (RL) agent (gray) applied to a Thermostatically Controlled Load (TCL).

While being a popular method, one of the fundamental drawbacks of Q-learning is its inefficient use of data, given that Q-learning discards the current data sample after every update. As a result, more observations are needed to propagate already known information through the state space. In order to overcome this drawback, batch RL techniques [21]–[24] can be used. In batch RL, a controller estimates a control policy based on a batch of experiences. These experiences can be a fixed set [23] or can be gathered online by interacting with the environment [25]. Given that batch RL algorithms can reuse past experiences, they converge faster compared to standard temporal difference methods like Q-learning [26] or SARSA [27]. This makes batch RL techniques suitable for practical implementations, such as demand response. For example, the authors of [28] combine Q-learning with eligibility traces in order to learn the consumer and time preferences of demand response applications. In [5], the authors use a batch RL technique to schedule a cluster of electric water heaters and in [29], Vandael *et al.* use a batch RL technique to find a day-ahead consumption plan of a cluster of electric vehicles. An excellent overview of batch RL methods can be found in [25] and [30].

Inspired by the recent developments in batch RL, in particular fitted Q-iteration by Ernst *et al.* [16], this paper builds upon the existing batch RL literature and contributes to the application of batch RL techniques to residential demand response. The contributions of our paper can be summarized as follows: (1) we demonstrate how fitted Q-iteration can be extended to the situation when a forecast of the exogenous data is provided; (2) we propose a policy adjustment method that exploits general expert knowledge about monotonicity conditions of the control policy; (3) we introduce a model-free Monte Carlo method to find a day-ahead consumption plan by making use of a novel metric based on Q-values.

This paper is structured as follows: Section II defines the building blocks of our batch RL approach applied to demand response. Section III formulates the problem as a Markov decision process. Section IV describes our model-free batch RL techniques for demand response. Section V demonstrates the presented techniques in a realistic demand response setting. To conclude, Section VI summarizes the results and discusses further research.

## II. Building Blocks: Model-Free Approach

Fig. 1 presents an overview of our model-free learning agent applied to a Thermostatically Controlled Load (TCL), where the gray building blocks correspond to the learning agent.

At the start of each day the learning agent uses a batch RL method to construct a control policy for the next day, given a batch of past interactions with its environment. The learning agent needs no a priori information on the model dynamics and considers its environment as a black box. Nevertheless, if a model of the exogenous variables, e.g., a forecast of the outside temperature, or reward model is provided, the batch RL method can use this information to enrich its batch. Once a policy is found, an expert policy adjustment method can be used to shape the policy obtained with the batch RL method. During the day, the learning agent uses an exploration-exploitation strategy to interact with its environment and to collect new transitions that are added systematically to the given batch.

In this paper, the proposed learning agent is applied to two types of TCLs. The first type is a residential electric water heater with a stochastic hot-water demand. The dynamic behavior of the electric water heater is modeled using a nonlinear stratified thermal tank model as described in [31]. Our second TCL is a heat-pump thermostat for a residential building. The temperature dynamics of the building are modeled using a second-order equivalent thermal parameter model [32]. This model describes the temperature dynamics of the indoor air and of the building envelope. However, to develop a realistic implementation, this paper assumes that the learning agent cannot measure the temperature of the building envelope and considers it as a hidden state variable.

In addition, we assume that both TCLs are equipped with a backup controller that guarantees the comfort and safety settings of its users. The backup controller is a built-in overrule mechanism that turns the TCL 'on' or 'off' depending on the current state and a predefined switching logic. The operation and settings of the backup controller are assumed to be unknown to the learning agent. However, the learning agent can measure the overrule action of the backup controller (see the dashed arrow in Fig. 1).

The observable state information contains sensory input data of the state of the process and its environment. Before this information is sent to the batch RL algorithm, the learning agent can apply feature extraction [33]. This feature extraction step can have two functions namely to extract hidden state information or to reduce the dimensionality of the state vector. For example, in the case of a heat-pump thermostat, this step can be used to extract a feature that represents the

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

hidden state information, e.g., the temperature of the building envelope. Alternatively, a feature extraction mapping can be used to find a low-dimensional representation of the sensory input data. For example, in the case of an electrical water heater, the observed state vector consists of the temperature sensors that are installed along the hull of the buffer tank. When the number of temperature sensors is large, it can be interesting to map this high-dimensional state vector to a low-dimensional feature vector. This mapping can be the result of an auto-encoder network or principal component analysis. For example in [34], Curran *et al.* indicate that when only a limited number of observations are available, a mapping to a low-dimensional state space can improve the convergence of the learning algorithm.

In this paper, the learning agent is applied to two relevant demand response business models: dynamic pricing and day-ahead scheduling [1], [35]. In dynamic pricing, the learning agent learns a control policy that minimizes its electricity cost by adapting its consumption profile in response to an external price signal. The solution of this optimal control problem is a closed-loop control policy that is a function of the current measurement of the state. The second business model relates to the participation in the day-ahead market. The learning agent constructs a day-ahead consumption plan and then tries to follow it during the day. The objective of the learning agent is to minimize its cost in the day-ahead market and to minimize any deviation between the day-ahead consumption plan and the actual consumption. In contrast to the solution of the dynamic pricing scheme, the day-ahead consumption plan is a feed-forward plan for the next day, i.e., an open-loop policy, which does not depend on future measurements of the state.

## III. Markov Decision Process Formulation

This section formulates the decision-making problem of the learning agent as a Markov decision process. The Markov decision process is defined by its $d$-dimensional state space $X \subset \mathbb{R}^d$, its action space $U \subset \mathbb{R}$, its stochastic discrete-time transition function $f$, and its cost function $\rho$. The optimization horizon is considered finite, comprising $T \in \mathbb{N} \setminus \{0\}$ steps, where at each discrete time step $k$, the state evolves as follows:

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

with $w_k$ a realization of a random disturbance drawn from a conditional probability distribution $p_{\mathcal{W}}(\cdot|x_k)$, $u_k \in U$ the control action, and $x_k \in X$ the state. Associated with each state transition, a cost $c_k$ is given by:

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

The goal is to find an optimal 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 starting from $x_1$ and following a policy $h$ is defined as follows:

$$J_T^h(x_1) = \mathop{\mathbb{E}}_{w_k \sim p_{\mathcal{W}}(\cdot|x_k)} \left[ \sum_{k=1}^{T} \rho(x_k, h(x_k), w_k) \right]. \tag{3}$$

A convenient way to characterize the policy $h$ is by using a state-action value function or Q-function:

$$Q^h(x, u) = \mathop{\mathbb{E}}_{w \sim p_{\mathcal{W}}(\cdot|x)} \Big[ \rho(x, u, w) + \gamma J_T^h(f(x, u, w)) \Big], \tag{4}$$

where $\gamma \in (0, 1)$ is the discount factor. The Q-function is the cumulative return starting from state $x$, taking action $u$, and following $h$ thereafter.

The optimal Q-function corresponds the best Q-function that can be obtained by any policy:

$$Q^*(x, u) = \min_h Q^h(x, u). \tag{5}$$

Starting from an optimal Q-function for every state-action pair, the optimal policy is calculated as follows:

$$h^*(x) \in \arg\min_{u \in U} Q^*(x, u), \tag{6}$$

where $Q^*$ satisfies the Bellman optimality equation [36]:

$$Q^*(x, u) = \mathop{\mathbb{E}}_{w \sim p_{\mathcal{W}}(\cdot|x)} \Big[ \rho(x, u, w) + \gamma \min_{u' \in U} Q^*\big(f(x, u, w), u'\big) \Big]. \tag{7}$$

The next three paragraphs give a formal description of the state space, the backup controller, and the cost function tailored to demand response.

### A. State Description

Following the notational style of [37], the state space $X$ is spanned by a time-dependent component $X_{\mathrm{t}}$, a controllable component $X_{\mathrm{ph}}$, and an uncontrollable exogenous component $X_{\mathrm{ex}}$:

$$X = X_{\mathrm{t}} \times X_{\mathrm{ph}} \times X_{\mathrm{ex}}. \tag{8}$$

*1) Timing:* The time-dependent component $X_{\mathrm{t}}$ describes the part of the state space related to timing, i.e., it carries timing information that is relevant for the dynamics of the system:

$$X_{\mathrm{t}} = X_{\mathrm{t}}^{\mathrm{q}} \times X_{\mathrm{t}}^{\mathrm{d}} \text{ with } X_{\mathrm{t}}^{\mathrm{q}} = \{1, ..., 96\}, X_{\mathrm{t}}^{\mathrm{d}} = \{1, ..., 7\}, \tag{9}$$

where $x_{\mathrm{t}}^{\mathrm{q}} \in X_{\mathrm{t}}^{\mathrm{q}}$ denotes the quarter in the day, and $x_{\mathrm{t}}^{\mathrm{d}} \in X_{\mathrm{t}}^{\mathrm{d}}$ denotes the day in the week. The rationale is that most consumer behavior tends to be repetitive and tends to follows a diurnal pattern.

*2) Physical Representation:* The controllable component $X_{\mathrm{ph}}$ represents the physical state information related to the quantities that are measured locally and that are influenced by the control actions, e.g., the indoor air temperature or the state of charge of an electric water heater:

$$x_{\mathrm{ph}} \in X_{\mathrm{ph}} \text{ with } \underline{x}_{\mathrm{ph}} < x_{\mathrm{ph}} < \bar{x}_{\mathrm{ph}}, \tag{10}$$

where $\underline{x}_{\mathrm{ph}}$ and $\bar{x}_{\mathrm{ph}}$ denote the lower and upper bounds, set to guarantee the comfort and safety of the end user.

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

*3) Exogenous Information:* The state description of the uncontrollable exogenous state is split into two components:

$$X_{\text{ex}} = X_{\text{ex}}^{\text{ph}} \times X_{\text{ex}}^{\text{c}}. \tag{11}$$

When the random disturbance $w_{k+1}$ is independent of $w_k$ there is no need to include exogenous variables in the state space. However, most physical processes, such as the outside temperature and solar radiation, exhibit a certain degree of autocorrelation, where the outcome of the next state depends on the current state. For this reason we include an exogenous component $x_{\text{ex}}^{\text{ph}} \in X_{\text{ex}}^{\text{ph}}$ in our state space description. This exogenous component is related to the observable exogenous information that has an impact on the physical dynamics and that cannot be influenced by the control actions, e.g., the outside temperature.

The second exogenous component $x_{\text{ex}}^{\text{c}} \in X_{\text{ex}}^{\text{c}}$ has no direct influence on the dynamics, but contains information to calculate the cost $c_k$. This work assumes that a deterministic forecast of the exogenous state information related to the cost $\hat{x}_{\text{ex}}^{\text{c}}$ and related to the physical dynamics, i.e., outside temperature and solar radiation, $\hat{x}_{\text{ex}}^{\text{ph}}$ is provided for the time span covering the optimization problem.

### B. Backup Controller

This paper assumes that each TCL is equipped with an overrule mechanism that guarantees comfort and safety constraints. The backup function $B : X \times U \longrightarrow U^{\text{ph}}$ maps the requested control action $u_k \in U$ taken in state $x_k$ to a physical control action $u_k^{\text{ph}} \in U^{\text{ph}}$:

$$u_k^{\text{ph}} = B(x_k, u_k). \tag{12}$$

The settings of the backup function $B$ are unknown to the learning agent, but the resulting action $u_k^{\text{ph}}$ can be measured by the learning agent (see the dashed arrow in Fig. 1).

### C. Cost Function

In general, RL techniques do not require a description of the cost function. However, for most demand response business models a cost function is available. This paper considers two typical cost functions related to demand response.

*1) Dynamic Pricing:* In the dynamic pricing scenario an external price profile is known deterministically at the start of the horizon. The cost function is described as:

$$c_k = u_k^{\text{ph}} \hat{x}_{k,\text{ex}}^{\text{c}} \Delta t, \tag{13}$$

where $\hat{x}_{k,\text{ex}}^{\text{c}}$ is the electricity price at time step $k$ and $\Delta t$ is the length of a control period.

*2) Day-Ahead Scheduling:* The objective of the second business case is to determine a day-ahead consumption plan and to follow this plan during operation. The day-ahead consumption plan should be minimized based on day-ahead prices. In addition, any deviation between the planned consumption and actual consumption should be avoided. As such, the cost function can be written as:

$$c_k = u_k \hat{x}_{k,\text{ex}}^{\text{c}} \Delta t + \alpha \left| u_k \Delta t - u_k^{\text{ph}} \Delta t \right|, \tag{14}$$

**Algorithm 1** Fitted Q-Iteration Using a Forecast of the Exogenous Data (Extended FQI)

**Input:** $\mathcal{F} = \{(x_l, u_l, x'_l, u_l^{\text{ph}})\}_{l=1}^{\#\mathcal{F}}$, $\{(\hat{x}_{l,\text{ex}}^{\text{ph}}, \hat{x}_{l,\text{ex}}^{\text{c}})\}_{l=1}^{\#\mathcal{F}}$, $T$

let $\widehat{Q}_0$ be zero everywhere on $X \times U$

**for** $l = 1, \ldots, \#\mathcal{F}$ **do**

  $\hat{x}'_l \leftarrow (x_{l,\text{t}}^{\text{q}\,\prime}, x_{l,\text{t}}^{\text{d}\,\prime}, x_{l,\text{ph}}', \hat{x}_{l,\text{ex}}^{\text{ph}\,\prime})$ ▷ *replace the observed exogenous part of the next state* $x_{l,\text{ex}}^{\text{ph}\,\prime}$ *by its forecast* $\hat{x}_{l,\text{ex}}^{\text{ph}\,\prime}$

**end for**

**for** $N = 1, \ldots, T$ **do**

  **for** $l = 1, \ldots, \#\mathcal{F}$ **do**

    $c_l \leftarrow \rho(\hat{x}_{l,\text{ex}}^{\text{c}}, u_l^{\text{ph}})$

    $Q_{N,l} \leftarrow c_l + \min_{u \in U} \widehat{Q}_{N-1}(\hat{x}'_l, u)$

  **end for**

  use regression to obtain $\widehat{Q}_N$ from $\mathcal{T}_{\text{reg}} = \left\{\left((x_l, u_l), Q_{N,l}\right), l = 1, \ldots, \#\mathcal{F}\right\}$

**end for**

**Ensure:** $\widehat{Q}^* = \widehat{Q}_T$

where $u_k$ is the planned consumption, $u_k^{\text{ph}}$ is the actual consumption, $\hat{x}_{k,\text{ex}}^{\text{c}}$ is the forecasted day-ahead price and $\alpha > 0$ is a penalty. The first part of (14) is the cost for buying energy at the day-ahead market, whereas the second part penalizes any deviation between the planned consumption and the actual consumption.

### D. Reinforcement Learning for Demand Response

When the description of the transition function and cost function is available, techniques that make use of the Markov decision process framework, such as approximate dynamic programming [38] or direct policy search [30], can be used to find near-optimal policies. However, in our implementation we assume that the transition function $f$, the backup controller $B$, and the underlying probability of the exogenous information $w$ are unknown. In addition, we assume that they are challenging to obtain in a residential setting. For these reasons, we present a model-free batch RL approach that builds on previous theoretical work on RL, in particular fitted Q-iteration [23], expert knowledge [39], and the synthesis of artificial trajectories [21].

## IV. Algorithms

Typically, batch RL techniques construct policies based on a batch of tuples of the form: $\mathcal{F} = \{(x_l, u_l, x'_l, c_l)\}_{l=1}^{\#\mathcal{F}}$, where $x_l = (x_{l,\text{t}}^{\text{q}}, x_{l,\text{t}}^{\text{d}}, x_{l,\text{ph}}, x_{l,\text{ex}}^{\text{ph}})$ denotes the state at time step $l$ and $x'_l$ denotes the state at time step $l+1$. However, for most demand response applications, the cost function $\rho$ is given a priori, and of the form $\rho(\hat{x}_{l,\text{ex}}^{c}, u_l^{\text{ph}})$. As such, this paper considers tuples of the form $(x_l, u_l, x'_l, u_l^{\text{ph}})$.

### A. Fitted Q-Iteration Using a Forecast of the Exogenous Data

Here we demonstrate how fitted Q-iteration [23] can be extended to the situation when a forecast of the exogenous

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

component is provided (Algorithm 1). The algorithm iteratively builds a training set $\mathcal{T}_{\text{reg}}$ with all state-action pairs $(x, u)$ in $\mathcal{F}$ as the input. The target values consist of the corresponding cost values $\rho(\hat{x}^{\text{c}}_{\text{ex}}, u^{\text{ph}})$ and the optimal Q-values, based on the approximation of the Q-function of the previous iteration, for the next states $\min_{u\in U} \widehat{Q}_{N-1}(\hat{x}'_l, u)$.

Since we consider a finite horizon problem with $T$ control periods, Algorithm 1 needs $T$ iterations until the Q-function contains all information about the future rewards. Notice that for $N = 1$ (first iteration), the Q-values in the training set correspond to their immediate cost $Q_{N,l} \leftarrow c_l$ (line 8 in Algorithm 1). In the subsequent iterations, Q-values are updated using the value iteration based on the Q-function of the previous iteration.

It is important to note that $\hat{x}'_l$ denotes the successor state in $\mathcal{F}$, where the observed exogenous information $x^{\text{ph}\,\prime}_{l,\text{ex}}$ is replaced by its forecast $\hat{x}^{\text{ph}\,\prime}_{l,\text{ex}}$ (line 3 in Algorithm 1). Note that in our algorithm the next state contains information on the forecasted exogenous data, whereas for standard fitted Q-iteration the next state contains past observations of the exogenous data. By replacing the observed exogenous parts of the next state by their forecasts, the Q-function of the next state assumes that the exogenous information will follow its forecast. In other words, the Q-value of the next state becomes biased towards the provided forecast of the uncontrollable exogenous data. Also notice that we recalculate the cost for each tuple in $\mathcal{F}$ by using the price for the next day $\hat{x}^{\text{c}}_{\text{ex}}$ (line 7 in Algorithm 1). The proposed algorithm is relevant for demand response applications that are influenced by exogenous weather data. Examples of these applications are heat-pump thermostats and air conditioning units.

In principle, any function approximator, such as a neural network [40], can be applied in combination with fitted Q-iteration. However, because of its robustness and fast calculation time, an ensemble of extremely randomized trees [23] is used to approximate the Q-function.

A discussion on the convergence properties of the proposed method and a comparison with Q-learning can be found in the appendix section.

### *B. Expert Policy Adjustment*

Given the Q-function from Algorithm 1, a near-optimal policy $\widehat{h}^*$ can be constructed by solving (6) for every state in the state space. However, in some cases, e.g., when $\mathcal{F}$ contains a limited number of observations, the resulting policy can be improved by using general prior knowledge about its shape.

In this section, we show how expert knowledge about the monotonicity of the policy can be exploited to regularize the policy. The method enforces monotonicity conditions by using convex optimization to approximate the policy, where expert knowledge is included in the form of extra constraints. These constraints can result directly from an expert or from a model-based solution. In order to define a convex optimization problem we use a fuzzy model with triangular membership functions [30] to approximate the policy. The centers of the triangular membership functions are located on an equidistant grid with $N$ membership functions along each dimension of the state space. This partitioning leads to $N^d$ state-dependent membership functions for each action. The parameter vector $\theta^*$ that approximates the original policy can be found by solving the following least-squares problem:

$$\theta^* \in \arg\min_{\theta} \sum_{l=1}^{\#\mathcal{F}} \left([F(\theta)](x_l) - \widehat{h}^*(x_l)\right)^2, \quad \text{s.t.} \quad \text{expert knowledge} \tag{15}$$

where $F(\theta)$ denotes an approximation mapping of a weighted linear combination of a set of triangular membership functions $\phi$ and $[F(\theta)](x)$ denotes the policy $F(\theta)$ evaluated at state $x$. The triangular Membership Functions (MFs) for each state variable $x_d$, with $d \in \{1, \ldots, \dim(X)\}$ are defined as follows:

$$\phi_{d,1}(x_d) = \max\left(0, \frac{c_{d,2} - x_d}{c_{d,2} - c_{d,1}}\right)$$

$$\phi_{d,i}(x_d) = \max\left[0, \min\left(\frac{x_d - c_{d,i-1}}{c_{d,i} - c_{d,i-1}}, \frac{c_{d,i+1} - x_d}{c_{d,i+1} - c_{d,i}}\right)\right],$$

$$\phi_{d,N_d}(x_d) = \max\left(0, \frac{x_d - c_{d,N_d-1}}{c_{d,N_d} - c_{d,N_d-1}}\right)$$

for $i = 2, \ldots, N_d - 1$. The centers of the MFs along dimension $d$ are denoted by $c_{d,1}, \ldots, c_{d,N_d}$ which must satisfy: $c_{d,1} < \cdots < c_{d,N_d}$. These centers are chosen equidistant along the state space in such a way that $x_d \in [c_{d,1}, c_{d,N_d}]$ for $d = 1, \ldots, \dim(X)$.

The fuzzy approximation of the policy allows us to add expert knowledge to the policy in the form of convex constraints of the least-squares problem (15), which can be solved using a convex optimization solver. Using the same notation as in [39], we can enforce monotonicity conditions along the $d$th dimension of the state space as follows:

$$\delta_d[F(\theta)](x_d) \leq \delta_d[F(\theta)]\left(x'_d\right) \tag{16}$$

for all state components $x_d \leq x'_d$ along the dimension $d$. If $\delta_d$ is -1 then $[F(\theta)]$ will be decreasing along the $d$th dimension of $X$, whereas if $\delta_d$ is 1 then $[F(\theta)]$ will be increasing along the $d$th dimension of $X$. Once $\widehat{h}^*$ is found, the adjusted policy $\widehat{h}^*_{\text{exp}}$, given this expert knowledge, can be calculated as $\widehat{h}^*_{\text{exp}}(x) = [F(\theta^*)](x)$.

### *C. Day-Ahead Consumption Plan*

This section explains how to construct a day-ahead schedule starting from the Q-function obtained by Algorithm 1 and using cost function (14). Finding a day-ahead schedule has a direct relation to two situations:

- a day-ahead market, where participants have to submit a day-ahead schedule one day in advance of the actual consumption [35];
- a distributed optimization process, where two or more participants are coupled by a common constraint, e.g., congestion management [9].

Algorithm 2 describes a model-free Monte Carlo method to find a day-ahead schedule that makes use of a metric based on Q-values. The method estimates the average return of a policy by synthesizing $p$ sequences of transitions of length $T$ from $\mathcal{F}$. These $p$ sequences can be seen as a proxy of the

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

actual trajectories that could be obtained by simulating the policy on the given control problem. Note that since we consider a stochastic setting, $p$ needs to be greater than 1. A sequence is grown in length by selecting a new transition among the samples of not-yet-used one-step transitions. Each new transition is selected by minimizing a distance metric with the previously selected transition.

In [21], Fonteneau *et al*. propose the following distance metric in $X \times U : \Delta((x, x'), (u, u')) = \|x - x'\| + \|u - u'\|$, where $\|\cdot\|$ denotes the Euclidean norm. It is important to note that this metric weighs each dimension of the state and action space equally. In order to overcome specifying weights for each dimension, we propose the following distance metric:

$$\left|\widehat{Q}^*\left(x_k^i, u_k^i\right) - \widehat{Q}^*(x_l, u_l)\right| + \xi \left\|x_k^i - x_l\right\|. \tag{17}$$

Here $\widehat{Q}^*$ is obtained by applying Algorithm 1 with cost function (14). The state $x_k^i$ and action $u_k^i$ denote the state and action corresponding to the $i$th trajectory at time step $k$. The control action $u_k^i$ is computed by minimizing the Q-function $\widehat{Q}^*$ in $x_k^i$ (see line 6). The next state $x'_{l^i}$ is found by taking the next state of the tuple that minimizes this distance metric (see line 8). The regularization parameter $\xi$ is a scalar that is included to penalize states that have similar Q-values, but have a large Euclidean norm in the state space. When the Q-function is strictly increasing or decreasing $\xi$ can be set to 0. The motivation behind using Q-values instead of the Euclidean distance in $X \times U$ is that Q-values capture the dynamics of the system and, therefore, there is no need to select individual weights.

Notice that once a tuple with the lowest distance metric is selected, this tuple is removed from the given batch (see line 12). As a result, this ensures that the $p$ artificial trajectories are distinct and thus can be seen as $p$ stochastic realizations that could be obtained by simulating the policy on the given control policy.

## V. Simulations

This section presents the simulation results of three experiments and evaluates the performance of the proposed algorithms. We focus on two examples of flexible loads, i.e., an electric water heater and a heat-pump thermostat. The first experiment evaluates the performance of extended FQI (Algorithm 1) for a heat-pump thermostat. The rationale behind using extended FQI for a heat-pump thermostat, is that the temperature dynamics of a building is influenced by exogenous weather data, which is less the case for an electric water heater. In the second experiment, we apply the policy adjustment method to an electric water heater. The final experiment uses the model-free Monte Carlo method to find a day-ahead consumption plan for a heat-pump thermostat. It should be noted that the policy adjustment method and model-free Monte Carlo method can also be applied to both heat-pump thermostat and electric water heater.

### A. Thermostatically Controlled Loads

Here we describe the state definition and the settings of the backup controller of the electric water heater and the heat-pump thermostat.

**Algorithm 2** Model-Free Monte Carlo Method Using a Q-Function to Find a Feed-Forward Plan

**Input:** $\mathcal{F} = \{(x_l, u_l, x'_l, u_l^{\text{ph}})\}_{l=1}^{\#\mathcal{F}}$, $\{(\hat{x}_{l,\text{ex}}^{\text{ph}}, \hat{x}_{l,\text{ex}}^{\text{c}})\}_{l=1}^{\#\mathcal{F}}$, $T$, $x_1$, $p$, $\xi$

1: $\mathcal{G} \leftarrow \mathcal{F}$
2: Apply Algorithm 1 with cost function (14), to obtain $\widehat{Q}^*$
3: **for** $i = 1, \ldots, p$ **do**
4: $\quad x_1^i \leftarrow x_1$
5: $\quad$ **for** $k = 1, \ldots, T$ **do**
6: $\quad\quad u_k^i \leftarrow \arg\min_{u \in U} \widehat{Q}^*(x_k^i, u)$
7: $\quad\quad \mathcal{G}_\text{s} \leftarrow \{(x_l, u_l, x'_l, u_l^{\text{ph}}) \in \mathcal{G} | u_l = u_k^i\}$
8: $\quad\quad \mathcal{H} \leftarrow \arg\min_{(x_l, u_l, x'_l, u_l^{\text{ph}}) \in \mathcal{G}_\text{s}} |\widehat{Q}^*(x_k^i, u_k^i) - \widehat{Q}^*(x_l, u_l)| + \xi \|x_k^i - x_l\|$
9: $\quad\quad l^i \leftarrow$ lowest index in $\mathcal{G}$ of the transitions in $\mathcal{H}$
10: $\quad\quad P_k^i \leftarrow u_k^i$
11: $\quad\quad x_{k+1}^i \leftarrow x'_{l^i}$
12: $\quad\quad \mathcal{G} \leftarrow \mathcal{G} \backslash \left\{(x_{l^i}, u_{l^i}, x'_{l^i}, u_{l^i}^{\text{ph}})\right\}$ ▷*do not reuse tuple*
13: $\quad$ **end for**
14: **end for**

**Ensure:** $p$ artificial trajectories: $[P_k^1]_{k=1}^T, \ldots, [P_k^p]_{k=1}^T$

*1) Electric Water Heater:* We consider that the storage tank of the electric water heater is equipped with $n_\text{s}$ temperature sensors. The full state description of the electric water heater is defined as follows:

$$x_k = \left(x_{k,t}^\text{q}, T_k^1, \ldots, T_k^i, \ldots, T_k^{n_\text{s}}\right), \tag{18}$$

where $x_{k,\text{t}}^\text{q}$ denotes the current quarter in the day and $T_k^i$ denotes the temperature measurement of the $i$th sensor. This work uses feature extraction to reduce the dimensionality of the controllable state space component by replacing it with with the average sensor measurement. As such the reduced state is defined as follows:

$$x_k = \left(x_{k,\text{t}}^\text{q}, \frac{\sum_{i=1}^{n_\text{s}} T_k^i}{n_\text{s}}\right). \tag{19}$$

More generic dimension reduction techniques, such as an auto-encoder network and a principal components analysis [41], [42] will be explored in future research.

The logic of the backup controller of the electric water heater is defined as:

$$B(x_k, u_k) = \begin{cases} u_k^{\text{ph}} = u_{\max} & \text{if } x_{k,\text{soc}} \leq 30\% \\ u_k^{\text{ph}} = u_k & \text{if } 30\% < x_{k,\text{soc}} < 100\%. \\ u_k^{\text{ph}} = 0 & \text{if } x_{k,\text{soc}} \geq 100\% \end{cases} \tag{20}$$

The electric heating element of the electric water heater can be controlled with a binary control action $u_k \in \{0, u_{\max}\}$, where $u_{\max} = 2.3$ kW is the maximum power. A detailed description of the nonlinear tank model of the electric water heater and the calculation of its state of charge $x_\text{soc}$ can be found in [31]. We use a set of hot-water demand profiles with a daily mean load of 100 liter obtained from [43] to simulate a realistic tap demand.

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

*2) Heat-Pump Thermostat:* Our second application considers a heat-pump thermostat that can measure the indoor air temperature, the outdoor air temperature, and the solar radiation. The full state of the heat-pump thermostat is defined as follows:

$$x_k = \left(x_{k,\mathrm{t}}^{\mathrm{q}}, T_{k,\mathrm{in}}, T_{k,\mathrm{out}}, S_k\right), \tag{21}$$

where the physical state space component consists of the indoor air temperature $T_{k,\mathrm{in}}$ and the exogenous state space component contains the outside air temperature $T_{k,\mathrm{out}}$ and the solar radiation $S_k$. The internal heat gains, caused by user behavior and electric appliances, cannot be measured and are excluded from the state. We included a measurement of the solar radiance in the state since solar energy transmitted through windows can significantly impact the indoor temperature dynamics [32]. In order to have a practical implementation, we consider that we cannot measure the temperature of the building envelope. We used a feature extraction technique based on past state observations by including a virtual building envelope temperature, which is a running average of the past $n_{\mathrm{r}}$ air temperatures:

$$x_k = \left(x_{k,\mathrm{t}}^{\mathrm{q}}, T_{k,\mathrm{in}}, \frac{\sum_{i=k-n_{\mathrm{r}}}^{k-1} T_{i,\mathrm{in}}}{n_{\mathrm{r}}}, T_{k,\mathrm{out}}, S_k\right). \tag{22}$$

The backup controller of the heat-pump thermostat is defined as follows:

$$B(x_k, u_k) = \begin{cases} u_k^{\mathrm{ph}} = u_{\max} & \text{if } T_{k,\mathrm{in}} \leq \underline{T}_k \\ u_k^{\mathrm{ph}} = u_k & \text{if } \underline{T}_k < T_{k,\mathrm{in}} < \overline{T}_k, \\ u_k^{\mathrm{ph}} = 0 & \text{if } T_{k,\mathrm{in}} \geq \overline{T}_k \end{cases} \tag{23}$$

where $\underline{T}_k$ and $\overline{T}_k$ are the minimum and maximum temperature settings defined by the end user. The action space of the heat pump is discretized in 10 steps, $u_k \in \{0, ..., u_{\max}\}$, where $u_{\max} = 3\mathrm{kW}$ is the maximum power. In the simulation section we define a minimum and maximum comfort setting of 19°C and 23°C. A detailed description of the temperature dynamics of the indoor air and building envelope can be found in [32]. The exogenous information consists of the outside temperature, the solar radiation, and the internal heat gains from a location in Belgium [1].

In the following experiments we define a control period of one quarter and an optimization horizon $T$ of 96 control periods.

### B. Experiment 1

The goal of the first experiment is to compare the performance of Fitted Q-Iteration (standard FQI) [23] to the performance of our extension of FQI (extended FQI), given by Algorithm 1. The objective of the considered heating system is to minimize the electricity cost of the heat pump by responding to an external price signal. The electricity prices are taken from the Belgian wholesale market [35]. We assume that a forecast of the outside temperature and of the solar radiance is available. Since the goal of the experiment is to assess the impact on the performance of FQI when a forecast is included, we assume perfect forecasts of the outside temperature and solar radiance.

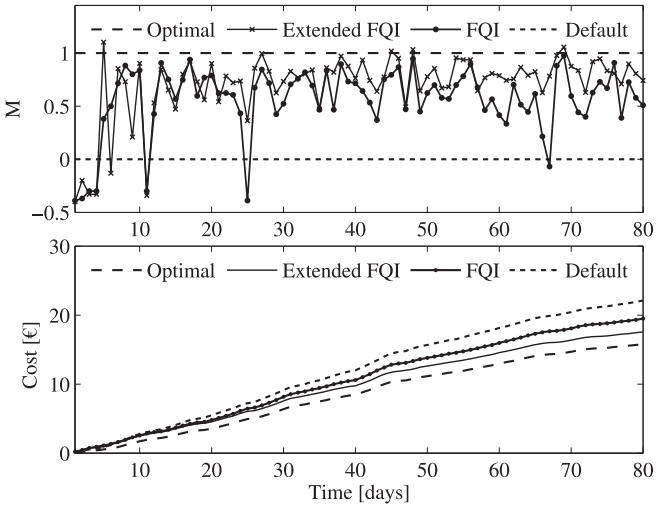


Fig. 2. Simulation results for a heat-pump thermostat and a dynamic pricing scheme using an optimal controller (Optimal), Fitted Q-Iteration (FQI), extended FQI, and a default controller (Default). The top plot depicts the performance metric $M$ and the bottom plot depicts the cumulative electricity cost.

*1) FQI Controllers:* Both FQI controllers start with an empty batch $\mathcal{F}$. At the end of each day, they add the tuples of the given day to their current batch and they compute a $T$-stage control policy for the next day. Both FQI controllers calculate the cost value of each tuple in $\mathcal{F}$ (line 7 in Algorithm 1). As such, FQI can reuse (or replay) previous observations even when the external prices change daily. The extended FQI controller uses the forecasted values of the outside temperature and solar radiance to construct the next states in the batch $\hat{x}_l' \leftarrow (x_{l,\mathrm{t}}^{\mathrm{q}\,\prime}, T_{l,\mathrm{in}}', \hat{T}_{l,\mathrm{out}}', \hat{S}_l')$, where $\hat{(\cdot)}$ denotes a forecast. In contrast, standard FQI uses observed values of the outside temperature and solar radiance to construct the next states in the batch.

The observed state information of both FQI controllers is defined by (22), where a handcrafted feature is used to represent the temperature of the building envelope ($n_{\mathrm{r}}$ set to 3). During the day, both FQI controllers use an $\varepsilon$-greedy exploration strategy. This exploration strategy selects a random control action with probability $\varepsilon_d$ and follows the policy with probability $1 - \varepsilon_d$. Since more interactions result in a better coverage of the state-action space, the exploration probability $\varepsilon_d$ is decreased on a daily basis, according to the harmonic sequence $1/d^n$, where $n$ is set to 0.7 and $d$ denotes the current day. As a result the agent becomes greedier, i.e., it selects the best action more often, as the number of tuples in the batch increases. A possible route of future work could be to include a Boltzmann exploration strategy that explores interesting state-action pairs based on the current estimate of the Q-function [38].

The exploration parameters and exploration probabilities of both FQI and extended FQI are identical. Moreover, we used identical disturbances, prices and weather conditions in both experiments.

*2) Results:* In order to compare the performance of the FQI controllers we define the following metric:

$$M = \frac{c_{\mathrm{fqi}} - c_{\mathrm{d}}}{c_{\mathrm{o}} - c_{\mathrm{d}}}, \tag{24}$$

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

where $c_{\text{fqi}}$ denotes the daily cost of the FQI controller, $c_{\text{d}}$ denotes the daily cost of the default controller and $c_{\text{o}}$ denotes the daily cost of the optimal controller. The metric $M$ corresponds to 0 if the FQI controller obtains the same performance as the default controller and corresponds to 1 if the FQI controller obtains the same performance as the optimal controller. The default controller is a hysteresis controller that switches on when the indoor air temperature is lower than 19°C and stops heating when the indoor air temperature reaches 20°C. The optimal controller is a model-based controller that has full information on the model parameters and has perfect forecasts of all exogenous variables, i.e., the outside temperature, solar radiation and internal heat gains. The optimal controller formalizes the problem as a mixed integer problem and uses a commercial solver [44]. The simulation results of the heating system for a simulation horizon of 80 days are depicted in Fig. 2. The top plot depicts the daily metric $M$, where the performance of the default controller corresponds to zero and the performance of the optimal controller corresponds to one. The bottom plot depicts the cumulative electricity cost of both FQI controllers, and of the default and optimal controller. The average metric $M$ over the simulation horizon is 0.56 for standard FQI and 0.71 for extended FQI, which is an improvement of 27%. The performance gap of 0.29 between extended FQI and the optimal controller is a reasonable result given that the model dynamics and disturbances are unknown, and that exploration is included.

This experiment demonstrated that fitted Q-iteration can be successfully extended to incorporate forecasted data. Extended FQI was able to decrease the total electricity cost with 19% compared to the default controller over the total simulation horizon, whereas standard FQI decreased the total electricity cost with 14%. It is important to note that the reduction of 19% is not the result of a lower energy consumption, since the energy consumption increased with 4% compared to the default controller when extended FQI was used.

### C. Experiment 2

The following experiment demonstrates the policy adjustment method for an electric water heater. As an illustrative example we used a sinusoidal price profile. As stated in (19), the state space of an electric water heater consists of two dimensions, i.e., the time component and the average temperature.

*1) Policy Adjustment Method:* As described in Section IV-B, the policy obtained with Algorithm 1 is represented using a grid of fuzzy membership functions. Our representation consists of $5 \times 5$ membership functions, located equidistantly along the state space. We enforce a monotonicity constraint along the second dimension, which contains the average sensor temperature, as follows:

$$[F(\theta)]\left(x_{\text{t}}^{q}, T_{\text{a}}\right) \leq [F(\theta)]\left(x_{\text{t}}^{q}, T_{\text{a}}'\right), \tag{25}$$

for all states $x_{\text{t}}^{q}$ and $T_{\text{a}}$, and where $T_{\text{a}} < T_{\text{a}}'$. These monotonicity constraints are added to the least-squares problem (15).

*2) Results:* The original policies obtained with fitted Q-iteration after 7, 14 and 21 days are presented in the left column of Fig. 3. It can be seen that the original policies, obtained by fitted Q-iteration, violate the monotonicity constraints along the second dimension in several states. The adjusted policies obtained by the policy adjustment method are depicted in the right column. The policy adjustment method was able to reduce the total electricity cost by 11% over 60 days compared to standard FQI. These simulation results indicate that when the number of tuples in the batch increases, the original and the adjusted policies converge. Furthermore, the results indicate that when the number of tuples in $\mathcal{F}$ is small, the expert policy adjustment method can be used to improve the performance of standard fitted Q-iteration.

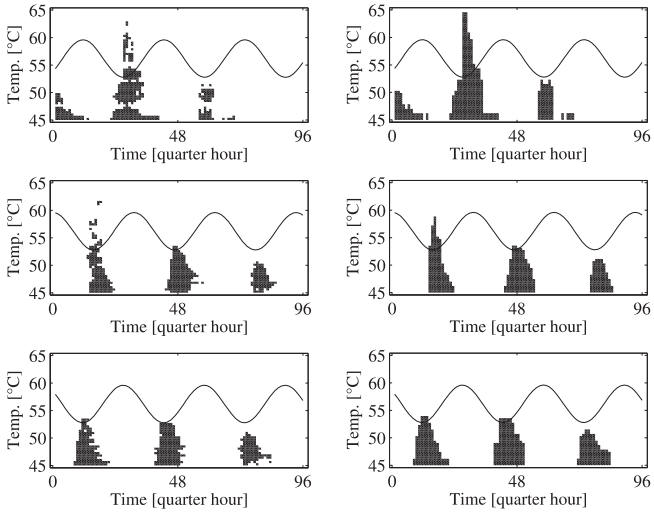


Fig. 3. Simulation results for an electric water heater and a dynamic pricing scheme. The left column depicts the original policies for day 7, 14, and 21 where the color corresponds to the control action, namely white (off) and black (on). The corresponding repaired policies are depicted in the right column. The price profile corresponding to each policy is depicted in the background.

### D. Experiment 3

The final experiment demonstrates the Model-Free Monte Carlo (MFMC) method (Algorithm 2) to find the day-ahead consumption plan of a heat-pump thermostat.

*1) MFMC Method:* First, the MFMC method uses Algorithm 1 with cost function (14) to calculate the Q-function. The parameter $\alpha$ was set to $10^3$ to penalize possible deviations between the planned consumption profile and the actual consumption. The resulting Q-function is then used as a metric to build $p$ distinct artificial trajectories (line 8 of Algorithm 2). The regularization parameter $\varepsilon$ was set to 0.1 to penalize states with identical Q-values, but with a large Euclidean norm in the state space. The parameter $p$, which indicates the number of artificial trajectories, was set to 4. Increasing the number of artificial trajectories beyond 4 did not significantly improved the performance of the MFMC method. A day-ahead consumption plan was obtained by taking the average of these 4 artificial trajectories.

*2) Results:* In order to assess the performance of our MFMC method, we introduce an optimal model-based controller. Similar as in experiment 1, the model-based controller has exact knowledge about the future exogenous variables and

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

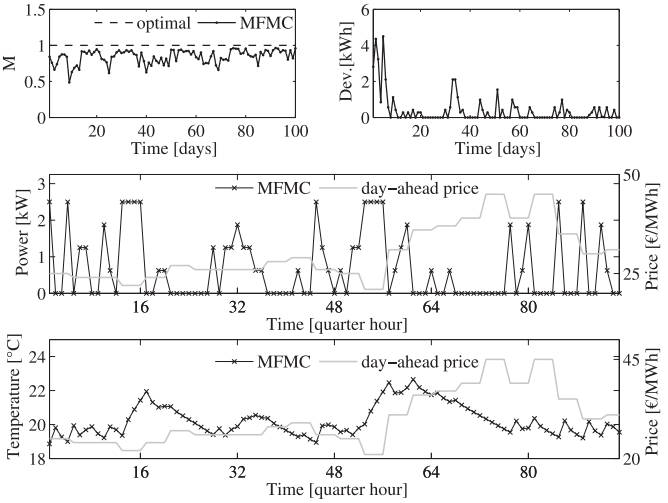


Fig. 4. The top plots depict the metric $M$ (left) and the daily deviation between the day-ahead consumption plan and actual consumption (right). The day-ahead consumption plan and the indoor temperature obtained with the Model-Free Monte Carlo (MFMC) method are depicted in the middle and bottom plot.

model equations. The solution of the model-based controller was found using a commercial solver [44]. Moreover, we define a performance metric $M = c_{\text{MFMC}}/c_{\text{o}}$, where $c_{\text{MFMC}}$ is the daily cost of the MFMC method and $c_{\text{o}}$ is the daily cost of the optimal model-based controller. If $M$ equals to 1, the performance of the MFMC controller is equal to the performance of the optimal model-based controller.

The results of an experiment, spanning 100 days, are depicted in Fig. 4. The experiment starts with an empty batch and the tuples of the current day are added to the batch at the end of each day. The top left plot depicts the daily metric $M$ of our MFMC method, where the metric of the optimal model-based controller corresponds to 1.

The right top plot indicates the absolute value of the daily imbalance between the day-ahead consumption plan and the actual followed consumption. This plot demonstrates that the daily imbalance decreases as number of observations (days) in $\mathcal{F}$ increases. The mean metric $M$ of the MFMC method over the whole simulation period, including the exploration phase is 0.81. Furthermore, the mean deviation relative to the consumed energy of the MFMC method over the whole simulation period corresponded to $4,6\%$. Since $\alpha$ was set to $10^3$, the solution of the optimal model-based controller resulted in a deviation of $0\%$. This implies that for the optimal model-based controller, the forward consumption plan and actual consumption are identical.

A representative result of a single day, obtained with a mature MFMC controller, is depicted in the middle and bottom plot of Fig. 4. As can be seen in the figure, the MFMC method minimizes its day-ahead cost and follows its day-head schedule during the day. These results demonstrate that the model-free Monte Carlo method can be successfully used to construct a forward consumption plan for the next day.

## VI. Conclusion

Driven by the challenges presented by the system identification step of model-based controllers, this paper has contributed to the application of model-free batch Reinforcement Learning (RL) to demand response. Motivated by the fact that some demand response applications, e.g., a heat-pump thermostat, are influenced by exogenous weather data, we have adapted a well-known batch RL technique, fitted Q-iteration, to incorporate a forecast of the exogenous data. The presented empirical results have indicated that the proposed extension of fitted Q-iteration was able to improve the performance of standard fitted Q-iteration by 27%. In general, batch RL techniques do not require any prior knowledge about the system behavior or the solution. However, for some demand response applications, expert knowledge about the monotonicity of the solution, i.e., the policy, can be available. Therefore, we have presented an expert policy adjustment method that can exploit this expert knowledge. The results of an experiment with an electric water heater have indicated that the policy adjustment method was able to reduce the cost objective by 11% compared to fitted Q-iteration without expert knowledge. A final challenge for model-free batch RL techniques is that of finding a consumption plan for the next day, i.e., an open-loop solution. In order to solve this problem, we have presented a model-free Monte Carlo method that uses a distance metric based on the Q-function. In a final experiment, spanning 100 days, we have successfully tested this method to find the day-ahead consumption plan of a residential heat pump.

Our future research in this area will focus on employing the presented algorithms in a realistic lab environment. We are currently testing the expert policy adjustment method on a converted electric water heater and an air conditioning unit with promising results. The preliminary findings of the lab experiments indicate that the expert policy adjustment and extended fitted Q-iteration can be successfully used in a real-world demand response setting.

## Appendix
## Convergence Properties and Comparison With Q-Learning

This section discusses the convergence properties of the fitted Q-iteration algorithm and the model-free Monte Carlo method. In addition, it provides an empirical comparison between a well-known reinforcement learning method, i.e., Q-learning, and fitted Q-iteration.

*Fitted Q-Iteration:* In this paper, we build upon the theoretical work of [16], [23], and [24], and we demonstrate how fitted Q-iteration can be adapted to work in a demand response setting. In [16], Ernst *et al.*, starting from [24], show that when a kernel-based regressor is used fitted Q-iteration, converges to an optimal policy. For the finite horizon case, they show that the solution of the $T$-stage optimal control problem yields a policy that approximates the true optimal policy. As the ensemble of extremely randomized trees [23], used in the current work, readjusts its approximator architecture to the output variables, it is not possible to guarantee convergence. However, empirically we observed that they perform better than a frozen set of trees.

An empirical evaluation of fitted Q-iteration with an ensemble of extremely randomized trees is presented in Fig. 5. In this

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

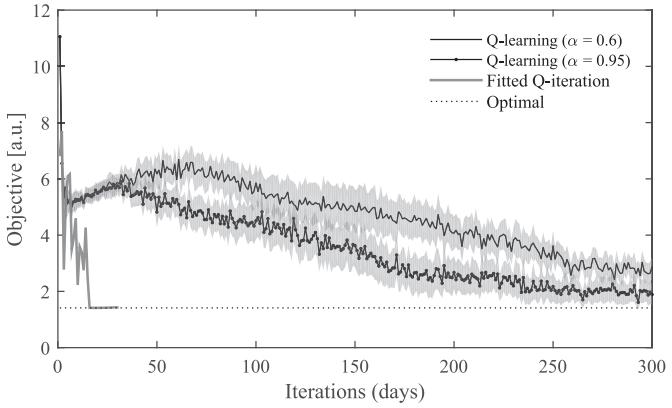


Fig. 5. Mean daily cost and standard deviation of 100 simulation runs with Q-learning for different learning rates $\alpha$, fitted Q-iteration and model-based solution (Optimal) during 300 iterations.

experiment, we implemented Q-learning [26] and we applied it to a heat-pump thermostat with a time-varying price profile as described in Section V. The intent of the experiment is to compare fitted Q-iteration with two other approaches namely Q-learning and the optimal solution. Since Q-learning discards the given tuple after each observation and assumes no model of the reward and the exogenous data is provided, we repeated the same day, i.e., identical price profile, disturbances and outside temperature, during 300 iterations. Specifically, Fig. 5 shows the mean daily cost and standard deviation of 100 simulation runs with Q-learning for different learning rates. Note that each simulation run uses a different random seed during the exploration step. As seen in Fig. 5, fitted Q-iteration converges to a near-optimal solution within 25 days. The solution of the model-based controller was found by solving the corresponding mixed integer problem with a commercial solver [44]. The model-based controller knows the exact future disturbance at the start of the optimization horizon. These empirical results indicate that the fitted Q-iteration algorithm clearly outperforms the standard Q-learning algorithm for the given demand response problem.

*Model-Free Monte-Carlo:* This paper uses a model-free Monte-Carlo (MC) method to construct a day-ahead consumption schedule of a flexible load based on the Q-function obtained from fitted Q-iteration. Specifically in [45], Fonteneau *et al.* infer bounds on the returns of the model-free MC method given a batch of tuples. They demonstrate that the lower and upper bounds converge at least linearly towards the true return of the policy, when the size of the batch grows, i.e., new tuples are added to the batch.

## ACKNOWLEDGMENT

The authors would like to thank Karel Macek of Honeywell and the reinforcement learning team from the University of Liège for their comments and suggestions.

## REFERENCES

[1] B. Dupont *et al.*, "LINEAR breakthrough project: Large-scale implementation of smart grid technologies in distribution grids," in *Proc. 3rd IEEE PES Innov. Smart Grid Technol. Conf. (ISGT Europe)*, Berlin, Germany, Oct. 2012, pp. 1–8.

[2] U.S. Energy Information Administration (EIA), "Annual energy outlook 2014 with projections to 2040," U.S. Dept. Energy, Washington, DC, USA, Tech. Rep. DOE/EIA-0383(2014), Apr. 2015. [Online]. Available: http://www.eia.gov/forecasts/aeo/pdf/0383(2014).pdf, accessed Mar. 21, 2015.

[3] S. Koch, J. L. Mathieu, and D. S. Callaway, "Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services," in *Proc. 17th IEEE Power Syst. Comput. Conf. (PSCC)*, Stockholm, Sweden, Aug. 2011, pp. 1–7.

[4] E. C. Kara, M. Berges, B. Krogh, and S. Kar, "Using smart devices for system-level management and control in the smart grid: A reinforcement learning framework," in *Proc. 3rd IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Tainan, Taiwan, Nov. 2012, pp. 85–90.

[5] F. Ruelens *et al.*, "Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning," in *Proc. 18th IEEE Power Syst. Comput. Conf. (PSCC)*, Wrocław, Poland, Aug. 2014, pp. 1–8.

[6] J. Tastu, P. Pinson, P.-J. Trombe, and H. Madsen, "Probabilistic forecasts of wind power generation accounting for geographically dispersed information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 480–489, Jan. 2014.

[7] E. Peeters, D. Six, M. Hommelberg, R. Belhomme, and F. Bouffard, "The ADDRESS project: An architecture and markets to enable active demand," in *Proc. 6th IEEE Int. Conf. Eur. Energy Market (EEM)*, Leuven, Belgium, May 2009, pp. 1–5.

[8] F. Bliek *et al.*, "PowerMatching city, a living lab smart grid demonstration," in *Proc. 1st IEEE PES Innov. Smart Grid Technol. Conf. (ISGT Europe)*, Gothenburg, Sweden, Oct. 2010, pp. 1–8.

[9] A. Molderink, V. Bakker, M. G. C. Bosman, J. L. Hurink, and G. J. M. Smit, "Management and control of domestic smart grid technology," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 109–119, Sep. 2010.

[10] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Gaithersburg, MD, USA, Oct. 2010, pp. 409–414.

[11] S. Vandael, B. Claessens, M. Hommelberg, T. Holvoet, and G. Deconinck, "A scalable three-step approach for demand side management of plug-in hybrid vehicles," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 720–728, Jun. 2013.

[12] J. Cigler, D. Gyalistras, J. Širokỳ, V.-N. Tiet, and L. Ferkl, "Beyond theory: The challenge of implementing model predictive control in buildings," in *Proc. 11th REHVA World Congr. (CLIMA)*, Prague, Czech Republic, 2013, pp. 1008–1018.

[13] J. L. Mathieu and D. S. Callaway, "State estimation and control of heterogeneous thermostatically controlled loads for load following," in *Proc. 45th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Maui, HI, USA, Jan. 2012, pp. 2002–2011.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.

[16] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 517–529, Apr. 2009.

[17] G. P. Henze and J. Schoenmann, "Evaluation of reinforcement learning control for thermal energy storage systems," *HVAC&R Res.*, vol. 9, no. 3, pp. 259–275, 2003.

[18] D. Lee and W. B. Powell, "An intelligent battery controller using bias-corrected Q-learning," in *Association for the Advancement of Artificial Intelligence*, J. Hoffmann and B. Selman, Eds. Palo Alto, CA, USA: AAAI Press, 2012.

[19] Y. Liang, L. He, X. Cao, and Z.-J. Shen, "Stochastic control for smart grid users with flexible demand," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2296–2308, Dec. 2013.

[20] H. V. Hasselt, "Double Q-learning," in *Proc. 24th Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2010, pp. 2613–2621. [Online]. Available: http://papers.nips.cc/paper/3964-double-q-learning.pdf

[21] R. Fonteneau, S. A. Murphy, L. Wehenkel, and D. Ernst, "Batch mode reinforcement learning based on the synthesis of artificial trajectories," *Ann. Oper. Res.*, vol. 208, no. 1, pp. 383–416, 2013.

[22] S. Adam, L. Busoniu, and R. Babuška, "Experience replay for real-time reinforcement learning control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 201–212, Mar. 2012.

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.

[23] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *J. Mach. Learn. Res.*, vol. 6, pp. 503–556, Apr. 2005.
[24] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, nos. 2–3, pp. 161–178, 2002.
[25] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement Learning: State-of-the-Art*, M. Wiering and M. van Otterlo, Eds. New York, NY, USA: Springer, 2012, pp. 45–73.
[26] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[27] G. A. Rummery and M. Niranjan, *On-Line Q-Learning Using Connectionists Systems*. Cambridge, U.K.: Cambridge Univ., 1994.
[28] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," Yahoo Lab., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2015. [Online]. Available: http://web.stanford.edu/class/ee292k/reports/ZhengWen.pdf
[29] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, Jul. 2015.
[30] L. Busoniu, R. Babuška, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: CRC Press, 2010.
[31] K. Vanthournout, R. D'hulst, D. Geysen, and G. Jacobs, "A smart domestic hot water buffer," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 2121–2127, Dec. 2012.
[32] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," in *Proc. IEEE Transm. Distrib. Conf. Expo.*, Chicago, IL, USA, Apr. 2008, pp. 1–5.
[33] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large scale dynamic programming," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 59–94, 1996.
[34] W. Curran, T. Brys, M. Taylor, and W. Smart, "Using PCA to efficiently represent state spaces," in *Proc. 12th Eur. Workshop Reinforcement Learn. (EWRL)*, Lille, France, 2015, pp. 1–8.
[35] *Belpex—Belgian Power Exchange*. [Online]. Available: http://www.belpex.be/, accessed Mar. 21, 2015.
[36] R. Bellman, *Dynamic Programming*. New York, NY, USA: Dover, 2013.
[37] D. P. Bertsekas, "Approximate dynamic programming," in *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2011, pp. 324–552.
[38] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.
[39] L. Busoniu, D. Ernst, R. Babuška, and B. De Schutter, "Exploiting policy knowledge in online least-squares policy iteration: An empirical study," *Autom. Comput. Appl. Math.*, vol. 19, no. 4, pp. 521–529, 2010.
[40] M. Riedmiller, "Neural fitted Q iteration—First experiences with a data efficient neural reinforcement learning method," in *Proc. 16th Eur. Conf. Mach. Learn. (ECML)*, vol. 3720. Porto, Portugal, Oct. 2005, pp. 317–328.
[41] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[42] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Barcelona, Spain, Jul. 2010, pp. 1–8.
[43] U. Jordan and K. Vajen, "Realistic domestic hot-water profiles in different time scales: Report for the international energy agency, solar heating and cooling task (IEA-SHC)," Institut für Thermische Energietechnik, Universität Marburg, Marburg, Germany, Tech. Rep., 2001.
[44] Gurobi Optimization. *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com/, accessed Mar. 21, 2015.
[45] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst, "Inferring bounds on the performance of a control policy from a sample of trajectories," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Nashville, TN, USA, Mar. 2009, pp. 117–123.

**Frederik Ruelens** received the M.Sc. degree (*cum laude*) in electrical engineering from Katholieke Universiteit Leuven, Belgium, in 2010. Since 2011, his Ph.D. research on demand response using reinforcement learning has been supported by an IWT Flanders scholarship. His research interests include machine learning, data science, reinforcement learning, and power systems.

**Bert J. Claessens** was born in Neeroeteren, Belgium. He received the M.Sc. and Ph.D. degrees in applied physics from the University of Technology of Eindhoven, The Netherlands, in 2002 and 2006, respectively. In 2006, he started working at ASML Veldhoven, The Netherlands, as a Design Engineer. Since 2010, he has been a Researcher with the Vlaamse Instelling voor Technologisch Onderzoek, Mol, Belgium. His research interests include algorithm development and data analysis.

**Stijn Vandael** was born in Hasselt, Belgium. He received the M.Sc. degree from Katholieke Universiteit Leuven (KU Leuven), Belgium, and the M.S. degree in industrial engineering from Katholieke Hogeschool Limburg, Belgium. He is currently pursuing the Ph.D. degree with the Department of Computer Science, KU Leuven, in close cooperation with the Department of Electrical Engineering. His research interests include coordination in multiagent systems, plug-in hybrid electric vehicles, and smart grids.

**Bart De Schutter** (M'08–SM'10) received the M.Sc. degree in electrotechnical-mechanical engineering and the Ph.D. degree (*summa cum laude* with congratulations of the examination jury) in applied sciences from Katholieke Universiteit Leuven, Leuven, Belgium, in 1991 and 1996, respectively. He is currently a Full Professor with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. His current research interests include control of intelligent transportation systems, hybrid systems control, discrete-event systems, and multiagent multilevel control. He is an Associate Editor of *Automatica* and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

**Robert Babuška** received the M.Sc. degree (*cum laude*) in control engineering from Czech Technical University, Prague, in 1990, and the Ph.D. degree (*cum laude*) from the Delft University of Technology, the Netherlands, in 1997. He has had faculty appointments with the Technical Cybernetics Department, Czech Technical University, and the Electrical Engineering Faculty of the Delft University of Technology. He is currently a Professor of Intelligent Control and Robotics with the Delft Center for Systems and Control. His research interests include reinforcement learning, neural and fuzzy systems, identification and state-estimation, nonlinear model-based and adaptive control, and dynamic multiagent systems. He has been working on applications of these techniques in the fields of robotics, mechatronics, and aerospace.

**Ronnie Belmans** (S'77–M'84–SM'89–F'05) received the M.S. degree in electrical engineering and the Ph.D. degree from the University of Leuven (Katholieke Universiteit Leuven), Heverlee, Belgium, in 1979 and 1984, respectively, and the Special Doctorate degree from RWTH, Aachen, Germany, in 1989 and 1993, respectively. He is currently a Full Professor with the Katholieke Universiteit Leuven, teaching electric power and energy systems. He is also a Guest Professor with the Imperial College of Science, Medicine, and Technology, London, U.K. His research interests include techno-economic aspects of power systems, power quality, and distributed generation. He is the Honorary Chairman of the Board of Elia, which is the Belgian transmission grid operator. He is also the President of the International Union for Electricity Applications. He is a Fellow of the Institute of Electrical Engineers, U.K.

Authorized licensed use limited to: San Jose State University. Downloaded on March 27,2023 at 20:58:12 UTC from IEEE Xplore. Restrictions apply.