Electric Power Systems Research 189 (2020) 106780

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Data-Driven Learning and Load Ensemble Control

Ali Hassan[a], Deepjyoti Deka[b], Michael Chertkov[c], Yury Dvorkin[*,a]

[a] Department of Electrical and Computer Engineering Tandon School of Engineering, New York University, New York, NY, USA
[b] Theory Division (T-5) Los Alamos National Laboratory, Los Alamos, NM, USA
[c] Applied Mathematics University of Arizona, Tucson, Arizona, USA

ARTICLE INFO

Keywords:
Markov Decision Process
Thermostatically Controlled Loads
Z-learning
Linearly Solvable MDP
TCL ensemble

ABSTRACT

Demand response (DR) programs aim to engage distributed small-scale flexible loads, such as thermostatically controllable loads (TCLs), to provide various grid support services. Linearly Solvable Markov Decision Process (LS-MDP), a variant of the traditional MDP, is used to model aggregated TCLs. Then, a model-free reinforcement learning technique called Z-learning is applied to learn the value function and derive the optimal policy for the DR aggregator to control TCLs. The learning process is robust against uncertainty that arises from estimating the passive dynamics of the aggregated TCLs. The efficiency of this data-driven learning is demonstrated through simulations on Heating, Cooling & Ventilation (HVAC) units in a testbed neighborhood of residential houses.

1. Introduction

Distribution grids are undergoing a rapid transition due to the massive deployment of distributed energy resources (DERs), e.g., PV arrays, electric vehicles, and energy storage units. The main factors fueling this expansion include significant decreases in the capital costs of DER technologies and incentives for DER installations offered by local electric power utilities, as well as by local and state authorities. For example, the state of California aims to reduce greenhouse gas emissions (GHG) by 40% below its 1990 levels in 2030 by means of increasing the share of electricity produced by renewable generation to 50%, doubling energy efficiency targets, and encouraging widespread transportation electrification [1]. Similarly, the state of NY set a target of zero-carbon power sector by 2040, along with the goal of reducing the 1990 levels of GHG emissions by 85% in 2050 [2]. On the other hand, the presence of DERs in distribution grids also imposes additional operational challenges, e.g. bidirectional power flows, voltage fluctuations, and, as a result, additional wear-and-tear on electric power equipment. Dealing with such challenges is crucial to ensure economic and reliable distribution grid operations and necessitates more flexibility. Demand Response (DR) is one way to provide this additional flexibility, which enrolls controllable loads in residential and commercial buildings to provide a broad range of distribution-level ancillary services (e.g. energy arbitrage, peak shaving, balancing regulation, congestion relief, capacity deferral, voltage support, [3]). Our efforts to explore this source of flexibility is motivated by the recent statistics that the U.S. building sector claims about 40% of the total electricity consumption [4] and still remains, to a large extent, unleveraged for distribution grid operations. The primary obstacle is in the current inability to accurately aggregate and synchronously operate a large ensemble of such small-scale loads, while taking into account their inherent techno- and socio-economic characteristics (e.g., dispatch limits, complex thermodynamics of building environments, and/or comfort preferences of building occupants). Therefore, to address these challenges, this paper focuses on mathematical modeling of an ensemble of thermostatically controlled loads (TCL), such as heat pumps, air conditioners, heating and ventilation systems, for its accurate representation in energy management (dispatch) tools used by DR aggregators or local electric power utilities, [3,5].

The primary challenge in modeling TCL ensembles is to simultaneously achieve a high level of accuracy and maintain computational tractability. Currently, there are two large groups of methods to model and forecast electricity consumption of TCL ensembles: (i) physics-based co-simulation of TCLs and building dynamics (e.g. using heat transport models, electromechanical considerations, Kirchoff's laws, evaporation, etc) and (ii) data-driven (e.g. statistical analyses and inference). The advantage of using the physics-based models is in their ability to describe buildings without prior observations. However, the performance of these models is highly sensitive to the number and accuracy of the underlying modeling choices and assumptions, as well as to input parameters. Physics-based models often require more inputs than existing data acquisition systems can provide [6], and therefore incur significant uncertainties in both model parameters and dynamic processes. Using such models for controlling an ensemble of TCLs may

* Corresponding author.
E-mail addresses: ah3909@nyu.edu (A. Hassan), deepjyoti@lanl.gov (D. Deka), chertkov@math.arizona.edu (M. Chertkov), dvorkin@nyu.edu (Y. Dvorkin).

https://doi.org/10.1016/j.epsr.2020.106780
Received 4 October 2019; Received in revised form 19 April 2020; Accepted 2 August 2020
Available online 13 October 2020
0378-7796/

lead to computational issues that would prevent their scalability and implementation for real-life decision-making. On the other hand, in lieu of the physics-based models, one can use machine learning and statistical modeling to perform data-driven studies of TCL and building dynamics using a vast amount of historical data available at the buildings equipped with smart meters. These reduced order models are trained using the historical energy consumption data and other parameters (e.g. weather conditions, daily operational schedules, and control functionality) [7,8]. This paper develops a data-driven model to accurately represent a TCL ensemble using historical data and to continuously improve the accuracy of model performance via learning.

Among data-driven methods, TCL ensembles have been modelled as virtual storage units with linear dynamics, [9–11], or as a Markov Decision Process (MDP) with probabilistic transitions, [12–17]. The MDP framework is particularly suitable for modeling large TCL ensembles, without sacrificing modeling accuracy or computational tractability. Thus, it produces high-quality solutions by means of using dynamic programming, which are both analytically and computationally tractable. The models in [12–17] model a TCL ensemble as a discrete-time, discrete-space Markov Process characterized by a given transition probability matrix with deterministic coefficients. However, in practice, it is hardly possible to estimate these coefficients accurately due to the imperfection or incompleteness of historical measurements and behavioral uncertainty of consumers. Therefore, the common caveat of current MDP models in [12–17] is that they ignore uncertainty on model parameters (e.g. transition probabilities). Since the inaccuracies stemming from the inability to compute model parameters in the MDP framework can be significant and can eliminate the benefits of using these resources for DR flexibility, this paper enhances the MDP framework with model-free reinforcement learning (RL), where the DR aggregator[1] interacts with the TCL ensemble and learns model parameters from both historical and streaming data (see Fig. 1). The main advantage of the model-free RL in the context of dispatch TCLs is in its ability to eliminate the need for knowing precise model parameters (e.g. parameters of the transition probability distribution underlying the MDP) because the optimal control policy can be learned from "experience". In the context of real-life DR applications, this "experience" can be obtained via indirect (passive) observations of the TCL ensemble or, in some cases, even individual TCLs by means of using advanced metering infrastructure or data crowdsourcing, [18].

Although there is a number of model-free RL techniques that can be used under the MDP framework, we exploit the property of TCL ensembles that allow for reducing a conventional MDP to a linearly-solvable MDP (LS-MDP). This reduction assumes that devices in the TCL ensemble are relatively heterogeneous and, therefore, explicit control actions on each TCL device (e.g. on/off decisions or power consumption) can be replaced by a distribution of potential future states of the TCL ensemble, [19–21]. Thus, the optimal policy derived from the LS-MDP is not a mapping of states to action variables, as in a conventional MDP, but is a mapping of a current state into a next-state distribution, which minimizes the expected next-state costs and the divergence cost between the default (e.g., without external control applied) and controlled (e.g. with external control applied) probability distributions [21,22]. The reduced LS-MDP problem is suitable for the Z-learning method, which is a modification of the common Q-learning method. In turn, the Z-learning method is capable of producing an accurate approximation of the original MDP at a faster convergence rate than the Q-learning method, [19–22], mainly because Z-learning does not require state-action values as needed in Q-learning.

This paper uses a LS-MDP to model a TCL ensemble and leverage the Z-learning method to find the optimal TCL dispatch policy. The Z-learning method samples transitions passively from the default (e.g. without external control) behavior of the system, but is able to learn the optimal policy by leveraging the specific structure of LS-MDP. Note that the available state transitions may not accurately reflect the underlying true distribution due to limited availability of data. Hence, we show that the Z-learning algorithm is robust to noise in the observed transitions and analyze its convergence in cases with and without noise. The case study is carried out on aggregated heating, ventilation, and air conditioning (HVAC) systems in a residential neighborhood with 100 homes, where data is sampled using the Net-Zero Energy Test Facility [23], operated by the National Institute of Standards and Technology (NIST).

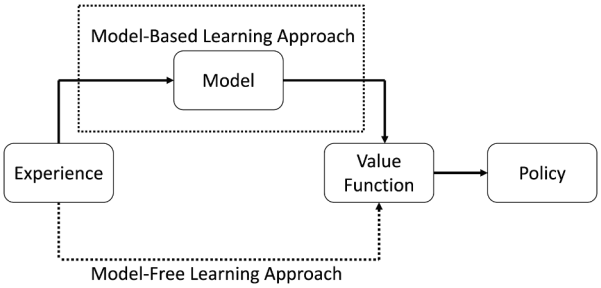

Fig. 1. Comparison between the model-based and model-free learning approaches.

The remainder of this paper is organized as follows. Section 2 presents a LS-MDP model for optimally dispatching a given TCL ensemble. Section 3 solves the LS-MDP model using dynamic programming and leverages the Z-learning approach to improve the solution accuracy. Section 4 presents the case study using real-life data from the NIST Test Facility to demonstrate the usefulness of the proposed approach. Section 5 concludes the paper.

2. Formulation

Similarly to [14–17], the MDP framework is leveraged to build the model for the control of the TCL ensemble. We define a LS-MDP for modeling a given TCL ensemble as a 5-tuple $\{\mathcal{T}, \mathcal{A}, U_t^\beta, \mathcal{P}_t^{\alpha\beta}, \overline{\mathcal{P}}^{\alpha\beta}\}$, where $\mathcal{T}$ is the set of time intervals, which constitute a planning horizon, $\mathcal{A}$ is the set of possible states, U_t^β is the utility of the aggregator in state $\beta \in \mathcal{A}$ at time $t \in \mathcal{T}$, $\overline{\mathcal{P}}^{\alpha\beta}$ and $\mathcal{P}_t^{\alpha\beta}$ are default (i.e. without control actions of the DR aggregator) and controlled (with control actions of the DR aggregator) transition probabilities from state $\beta \in \mathcal{A}$ to $\alpha \in \mathcal{A}$. The states in set $\mathcal{A} = \{\alpha, \beta, ...\}$ are obtained by discretizing the range of power consumption for each TCL ensemble given the operating range of TCL devices in the ensemble. For any given state $\beta \in \mathcal{A}$ at time $t \in \mathcal{T}$, the probability of the transition of the TCL ensemble to the next state $\alpha \in \mathcal{A}$ at time $t+1 \in \mathcal{T}$ is characterized by $\mathcal{P}_t^{\alpha\beta}$. Fig. 2 displays all possible transitions from the current state β at time t to all possible next states α at time $t+1$. Note that the ensemble can remain in the same state β at time $t+1$ such that $\alpha = \beta$. The default transition probabilities, represented by parameter $\overline{\mathcal{P}}^{\alpha\beta}$, corresponds to the internal dynamics of the TCL ensemble without actions of the aggregator and are typically estimated from historical data (see [15]). The TCL ensemble is then optimized as:

$$\min_{\rho,\mathcal{P}} \mathbb{E}_\rho \sum_{t\in\mathcal{T}-1} \sum_{\alpha\in\mathcal{A}} \left(-U_{t+1}^\alpha + \sum_{\beta\in\mathcal{A}} \gamma \log \frac{\mathcal{P}_t^{\alpha\beta}}{\overline{\mathcal{P}}^{\alpha\beta}} \right) \tag{1a}$$

$$\rho_{t+1}^\alpha = \sum_{\beta\in\mathcal{A}} \mathcal{P}_t^{\alpha\beta} \rho_t^\beta, \quad \forall\, \alpha \in \mathcal{A}, t \in \mathcal{T} \tag{1b}$$

$$\sum_{\alpha\in\mathcal{A}} \mathcal{P}_t^{\alpha\beta} = 1, \quad \forall\, \beta \in \mathcal{A}, t \in \mathcal{T}-1, \tag{1c}$$

where $\rho_{t+1}^\alpha \geq 0$ and $\rho_t^\beta \geq 0$ are decision variables, which characterize the probability that the TCL ensemble is operated in states α and β at time $t+1$ and t, and are related via transition probabilities $\mathcal{P}_t^{\alpha\beta}$.

[1] Alternatively, TCL ensembles can be aggregated and operated by utilities.

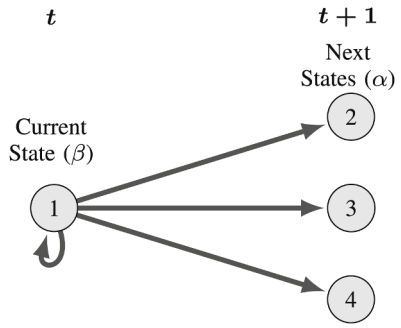

Fig. 2. A schematic representation of the Markov Process displaying transitions from a current state (β) at time t to the possible future next states (α) at time $t+1$. Note that the ensemble can remain in the same state β at time $t+1$ such that $\alpha = \beta$.

Eq. (1a) represents the objective function of the DR aggregator that controls the TCL ensemble and aims to maximize its expected utility or minimize its expected cost of energy ($-U^{\alpha}_{t+1}$) and to minimize the discomfort cost for the TCL ensemble. The discomfort cost is computed using the Kullback-Leibler (KL) divergence, weighted by parameter γ. This divergence penalizes the difference between the transition decisions made by the DR aggregator ($\mathcal{P}^{\alpha\beta}_t$) and the default transitions of the TCL ensemble ($\overline{\mathcal{P}}^{\alpha\beta}$), under the assumption that the latter represents first-choice preferences of TCL users. Parameter γ can influence the KL divergence and thus encourage or discourage deviations from the default behavior of the TCL ensemble. The choice of the KL divergence for the penalty cost is motivated by its extensive use for modeling randomness of discrete and continuous time-series [24]. Eq. (1b) describes the temporal evolution of the TCL ensemble from time t to $t+1$ over time horizon $\mathcal{T}$. Eq. (1c) imposes the integrality constraint on the transition decisions optimized by the DR aggregator such that their total probability is equal to one.

After solving (1) as described later in Section 3.1, the active power (p_t) consumed by the TCL ensemble can be computed using optimized decisions ρ^{β}_t and rated active power $p^{\beta,rated}$ at each state, e.g. $p_t = \sum_{\beta\in\mathcal{A}} p^{\beta,rated}\rho^{\beta}_t$, $\forall\, t \in \mathcal{T}$.

2.1. Relation to Other Methods

The LS-MDP in (1) can be related to linear dynamical TCL models in other data-driven methods, [9,10]. Consider the following linear dynamics for the TCL ensemble, [11]:

$$S_{t+1} = S_t + (u_t + P_t)\Delta t, \tag{2}$$

where S_t is the energy state of the TCL ensemble, P_t is the normal power consumed and u_t is the power change sought by control actions. Let $P_t \sim N(\mu_{P_t}, \sigma_{P_t})$ and consider the KL divergence between S^0_{t+1} (without control) and S_{t+1} (with control). Using [25] leads to:

$$KL(S^0_{t+1}\|S_{t+1}) = \frac{u^2(t)}{2\sigma^2_P\Delta^2 t}, \tag{3}$$

which is the quadratic cost for control used in linear systems, i.e. the quadratic discomfort cost for control in linear dynamics with Gaussian uncertainties is equivalent to the discrete-time KL cost in the LS-MDP. However, the discrete nature of LS-MDP transitions simplifies modeling, even for complex state transitions and non-Gaussian uncertainties.

3. Z-learning in LS-MDP

3.1. Solving LS-MDP

The optimization in Eq. (1) is a Linearly Solvable MDP (LS-MDP) as introduced by [19]. The optimal policy for Eq. (1) is computed using techniques from dynamic programming [26]. The Bellman equation for the LS-MDP in (4) can be derived from the Bellman equation for the traditional MDP explained in Appendix A and leads to:

$$\frac{1}{\gamma}\varphi^{\beta}_t = \frac{1}{\gamma}\min_{\mathcal{P}}\left(-U^{\beta}_t + \mathbb{E}_{\mathcal{P}^{\alpha\beta}_t}\left[\gamma\log\frac{\mathcal{P}^{\alpha\beta}_t}{\overline{\mathcal{P}}^{\alpha\beta}} + \varphi^{\alpha}_{t+1}\right]\right), \tag{4}$$

where φ^{β}_t is the value function of the TCL ensemble at present state β at time t and φ^{α}_{t+1} is the value function at next state α at time $t+1$. By introducing desirability function $z^{\beta}_t = \exp(\frac{-\varphi^{\beta}_t}{\gamma})$ in (4) we obtain:

$$\begin{aligned} -\log(z^{\beta}_t) &= \frac{1}{\gamma}\min_{\mathcal{P}}\left(-U^{\beta}_t + \gamma\mathbb{E}_{\mathcal{P}^{\alpha\beta}_t}\left[\log\frac{\mathcal{P}^{\alpha\beta}_t}{\overline{\mathcal{P}}^{\alpha\beta}} - \log(z^{\alpha}_{t+1})\right]\right) \\ &= \frac{1}{\gamma}\min_{\mathcal{P}}\left(-U^{\beta}_t + \gamma\mathbb{E}_{\mathcal{P}^{\alpha\beta}_t}\left[\log\frac{\mathcal{P}^{\alpha\beta}_t}{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}\right]\right) \end{aligned} \tag{5}$$

After introducing a normalization term defined as $\mathcal{G}^{\beta}_t(z) = \sum_{\alpha}\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}$, (5) can be recast as:

$$\begin{aligned} -\log(z^{\beta}_t) &= \frac{1}{\gamma}\min_{\mathcal{P}}\left(-U^{\beta}_t + \gamma\mathbb{E}_{\mathcal{P}^{\alpha\beta}_t}\left[\log\frac{\mathcal{P}^{\alpha\beta}_t\mathcal{G}^{\beta}_t(z)}{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}\mathcal{G}^{\beta}_t(z)}\right]\right) \\ &= \left(\frac{-U^{\beta}_t}{\gamma} + \min_{\mathcal{P}}KL\left[\mathcal{P}^{\alpha\beta}_t\middle\|\frac{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}{\mathcal{G}^{\beta}_t(z)}\right] - \log\mathcal{G}^{\beta}_t(z)\right) \end{aligned} \tag{6}$$

The KL divergence provides the expectation of the log-difference between the two distributions such that $KL[p_1\|p_2] = \mathbb{E}_{p_1}[\log\frac{p_1}{p_2}]$. It is zero if and only if the two distributions are same. Therefore, it follows from Eq. (6) that the optimal policy is achieved when the KL divergence term in Eq. (6) is minimal, i.e. it is equal to zero. Hence, by equating the two distributions in the KL divergence, the optimal policy follows as:

$$\mathcal{P}^{\alpha\beta}_t = \frac{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}{\mathcal{G}^{\beta}_t(z)} = \frac{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}{\sum_{\alpha}\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}, \tag{7}$$

The optimal policy in Eq. (7) depends on the uncontrolled transition probability ($\overline{\mathcal{P}}^{\alpha\beta}$) and the desirability function of the TCL ensemble at the next state (z^{α}_{t+1}). The optimal policy reduces the Bellman equation in (6) to the following form:

$$-\log(z^{\beta}_t) = \{\frac{-U^{\beta}_t}{\gamma} - \log\mathcal{G}^{\beta}_t(z)\} \tag{8}$$

$$\log(z^{\beta}_t) = \left\{\frac{U^{\beta}_t}{\gamma} + \log\left[\sum_{\alpha}\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}\right]\right\} \tag{9}$$

Exponentiating Eq. (9) converts the Bellman equation to the following reduced form:

1: Initialize $z^{\beta}_t = 1\ \forall\beta\in\mathcal{A}, t\in\mathcal{T}$
2: $z^{\beta}_{|\mathcal{T}|} = \exp\left(\frac{U^{\beta}_{|\mathcal{T}|}}{\gamma}\right)$, where $|\mathcal{T}|$:=final time
3: Set $t = |\mathcal{T}-1|$
4: **for** $t \leftarrow |\mathcal{T}-1|$ **to** 1 **do**
5: $\quad z^{\beta}_t = \exp\left(\frac{U^{\beta}_t}{\gamma}\right)\sum_{\alpha}\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}$
6: **end for**
7: **for** all $t\in\mathcal{T}-1$ **do**
8: $\quad$ **for** all $\beta\in\mathcal{A}$ **do**
9: $\quad\quad \mathcal{P}^{\alpha\beta}_t = \frac{\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}{\sum_{\alpha}\overline{\mathcal{P}}^{\alpha\beta}z^{\alpha}_{t+1}}$
10: $\quad$ **end for**
11: **end for**

Algorithm 1. Solving a LS-MDP

1: Initialize $z_t^\beta = 1\ \forall \beta \in \mathcal{A}, t \in \mathcal{T}$
2: $z_{|\mathcal{T}|}^\beta = \exp\left(\frac{U_{|\mathcal{T}|}^\beta}{\gamma}\right)$, where $|\mathcal{T}|$:=final time
3: **repeat**
4: Set k = current sample at state α from passive dynamics $\overline{\mathcal{P}}$
5: Starting with time $t = |\mathcal{T}-1|$
6: **for** $t \leftarrow |\mathcal{T}-1|$ **to** 1 **do**
7: $\hat{z}_{t,k}^\beta \leftarrow (1-\eta_k)\hat{z}_{t,k-1}^\beta + \eta_k \exp\left(\frac{U_t^\beta}{\gamma}\right)\hat{z}_{t+1,k-1}^\alpha$
8: **end for**
9: **until** convergence

Algorithm 2. Z-learning

$$z_t^\beta = \exp\left(\frac{U_t^\beta}{\gamma}\right)\sum_\alpha \overline{\mathcal{P}}^{\alpha\beta} z_{t+1}^\alpha \tag{10}$$

Given the Bellman equation in (10) and optimal policy in (7), the LS-MDP is solved as described in Algorithm 1. Eq. (10) is linear and thus can be represented in a matrix form as $\boldsymbol{z}_t = \boldsymbol{U}_t \overline{\mathcal{P}} \boldsymbol{z}_{t+1}$, where $\boldsymbol{z}_t$ is a vector with elements z_t^β, $\overline{\mathcal{P}}$ is a matrix with entries $\overline{\mathcal{P}}^{\alpha\beta}$, and $\boldsymbol{U}_t$ is a diagonal matrix with elements $\exp\left(\frac{U_t^\beta}{\gamma}\right)$ along its main diagonal [19–22,27,28].

3.2. Z-learning

Although the LS-MDP solves the optimization problem for the TCL ensemble efficiently, it requires knowledge about the model of the environment. Since the model is estimated from the historical data (e.g. values of the default transitions in $\overline{\mathcal{P}}^{\alpha\beta}$), which is limited and imperfect, it may introduce inaccuracies. This motivates the use of model-free learning techniques to robustly solve the optimization problem in (1). Using Z-learning, a model-free learning method, returns stochastic approximations $\hat{z}$ of the optimal value function in Eq. (10). Thus, $\hat{z}$ is updated as

$$\hat{z}_{t,k}^\beta \leftarrow (1-\eta_k)\hat{z}_{t,k-1}^\beta + \eta_k \exp\left(\frac{U_t^\beta}{\gamma}\right)\hat{z}_{t+1,k-1}^\alpha \tag{11}$$

where η_k is a decaying learning rate and α is the state observed at sample k by transitioning from previous state β. Z-learning updates the value function at the present state based on the sample providing next-state information instead of averaging over all the future possible states as in the LS-MDP. Unlike in Q-learning, there is no optimization of actions during the iterations in Z-learning. Instead, the samples for Z-learning are passively collected from the underlying distribution discretized in $\overline{\mathcal{P}}^{\alpha\beta}$. Then, $\hat{z}_{t,k}^\beta$ are updated by using the specific KL divergence form of the optimal policy, which enables faster computations.

The proposed application of the Z-learning algorithm to dispatching TCLs is detailed in Algorithm 2. First, the algorithm is initialized with $z_t^\beta = 1$ for all states $\beta \in \mathcal{A}$ and time periods $t \in \mathcal{T}$. Next, it computes the desirability function for the final time $|\mathcal{T}|$. Then, it iteratively computes the desirability function for the remaining time intervals (from $t = 1$ to $t = |\mathcal{T}-1|$) using samples generated from the passive dynamics and updates the desirability function until a chosen convergence criterion is achieved. In this paper, the convergence criterion is defined as the difference between two successive values of the desirability function.

Note that the state transitions in the samples used in Z-learning may be corrupted by noise as well. The noise in the passive dynamics is modelled as the error term $\epsilon^{\alpha\beta} \in \mathbb{R}^{n\times n}$, where $n = |\mathcal{A}|$:

$$\overline{\mathcal{P}}^{\alpha\beta} = \overline{\mathcal{P}}^{\alpha\beta} + \epsilon^{\alpha\beta} \tag{12}$$

where $\epsilon^{\alpha\beta}$ can be modelled by a zero-mean, normal distribution with variance σ_n^2, i.e. $\epsilon^{\alpha\beta} \sim N(0, \sigma^2)$ (other parametric distributions are also suitable). To ensure that every row in the transition probability matrix remains equal to one i.e. $\sum_{\alpha\in\mathcal{A}} \overline{\mathcal{P}}^{\alpha\beta} = 1,\ \forall\ \beta \in \mathcal{A}$, every row in $\epsilon^{\alpha\beta}$ must be equal to zero, i.e. $\sum_{\alpha\in\mathcal{A}} \epsilon^{\alpha\beta} = 0,\ \forall\ \beta \in \mathcal{A}$[2]. $\overline{\mathcal{P}}^{\alpha\beta}$ can be extended to capture noise scenarios by defining a set of N probability distributions as $\overline{\mathcal{P}}_n^{\alpha\beta},\ \forall\ n \in [1, N]$, such that $\overline{\mathcal{P}}_n^{\alpha\beta}$ is characterized as:

$$\frac{1}{N}(\mathbb{E}[\overline{\mathcal{P}}_1^{\alpha\beta}] + \mathbb{E}[\overline{\mathcal{P}}_2^{\alpha\beta}] + \cdots + \mathbb{E}[\overline{\mathcal{P}}_N^{\alpha\beta}]) \approx \overline{\mathcal{P}}^{\alpha\beta}, \tag{13}$$

where Eq. (13) ensures that the expected value of all N distributions is close to the passive dynamics of the TCL ensemble given by $\overline{\mathcal{P}}^{\alpha\beta}$. At each Z-learning iteration, one out of N distributions is selected with probability to update the value function. Note that despite the noise in the transition probability matrix, the same Algorithm 2 for Z-learning is used and, as shown in Section 4, performs efficiently and robustly.

3.3. Convergence of Z-learning

The convergence of Z-learning can be assessed using the optimal LS-MDP policy in (7) by proving that the Z-update in (11) asymptotically converges to (10). Let $\Delta\hat{z}_{t,k}^\beta = z_t^\beta - z_{t,k}^\beta$ be the optimality at the k^{th} iteration of Z-learning. Using (10)-(11) leads to:

$$\Delta\hat{z}_{t,k}^\beta = (1-\eta_k)\Delta\hat{z}_{t,k-1}^\beta + \eta_k \exp\left(\frac{U_t^\beta}{\gamma}\right)\left(\mathbb{E}_{\overline{\mathcal{P}}}[z_{t+1}^\alpha] - \sum_\alpha \mathbb{1}_{\alpha_k=\alpha}\hat{z}_{t+1,k-1}^\alpha\right),$$

where the indicator function $\mathbb{1}$ is 1, if state α is observed in the k^{th} iteration, and 0 otherwise. Consider $t = |\mathcal{T}| - 1$, the final time-interval for updating z-values. $\hat{z}_\mathcal{T} = z_\mathcal{T}$ is of course directly determined using $\mathcal{U}_\mathcal{T}$. It is clear that $\mathbb{E}_{\overline{\mathcal{P}}}[z_\mathcal{T}^\alpha] - \sum_\alpha \mathbb{1}_{\alpha_k=\alpha} z_{\mathcal{T},k}^\alpha$ is a random variable with mean 0 and a finite variance. Then, if learning rates η_k are selected such that $\sum_k \eta_k = \infty$ and $\sum_k \eta_k^2 < \infty$, it follows that $\lim_k \Delta\hat{z}_{\mathcal{T}-1,k}^\beta \to 0$, see [30]. Following similarly for $t = |\mathcal{T}| - 2, \ldots, 1$, it returns $\lim_k \Delta\hat{z}_{t,k}^\beta \to 0$. Thus Z-update (11) converges to the solution of (10). Note that the convergence also holds if a finite variance noise is allowed in the transition probability matrix (see Eq. (12)).

4. Case Study

4.1. Data

We use data from the Net-Zero Energy Test Facility, [23], which is a single-family, three-floor, net-zero-energy house, with the total area of 386 (4156) m^2 (ft^2), located in Gaithersburg, MD. To create an ensemble, this case study considers a neighbourhood with 100 houses with parameters and historical data obtained based on adding random noise to the data obtained from the Net-Zero Energy Test Facility. The random noise is limited in its magnitude by 20% of the original values because 100 houses are assumed to be located in close proximity and

[2] Other methods can be used to capture noise, such as Interval Markov Chains, where actual transition probabilities lie in intervals [29].

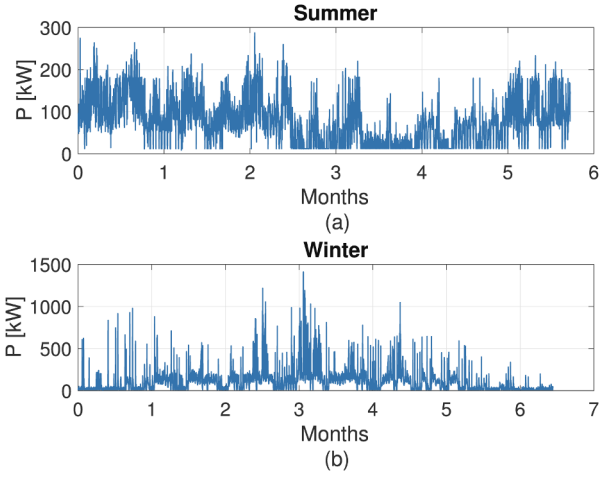

Fig. 3. Aggregated HVAC power consumption of 100 houses.

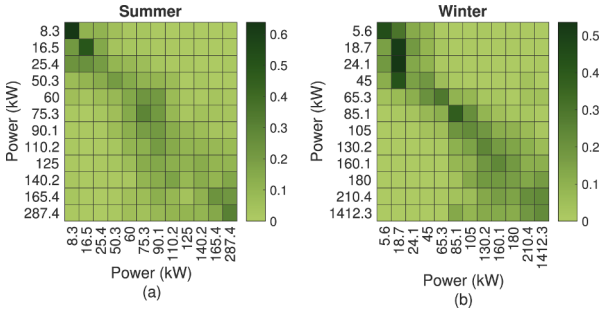

Fig. 4. Default transition probability matrix with 12 states constructed from the power profiles in Fig. 3, where color density indicates the probability value in the sidebar.

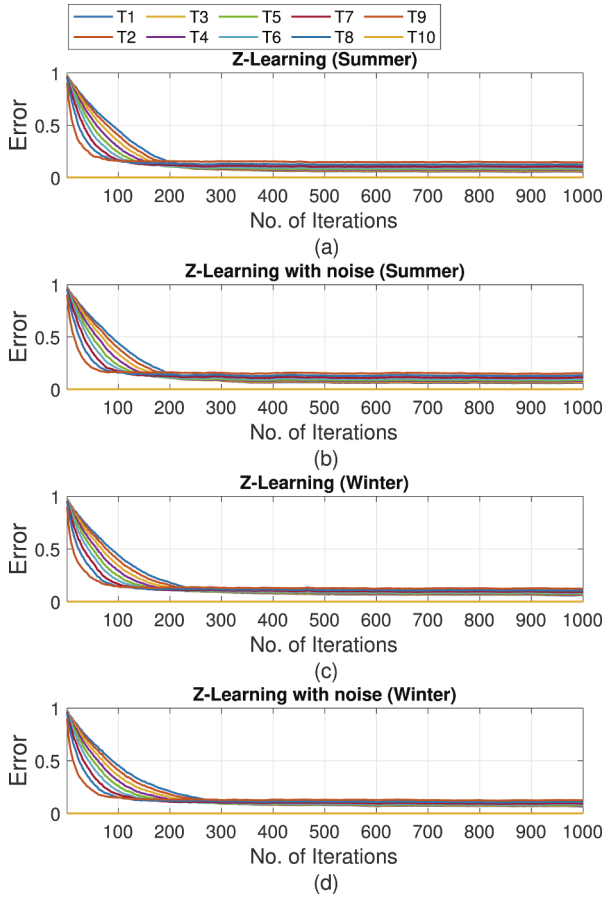

Fig. 5. Comparing Z-learning performance with and without noise during the summer and winter seasons for hourly time periods T1-T10.

function similarly. For these 100 houses, we extract HVAC data and assume that all HVACs are operated by the same DR aggregator. Fig 3 shows the aggregated HVAC power consumption for both summer and winter seasons in the period from July 1, 2013 to June 30, 2014. These profiles were discretized in 12 Markovian states and Fig. 4 displays the resulting transition probability matrices ($\overline{\mathcal{P}}^{\alpha\beta}$). These transition probability matrices are used to dispatch the TCL ensemble over the time horizon of 10 hourly intervals.

The case study solves the TCL optimization problem in Eq. (1) using Algorithms 1 and 2 and compare their performance in terms of the value function using the error metric:

$$\text{Error} = \frac{\sum_{\beta\in\mathcal{A}} |\varphi_t^{\beta_{\text{LS-MDP}}} - \varphi_t^{\beta_{\text{Z-learning}}}|}{\sum_{\beta\in\mathcal{A}} (\varphi_t^{\beta_{\text{LS-MDP}}})}, \tag{14}$$

which computes the relative difference between the Z-learning and LS-MDP values. Moreover, the Z-learning algorithm is run for two cases: (a) without noise added to the passive dynamics and (b) with noise. The learning rate for the Z-learning algorithms is set to decay as $\eta_k = \frac{1000}{1000+k}$, where k is a sample number.

4.2. Results

Fig. 5 describes the error convergence of the Z-learning algorithm with and without noise for each hourly time period. As the number of learning iterations increases, the resulting error reduces. The rate of convergence differs for the winter and summer seasons. For instance, the 10% error for all time period is achieved within 225 and 245 learning iterations. Similarly, the effect of noise on the learning rate is more visible during the winter season, where the number of learning iterations required to achieve the 10% error increases from 245 to 290 iterations. In contrast, in the summer case, adding noise does not affect the convergence rate and Z-learning achieve the 10% error in 225 learning iterations. The slower convergence rate in the winter case is explained by the fact that a greater power consumption being approximated using the same number of discrete states in the transition probability matrix, which requires more exploration of the model environment, especially when noise samples noticeably deviate from the default behavior defined by the passive dynamics.

Given the outcomes of Z-learning, the estimated transition probabilities are obtained as shown in Fig. 6. The estimated matrices for the cases with and without noise do not differ significantly. Thus, the Root-mean-square difference of elements between is 0.0101% and 0.0068% for the summer and winter seasons. Notably, this difference changes only slightly when compared to the default transition matrices in Fig. 2. In the case of winter season shown in Fig. 6 (c) and (d), the difference is 0.0017% and 0.0055% for the case without noise and with noise. The difference for the summer season in Fig. 6 (a) and (b) increases relative to the winter season and is 0.0023% and 0.01% for the case without noise and with noise. The result of using Z-learning is that as the number of iterations and samples increases, its outcomes will converge to the LS-MDP values.

Based on the transition probability matrices obtained with the LS-MDP and Z-learning, Fig. 7 compares the power dispatch of the TCL ensemble. Both Z-learning results with and without noise accurately approximate the benchmark LS-MDP solution. The maximum difference observed for the case with noise is 4.17 kW for the summer season and 13.7 kW for the winter season, and for the case without noise is 13.7 kW for the summer season and 13.9 kW for the winter season. These differences are relatively small given the summer and winter peaks of 287.4 kW and 1412.3 kW. The power dispatch of the TCL ensemble at

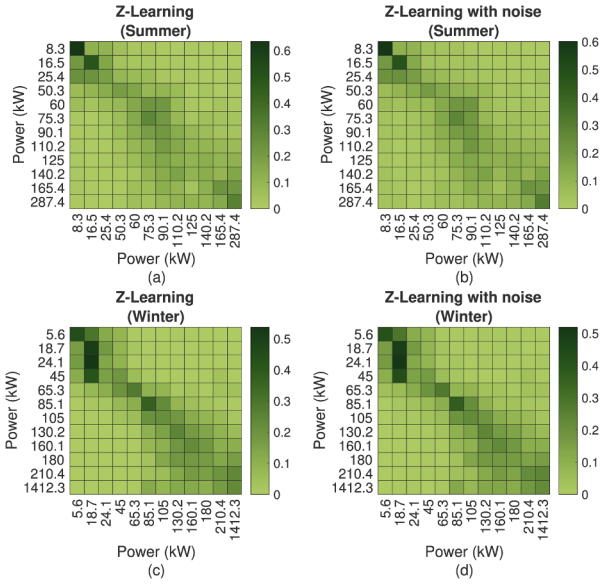

Fig. 6. Estimated transition probabilities for the summer and winter seasons with and without noise.

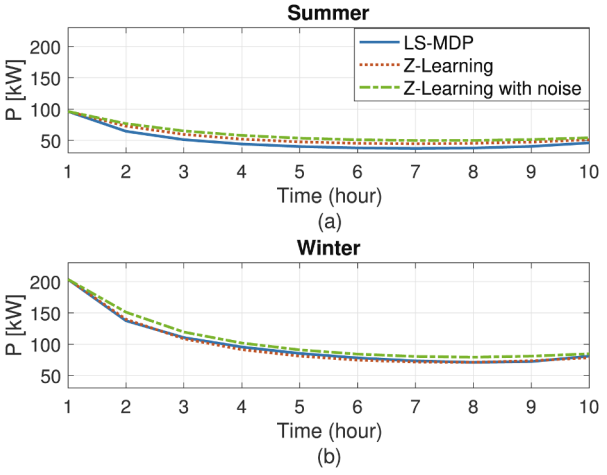

Fig. 7. Comparison of the TCL ensemble dispatch decisions for the LS-MDP and Z-learning solutions.

every iteration during Z-learning is shown in Fig. 8, where values stabilize as the number of iterations continues to increase. Similarly, Fig. 9 compares the value function of each method that represents the operating cost of the TCL ensemble. The values of the operating cost for both Z-learning with and without noise are slightly greater than the optimal value provided by the LS-MDP, because Z-learning approximates the optimal solution for the optimization problem that minimizes the objective function (i.e. the operating cost). Notably, the operating cost is comparatively high when $\mathcal{P}_t^{\alpha\beta} = \overline{\mathcal{P}}^{\alpha\beta}$ (no control taken), which shows the importance of controlling the TCL ensemble to lower the cost.

5. Conclusion

This paper presents a data-driven learning method for the control of TCL ensemble using the MDP and Z-learning approaches. The results show the importance of moving from model-based methods to model-free methods to bridge the gap between real environment and its model. The importance of modelling uncertainty to provide more robust

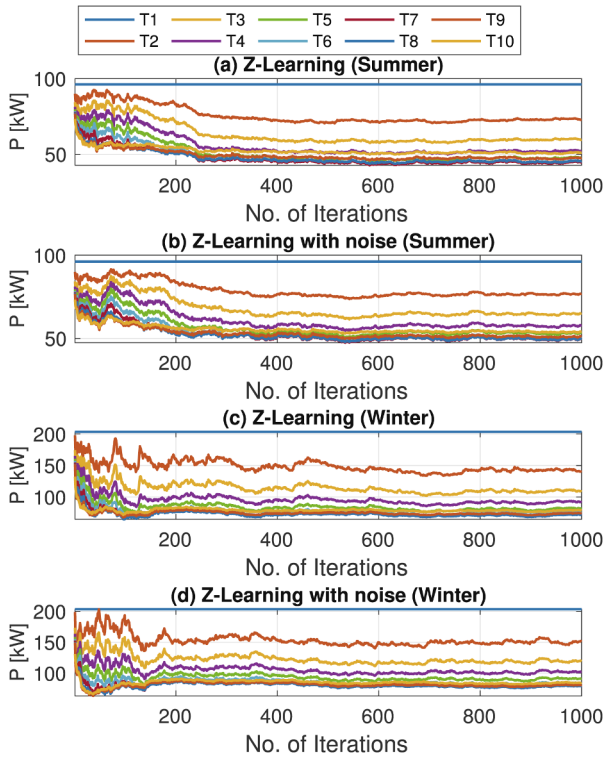

Fig. 8. Dispatch decisions for the TCL ensemble obtained with Z-learning for hourly time periods T1-T10.

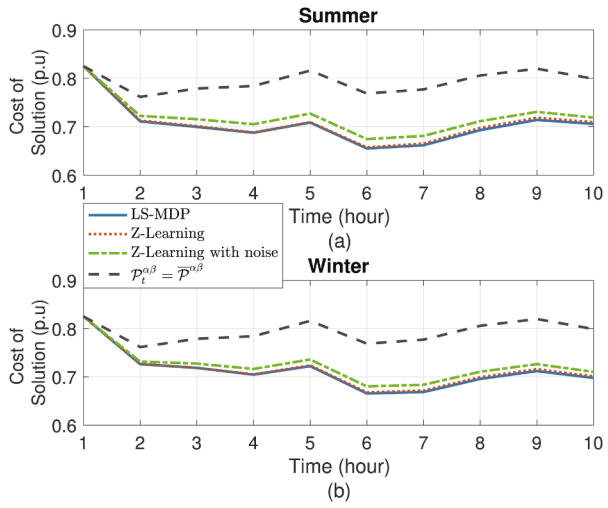

Fig. 9. Comparison of the solution cost for the LS-MDP and Z-learning solutions.

solutions is demonstrated by comparing the TCL ensemble injections and cost of the solution. In future, we will also consider the related problem of TCL optimization under uncertain energy prices and analyze the regret associated with online learning based schemes [31].

Declaration of Competing Interest

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

Appendix A. Bellman Equation Derivation for LS-MDP

The Bellman equation for a finite-horizon MDP is [32]:

$$\frac{1}{\gamma}\varphi_t^{\beta} = \frac{1}{\gamma}\min_{u}\left\{l_t^{\beta}(u) + \mathbb{E}_{\mathcal{P}_t^{\alpha\beta}(u)}[\varphi_{t+1}^{\alpha}]\right\}, \tag{15}$$

where $l_t^{\beta}(u)$ represents the immediate cost that the agent pays at time t for taking action u at state β and $\mathbb{E}_{\mathcal{P}_t^{\alpha\beta}(u)}[\varphi_{t+1}^{\alpha}]$ is the expectation of φ_{t+1}^{α} taken with respect to $\mathcal{P}_t^{\alpha\beta}(u)$:

$$\mathbb{E}_{\mathcal{P}_t^{\alpha\beta}(u)}[\varphi_{t+1}^{\alpha}] = \sum_{\alpha}\mathcal{P}_t^{\alpha\beta}(u)\varphi_{t+1}^{\alpha}, \tag{16}$$

Eq. (15) implicate the search over all actions u for each new state α. However, this can be time consuming due to the exponential growth of future states. The LS-MDP offers a solution for this problem, which uses the transition probabilities instead of the symbolic actions, where the agent can directly specify the probability of transition from the current state to any possible future state. The Bellman equation for choosing $\mathcal{P}_t^{\alpha\beta}$ by the agent is:

$$\frac{1}{\gamma}\varphi_t^{\beta} = \frac{1}{\gamma}\min_{\mathcal{P}}\left\{U_t^{\beta} + \gamma\mathbb{E}_{\mathcal{P}_t^{\alpha\beta}}\left[\log\frac{\mathcal{P}_t^{\alpha\beta}}{\overline{\mathcal{P}}^{\alpha\beta}}\right] + \mathbb{E}_{\mathcal{P}_t^{\alpha\beta}}[\varphi_{t+1}^{\alpha}]\right\}, \tag{17}$$

where U_t^{β} represents the state cost and $\mathbb{E}_{\mathcal{P}_t^{\alpha\beta}}$ means the statistical expectation of α taken with respect to the controlled transition distribution $\mathcal{P}_t^{\alpha\beta}$. Eq. (17) represents the Bellman equation for LS-MDP.

References

[1] CPUC, California's Distributed Energy Resources Action Plan: Aligning Vision and Action, May 2017.
[2] NYSERDA, New York Clean Energy Industry Report, 2019.
[3] D.S. Callaway, I.A. Hiskens, Achieving controllability of electric loads, Proceedings of the IEEE 99 (1) (2011) 184–199.
[4] EIA, U.S. Energy Information Administration (EIA) Monthly Energy Review, Aug 2019.
[5] Ning Lu, D.P. Chassin, A state-queueing model of thermostatically controlled appliances, IEEE Transactions on Power Systems 19 (3) (2004) 1666–1673.
[6] D. Coakley, P. Raftery, M. Keane, A review of methods to match building energy simulation models to measured data, Renewable and Sustainable Energy Reviews 37 (2014) 123–141.
[7] T. Samad, E. Koch, P. Stluka, Automated demand response for smart buildings and microgrids: The state of the practice and research challenges, Proceedings of the IEEE 104 (4) (2016) 726–744.
[8] D.P. Chassin, J. Stoustrup, P. Agathoklis, N. Djilali, A new thermostat for real-time price demand response: Cost, comfort and energy impacts of discrete-time control without deadband, Applied Energy 155 (2015) 816–825.
[9] D.S. Callaway, Tapping the energy storage potential in electric loads to deliver load following and regulation, Energy Conversion and Management 50 (5) (2009) 1389–1400.
[10] J.L. Mathieu, M.G. Vayá, G. Andersson, Uncertainty in the flexibility of aggregations of demand response resources, IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, (2013), pp. 8052–8057.
[11] M. Vrakopoulou, B. Li, J.L. Mathieu, Chance constrained reserve scheduling using uncertain controllable loads part I: Formulation and scenario-based analysis, IEEE Transactions on Smart Grid 10 (2) (2017) 1608–1617.
[12] S. Koch, J.L. Mathieu, D.S. Callaway, Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services, in Proceedings of Power Systems Computation Conference, (2011).
[13] A. Bušić, S. Meyn, Ordinary differential equation methods for markov decision processes and application to Kullback–Leibler control cost, SIAM Journal on Control and Optimization 56 (1) (2018) 343–366.
[14] M. Chertkov, V.Y. Chernyak, D. Deka, Ensemble control of cycling energy loads: Markov decision approach, Energy Markets and Responsive Grids: Modeling, Control, and Optimization, The IMA Volumes in Mathematics and its Applications, vol 162, New York, 2018, pp. 363–382.
[15] R. Pop, A. Hassan, K. Bruninx, M. Chertkov, Y. Dvorkin, A markov process approach to ensemble control of smart buildings, 2019 IEEE Milan PowerTech (2019) 1–6.
[16] M. Chertkov, D. Deka, Y. Dvorkin, Optimal ensemble control of loads in distribution grids with network constraints, 2018 Power Systems Computation Conference (PSCC) (2018) 1–7.
[17] A. Hassan, R. Mieth, M. Chertkov, D. Deka, Y. Dvorkin, Optimal load ensemble control in chance-constrained optimal power flow, IEEE Transactions on Smart Grid 10 (5) (2019) 5186–5195.
[18] M.D. Wagy, J.C. Bongard, J.P. Bagrow, P.D.H. Hines, Crowdsourcing predictors of residential electric energy usage, IEEE Systems Journal 12 (4) (2018) 3151–3160.
[19] E. Todorov, Linearly-solvable markov decision problems, in: B. Schölkopf, J.C. Platt, T. Hoffman (Eds.), Adv. in Neural Inf. Proc. Syst. MIT Press, 2007, pp. 1369–1376.
[20] E. Todorov, Efficient computation of optimal actions, Proceedings of the National Academy of Sciences 106 (28) (2009) 11478–11483.
[21] K. Dvijotham, E. Todorov, Linearly Solvable Optimal Control, John Wiley & Sons, Ltd, pp. 119–141.
[22] A. Jonsson, V. Gómez, Hierarchical linearly-solvable markov decision problems, Proceedings of the Twenty-Sixth International Conference on International Conference on Automated Planning and Scheduling, ICAPS'16, AAAI Press, 2016, pp. 193–201.
[23] W. Healy et al., Net Zero Energy Residential Test Facility Instrumented Data; Year 1, 2018.
[24] T. Warren Liao, Clustering of time series data-a survey, Pattern Recogn. 38 (11) (2005) 1857–1874.
[25] C. Bishop, Pattern recognition and machine learning, springer, 2006.
[26] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, 2018.
[27] S.P. Meyn, P. Barooah, A. Bušić, Y. Chen, J. Ehren, Ancillary service to the grid using intelligent deferrable loads, IEEE Transactions on Automatic Control 60 (11) (2015) 2847–2862.
[28] S. Meyn, P. Barooah, A. Bušić, J. Ehren, Ancillary service to the grid from deferrable loads: The case for intelligent pool pumps in florida, 52nd IEEE Conference on Decision and Control, (2013), pp. 6946–6953.
[29] K. Chatterjee, K. Sen, T.A. Henzinger, Model-checking ω-regular properties of interval markov chains, in: R. Amadio (Ed.), Foundations of Software Science and Computational Structures, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 302–317.
[30] T. Jaakkola, M.I. Jordan, S.P. Singh, Convergence of stochastic iterative dynamic programming algorithms, Advances in neural information processing systems, (1994), pp. 703–710.
[31] G. Neu, V. Gómez, Fast rates for online learning in linearly solvable markov decision processes, arXiv preprint arXiv:1702.06341 (2017).
[32] R. Bellman, A markovian decision process, Journal of Mathematics and Mechanics 6 (5) (1957) 679–684.