

Lawrence Berkeley National Laboratory

LBL Publications

Title

Reinforcement learning for building controls: The opportunities and challenges

Permalink

<https://escholarship.org/uc/item/5rh8s2hc>

Authors

Wang, Zhe

Hong, Tianzhen

Publication Date

2020-07-01

DOI

10.1016/j.apenergy.2020.115036

Peer reviewed



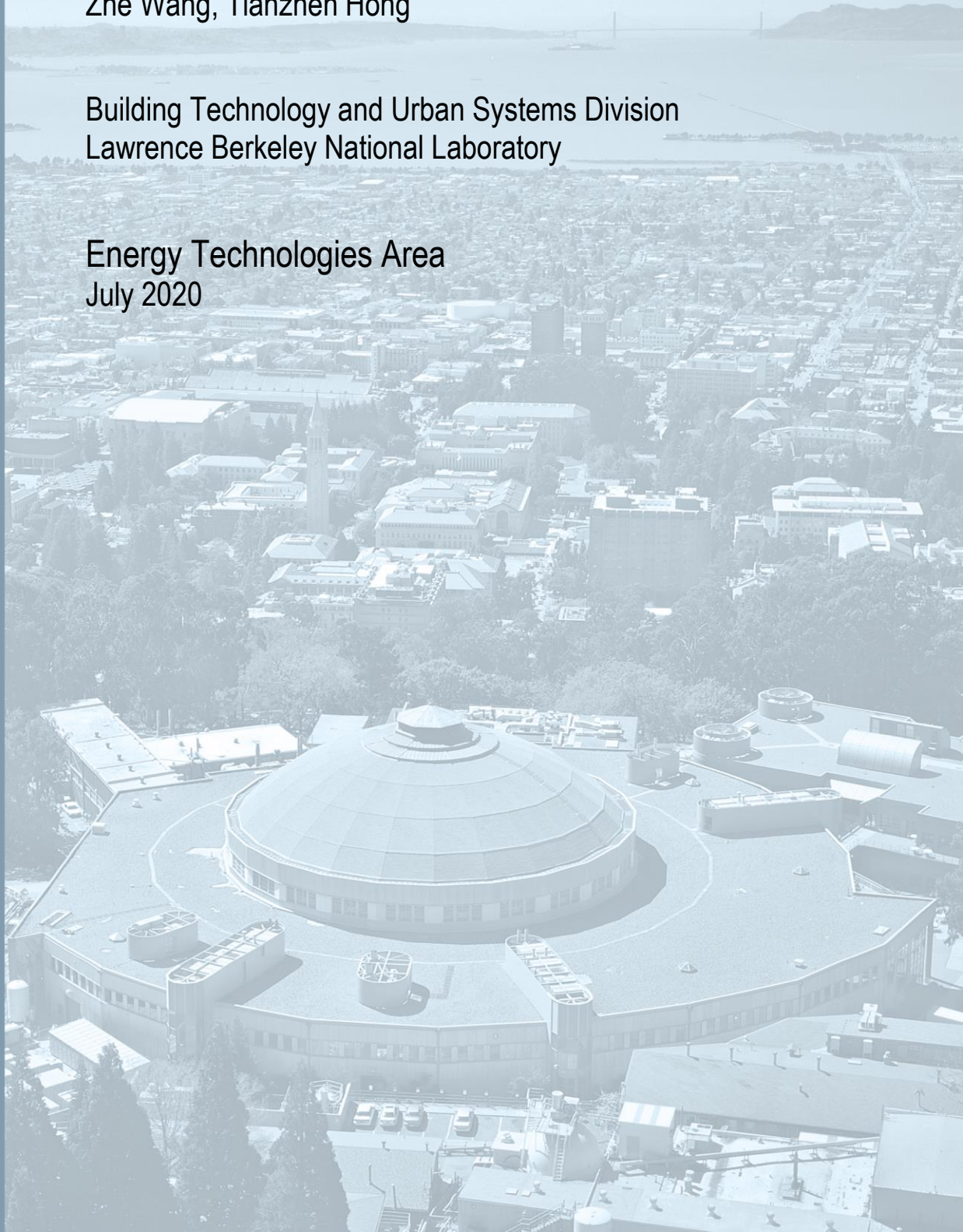
Lawrence Berkeley National Laboratory

Reinforcement Learning for Building Controls: The opportunities and challenges

Zhe Wang, Tianzhen Hong

Building Technology and Urban Systems Division
Lawrence Berkeley National Laboratory

Energy Technologies Area
July 2020



Disclaimer:

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Reinforcement Learning for Building Controls: The opportunities and challenges

Zhe Wang, Tianzhen Hong*

Building Technology and Urban Systems Division
Lawrence Berkeley National Laboratory
One Cyclotron Road, Berkeley, CA 94720, USA
* *Corresponding author: T. Hong, thong@lbl.gov*

ABSTRACT

Building controls are becoming more important and complicated due to the dynamic and stochastic energy demand, on-site intermittent energy supply, as well as energy storage, making it difficult for them to be optimized by conventional control techniques. Reinforcement Learning (RL), as an emerging control technique, has attracted growing research interest and demonstrated its potential to enhance building performance while addressing some limitations of other advanced control techniques, such as model predictive control. This study conducted a comprehensive review of existing studies that applied RL for building controls. It provided a detailed breakdown of the existing RL studies that use a specific variation of each major component of the Reinforcement Learning: algorithm, state, action, reward, and environment. We found RL for building controls is still in the research stage with limited applications (11%) in real buildings. Three significant barriers prevent the adoption of RL controllers in actual building controls: (1) the training process is time consuming and data demanding, (2) the control security and robustness need to be enhanced, and (3) the generalization capabilities of RL controllers need to be improved using approaches such as transfer learning. Future research may focus on developing RL controllers that could be used in real buildings, addressing current RL challenges, such as accelerating training and enhancing control robustness, as well as developing an open-source testbed and dataset for performance benchmarking of RL controllers.

KEYWORDS

Building controls; Reinforcement Learning; Machine Learning; Optimization; Building performance

1. Introduction

People spend more than 85% of their time in buildings [1]. At the same time, buildings consume about 40% of total primary energy in countries like the United States [2]. Well-performing building controls are capable of delivering a healthy and comfortable indoor environment in an energy- and carbon-efficient way. However, building controls are becoming complicated because in addition to traditional services such as lighting and HVAC, modern building energy systems must respond to on-site intermittent renewables, energy storage, electric vehicle charging, and more. Furthermore, buildings need to respond to grid signals by shifting the load to improve grid stability and security, adding a layer of complexity to building controls. The U.S. Department of Energy launched an initiative on Grid-interactive Efficient Buildings (GEB), which aims to develop and integrate technologies for grid responsive buildings to achieve lower energy use (energy efficiency), flexible loads (demand flexibility), and resilience (e.g., running in low power mode under constrained conditions such as heatwaves). Smart building controls play a critical role in GEB [3].

May [4] argued that advanced building controls need to function well in the following three aspects to make buildings smart and intelligent: first, they must balance the trade-off between multiple goals, such as occupant comfort, energy conservation, grid flexibility and carbon emission reduction; second, they must adapt autonomously to the environment and its occupants; and third, they must feed occupant feedback into the control logic (human-in-the-loop). Unfortunately, those functions are difficult to achieve using conventional building control techniques.

The most conventional building control is rule-based feedback control, which includes two steps: (1) rely on some pre-determined schedules to select the setpoints (e.g., temperature setpoint), and (2) track the setpoints using techniques such as Proportional-Integral-Derivative (PID) control [5]. The rule-based prescriptive approach can maintain occupant comfort by maintaining a comfort range. Additionally, it is possible to reduce energy consumption and carbon emissions by adjusting the setpoints based on heuristic rules; for example, relaxing the temperature setpoint band during unoccupied hours or demand response events. ASHRAE Guideline 36 summarized those rules [6], which could represent the state of the art of this approach adopted by industry.

The prescriptive and feedback-based reactive control strategy is simple and effective, but not optimal, for two reasons. First, predictive information is not taken into consideration, leading to sub-optimal performance. For instance, if the coming day is predicted to be hot, it might be more energy efficient to pre-cool the building in advance. Second, the control sequence (such as those parameters in the PID controller) is fixed and predetermined, so it is not customized to a specific building and climate condition. To improve building control performance, Model Predictive Control (MPC) has been explored.

The three words in the Model Predictive Control correspond to its three critical steps. “Model” corresponds to the development and identification of models that characterize the thermal and energy dynamics of buildings and systems. “Predictive” corresponds to the disturbance prediction, such as weather or occupancy prediction in the building context. “Control” corresponds to solving the optimization problem by feeding the predictive information into the developed model. Since it was initially proposed in the 1970s in the chemical and petrochemical industries, MPC has been successfully applied in many fields [7]. In the building industry, MPC has been used to control radiant ceiling heating [8], floor heating [9], intermittent heating [10] and ventilation [11], and to optimize cold water thermal storage systems [12]. MPC has proved its potential to save energy in both simulation [13] and experimental tests on real buildings [8].

The major challenge of MPC is that it is labor-intensive and requires expertise to use. It might be cost-effective to develop and calibrate a model for a car or an airplane that can be generalized and used for many cars and airplanes. Still, every building and its energy systems are unique, so it is difficult to generalize a standard building energy model for various buildings. As a result, despite the promising results, MPC has not yet been widely adopted by the building industry [14].

Empowered by big data, powerful computing, and algorithm advancement, Machine Learning (ML) has been used in almost every stage of the building lifecycle and has demonstrated its potential to enhance building performance [15]. As a branch of machine learning specifically for control problems, Reinforcement Learning (RL) is becoming a promising method to revolutionize building controls. RL is data-driven, which could help users avoid the tedious work of developing and calibrating a detailed model, as is required by MPC. Additionally, RL could leverage the recent and rapid developments in the machine learning field, such as deep learning and feature encoding, to make better control decisions. RL has been successfully applied in other areas, ranging from gaming [16] to robotics [17]. It is the time to explore whether or not RL could be used to optimize building controls to achieve energy efficiency, demand flexibility, and resiliency, which is a new but rapidly developing area. The objectives of this paper are threefold. First, we introduce the general framework of RL and how this framework would fit into the building control field. Secondly, we provide a detailed breakdown of the existing studies that use a specific variation of each major component of the Reinforcement Learning: algorithm, state, action, reward, and environment. Last, we discuss the current challenges and future

research opportunities of RL for building controls.

2. Methods and Objectives

2.1 Reinforcement learning for building controls

Reinforcement learning is a branch of machine learning that is specialized in solving control, or sequential decision making, problems. As shown in Figure 1, the three categories of machine learning problems differentiate from each other in terms of the kinds of feedback the agent/algorithm will receive after they make a decision/prediction. For supervised learning, the agent will immediately know how accurate its prediction is compared with the ground truth given by the label data. And this information will be used to update and improve the predictor. For unsupervised learning, no feedback is provided as the dataset is unlabeled. Reinforcement learning lies in the middle between the two scenarios, which receives delayed feedback.

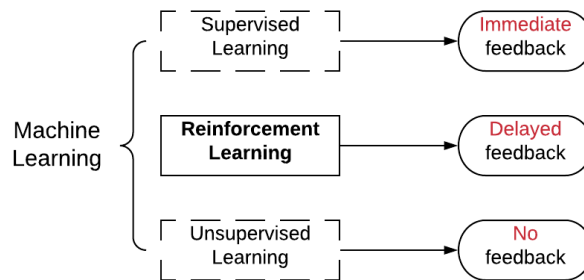


Figure 1. Three types of machine learning problems

To better understand the concept of delayed feedback, we need to dive deep into the Markov Decision Process (MDP), which is the mathematical foundation of RL. MDP is formed by a tuple (S, A, P, R) , as shown in Figure 2.

- **S: State**

The state is a mathematical description of the environment that is relevant and informative to the decision to be made. States in RL are similar to features in supervised or unsupervised learning. Taking HVAC controls as an example, current room temperature could be the state the HVAC controller wants to consider. Additionally, the predicted outdoor temperature of the next time step might also be another state variable, as this information could inform the controller to make a better decision.

- **A: Action**

Action is the decision made by the controller in terms of how to control the environment. In the example of HVAC control, the action could be adjusting indoor temperature setpoint, supply air temperature, fan speed, etc.

- **Environment**

Environment is the target of the control, which is mathematically represented by the following two functions:

- **P: Transition Probability**

The transition probability predicts how the environment will evolve if we take action a_t at state s_t , i.e., mapping the state and action of the current time step to the state of the next time step.

- **R: Reward Function**

The reward function predicts the immediate rewards of taking action a_t at state s_t , i.e., mapping the state and action to the rewards.

- **Controller/Agent**

The goal of the controller is to find the optimal **Policy** (π), which outputs an optimal action for each state. There are primarily two approaches to achieve this goal:

- **Model-based RL**

If the characteristic of the environment is known to the controller, i.e., the **transition probability** and the **reward function** are known, we could use value iteration or policy iteration to find the optimal policy.

- **Model-free RL**

In most scenarios, the behaviors of the environment are unknown to the agent. The controller needs to find out the optimal policy without modeling the environment. Model-free RL is similar to the concept of end-to-end machine learning, as they both skip some intermediate steps (modeling the environment in RL context) and achieve the goal directly.

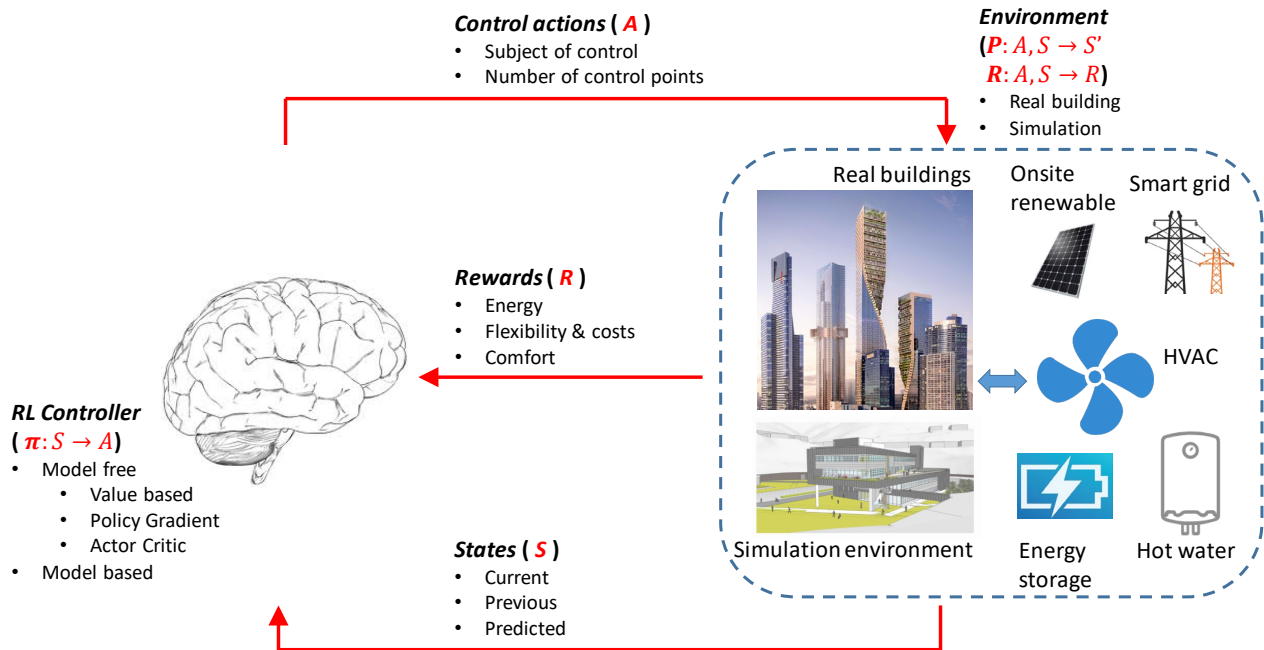


Figure 2. Reinforcement learning for building controls

Given the RL framework, we can better understand the concept of delayed feedback. Because the feedback is delayed, the control problem becomes complicated. Under the context of RL, any action leads to two consequences, receiving an immediate reward and arriving at a new state. The control agent could not simply select the action corresponding to the highest reward; instead, it needs to consider the delayed future rewards corresponding to the new state. For instance, the action of pre-cooling might lead to higher immediate energy consumption, but in the long term, the new state saves utility costs. The strength of RL lies in its ability to optimize the trade-off between short-term and long-term benefits. To differentiate the long-term benefits from the short-term ones, the concept of **Value** is introduced. Value is defined as the accumulated ‘benefits’ of future multiple steps. On the contrary, reward is defined as the immediate ‘benefits’ of taking the selected action at the current time step. In other words, value is the accumulated rewards of multiple future steps until the end.

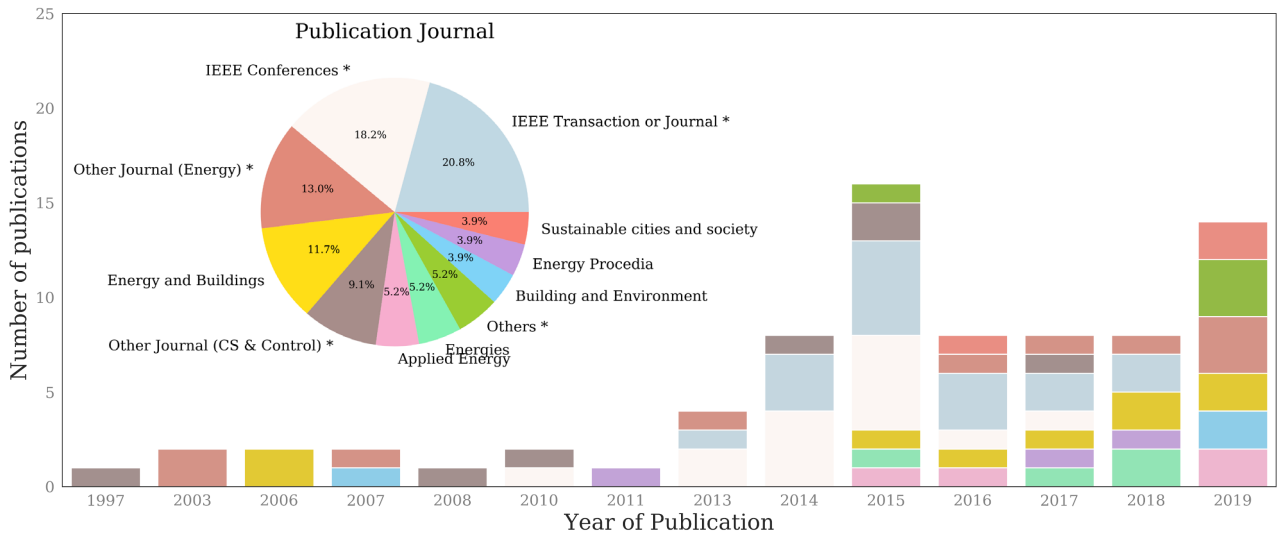
As observed in Figure 2, there are five major components in RL settings: controller, states, actions, rewards, and the environment. Varieties in the five components (such as different algorithms or different states to represent the environment) lead to different RL implementation, which results in different control performance. The ultimate goal of this study is to conduct a tutorial survey and a comprehensive review of existing studies using RL for building controls. By surveying how current researchers select state and action variables, determine reward function, and choose algorithms, we aim to present an overview of the current applications of RL for building controls.

2.2 Literature search

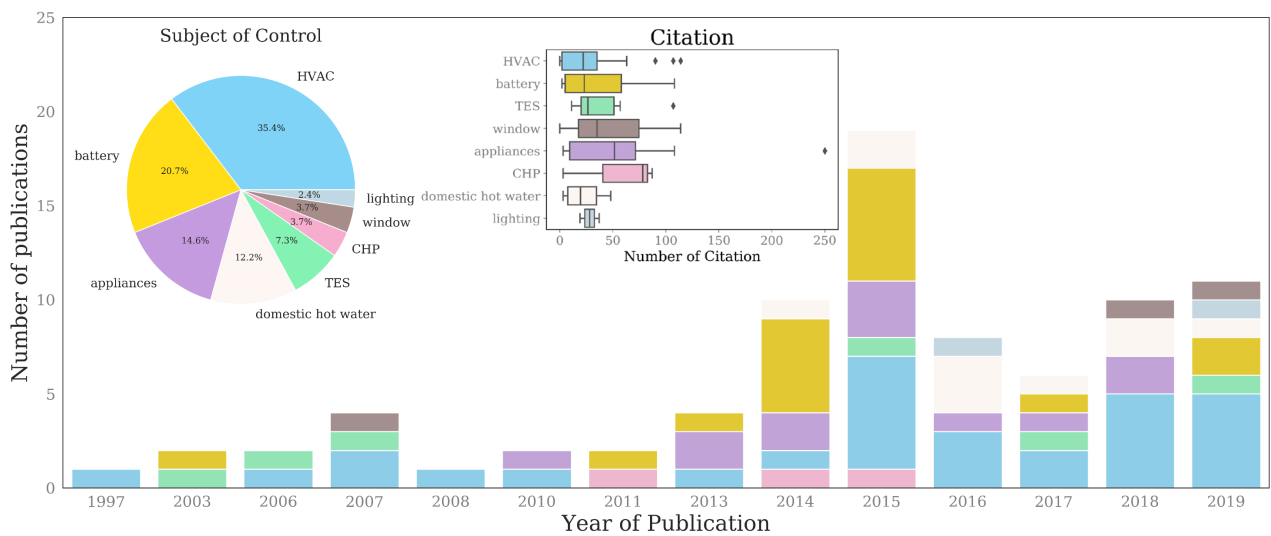
We conducted a literature search on the academic search platform *Web of Science* using the topic structure and keywords shown in Equation 1, where the symbol “*” is used to search for terms in both singular and plural forms. The *Web of Science* platform could retrieve papers from both the traditional built environment field and the computer science field. We did not down select or filter out any papers that applied Reinforcement Learning in the buildings field.

$$topic = (reinforcement\ learning) \text{ AND } [(building * \text{ OR } house * \text{ OR } home \text{ OR } residential *) \text{ AND } control] \quad (\text{Eq. 1})$$

The literature search was conducted in December of 2019. With the search structure and keywords listed in Equation 1, 77 articles on this topic were found and reviewed. The 77 studies examined in this paper are listed in Table A1 of the Appendix. Figure 3 summarizes the papers based on their publication journals and the control subjects.



(a) Publication journal ¹



(b) Subject of control ²

Figure 3. Summary of articles searched

As shown in Figure 3, the publication on this topic jumped between 2014 and 2015, and then stayed stable. Applying RL for building controls is an interdisciplinary field: half of the papers are from journals focused on computer science, artificial intelligence, and controls. The remaining publications are from journals focused on buildings, energy, and the environment. To follow the latest progress, researchers need to watch journals from both fields. In terms of the control subject, 35% of studies used RL to control HVAC, and the proportion increased to 50% after 2015. Other popular subjects include the charging/discharging of batteries and scheduling of home appliances.

In addition to the number of articles published, another important index for estimating the quality and influence of those publications is the number of citations,³ which was listed in Figure 3b. Articles in this field, in general, are cited between 20–70 times per paper. The most cited paper in this field was published in 2010 using tabular Q-learning for home appliance scheduling [18]. This article was published in First IEEE International Conference on Smart Grid Communications and has been cited 250 times. Papers using RL for HVAC controls were not cited as many times as those focused on other fields. A possible reason is those papers were published more recently, between 2015 and 2019. Table 1 lists the most highly cited papers in each subject, so hopefully, that list can direct readers to the most influential papers on each subject.

¹ As studies using reinforcement learning for building controls were published in too many journals to be exhaustively presented, we only list those journals that published at least two papers and group the remaining into a small number of combined categories. The combined categories are noted with a star.

² An article might be counted twice if it controls multiple building components.

³ The number of citations were retrieved from Google Scholar until December 23, 2019.

Table 1. Highly cited papers in each subject

Subject	Number of Citations	Most Highly Cited Paper	Journal/Conference of Publication
HVAC	114	Dalamagkidis et al. (2007) [19]	Building and Environment
Batteries	108	Wei et al. (2014) [20]	IEEE Transactions on Industrial Electronics
Appliances	250	O'Neill et al. (2010) [18]	IEEE Conference on Smart Grid Communications
Domestic Hot Water	48	Ruelens et al. (2014) [21]	IEEE Conference on Power Systems Computation
Thermal Energy Storage	107	Liu and Henze (2006) [22]	Energy and Buildings
Combined Heat and Power	87	Jiang and Fei (2014) [23]	IEEE Transactions on Smart Grid
Windows	114	Dalamagkidis et al. (2007) [19]	Building and Environment
Lighting	37	Cheng et al. (2016) [24]	Energy and Buildings

2.3 Previous reviews

Three literature review studies were found in the literature search. They summarized the applications of reinforcement learning in building controls for three specific purposes: occupant comfort, energy savings, and demand response.

Han et al. (2019) [25] reviewed the application of reinforcement learning for occupant comfort management. Thirty-three empirical studies on this topic have been identified and reviewed. Among the papers reviewed, value-based Q-learning was found to dominate the learning algorithms. The majority of papers sought to maintain comfortable indoor temperature, while other important aspects of occupant comfort, such as indoor air quality and visual comfort, are rarely studied. Another interesting finding is how the occupant comfort should be defined, as only 5 out of 33 studies include occupant feedback in the control loop. At the end of the paper, the authors proposed some future research trends. First, multi-agent reinforcement learning needs to be further explored because there might be multiple occupants present in the environment. Additionally, because reinforcement learning is computationally demanding, how to better integrate the computation platforms with the building management system is also important for the application of RL in buildings.

Mason and Grijalva (2019) [26] reviewed the application of reinforcement learning for building energy management, including HVAC, water heater, appliances, lighting, photovoltaics (PV), batteries and the electrical grid. It was found that RL can typically provide savings of about 10% for HVAC and about 20% for water heaters. However, the vast majority of current studies are in simulation only. Several future research trends on this topic have been identified. First, Deep Reinforcement Learning was believed to be promising due to its capability to learn more complex policy under sophisticated environments. Second, as building operation has multiple goals (e.g., energy, comfort, cost), multi-objective RL demands further investigation, such as Pareto Q learning. Third, in the scenario of controlling a community of homes or in the campus/urban scale, multi-agent RL is needed. Last, transfer learning is crucial for the large adoption of RL for building controls, as it is time-consuming and computationally demanding, if not totally impossible, to train an RL controller for each building. Transfer Learning is defined as the process of applying the knowledge learned from one task to a related, but different, task [27]. Transfer learning is important because training the controller is a time-consuming process and requires expertise. Rather than training the RL controller on every individual building, it would be more efficient and scalable if we could train the RL controller on a small number of buildings and then apply them to larger building stocks. Transfer learning technique has been used in MPC based building controls [28], however, no successful application of transfer learning is found in the RL-based building controls.

Vázquez-Canteli and Nagy (2019) [29] reviewed the use of reinforcement learning for demand response applications. In total, 105 articles were reviewed. Some common research gaps were identified; for example, only a small fraction of studies reviewed have been tested in physical systems, and very few studies include

occupant feedback into the control loop. Additionally, Vázquez-Canteli and Nagy (2019) pointed out that most of the studies are not easily reproducible, and the performance of controllers are not comparable due to the different thermal dynamics and properties of different testbeds. Based on those gaps, two future research needs were identified. First, standardized control problems—as well as integrated software tools that include both building simulation and machine learning features—are needed to help researchers investigate their control approaches and compare them directly to other approaches. And second, the applicability of reinforcement learning in multi-agent systems needs to be further explored, especially for grid operation and optimization.

2.4 Research gaps and objectives

As introduced in Section 2.1, “reinforcement learning” is a broad and ambiguous term. A careful selection of states, actions, and algorithms is crucial for the performance of RL controllers. Meanwhile, different environmental settings make the comparison between various studies very challenging, if not impossible. As RL attracts increasing research and practical attention, it is necessary to comprehensively review and summarize which states, actions, and algorithms were selected, and how the environment was set up in existing studies. Such a review could help new researchers better understand the progress and identify research gaps in existing studies. In this study, we aim to provide a detailed breakdown of the existing studies that use a specific variation of each major component of the Reinforcement Learning, from the selection of algorithm, state, action, value approximation to the design of environment. Such a comprehensive breakdown has never been done before. Even though three review studies were found on similar topics, they only focus on a specific topic of building controls, which might not be able to provide a comprehensive overview of this topic.

The goal of this study was to conduct a comprehensive survey of studies that applied RL for building controls. When designing an RL controller, many decisions need to be made; among them, how the state and action space is determined, how the reward function is designed, which algorithm is used, and where the training data come from. The object of this study was to dive deep into those subtle but essential variations. By reviewing the existing studies, we aim to present a whole picture of: (1) which areas/approaches have been extensively studied and which have not, and (2) which approach works and which does not, and why. We believe this work could help researchers learn from existing studies, get inspiration from current research trends, choose which research gaps to address, and improve the design of their own RL controller.

We organize our review around five topics: algorithms, states, actions, rewards, and the environment—each corresponding to a key component in the RL framework, as presented in Figure 2. The result of this survey will be presented in Section 3. A full list of reviewed papers is shown in Table A1, so readers can easily retrieve key information as needed. In Section 4, we discuss some advanced topics of RL controllers, including how to speed up training, how to guarantee security, and how to evaluate performance. Conclusions are presented in Section 5.

3 Survey on Reinforcement Learning for Building Controls

Before diving deep into the survey results, we started with the mathematical formulation of the RL problem, which is shown in equations 2 and 3. The sequence of states and actions $s_1, a_1, \dots, s_T, a_T$ is called trajectory \mathbf{r} , which is determined by the transition probability $p(s_{t+1}|s_t, a_t)$ and the policy $\pi_\theta(a_t|s_t)$. The transition probability $p(s_{t+1}|s_t, a_t)$ and the reward function $r(s_t, a_t)$ are the characteristics of the environment. Given $p(s_{t+1}|s_t, a_t)$ and $r(s_t, a_t)$, the goal of the agent is to find the optimal control policy $\pi_\theta(a_t|s_t)$ that could result in the trajectory $s_1, a_1, \dots, s_T, a_T$ with highest accumulative rewards $E_{\mathbf{r} \sim p_\theta(\mathbf{r})}[\sum_t r(s_t, a_t)]$. The expectation operator $E_{\mathbf{r} \sim p_\theta(\mathbf{r})}$ is introduced because both the environment and the policy could be stochastic.

$$p_\theta(s_1, a_1, \dots, s_T, a_T) = p(s_1) \prod_{t=1}^T \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \quad (\text{Eq. 2})$$

$$\max_{\theta} \{E_{\mathbf{r} \sim p_\theta(\mathbf{r})}[\sum_t r(s_t, a_t)]\} \quad (\text{Eq. 3})$$

3.1 Algorithms

As introduced in Section 2.1, there are two major categories of RL algorithms: model-based RLs and model-free RLs. A model-based RL learns the characteristics of the environment $p(s_{t+1}|s_t, a_t)$ and $r(s_t, a_t)$ first, if they are not known in advance. Then the learned $p(s_{t+1}|s_t, a_t)$ and $r(s_t, a_t)$ can be used to find the optimal policy. This approach is called “model-based RL” because the process of learning $p(s_{t+1}|s_t, a_t)$ and $r(s_t, a_t)$ is primarily developing a model for the environment. The model could be a data-driven model, such

as a deep neural network, or a physics-based model, such as thermal resistance–thermal capacity model. In this regard, a model-based RL is similar to the MPC technique discussed in the Introduction section, as MPC could be developed from not only physics-based or reduced-order models [30], but also pure data-driven models [31].

However, learning an accurate model is time-consuming and requires expertise. And a more accurate model might not necessarily lead to better control [32]. The model-free RL skips the process of having to learn a model. Instead, it explores the optimal control policy by learning from the interaction with the environment. There are three approaches to find the optimal control policy without learning the model: policy gradient, actor-critic, and value-based.

The **Policy Gradient** method directly differentiates the accumulated rewards $E_{r \sim p_{\theta}(r)}[\sum_t r(s_t, a_t)]$ (referred to as $J(\theta)$) with respect to θ . After some mathematical tricks, the gradient of accumulated rewards could be rewritten as Equation 4 [33]. Once $\nabla_{\theta}J(\theta)$ is calculated, we could update θ using Equation 5 to increase the accumulated reward $J(\theta)$. This approach is named “policy gradient” because it uses the gradient of policy $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$ to update the policy. After the policy is updated, we run the new policy with the environment to collect a new trajectory and rewards $s_1, a_1, r_1, \dots, s_T, a_T, r_T$, and use the new trajectory to calculate $\nabla_{\theta}J(\theta)$. Thanks to the advancement of deep learning, the implementation of the policy gradient is convenient, using automatic differentiation packages [34] such as TensorFlow, Pytorch, and others.

$$\nabla_{\theta}J(\theta) \approx \frac{1}{N} \sum_{i=1}^N [(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)) (\sum_{t=1}^T r(s_t, a_t))] \quad (\text{Eq. 4})$$

$$\theta \leftarrow \theta + \alpha \nabla_{\theta}J(\theta) \quad (\text{Eq. 5})$$

As the log-likelihood term of $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$ essentially quantifies how likely the action a_t will be selected given the current s_t , the policy gradient algorithm could be interpreted as: increasing the chance of taking action a_t if a_t will result in a higher accumulated reward $\sum_{t=1}^T r(s_t, a_t)$.

The **Actor-Critic** algorithm enhances the policy gradient approach by replacing the accumulated rewards $\sum_{t=1}^T r(s_t, a_t)$ with a value approximation function. We introduced $\sum_t r(s_t, a_t)$ because we need to evaluate the policy π , and use this evaluation to improve the policy. However, this evaluation has a high variance. Because the environment is stochastic in most cases, the trajectory $s_1, a_1, r_1, \dots, s_T, a_T, r_T$ is just one of many outcomes. Therefore using $\sum_{t=1}^T r(s_t, a_t)$ in Equation 4 is essentially using only one sample of trajectory to estimate the performance of the policy π_{θ} . Though the single-sample estimator is unbiased, it has a very high variance. To address this issue, Actor-Critic algorithm introduces a value estimator $Q^{\pi}(s_t, a_t)$ to replace the single sample estimator $\sum_{t=1}^T r(s_t, a_t)$ as the evaluation of the policy; and then use $Q^{\pi}(s_t, a_t)$ to update and improve the policy. $Q_{\phi}^{\pi}(s_t, a_t)$ is fitted with the sampled reward sums. This approach is named “Actor-Critic” because in addition to the policy function $\pi_{\theta}(a_t | s_t)$ (called actor), the value estimation function $Q_{\phi}^{\pi}(s_t, a_t)$ (called critic) is introduced.

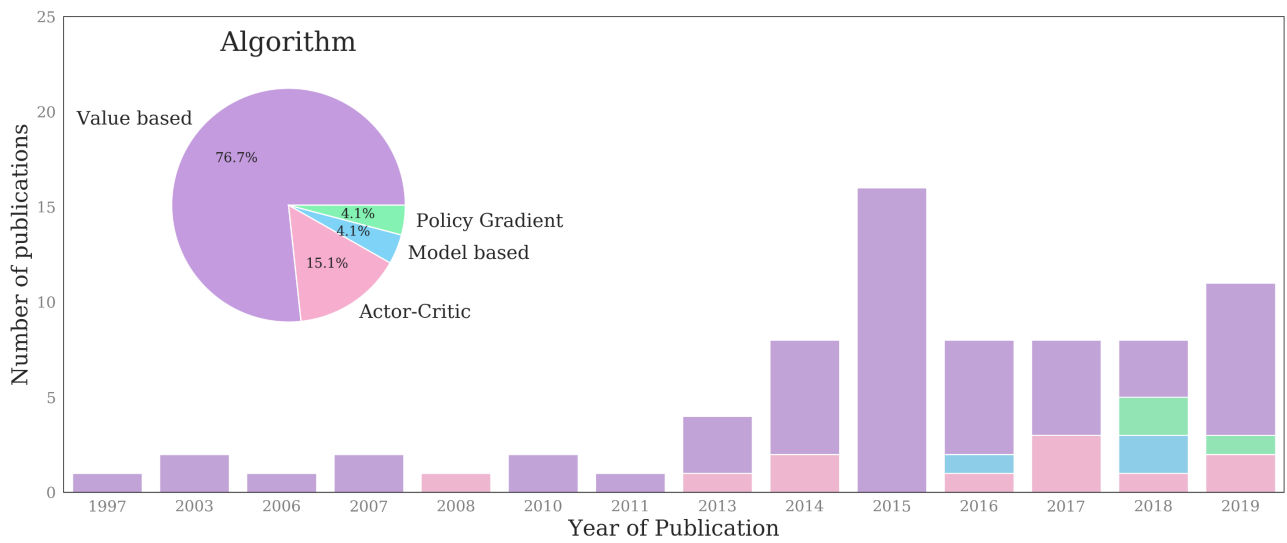
The third model-free RL algorithm type is **Value-Based**. A value-based RL learns the value function without explicitly representing the policies. The idea behind the value-based approach is: once we can evaluate every action-state pair (s_t, a_t) at any time step, there is no need to calculate $\nabla_{\theta}J(\theta)$ to improve the policy ($\theta \leftarrow \theta + \alpha \nabla_{\theta}J(\theta)$). Instead, we can directly use (s_t, a_t) to select our action by using an argmax operation as shown in Equation 6. In this way, we do not need to explicitly represent the policy function. The value function of action-state pairs is called the Q function. This approach is known as Q-learning.

$$\pi'(a_t | s_t) = \begin{cases} 1 & \text{if } a_t = \operatorname{argmax}_{a_t} Q(s_t, a_t) \\ 0 & \text{otherwise} \end{cases} \quad (\text{Eq. 6})$$

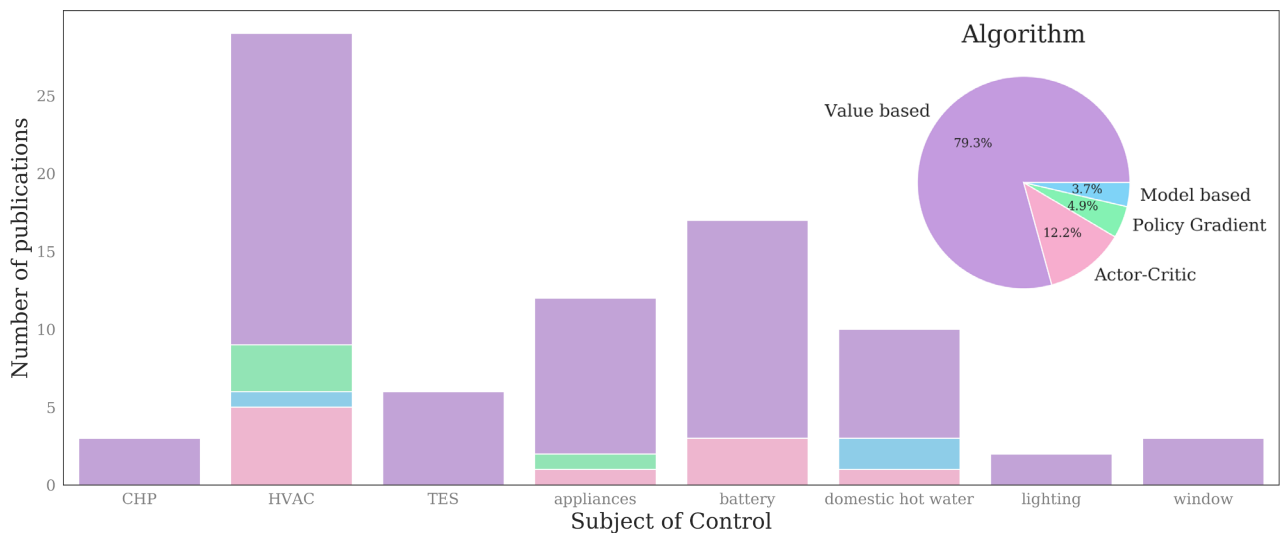
We summarized which approach is most widely used for the purpose of building controls in Table 2 and Figure 4. Table 2 excluded three reviews and one tutorial from the 77 studies. It is clear the value-based approach dominates our field. The reason is that the value approximation function is needed because it significantly reduces the variance, compared with the single-sample estimation $\sum_t r(s_t, a_t)$. However, the policy function is not necessary, as we could directly use the argmax operation to represent and improve the policy. Therefore, the value-based approach balances well the trade-off of performance and simplicity.

Table 2. RL algorithms used for building controls

Algorithm		$\pi_{\theta}(a_t s_t)$	$V_{\phi}(s_t)$ or $Q_{\phi}(s_t, a_t)$	Popularity
Model-free	Policy Gradient	√	×	3 out of 73 studies
	Value-Based	×	√	56 out of 73 studies
	Actor-Critic	√	√	11 out of 73 studies
Model-based				3 out of 73 studies



(a) algorithms by year



(b) algorithms by control subject⁴

Figure 4. Algorithms of RL for building controls

However, shown in Figure 4, the policy gradient and actor-critic approaches have become increasingly popular in recent years, especially since 2017. The reason behind this trend is researchers gradually realized an explicitly represented policy function $\pi_{\theta}(a_t|s_t)$ could help to transfer the knowledge learned from one

⁴ A paper might be counted twice if it controls more than one building system.

building to another, i.e., facilitate transfer learning. To persuade the industry to adopt RL, it is critical to convince users that what a controller learned from one building can be generalized to another. The value function (a mapping from state-action pairs to value) is not suitable to transfer, because different clients might have different control goals and utility structures. However, the policy function (a mapping from state to action) is more transferable; for instance, no matter what the goal is, turning on the heating when the indoor temperature is low remains the same for almost every building. Therefore, increasingly studies are starting to explore the possibility of using policy gradient [32] or actor-critic [35] methods to facilitate transfer learning. If an actor-critic or value-based approach is selected, the next question is how to represent the value function. The simplest idea is to use a table to record the value associated with each action-state pair. As shown in Figure 5, about 42% of studies adopted this simple solution. However, when the number of action/state variables increase, or if the state/action variables are continuous rather than discrete, storing the value of each action-state pair in a table becomes infeasible. To solve this problem, value function estimators have been proposed. The most widely used estimator is a deep neural network, thanks to the rapid development of deep learning. As observed in Figure 5, using a deep neural network as a value function approximation accounts for more than 50% of studies in this field since 2018.

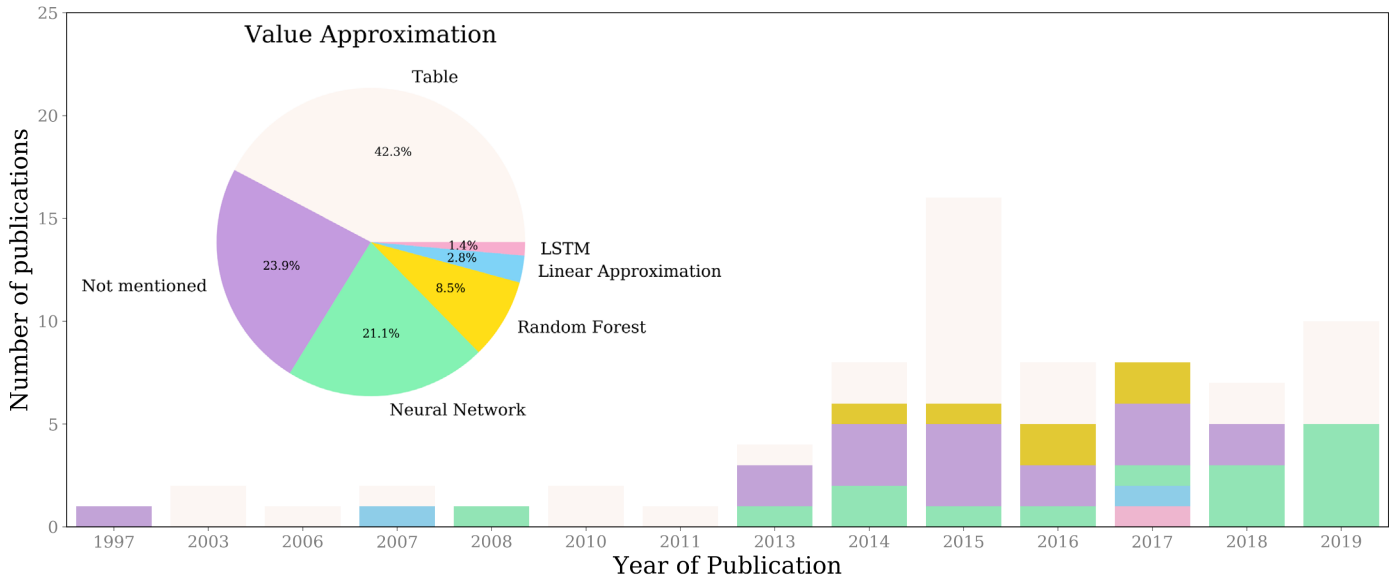


Figure 5. Value functions of RL for building controls

Another important aspect RL practitioners need to consider is the balance between exploration and exploitation. RL implements the trial-and-error approach to find the optimal control policy. It tries different control policies, evaluates them, and selects the most rewarding one. However, if the controller only focuses on improving itself using either policy gradient (Equation 5) or argmax (Equation 6), the controller might be locked in local optimal if it fails to explore the entire action space. Utilizing the currently known knowledge is called exploitation, and exploring the new action space is called exploration. A good controller needs to guarantee that it has explored the whole action space and avoided the local optimal. Two exploration strategies are popular in the RL field: ϵ -greedy and Boltzmann Exploration (a.k.a. softmax exploration).

The ϵ -greedy approach, as presented in equations 7 and 8, selects the currently known optimal action with the probability of $1 - \epsilon$ and selects a random action with the probability of ϵ . The controller with a higher ϵ explores more.

$$P(a_i = \text{argmax}(Q(a_i))) = 1 - \epsilon \quad (\text{Eq. 7})$$

$$P(a_i = \text{random}) = \epsilon \quad (\text{Eq. 8})$$

The **Boltzmann** approach, as presented in Equation 9, selects the action based on the action performance $Q(a_i)$ and τ . τ is also called the “temperature” factor, specifying how random the selection is. When τ is high, all possible actions will be explored almost equally. When τ is small, actions with high $Q(a_i)$ value are more likely to be selected.

$$P(a_i) = \frac{\exp(\frac{Q(a_i)}{\tau})}{\sum_{i=1}^n \exp(\frac{Q(a_i)}{\tau})} \quad (\text{Eq. 9})$$

In practice, the controller tends to explore more at the beginning of training and exploit more when the majority of action space has been explored already. This strategy could be easily implemented by reducing ε or τ for ε – greedy and Boltzmann Exploration, respectively.

Figure 6 surveyed the exploration methods used in the RL controller. About 60% of studies selected ε – greedy, which is three times as popular as Boltzmann Exploration. ε – greedy is more popular because its simplicity does not sacrifice its performance.

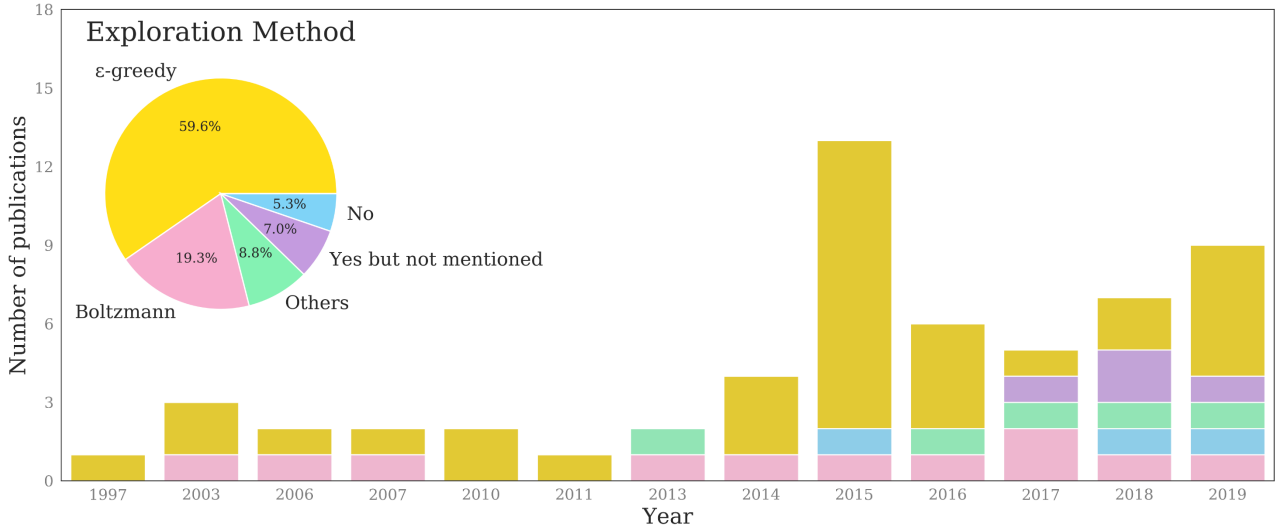


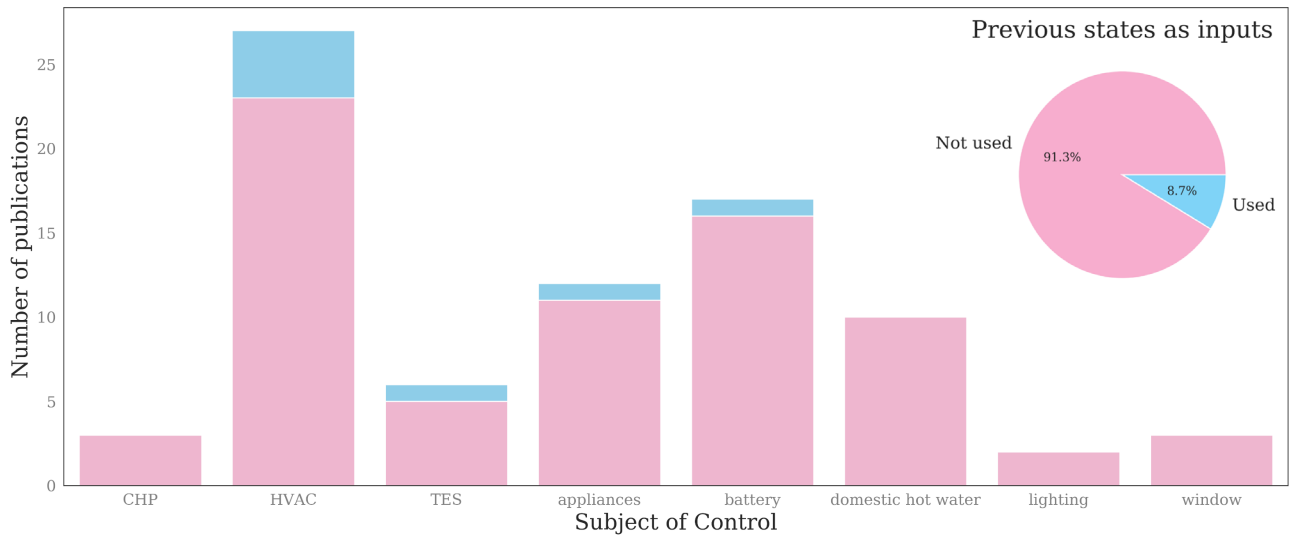
Figure 6. Exploration method of RL for building controls

3.2 States

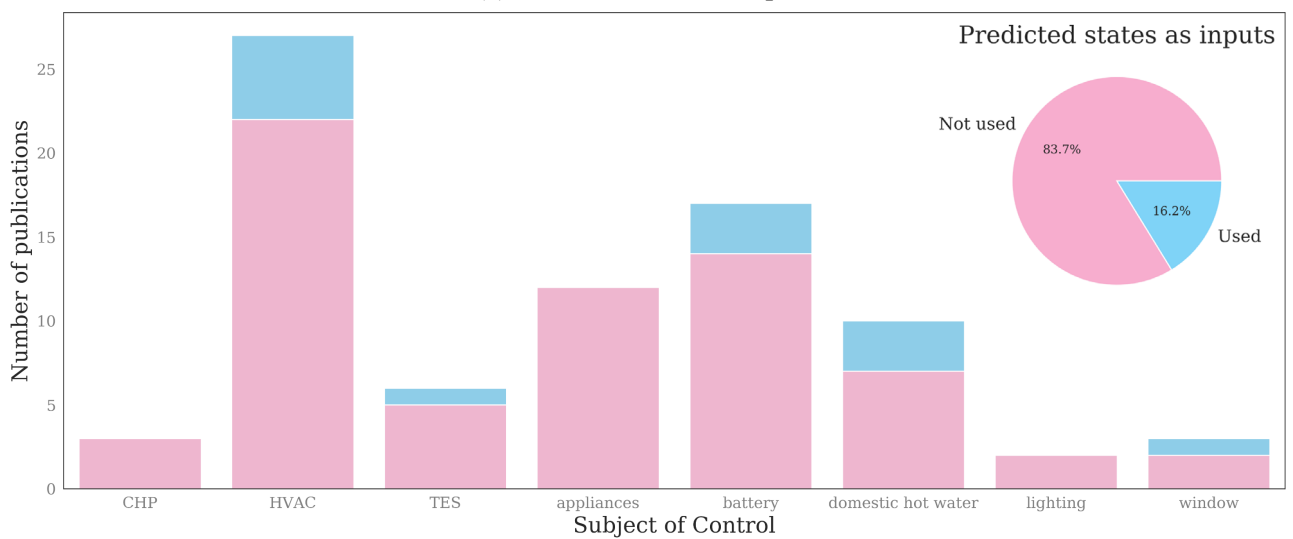
The selection of states is another crucial step for RL learning. If unnecessary states are selected, the RL controller suffers from the curse of dimensionality. Contrarily, if some important states are not selected as inputs of the controller, it is impossible for the controller to make optimal decisions, regardless of how good the algorithm is.

As introduced in Section 2, RL is mathematically formed as an MDP, in which the Markovian Property must be held. Markovian Property represents the behavior that future states purely depend on the current states, which, unfortunately, does not hold for building thermal dynamics, because of the thermal mass. To solve this problem, historical states need to be included in the MDP, especially if thermal dynamics are involved. As shown in Figure 7a, only one out of six studies of RL for HVAC control considered historical states. The remaining studies might be problematic because they train their controllers using RL but cannot guarantee that the Markovian Property holds.

One reason that the majority of studies do not consider historical states is the curse of dimensionality. For instance, in the Fuselli and De Angelis (2013) study, states of the previous two time steps were considered in the critic network, markedly increasing the number of inputs [36]. Some solutions have been proposed to address the curse of dimensionality; for example, Ruelens et al. (2015) [37] used an auto-encoder to compress the previous ten indoor temperatures and control signals into six hidden states. They then used the six hidden states to develop their RL controller. The auto-encoder is a deep neural network-based dimension reduction technique.



(a) Historical states as inputs



(b) Predicted states as inputs

Figure 7. States used in RL controller for building controls

In addition to the historical states, predicted states could also help to improve controller performance. For instance, a weather forecast could be used to inform the operation of pre-cooling or pre-heating. Integrating predicted information into control is a crucial idea introduced by MPC, and this also can be used in an RL controller. As shown in Figure 7b, only about 16% of studies use predicted information. Ruelens et al. (2016) [38] found that including weather forecasts as states could improve the performance of RL controllers by 27%. Similarly, de Gracia et al. (2015) [39] used weather forecasts to improve the energy performance of a ventilated double skin façade controller. However, energy performance is very sensitive to the accuracy of the weather forecast. Using real weather data could save 18% energy than using predicted weather data. Actual weather forecasts unavoidably have some prediction errors; how those forecast errors influence the RL performance demands further investigation.

3.3 Actions

The selection of control variables is the third decision RL practitioners need to make. Too many control points is problematic due to the curse of dimensionality. As shown in Figure 8, 70% of existing studies control fewer than four points. More than ten control points are included in 13.8% of studies, mostly because those controllers are designed for multi-building optimization.

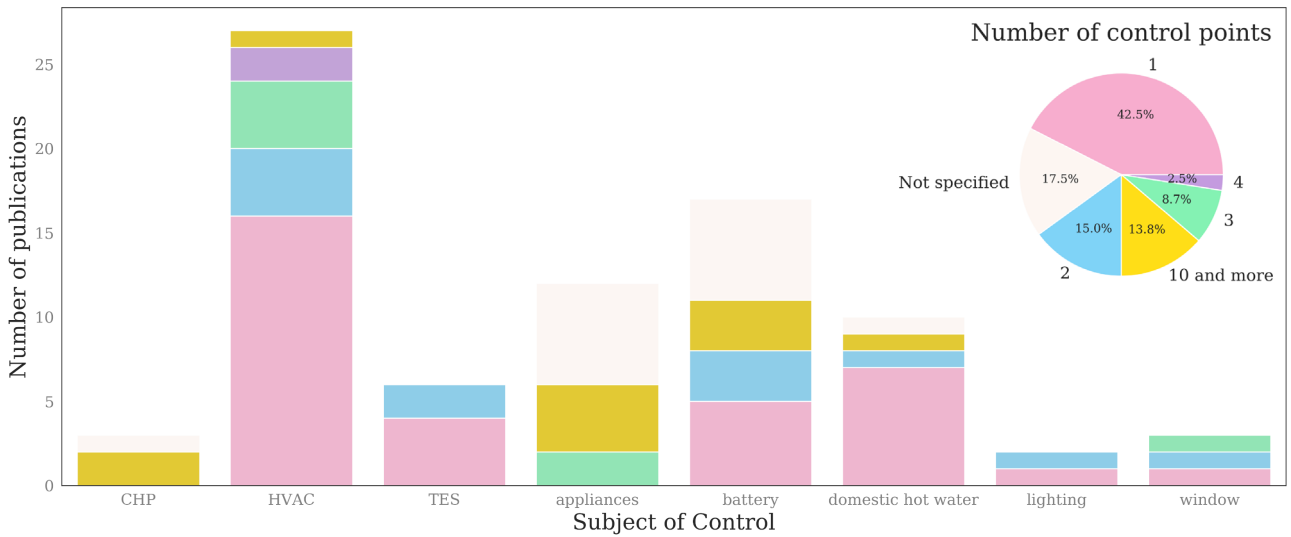




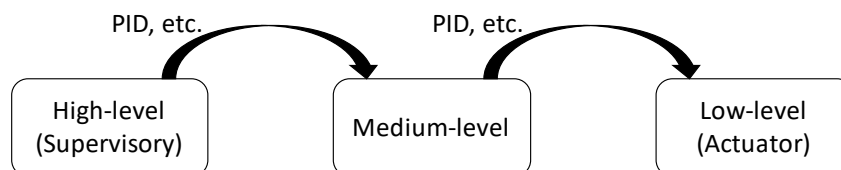


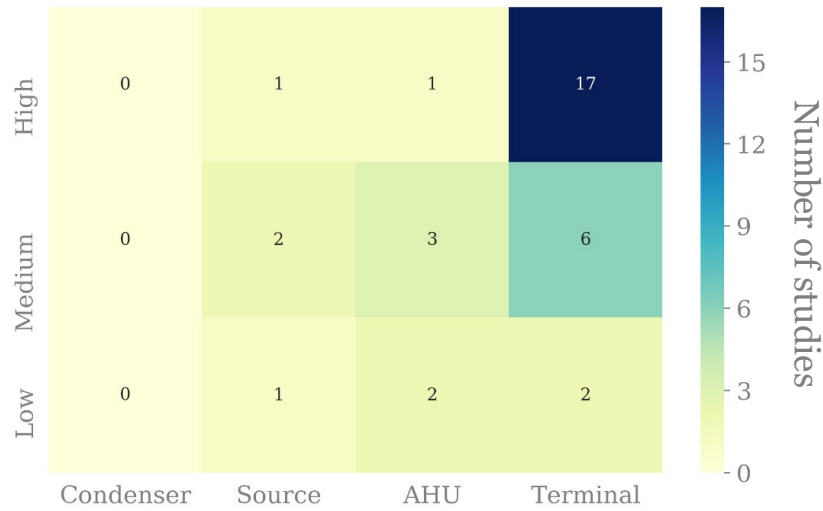
Figure 8. Number of control points of RL for building controls

HVAC control is more complicated than that for other building components, including batteries or lighting. Figure 9a illustrates why it is challenging to design an HVAC controller that could be used in every building. HVAC control is complex for two reasons. First, there are different components: a terminal, an air handling unit, a heating/cooling source, and a condenser; and for each component, there are different device types; for instance, the terminal could be a variable air volume (VAV) box or baseboard radiator. Second, for each device, there are different levels of controls. The controller could directly control the actuator level, or the setpoint (aka supervisory control). If the supervisory control is selected, conventional controllers are needed to control the actuator to track the setpoint. Figure 9b shows the control variables surveyed from existing studies. The majority of studies were controlling the HVAC terminals at a high level, i.e., the room temperature setpoint. At this level, pre-cooling or pre-heating strategies are optimized to shift the load. Additionally, six studies are about medium-level control of the terminal, such as the supply air temperature or flow rate of the VAV box. Very few current studies are controlling the actuator directly.

	 Condenser	 Source	 AHU	 Terminal
Equipment type	Cooling tower, air source, ground source			VAV box, Fan coil unit, radiator
High (ultimate goal)				Room temp sp.
Medium	Cond. water temp Cond. water flow	Supply water temp Supply water flow	Supply air temp. Supply air flow rate	Supply air temp. Supply air flow rate
Low (actuator)	Fan speed Pump speed	Compressor speed etc.	Cooling valve Economizer	Reheat valve Fan/damper



(a) Complexity of HVAC control



(b) Complexity of HVAC control

Figure 9. Control points of RL for HVAC control: temp is short for temperature, sp is short for set-point

3.4 Rewards

The reward function is designed based on the optimization goal. Figure 10 shows results from surveys of the optimization goal of existing studies. The surveys found three major goals for building controls: occupant comfort, energy conservation, and load flexibility. In this study, we consider load flexibility and cost reduction to be the same goals because cost reduction is achieved by shifting the load from the periods with high utility prices to periods with lower prices.

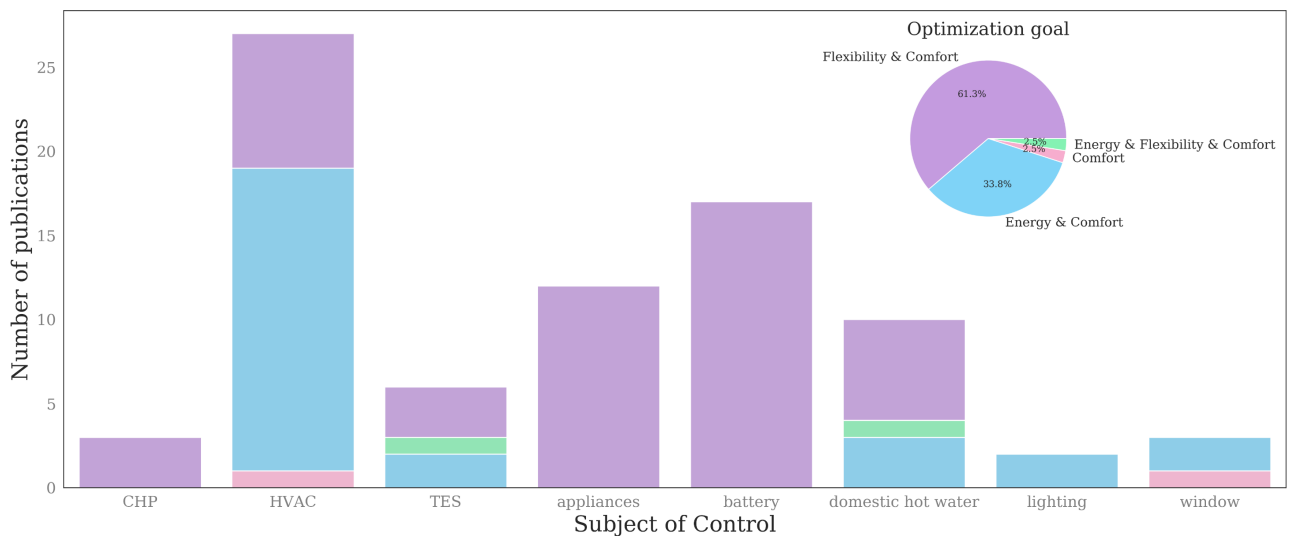


Figure 10. Optimization goal of RL controller for buildings

As observed in Figure 10, occupant comfort is a prerequisite of load flexibility and energy conservation, and all studies list occupant comfort as at least one of their control goals. Additionally, more than 60% of studies aim to enhance load flexibility. All RL controllers for combined heat and power, appliance scheduling and battery operation aim to improve load flexibility. Energy conservation is another common goal, and this goal is mostly achieved by controlling HVAC. Some other important goals, such as carbon reduction, are missing in current studies using RL for building controls.

When multiple goals exist, the next question is how the reward function should be formulated so that different goals can be considered simultaneously. The first approach is to use the weighted sum of different optimization goals. Chen et al. (2019) [32] integrated the comfort and energy targets by formulating the cost function as Equation 10, where the weight η is tunable: η has a higher value during occupied hours than non-occupied hours.

$$\min(\eta C_{comfort} + C_{energy})$$

$$(Eq. 10)$$

A second approach is to form the multi-objective optimization as a constrained optimization problem. Leurs et al. (2016) [40] set a lower and upper temperature boundary to guarantee occupant comfort. To make sure the comfort constraint would be satisfied, a conventional controller that could overwrite the RL controller when the temperature is close to or beyond the boundary was set up as a backup. The idea of a backup controller enhances the reliability of the RL controller. It also was used in Costanzo et al. (2016) [41], Ruelens et al. (2016) [42], and De Somer et al. (2017) [43]. In addition to the hard constraint, Yu and Dexter (2010) [44] implemented a soft constraint to co-optimize the comfort and energy goal by posing a penalty if the indoor temperature is outside the comfort range.

3.5 Environment

An RL controller is trained through trial-and-error approaches, which means an RL controller tries different policies, evaluates their performances, and then uses the evaluation to improve its policy. The trial-and-error method requires that the environment run the policy generated by the controller. This type of learning is called **on-policy** learning, i.e., the policy output of the controller is being carried out by the environment. However, on-policy training is challenging to implement in real buildings. A building operator would not allow an RL controller to test some random policy on an actual building because those random policies might mess up the built environment. Therefore, the idea of **off-policy** learning has been proposed. In off-policy training, controllers learn by observing the trajectory $s_1, a_1, r_1, \dots, s_T, a_T, r_T$ generated from other policies. Policy gradient and actor-critic methods require on-policy learning, while some value-based algorithms allow off-policy learning.⁵ This is one reason why a value-based approach is more popular than a policy gradient or actor-critic approach, because off-policy learning is more flexible than on-policy learning.

Though off-policy learning is more flexible, it is not as effective as on-policy learning, because the action space cannot be fully explored, and the optimal policy might be overlooked by the current policy. Additionally, training an RL controller demands a huge amount of data. Using measured data alone might be inadequate.

Therefore, some researchers use simulation to create a virtual environment. The RL controller learns by interacting with the virtual environment. Figure 11 shows results from surveys of the simulation platform used to train an RL controller. MATLAB and EnergyPlus are the most popular simulation platforms for this purpose.

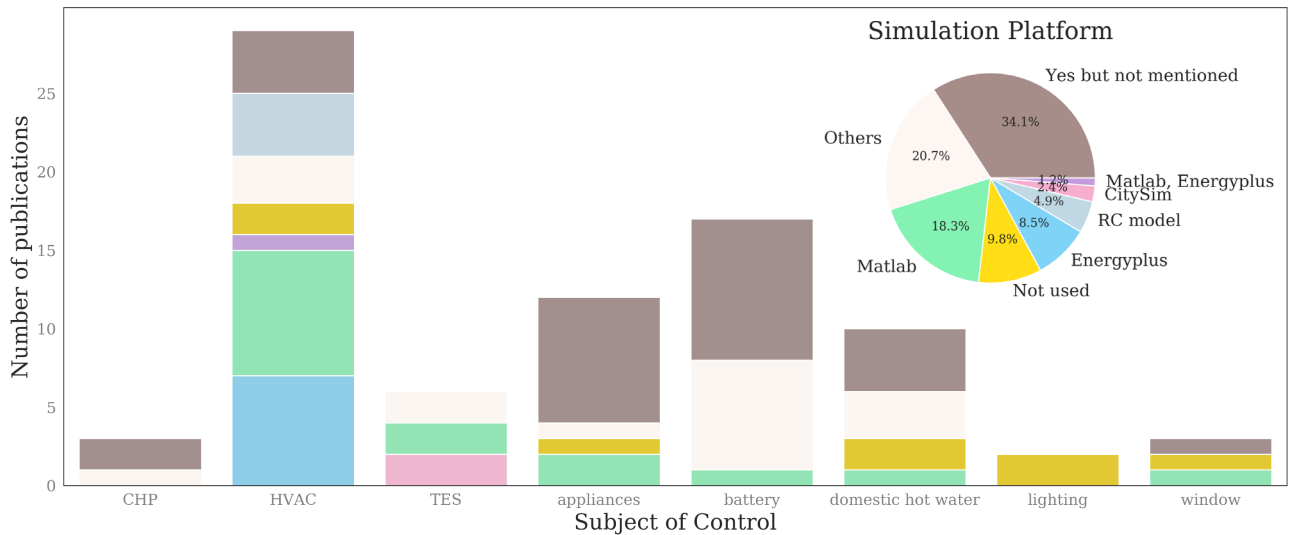


Figure 11. Environment to train the RL controller

3.6 Application in real buildings

Whether the RL controllers have been implemented in actual buildings and how they perform compared with conventional controllers are two key performance indicators of RL. Among the 77 studies reviewed in this paper, only nine controllers were implemented in real buildings: 3 domestic hot water controllers, 3 HVAC controllers, 2 lighting controllers and 1 window controller. Three studies reported energy/cost savings or comfort improvements compared with other controllers: In De Somer et al. (2017), the hot water controller

⁵ SARSA (State-Action-Reward-State-Action) is an on-policy value-based RL, while Q-learning is an off-policy value-based RL.

[38] saved operational cost by 15% after 40 days of training. In Kazmi et al. (2018), the hot water controllers were implemented in 32 Dutch houses [45] and found, compared with the fixed schedule or fixed setpoint control, the RL controller reduced energy consumption by almost 20% while maintaining occupant comfort. Extrapolated to a year, the RL controller has the potential to reduce household energy consumption by up to 200 kilowatt-hours (kWh). May (2019) [4] compared an RL controller with the manual occupant control of windows and found the RL controller could significantly improve thermal comfort and indoor air quality by 90%. However, how the improvements were quantified was not described in detail.

3.7 Discount factor

Another component of a typical MDP that was not discussed in Section 2 is the discount factor γ . The discount factor will revise the accumulated rewards to $E_{r \sim p_{\theta}(r)}[\sum_t \gamma^{t-1} r(s_t, a_t)]$. As γ is usually less than 1, future rewards are not as valuable as current rewards.

The discount factor is introduced mostly to guarantee convergence of MDP. Though primarily for mathematical purposes, the discount factor could be explained in two ways: (1) immediate rewards are more valuable because immediate rewards can generate interest if the reward is in a monetary form, and (2) due to the existence of uncertainty, future benefits are associated with higher risks, and accordingly need to be discounted. Because if you are uncertain about what will happen in the future, it is not a bad idea to discount future potential rewards a bit.

Even though the discount factor seems to make sense in some way, determining the proper value of γ remains a question. Vázquez-Canteli et al. (2017) [46] discussed how different values of γ would influence the behaviors of an RL controller for a heat pump. Higher γ values assign greater importance to achieve long-term rewards and accordingly lead to more frequent operation of heat pumps when the outdoor temperature is high (higher coefficient of performance [COP]). Pre-heating during the periods with a high COP consumes more energy at the current time step, while saving energy for upcoming time steps. If γ is small, the discounted future savings could not justify the current costs. Therefore, pre-heating could happen more only when γ is adequately high.

4 Discussion

4.1 Accelerate training

As introduced in the previous section, training an RL controller is data- and time-demanding. An early study conducted by Henze and Dodier (2003) [47] showed 30 years of data was required to train the RL controller. With the advancement of the RL algorithm, the size of the training data has reduced significantly. The Yang et al. (2015) study found three years of training data to be adequate to guarantee that RL controllers outperform rule-based controllers [48]. How to use fewer data to achieve high performance with less training time is a crucial research question in this field.

The first approach researchers have proposed to accelerate training is to reduce the dimension of state-action space. The optimization problem becomes more complicated when the number of state and action variables increases, which demands more data and longer training time. Guan et al. (2015) [49] used one state variable, the net power (defined as the difference between the electricity load and PV generation), to replace two state variables, load and generation. Additionally, auto-encoder, a neural network-based dimension reduction technique, was employed in the Ruelens et al. (2016) [42] study. Zhou et al. (2019) [50] implemented Fuzzy Q-learning, which used fuzzy rules to discretize continuous states-actions and to reduce dimensions. Yoon and Moon (2019) [51] used Gaussian Process Regression (GPR) to compress six states into two.

The second approach is to decouple a complicated problem into multiple simpler problems. Ruelens et al. (2014) [21] decoupled a complicated problem with multiple action variables into multiple sub-problems, where each problem contained only one action variable. The multiple sub-problems communicate and collaborate in a multi-agent system. The authors claimed this approach brought in two benefits. First, the problem is more tractable and faster to train, as the number of state and action variables decreases for each sub-problem. Second, it provides a realistic decentralized solution with good scalability qualities. We will discuss the decentralized controller in more detail in Section 4.4. Zhang and van der Schaar (2014) [52] proposed the idea of decoupling the system dynamics into the known part and the unknown part, to speed the training. By decomposing the transition dynamics into these two parts, only the unknown part of the dynamics needs to be learned. By exploiting the partial information about the system dynamics that is already known, the convergence speed was increased by 30% compared to conventional Q-learning.

Similarly, Kim et al. (2015) [53] also defined and used post-decision state (PDS) to better utilize the known dynamics, using RL to only learn unknown dynamics. The third way to decouple a complicated problem is to

leverage supervisor control, i.e., controlling the setpoint only and leaving the setpoint tracking to conventional controllers such as a PID. In this way, the complexity of controlling the actuator could be left aside. As a result, fewer data and time are needed to train the controller.

Li and Xia (2015) [54] used a multi-stage approach to speed the training. The idea is to first discretize the state-action space at low grid density, once the RL controller converges at the coarse discretization, then to implement a finer discretization. The simulation result shows that the multi-grid method helps accelerate the convergence of Q-learning. It is suggested to be applied to other problems where Q-learning of a single grid density is unsatisfying.

Sun et al. (2013) [55] proposed an event-based approach to accelerate training. The event-based approach only updates decision variables when certain events happen. The events are defined as a set of transitions of disturbance variables such as occupancy, outdoor weather, and energy price. The authors claim this approach is effective and that it can save 70% of computational time for a building with 24 rooms. However, the event-based approach could only find the suboptimal solution, which consumes 0.5% more energy than the traditional RL controller. The event-based approach was also adopted in the Sun et al. (2015) study to accelerate training [56].

4.2 Control security/safety/robustness

As RL controllers learn the optimal policy by testing new policies and evaluating the outcomes, it is possible that some tested policies might lead to an undesirable outcome, such as too cold or too hot temperatures. The control security/safety/robustness in this section refers to minimizing or even eliminating the chance of generating control signals that might lead to undesirable outcomes during the training, testing, or implementation phases. How to avoid those undesirable outcomes and guarantee control security is a key challenge of implementing RL in real buildings.

A commonly used approach to enhance control security is to set up a backup controller, like in studies [40], [41], [42] and [43]. When the temperature is close to or about to go beyond the comfort boundary, the backup controller is activated to overwrite the RL controller.

The second approach is to pre-train the controller to make it safe enough to be implemented in real buildings. We could use simulators rather than real buildings to pre-train the controllers. Or, we could use some “expert” knowledge to pre-train the controller. The “expert” knowledge could be a common practice or industry standard in this field. The Jia et al. (2018) study found that, with some guidance from “expert” policy, the RL controller’s performance could be significantly improved [57].

Additionally, we could also use the optimal policy resulting from other optimization methods to pre-train the controller. Fuselli and De Angelis (2013) [36] used the optimal solution found from Particle Swarm Optimization (PSO)⁶ to pre-train the actor network. Wang et al. (2015) [59] first optimized the charging/discharging of the storage system by solving a model-based convex optimization problem, and then used the optimized results to pre-train the RL. Chen et al. (2019) [32] used the optimization result of MPC to pre-train the actor network.

As the simulator or the model-based optimization is used only for pre-training, either the simulator or the model does not need to be very accurate. It is okay if the optimal policy found from other optimization methods is not the global optimal solution because RL will further improve the policy by fine-tuning it to better adapt to the environment.

4.3 Multi-agent problem

Multi-agent systems are common for building- or campus-level control problems. Based on the information availability and optimization goal, there are four different types of multi-agent optimization problems [60]:

- **Centralized:** One agent with available information about the whole environment makes decisions.
- **Decentralized:** Multiple agents who only perceive their environments make decisions.
- **Cooperative:** Agents are allowed to share their observations about the environment, and this type aims to maximize the rewards of all agents.
- **Non-cooperative:** Agents do not share observations, and only consider their interests.

Ruelens et al. (2014) [21] proposed a decentralized multi-agent solution to coordinate the operation of domestic

⁶ PSO is a technique developed by Eberhard and Kennedy and inspired by certain social behaviors exhibited in bird and fish groups that is used to explore a solutions space for finding parameters that are required to optimize a specific aspect of the problem [58].

hot water heating of multiple households. Raju et al. (2015) [61] proposed Coordinated Q-Learning to optimize the micro-grid operation. In the Raju et al. (2015) framework, the agent first learns optimal single-agent policy when acting alone in the environment using conventional Q-learning. Then the coordinated Q-learning algorithm detects if any other agents' operation would lead to any reward changes for the selected state-action pair. If no reward changes occur, then this state-action pair is marked as a "safe" state, and the Q-value does not need to be updated. If any changes are detected in the rewards, then this state-action pair is marked as a "dangerous" state, where the Q-value and corresponding optimal action will depend on other agents. For "dangerous" states, any interference from other agents will be reflected in the rewards. The Q-values of "dangerous" states need to be updated accordingly. The authors argued that distinguishing between "safe" and "dangerous" states could save computation by avoiding recalculating the Q-value of "safe" state-action pairs. Sun et al. (2015) explored the possibility of using the Lagrangian Relaxation (LR)-based method to co-optimize the fresh air unit (FAU) at the building level and the fan coil unit (FCU) at the room level. The Lagrangian multipliers were introduced to decouple the two-level problem into sub-problems of FAU control and FCU control. The Lagrangian multipliers would be updated in a building-level dual problem to make sure the chiller capacity constraint is being considered, and finally, be forced with iteration.

4.4 Performance evaluation

Given that there are so many approaches, as introduced in Section 3, a natural question is which performs better under the context of building controls. Al-Jabery et al. (2016) [62] compared the actor-critic approach with the value-based approach for domestic hot water control and found Q-learning performs better (\$466 annual savings compared with \$367). Al-Jabery et al. (2016) [62] claimed that Q-learning is simpler, more robust, and more easily deployable. Mocanu et al. (2018) [63] found both Deep Q-learning and Deep Policy Gradient performs better than the tabular Q-learning, and Deep Policy Gradient is more suited to perform scheduling of energy resources than Deep Q-learning.

In terms of cross-study comparison, different studies use different benchmarks to evaluate the performance of their RL controller, making cross-study comparison difficult. Yang et al. (2015) [48] compared the RL controller with a "rule-based controller." Barrett and Linder (2015) [64] selected the "always on" and "programmable control" as comparison benchmarks. Wang et al. (2017) [65] compared their RL controller with a "fixed setpoint" controller. Similarly, the Kazmi et al. (2018) [45] controller is compared with a "fixed schedule, fixed setpoint" control. Chen et al. (2018) [66] benchmarked their controller with a "rule-based heuristic" control strategy. Kazmi et al. (2019) [67] used a rule-based dead-band controller as the benchmark. Ahn and Park (2019) [68] claimed their controller saved 15.7% energy compared with the fixed pre-determined schedule on OA damper position and temperature setpoint. Different studies use different comparison baselines, and some of those baselines are too simple to justify the performance of an RL controller. To enable the selection and performance comparison of a different RL controller, an open-sourced and well-recognized control testbed is needed. One reason why RL developed fast in the past decade is that OpenAI Gym provides a common benchmark with a wide variety of different environments [69]. A similar platform in the building controls area is needed. The good news is some efforts have been taken. For example, the Open Building controls project is targeting this goal [70]. Additionally, the OpenAI Gym environment, CityLearn, was developed and open-sourced by the UT Austin team, for the easy implementation of reinforcement learning controllers in a multi-agent demand response setting [71]. The CityLearn Challenge has been announced and aims to compare different RL algorithms that are capable of coordinating multiple buildings to maximize demand response potential [72].

4.5 Contribution and implication

In this study, we conducted a comprehensive review of studies applying RL for building controls. We dove deep into subtle but important choices researchers need to make to develop an RL controller, such as how the state and action space is determined, how the reward function is designed, which algorithm is used, where the training data come from, and other factors. This comprehensive review helps researchers better understand the general RL framework, and more importantly, the progress and challenges of applying RL for building controls, as well as helps identify the research gap and design specific RL controllers.

5 Conclusion

This article reviewed a broad set of studies using reinforcement learning for building controls. The number of papers published on this topic jumped in 2015 and has remained stable since then. Reinforcement learning has demonstrated its potential to enhance building controls, although some key challenges remain to be addressed. We surveyed existing studies from five perspectives: algorithms, states, actions, rewards, and the environment, each corresponding to one of the five key components of an RL controller. Significant findings include (1)

Algorithm: 77% of existing studies used value-based RL algorithms, among which Q-learning is the most popular. Actor-critic and policy gradient approaches have become more frequently used since 2017 due to their ability to facilitate transfer learning. (2) **States:** 91% of studies did not include historical states. As the Markovian property might not hold in building thermal dynamics, the RL controller might fail to converge to the optimal. 83% of the studies did not include predicted states, even though the predicted states (e.g., weather forecast) might provide valuable information for optimization. (3) **Actions:** Fewer than four variables were controlled in 69% of the studies. A majority of HVAC controllers adopted supervisory control, i.e., controlling the setpoint rather than controlling the actuator directly. In this case, conventional controllers are still needed to track the setpoint. (4) **Rewards:** 97% of studies have multiple objectives: either enhancing flexibility (61%) or conserving energy (34%) or both (3%) while maintaining occupant comfort. (5) **Environments:** 90% of studies used simulators to generate data for training the RL controller.

Even though RL-based building controls have attracted increasing research interest, the RL controller is still in the research and development stage, with limited adoption in actual buildings. Among the 77 studies we surveyed, only 11% of RL controllers were implemented and tested in an actual building. Significant barriers limiting the applications of RL controller for real building controls include (1) the training process is time-consuming and data-demanding, (2) the security of controls needs to be addressed, i.e., making sure the RL controller would not mess up the building controls, especially during the training stage, (3) it is yet unknown how to implement the transfer learning so that controllers trained by a small number of buildings could be generalized and used for other buildings, and (4) a data-rich, open-sourced, and interoperable virtual testbed is needed to facilitate cross-study validations and benchmarking of performance of RL controllers.

Conflict of Interest

The authors declare no competing interest in the reported work.

Acknowledgments

This work was supported by Laboratory Directed Research and Development (LDRD) funding from Lawrence Berkeley National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Appendix

Table A1. Summary of existing studies on reinforcement learning for building controls

	Control objectives	Control subject	Algorithm	Exploration	Simulation environment	Length of data for training	Implementation in real buildings
Anderson et al. (1997) [73]	Energy & Comfort	HVAC	Value iteration	ϵ -greedy	Not introduced in detail		No
Henze and Dodier (2003) [47]	Flexibility & Comfort	battery, PV	Tabular Q-learning	Boltzmann, ϵ -greedy	Not introduced in detail	30 years	No
Henze and Schoenmann (2003) [74]	Flexibility & Comfort	TES	Tabular Q-learning	ϵ -greedy	Not introduced in detail		No
Liu and Henze (2006) [75] [22]	Flexibility & Comfort	HVAC, TES	Tabular Q-learning	Boltzmann, ϵ -greedy	Matlab/Simulink	3000-6000 days	No
Liu and Henze (2007) [76]	Flexibility & Comfort	HVAC, TES	Tabular Q-learning	Boltzmann	Matlab/Simulink implementing RC (2R3C) model	6000 days	No
Dalamagkidis et al. (2007) [19]	Energy & Comfort	HVAC, window		ϵ -greedy	Matlab/Simulink	4 years	No
Du and Fei (2008) [77]	Energy & Comfort	HVAC			Not introduced in detail		No
O'Neill et al. (2010) [18]	Flexibility & Comfort	appliances	Tabular Q-learning	ϵ -greedy	Not introduced in detail		No
Yu and Dexter (2010) [44]	Energy & Comfort	HVAC	Fuzzy Q-learning	ϵ -greedy		30 days	No
Jiang and Fei (2011) [78]	Flexibility & Comfort	CHP, battery	Tabular Q-learning	ϵ -greedy	Not introduced in detail		No
Liang et al. (2013) [79]	Flexibility & Comfort	appliances	Temporal Difference Learning	boltzmann			No
Kaliappan and Sathiakumar (2013) [80]	Flexibility & Comfort	appliances			Matlab		No
Fuselli and De Angelis (2013) [36]	Flexibility & Comfort	battery		Random perturbation	Not introduced in detail		No
Sun et al. (2013) [55]	Flexibility & Comfort	HVAC	Tabular Q-learning		Matlab		No
Li and Jayaweera (2013) [81]	Flexibility & Comfort	appliances	Tabular Q-learning	ϵ -greedy		10000 time steps	No
Wei and Liu (2014) [82]	Flexibility & Comfort	battery			Not introduced in detail		No
Zhang and van der Schaar (2014) [52]	Flexibility & Comfort	battery			Not introduced in detail		No
Wei and Liu (2014) [20]	Flexibility & Comfort	battery					No
Li and Jayaweera (2014) [83]	Flexibility & Comfort	battery		ϵ -greedy		10000 time steps	No
Jiang and Fei (2014) [23]	Flexibility & Comfort	CHP, appliances, battery		ϵ -greedy	self-coded in Java		No
Ruelens et al. (2014) [21]	Flexibility & Comfort	domestic hot water	Fitted Q-iteration	boltzmann		40-45 days	No
Fazenda and Veeramachaneni (2014) [84]	Energy & Comfort	HVAC	Wire fitted neural network		Matlab	50 days	No
Wen et al. (2015) [85]	Flexibility & Comfort	appliances	Tabular Q-learning				No
Kim et al. (2015) [53]	Flexibility & Comfort	appliances		Not used			No
Rayati et al. (2015) [86]	Flexibility & Comfort	appliances, CHP, domestic hot water		ϵ -greedy	Not introduced in detail		No
Wang et al. (2015)	Flexibility	battery	Fitted Q-		Matlab		No

[59]	& Comfort		iteration				
Raju et al. (2015) [61]	Flexibility & Comfort	battery	Coordinated Q-learning		self-coded in Python		No
Berlink et al. (2015) [87]	Flexibility & Comfort	battery	Tabular Q-learning	ϵ -greedy	Simulation with historical data		No
Guan et al. (2015) [49]	Flexibility & Comfort	battery	Temperoral Difference	ϵ -greedy	Simulation with historical data		No
Qiu et al. (2015) [88]	Flexibility & Comfort	battery	Tabular Q-learning	ϵ -greedy	Simulation with historical data		No
Sekizaki et al. (2015) [89]	Flexibility & Comfort	battery, domestic hot water		ϵ -greedy	Simulation with historical data		No
Yang et al. (2015) [48]	Energy & Comfort	HVAC	Batch Q-learning with Memory Replay	ϵ -greedy	Matlab/Simulink	3 years	No
Ruelens et al. (2015) [37]	Energy & Comfort	HVAC	Fitted Q-iteration	Boltzmann	RC model (1R1C for air and the building envelope)		No
Barrett and Linder (2015) [64]	Energy & Comfort	HVAC	Tabular Q-learning	ϵ -greedy	Not introduced in detail		No
Li and Xia (2015) [54]	Energy & Comfort	HVAC	Tabular Q-learning	ϵ -greedy	Matlab, Energyplus		No
Sun et al. (2015) [90]	Flexibility & Comfort	HVAC	Tabular Q-learning	ϵ -greedy	Matlab		No
Sun et al. (2015) [56]	Energy & Comfort	HVAC	Tabular Q-learning	ϵ -greedy	Matlab		No
de Gracia et al. (2015) [39]	Energy & Comfort	TES	SARSA	ϵ -greedy	Self-coded numerical equation		No
Sheikhi et al. (2016) [91]	Flexibility & Comfort	appliances	Tabular Q-learning	ϵ -greedy	Matlab		No
Kazmi et al. (2016) [92]	Energy & Comfort	domestic hot water	Hybrid Ant-Colony Optimization was used to find the optimal control solution				Yes
Al-Jabery et al. (2016) [62]	Flexibility & Comfort	domestic hot water		Random selection	Matlab		No
Ruelens et al. (2016) [42]	Flexibility & Comfort	domestic hot water	Fitted Q-iteration	Boltzmann		40 days	Yes
Leurs et al. (2016) [40]	Flexibility & Comfort	HVAC	Fitted Q-iteration	Boltzmann	RC model (2R2C)		No
Ruelens et al. (2016) [38]	Flexibility & Comfort	HVAC	Fitted Q-iteration	ϵ -greedy	RC Model (Second order)		No
Costanzo et al. (2016) [41]	Flexibility & Comfort	HVAC	Fitted Q-iteration	ϵ -greedy	RC model	20 days	Yes
Cheng et al. (2016) [24]	Energy & Comfort	lighting	Tabular Q-learning	ϵ -greedy	Not used		Yes
Bahrami et al. (2017) [93]	Flexibility & Comfort	appliances		Boltzmann	Not introduced in detail		No
Mbuwir et al. (2017) [94]	Flexibility & Comfort	battery	Fitted Q-iteration		Simulation with historical data		No
De Somer et al. (2017) [43]	Flexibility & Comfort	domestic hot water	Fitted Q-iteration	Not introduced in detail	Simulation with historical data	2 months	No
Wang et al. (2017) [65]	Energy & Comfort	HVAC			EnergyPlus		No
Schmidt et al. (2017) [95]	Energy & Comfort	HVAC	Fitted Q-iteration	Gaussian noise			No
Vázquez-Canteli et al. (2017) [46]	Energy & Comfort	TES	Fitted Q-iteration	Boltzmann	CitySim		No
Hurtado et al. (2017) [60]	Flexibility & Comfort				Matlab/Simulink		No
Zhang et al. (2017) [35]	Flexibility & Comfort			ϵ -greedy	Not introduced in detail		No

Remani et al. (2018) [96]	Flexibility & Comfort	appliances	Fitted Q-iteration	ϵ -greedy	Not introduced in detail		No
Kazmi et al. (2018) [45]	Energy & Comfort	domestic hot water		as part of the reward function	Not used		Yes
Claessens et al. (2018) [97]	Energy & Flexibility & Comfort	district heating	Fitted Q-iteration	Boltzmann	Not introduced in detail	60 days	No
Kontes et al. (2018) [98]	Energy & Comfort	HVAC		Not introduced in detail	EnergyPlus		No
Zhang et al. (2018) [99]	Energy & Comfort	HVAC		Not introduced in detail	Not used		No
Jia et al. (2018) [57]	Energy & Comfort	HVAC		Not used	EnergyPlus		No
Mocanu et al. (2018) [63]	Flexibility & Comfort	HVAC, appliances	Deep Q-learning, Deep policy gradient	Not used	Not used		Yes
Chen et al. (2018) [66]	Energy & Comfort	HVAC, window	Q-learning	ϵ -greedy	Not introduced in detail		No
Zhou et al. (2019) [50]	Flexibility & Comfort	battery	Fuzzy Q-learning	ϵ -greedy	Not introduced in detail		No
Odonkor et al. (2019) [100]	Flexibility & Comfort	battery	Deep Deterministic Policy Gradients with experience replay		Not introduced in detail		No
Kazmi et al. (2019) [67]	Energy & Comfort	domestic hot water	Monte Carlo with Exploring Starts	ϵ -greedy	Not introduced in detail		No
Ahn and Park (2019) [68]	Energy & Comfort	HVAC		ϵ -greedy	EnergyPlus on DOE reference model		No
Zhang et al. (2019) [101]	Energy & Comfort	HVAC	Asynchronous advantage actor- critic	ϵ -greedy	EnergyPlus on real buildings		No
Lu et al. (2019) [102]	Comfort	HVAC	Tabular Q-learning	ϵ -greedy	ASHRAE database		No
Yoon and Moon (2019) [51]	Energy & Comfort	HVAC	Double Deep Q-learning with experience replay	Not introduced in detail	EnergyPlus		No
Chen et al. (2019) [32]	Energy & Comfort	HVAC	Differentiable MPC; REINFORCE	Not used	EnergyPlus		Yes
Park et al. (2019) [103]	Energy & Comfort	lighting	Value iteration	limited state and action, explore all of them	Not used		Yes
Vázquez-Canteli et al. (2019) [104]	Energy & Flexibility & Comfort	TES	Fitted Q-iteration	Boltzmann	CitySim	40 days	No
May (2019) [4]	Comfort	window	SARSA	Not used	Not used		Yes

References

- [1] N. E. Klepeis *et al.*, "The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants," *J. Expo. Sci. Environ. Epidemiol.*, vol. 11, no. 3, pp. 231–252, Jul. 2001, doi: 10.1038/sj.jea.7500165.
- [2] U. S. Energy Information Administration, "Monthly Energy Review November 2019." US EIA, Nov-2019, [Online]. Available: https://www.eia.gov/totalenergy/data/monthly/pdf/sec2_3.pdf.
- [3] A. Roth and J. Reyna, "Grid-Interactive Efficient Buildings Technical Report Series: Whole-Building Controls, Sensors, Modeling, and Analytics," NREL/TP-5500-75478, DOE/GO-102019-5230, 1580329, Dec. 2019. doi: 10.2172/1580329.
- [4] R. May, "The reinforcement learning method : A feasible and sustainable control strategy for efficient occupant-centred building operation in smart cities," 2019, Accessed: 23-Dec-2019. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:du-30613>.
- [5] Guang Geng and G. M. Geary, "On performance and tuning of PID controllers in HVAC systems," in *Proceedings of IEEE International Conference on Control and Applications*, 1993, pp. 819–824 vol.2, doi: 10.1109/CCA.1993.348229.
- [6] The American Society of Heating, Refrigerating and Air-Conditioning Engineers, "Guideline 36-2018. High Performance Sequences of Operation for HVAC Systems." A.S.H.R.A.E., 2018.
- [7] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4, pp. 667–682, May 1999, doi: 10.1016/S0098-1354(98)00301-9.
- [8] S. Prívvara, J. Široký, L. Ferkl, and J. Cigler, "Model predictive control of a building heating system: The first experience," *Energy Build.*, vol. 43, no. 2, pp. 564–572, Feb. 2011, doi: 10.1016/j.enbuild.2010.10.022.
- [9] H. Karlsson and C.-E. Hagentoft, "Application of model based predictive control for water-based floor heating in low energy residential buildings," *Build. Environ.*, vol. 46, no. 3, pp. 556–569, Mar. 2011, doi: 10.1016/j.buildenv.2010.08.014.
- [10] I. Hazyuk, C. Ghiaus, and D. Penhouet, "Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part II – Control algorithm," *Build. Environ.*, vol. 51, pp. 388–394, May 2012, doi: 10.1016/j.buildenv.2011.11.008.
- [11] S. Yuan and R. Perez, "Multiple-zone ventilation and temperature control of a single-duct VAV system using model predictive strategy," *Energy Build.*, vol. 38, no. 10, pp. 1248–1261, Oct. 2006, doi: 10.1016/j.enbuild.2006.03.007.
- [12] Y. Ma, F. Borrelli, B. Hancey, A. Packard, and S. Bortoff, "Model Predictive Control of thermal energy storage in building cooling systems," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 392–397, doi: 10.1109/CDC.2009.5400677.
- [13] B. Paris, J. Eynard, S. Grieu, T. Talbert, and M. Polit, "Heating control schemes for energy management in buildings," *Energy Build.*, vol. 42, no. 10, pp. 1908–1917, Oct. 2010, doi: 10.1016/j.enbuild.2010.05.027.
- [14] G. D. Kontes *et al.*, "Simulation-Based Evaluation and Optimization of Control Strategies in Buildings," *Energies*, vol. 11, no. 12, p. 3376, Dec. 2018, doi: 10.3390/en1123376.
- [15] T. Hong, Z. Wang, X. Luo, and W. Zhang, "State-of-the-Art on Research and Applications of Machine Learning in the Building Life Cycle," *Energy Build.*, p. 109831, Feb. 2020, doi: 10.1016/j.enbuild.2020.109831.
- [16] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.
- [17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4–5, pp. 421–436, Apr. 2018, doi: 10.1177/0278364917710318.
- [18] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential Demand Response Using Reinforcement Learning," in *2010 First IEEE International Conference on Smart Grid Communications*, 2010, pp. 409–414, doi: 10.1109/SMARTGRID.2010.5622078.
- [19] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Build. Environ.*, vol. 42, no. 7, pp. 2686–2698, Jul. 2007, doi: 10.1016/j.buildenv.2006.07.010.
- [20] Q. Wei, D. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart

residential environments," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2509–2518, Apr. 2015, doi: 10.1109/TIE.2014.2361485.

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

[22] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis," *Energy Build.*, vol. 38, no. 2, pp. 148–161, Feb. 2006, doi: 10.1016/j.enbuild.2005.06.001.

[23] B. Jiang and Y. Fei, "Smart Home in Smart Microgrid: A Cost-Effective Energy Ecosystem With Intelligent Hierarchical Agents," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 3–13, Jan. 2015, doi: 10.1109/TSG.2014.2347043.

[24] Z. Cheng, Q. Zhao, F. Wang, Y. Jiang, L. Xia, and J. Ding, "Satisfaction based Q-learning for integrated lighting and blind control," *Energy Build.*, vol. 127, pp. 43–55, Sep. 2016, doi: 10.1016/j.enbuild.2016.05.067.

[25] M. Han *et al.*, "A review of reinforcement learning methodologies for controlling occupant comfort in buildings," *Sustain. Cities Soc.*, vol. 51, p. 101748, Nov. 2019, doi: 10.1016/j.scs.2019.101748.

[26] K. Mason and S. Grijalva, "A Review of Reinforcement Learning for Autonomous Building Energy Management," *ArXiv190305196 Cs Stat*, Mar. 2019, Accessed: 26-Nov-2019. [Online]. Available: <http://arxiv.org/abs/1903.05196>.

[27] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains: A Survey," *J. Mach. Learn. Res.*, vol. 10, no. Jul, pp. 1633–1685, 2009.

[28] Y. Chen, Z. Tong, Y. Zheng, H. Samuelson, and L. Norford, "Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings," *J. Clean. Prod.*, vol. 254, p. 119866, May 2020, doi: 10.1016/j.jclepro.2019.119866.

[29] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Appl. Energy*, vol. 235, pp. 1072–1089, Feb. 2019, doi: 10.1016/j.apenergy.2018.11.002.

[30] D. H. Blum, K. Arendt, L. Rivalin, M. A. Piette, M. Wetter, and C. T. Veje, "Practical factors of envelope model setup and their effects on the performance of model predictive control for building heating, ventilating, and air conditioning systems," *Appl. Energy*, vol. 236, pp. 410–425, Feb. 2019, doi: 10.1016/j.apenergy.2018.11.093.

[31] Y. Chen, Z. Tong, W. Wu, H. Samuelson, A. Malkawi, and L. Norford, "Achieving natural ventilation potential in practice: Control schemes and levels of automation," *Appl. Energy*, vol. 235, pp. 1141–1152, Feb. 2019, doi: 10.1016/j.apenergy.2018.11.016.

[32] B. Chen, Z. Cai, and M. Bergés, "Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, New York, NY, USA, 2019, pp. 316–325, doi: 10.1145/3360322.3360849.

[33] S. Levine, "CS 285: Deep Reinforcement Learning," *CS 285 at UC Berkeley: Deep Reinforcement Learning*. <http://rail.eecs.berkeley.edu/deeprlcourse/> (accessed Jan. 02, 2020).

[34] A. Güne, G. Baydin, B. A. Pearlmutter, and J. M. Siskind, "Automatic Differentiation in Machine Learning: a Survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, 2018.

[35] X. Zhang, T. Bao, T. Yu, B. Yang, and C. Han, "Deep transfer Q-learning with virtual leader-follower for supply-demand Stackelberg game of smart grid," *Energy*, vol. 133, pp. 348–365, Aug. 2017, doi: 10.1016/j.energy.2017.05.114.

[36] D. Fuselli *et al.*, "Action dependent heuristic dynamic programming for home energy resource scheduling," *Int. J. Electr. Power Energy Syst.*, vol. 48, pp. 148–160, Jun. 2013, doi: 10.1016/j.ijepes.2012.11.023.

[37] F. Ruelens, S. Iacovella, B. J. Claessens, and R. Belmans, "Learning Agent for a Heat-Pump Thermostat with a Set-Back Strategy Using Model-Free Reinforcement Learning," *Energies*, vol. 8, no. 8, pp. 8300–8318, Aug. 2015, doi: 10.3390/en8088300.

[38] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuška, and R. Belmans, "Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2149–2159, Sep. 2017, doi: 10.1109/TSG.2016.2517211.

[39] A. de Gracia, C. Fernández, A. Castell, C. Mateu, and L. F. Cabeza, "Control of a PCM ventilated facade using reinforcement learning techniques," *Energy Build.*, vol. 106, pp. 234–242, Nov. 2015, doi: 10.1016/j.enbuild.2015.06.045.

- [40] T. Leurs, B. J. Claessens, F. Ruelens, S. Weckx, and G. Deconinck, "Beyond theory: Experimental results of a self-learning air conditioning unit," in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016, pp. 1–6, doi: 10.1109/ENERGYCON.2016.7513916.
- [41] G. T. Costanzo, S. Iacovella, F. Ruelens, T. Leurs, and B. J. Claessens, "Experimental analysis of data-driven control for a building heating system," *Sustain. Energy Grids Netw.*, vol. 6, pp. 81–90, Jun. 2016, doi: 10.1016/j.segan.2016.02.002.
- [42] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuška, and R. Belmans, "Reinforcement Learning Applied to an Electric Water Heater: From Theory to Practice," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3792–3800, Jul. 2018, doi: 10.1109/TSG.2016.2640184.
- [43] O. De Somer, A. Soares, K. Vanthournout, F. Spiessens, T. Kuijpers, and K. Vossen, "Using reinforcement learning for demand response of domestic hot water buffers: A real-life demonstration," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pp. 1–7, doi: 10.1109/ISGTEurope.2017.8260152.
- [44] Z. Yu and A. Dexter, "Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning," *Control Eng. Pract.*, vol. 18, no. 5, pp. 532–539, May 2010, doi: 10.1016/j.conengprac.2010.01.018.
- [45] H. Kazmi, F. Mehmood, S. Lodeweyckx, and J. Driesen, "Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems," *Energy*, vol. 144, pp. 159–168, Feb. 2018, doi: 10.1016/j.energy.2017.12.019.
- [46] J. Vázquez-Canteli, J. Kämpf, and Z. Nagy, "Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration," *Energy Procedia*, vol. 122, pp. 415–420, Sep. 2017, doi: 10.1016/j.egypro.2017.07.429.
- [47] G. P. Henze and R. H. Dodier, "Adaptive Optimal Control of a Grid-Independent Photovoltaic System," presented at the ASME Solar 2002: International Solar Energy Conference, 2009, pp. 139–148, doi: 10.1115/SED2002-1045.
- [48] L. Yang, Z. Nagy, P. Goffin, and A. Schlueter, "Reinforcement learning for optimal control of low exergy buildings," *Appl. Energy*, vol. 156, pp. 577–586, Oct. 2015, doi: 10.1016/j.apenergy.2015.07.050.
- [49] Chenxiao Guan, Y. Wang, Xue Lin, S. Nazarian, and M. Pedram, "Reinforcement learning-based control of residential energy storage systems for electric bill minimization," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, 2015, pp. 637–642, doi: 10.1109/CCNC.2015.7158054.
- [50] S. Zhou, Z. Hu, W. Gu, M. Jiang, and X.-P. Zhang, "Artificial intelligence based smart energy community management: A reinforcement learning approach," *CSEE J. Power Energy Syst.*, vol. 5, no. 1, pp. 1–10, Mar. 2019, doi: 10.17775/CSEEJPES.2018.00840.
- [51] Y. R. Yoon and H. J. Moon, "Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling," *Energy Build.*, vol. 203, p. 109420, Nov. 2019, doi: 10.1016/j.enbuild.2019.109420.
- [52] Y. Zhang and M. van der Schaar, "Structure-aware stochastic load management in smart grids," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 2643–2651, doi: 10.1109/INFOCOM.2014.6848212.
- [53] B.-G. Kim, Y. Zhang, M. van der Schaar, and J.-W. Lee, "Dynamic Pricing and Energy Consumption Scheduling With Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sep. 2016, doi: 10.1109/TSG.2015.2495145.
- [54] B. Li and L. Xia, "A multi-grid reinforcement learning method for energy conservation and comfort of HVAC in buildings," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 444–449, doi: 10.1109/CoASE.2015.7294119.
- [55] B. Sun, P. B. Luh, Q.-S. Jia, and B. Yan, "Event-based optimization with non-stationary uncertainties to save energy costs of HVAC systems in buildings," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, 2013, pp. 436–441, doi: 10.1109/CoASE.2013.6654055.
- [56] B. Sun, P. B. Luh, Q.-S. Jia, and B. Yan, "Event-Based Optimization Within the Lagrangian Relaxation Framework for Energy Savings in HVAC Systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1396–1406, Oct. 2015, doi: 10.1109/TASE.2015.2455419.
- [57] R. Jia, M. Jin, K. Sun, T. Hong, and C. Spanos, "Advanced Building Control via Deep Reinforcement Learning," *Energy Procedia*, vol. 158, pp. 6158–6163, Feb. 2019, doi: 10.1016/j.egypro.2019.01.494.

- [58] Eberhart and Yuhui Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 81–86 vol. 1, doi: 10.1109/CEC.2001.934374.
- [59] Y. Wang, X. Lin, and M. Pedram, "A Near-Optimal Model-Based Control Algorithm for Households Equipped With Residential Photovoltaic Power Generation and Energy Storage Systems," *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 77–86, Jan. 2016, doi: 10.1109/TSTE.2015.2467190.
- [60] L. A. Hurtado, E. Mocanu, P. H. Nguyen, M. Gibescu, and R. I. G. Kamphuis, "Enabling Cooperative Behavior for Building Demand Response Based on Extended Joint Action Learning," *IEEE Trans. Ind. Inform.*, vol. 14, no. 1, pp. 127–136, Jan. 2018, doi: 10.1109/TII.2017.2753408.
- [61] L. Raju, S. Sankar, and R. S. Milton, "Distributed Optimization of Solar Micro-grid Using Multi Agent Reinforcement Learning," *Procedia Comput. Sci.*, vol. 46, pp. 231–239, Jan. 2015, doi: 10.1016/j.procs.2015.02.016.
- [62] K. Al-jabery, Z. Xu, W. Yu, D. C. Wunsch, J. Xiong, and Y. Shi, "Demand-Side Management of Domestic Electric Water Heaters Using Approximate Dynamic Programming," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 36, no. 5, pp. 775–788, May 2017, doi: 10.1109/TCAD.2016.2598563.
- [63] E. Mocanu *et al.*, "On-Line Building Energy Optimization Using Deep Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019, doi: 10.1109/TSG.2018.2834219.
- [64] E. Barrett and S. Linder, "Autonomous HVAC Control, A Reinforcement Learning Approach," in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2015, pp. 3–19, doi: 10.1007/978-3-319-23461-8_1.
- [65] Y. Wang, K. Velswamy, and B. Huang, "A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems," *Processes*, vol. 5, no. 3, p. 46, Sep. 2017, doi: 10.3390/pr5030046.
- [66] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of HVAC and window systems for natural ventilation through reinforcement learning," *Energy Build.*, vol. 169, pp. 195–205, Jun. 2018, doi: 10.1016/j.enbuild.2018.03.051.
- [67] H. Kazmi, J. Suykens, A. Balint, and J. Driesen, "Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads," *Appl. Energy*, vol. 238, pp. 1022–1035, Mar. 2019, doi: 10.1016/j.apenergy.2019.01.140.
- [68] K. U. Ahn and C. S. Park, "Application of deep Q-networks for model-free optimal control balancing between different HVAC systems," *Sci. Technol. Built Environ.*, vol. 0, no. 0, pp. 1–14, Oct. 2019, doi: 10.1080/23744731.2019.1680234.
- [69] G. Brockman *et al.*, "OpenAI Gym," Jun. 2016, Accessed: 02-Jan-2020. [Online]. Available: <https://arxiv.org/abs/1606.01540v1>.
- [70] M. Wetter, J. Hu, M. Grahovac, B. Eubanks, and P. Haves, "OpenBuildingControl: Modeling feedback control as a step towards formal design, specification, deployment and verification of building control sequences," in *Proc. of Building Performance Modeling Conference and SimBuild*, Chicago, IL, USA, 2018, pp. 775–782.
- [71] J. R. Vázquez-Canteli, J. Kämpf, G. Henze, and Z. Nagy, "CityLearn v1.0: An OpenAI Gym Environment for Demand Response with Deep Reinforcement Learning," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, New York, NY, USA, 2019, pp. 356–357, doi: 10.1145/3360322.3360998.
- [72] "www.citylearn.net." <https://sites.google.com/view/citylearnchallenge> (accessed Mar. 27, 2020).
- [73] C. W. Anderson, D. C. Hittle, A. D. Katz, and R. M. Kretchmar, "Synthesis of reinforcement learning, neural networks and PI control applied to a simulated heating coil," *Artif. Intell. Eng.*, vol. 11, no. 4, pp. 421–429, Oct. 1997, doi: 10.1016/S0954-1810(97)00004-6.
- [74] G. P. Henze and J. Schoenmann, "Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems," *HVACR Res.*, vol. 9, no. 3, pp. 259–275, Jul. 2003, doi: 10.1080/10789669.2003.10391069.
- [75] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation," *Energy Build.*, vol. 38, no. 2, pp. 142–147, Feb. 2006, doi: 10.1016/j.enbuild.2005.06.002.
- [76] S. Liu and G. P. Henze, "Evaluation of Reinforcement Learning for Optimal Control of Building Active and Passive

- Thermal Storage Inventory," *J. Sol. Energy Eng.*, vol. 129, no. 2, pp. 215–225, May 2007, doi: 10.1115/1.2710491.
- [77] D. Du and M. Fei, "A two-layer networked learning control system using actor–critic neural network," *Appl. Math. Comput.*, vol. 205, no. 1, pp. 26–36, Nov. 2008, doi: 10.1016/j.amc.2008.05.062.
- [78] B. Jiang and Y. Fei, "Dynamic Residential Demand Response and Distributed Generation Management in Smart Microgrid with Hierarchical Agents," *Energy Procedia*, vol. 12, pp. 76–90, Jan. 2011, doi: 10.1016/j.egypro.2011.10.012.
- [79] 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, doi: 10.1109/TSG.2013.2263201.
- [80] A. T. Kaliappan, S. Sathiakumar, and N. Parameswaran, "Flexible power consumption management using Q learning techniques in a smart home," in *2013 IEEE Conference on Clean Energy and Technology (CEAT)*, 2013, pp. 342–347, doi: 10.1109/CEAT.2013.6775653.
- [81] D. Li and S. K. Jayaweera, "Reinforcement learning aided smart-home decision-making in an interactive smart grid," in *2014 IEEE Green Energy and Systems Conference (IGESC)*, 2014, pp. 1–6, doi: 10.1109/IGESC.2014.7018632.
- [82] Q. Wei, D. Liu, G. Shi, Y. Liu, and Q. Guan, "Optimal self-learning battery control in smart residential grids by iterative Q-learning algorithm," in *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2014, pp. 1–7, doi: 10.1109/ADPRL.2014.7010630.
- [83] D. Li and S. K. Jayaweera, "Machine-Learning Aided Optimal Customer Decisions for an Interactive Smart Grid," *IEEE Syst. J.*, vol. 9, no. 4, pp. 1529–1540, Dec. 2015, doi: 10.1109/JSYST.2014.2334637.
- [84] P. Fazenda, K. Veeramachaneni, P. Lima, and U.-M. O'Reilly, "Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems," *J. Ambient Intell. Smart Environ.*, vol. 6, no. 6, pp. 675–690, Jan. 2014, doi: 10.3233/AIS-140288.
- [85] Z. Wen, D. O'Neill, and H. Maei, "Optimal Demand Response Using Device-Based Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015, doi: 10.1109/TSG.2015.2396993.
- [86] M. Rayati, A. Sheikhi, and A. M. Ranjbar, "Applying reinforcement learning method to optimize an Energy Hub operation in the smart grid," in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2015, pp. 1–5, doi: 10.1109/ISGT.2015.7131906.
- [87] H. Berlink, N. Kagan, and A. H. Reali Costa, "Intelligent Decision-Making for Smart Home Energy Management," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 331–354, Dec. 2015, doi: 10.1007/s10846-014-0169-8.
- [88] X. Qiu, T. A. Nguyen, and M. L. Crow, "Heterogeneous Energy Storage Optimization for Microgrids," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1453–1461, May 2016, doi: 10.1109/TSG.2015.2461134.
- [89] S. Sekizaki, T. Hayashida, and I. Nishizaki, "An intelligent Home Energy Management System with classifier system," in *2015 IEEE 8th International Workshop on Computational Intelligence and Applications (IWCIA)*, 2015, pp. 9–14, doi: 10.1109/IWCIA.2015.7449452.
- [90] Y. Sun, A. Somani, and T. E. Carroll, "Learning based bidding strategy for HVAC systems in double auction retail energy markets," in *2015 American Control Conference (ACC)*, 2015, pp. 2912–2917, doi: 10.1109/ACC.2015.7171177.
- [91] A. Sheikhi, M. Rayati, and A. M. Ranjbar, "Demand side management for a residential customer in multi-energy systems," *Sustain. Cities Soc.*, vol. 22, pp. 63–77, Apr. 2016, doi: 10.1016/j.scs.2016.01.010.
- [92] H. Kazmi, S. D'Oca, C. Delmastro, S. Lodeweyckx, and S. P. Corgnati, "Generalizable occupant-driven optimization model for domestic hot water production in NZEB," *Appl. Energy*, vol. 175, pp. 1–15, Aug. 2016, doi: 10.1016/j.apenergy.2016.04.108.
- [93] S. Bahrami, V. W. S. Wong, and J. Huang, "An Online Learning Algorithm for Demand Response in Smart Grid," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4712–4725, Sep. 2018, doi: 10.1109/TSG.2017.2667599.
- [94] B. V. Mbuwir, F. Ruelens, F. Spiessens, and G. Deconinck, "Battery Energy Management in a Microgrid Using Batch Reinforcement Learning," *Energies*, vol. 10, no. 11, p. 1846, Nov. 2017, doi: 10.3390/en10111846.
- [95] M. Schmidt, M. V. Moreno, A. Schülke, K. Macek, K. Mařík, and A. G. Pastor, "Optimizing legacy building operation: The evolution into data-driven predictive cyber-physical systems," *Energy Build.*, vol. 148, pp. 257–279, Aug. 2017, doi: 10.1016/j.enbuild.2017.05.002.
- [96] T. Remani, E. A. Jasmin, and T. P. I. Ahamed, "Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3283–3294, Sep. 2019, doi: 10.1109/JSYST.2018.2855689.

- [97] B. J. Claessens, D. Vanhoudt, J. Desmedt, and F. Ruelens, "Model-free control of thermostatically controlled loads connected to a district heating network," *Energy Build.*, vol. 159, pp. 1–10, Jan. 2018, doi: 10.1016/j.enbuild.2017.08.052.
- [98] G. D. Kontes *et al.*, "Simulation-Based Evaluation and Optimization of Control Strategies in Buildings," *Energies*, vol. 11, no. 12, p. 3376, Dec. 2018, doi: 10.3390/en11123376.
- [99] Z. Zhang, C. Ma, and R. Zhu, "Thermal and Energy Management Based on Bimodal Airflow-Temperature Sensing and Reinforcement Learning," *Energies*, vol. 11, no. 10, p. 2575, Oct. 2018, doi: 10.3390/en11102575.
- [100] P. Odonkor and K. Lewis, "Automated Design of Energy Efficient Control Strategies for Building Clusters Using Reinforcement Learning," *J. Mech. Des.*, vol. 141, no. 2, Feb. 2019, doi: 10.1115/1.4041629.
- [101] Z. Zhang, A. Chong, Y. Pan, C. Zhang, and K. P. Lam, "Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning," *Energy Build.*, vol. 199, pp. 472–490, Sep. 2019, doi: 10.1016/j.enbuild.2019.07.029.
- [102] S. Lu, W. Wang, C. Lin, and E. C. Hameen, "Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884," *Build. Environ.*, vol. 156, pp. 137–146, Jun. 2019, doi: 10.1016/j.buildenv.2019.03.010.
- [103] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, "LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning," *Build. Environ.*, vol. 147, pp. 397–414, Jan. 2019, doi: 10.1016/j.buildenv.2018.10.028.
- [104] J. R. Vázquez-Canteli, S. Ulyanin, J. Kämpf, and Z. Nagy, "Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities," *Sustain. Cities Soc.*, vol. 45, pp. 243–257, Feb. 2019, doi: 10.1016/j.scs.2018.11.021.