2018 Building Performance Modeling Conference and
SimBuild co-organized by ASHRAE and IBPSA-USA
Chicago, IL
September 26-28, 2018

A DEEP REINFORCEMENT LEARNING APPROACH TO USING WHOLE BUILDING ENERGY MODEL FOR HVAC OPTIMAL CONTROL

Zhiang Zhang[1], Adrian Chong[2], Yuqi Pan[3], Chenlu Zhang[1], Siliang Lu[1], and Khee Poh Lam[1,2]
[1]Carnegie Mellon University, Pittsburgh, PA, USA
[2]National University of Singapore, Singapore
[3]Ghafari Associates, MI, USA

ABSTRACT

Whole building energy model (BEM) is difficult to be used in the classical model-based optimal control (MOC) because of its high-dimension nature and intensive computational speed. This study proposes a novel deep reinforcement learning framework to use BEM for MOC of HVAC systems. A case study based on a real office building in Pennsylvania is presented in this paper to demonstrate the workflow, including building modeling, model calibration and deep reinforcement learning training. The learned optimal control policy can potentially achieve 15% of heating energy saving by simply controlling the heating system supply water temperature.

INTRODUCTION

Whole building energy model (BEM) has been widely used in building design stage for design optimization and code compliance. As BEM contains rich information related to building system control and operation, extending the life-cycle of BEM from building design stage to building operation stage has been a popular concept in recent years.

One way to integrate BEM into building operation is model-based optimal control (MOC) of HVAC systems. However, most BEM engines are high-order and non-differentiable models so conventional MOC algorithms cannot be directly applied. Furthermore, most BEM engines have intensive computational demand and hence it can be too slow to meet real-time control requirement. For example, the EnergyPlus based model predictive control (MPC) must limit its prediction horizon to one time-step to save computational cost if it is for real-time control (Zhang and Lam 2017), or the algorithm can only be used offline to pre-calculate the optimal control actions once a day(Ascione et al. 2016).

Reinforcement learning control provides a suitable solution to use BEM in the MOC of HVAC systems because the optimal control policy is developed by reinforcement learning using the model (a.k.a. simulator) off-line. Conventional studies of MOC with reinforcement learning for HVAC systems usually use simple reinforcement learning methods, such as tabular setting Q-learning (Liu and Henze 2006) and linear state-value function approximation (Dalamagkidis et al. 2007). Such methods require careful design of the control algorithm and cannot easily scale to large complicated buildings.

Deep reinforcement learning (DRL) has been recently proposed for optimal control (Mnih et al. 2013) which integrates deep learning into the conventional reinforcement learning framework. Compared to the conventional reinforcement learning, DRL can solve complicated control problems and make "end-to-end" control possible (i.e. raw observations as inputs, raw control actions as outputs). However, DRL-based MOC for HVAC systems is studied at an infant stage of research with very limited existing studies using only simple hypothetical building models as case studies (Wei, Wang, and Zhu 2017; Wang, Velswamy, and Huang 2017; Li et al. 2017).

This study proposes a framework that uses BEM for MOC of HVAC systems by DRL that can be deployed in a real building. The processes, including building modeling, model calibration and DRL training, are explained by using a real-life office building as the case study. The energy efficiency and thermal comfort performance of the proposed control method are analyzed. In addition, the optimization problems of DRL training are discussed.

DRL BACKGROUNDS

Reinforcement Learning (RL)

According to Sutton and Barto (2017), a standard RL problem involves a learning agent interacts with the environment in a number of discrete steps to learn how to maximize the returned reward from the environment. In this case, the learning agent (hereafter called agent) is a virtual entity that performs the control actions, the environment is the controlled HVAC system, and the reward is the human-designed feedback from the environment representing how good the current environment state is. Agent-environment interactions of one step can be expressed as a tuple $(S_t, A_t, S_{t+1}, R_{t+1})$, where S_t is the environment's state at time t, A_t is the control action of agent at the time t, S_{t+1} is the resulting environment' state af-

ter the agent has taken the action, R_{t+1} is the reward received by the agent from the environment. Ultimately, the goal of the RL control is to learn an optimal control policy $\pi : S_t \rightarrow A_t$ that maximizes the accumulated future reward $\sum_t^{T_\infty} R_t$.

The above-mentioned standard RL problem is a Markov decision process (MDP) if it obeys the Markov property. Most RL algorithms implicitly assume the environment is a MDP. However, empirically, problems can still be solved even though the environment is not strictly MDP, such as Atari games (Mnih et al. 2013).

In RL, there are three important concepts including state-value function, action-value function and advantage function, as shown in Equations (1), (2), and (3):

$$v_\pi(s) \doteq \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\Big], \tag{1}$$

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\Big], \tag{2}$$

$$a_\pi(s,a) = q_\pi(s,a) - v_\pi(s), \tag{3}$$

where γ is the reward discount factor (Sutton and Barto 2017). Intuitively, state-value function represents how good is a state, action-value function represents how good is an action, and advantage function describes how good is an action with respect to the state. Note that $\mathbb{E}_{a\sim\pi(s)}[a_\pi(s,a)] = 0$. For the optimal policy π^*, there is

$$\begin{aligned} v_{\pi^*}(s) &= \max_a q_{\pi^*}(s,a) \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi^*}(S_{t+1}) | S_t = s, A_t = a]. \end{aligned} \tag{4}$$

In deep reinforcement learning, deep learning models are used to represent value functions and policy function, i.e. $v_\pi(s,\theta)$, $q_\pi(s,a,\theta)$ and $\pi(s,a,\theta)$ where θ is function parameters.

Policy Gradient

Policy gradient (Sutton and Barto 2017) is one type of RL training method that is used in this study. The goal of the policy gradient method is to learn the parameter θ in $\pi_\theta(s,a) = Pr(a|s,\theta)$ that maximizes average reward per time step $J(\theta)$:

$$J(\theta) = \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_\theta(s,a), \tag{5}$$

where $d_{\pi_\theta}(s)$ is stationary distribution for the state s of the Markov chain starting from s_0 following the policy π_θ, and R_s^a is the reward of the agent at the state s taking the action a.

We can use gradient descent to maximize Equation (5). The gradient of $J(\theta)$ with respect to θ is given by:

$$\nabla_\theta J(\theta) = \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} \tag{6a}$$

$$= \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a) \tag{6b}$$

$$= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) q_{\pi_\theta}(s,a)] \tag{6c}$$

$$= \mathbb{E}_{\pi_\theta}\big[\nabla_\theta \log \pi_\theta(s,a)\big(q_{\pi_\theta}(s,a) - v(s)\big)\big] \tag{6d}$$

$$= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) a_{\pi_\theta}(s,a)], \tag{6e}$$

where Equation (6c) follows from the policy gradient theorem, (6d) is obtained by subtracting a zero-valued "baseline function" $\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) v(s)]$ to reduce the variance of q_{π_θ}.

The policy gradient in the form of Equation (6e) is called advantage actor critic (A2C).

Asynchronous Advantage Actor Critic (A3C)

The state-of-the-art deep reinforcement learning variation of the A2C, Asynchronous Advantage Actor Critic (A3C) (Mnih et al. 2016), is the specific deep reinforcement learning method used in this study.

In A3C method, rather than having only one agent to interacts with the environment, a number of agents interact the copy of the same environment independently, but update the same global action-value or policy function network asynchronously. Also asynchronously, the agents update their own action-value or policy function network to be the same as the global one in a certain frequency. The purpose of this method is to ensure the $(S_t, A_t, S_{t+1}, R_{t+1})$ tuples used to train the global network are independent. Compared to the non-asynchronous methods, A3C significantly reduces memory usage and training time. Details of the algorithm can be found in (Mnih et al. 2016).

According to Equation (6d), two deep neural network models are needed for A3C, including $\pi_\theta(s,a)$ to approximate the policy and $v_{\theta_v}(s)$ to approximate the state-value function. One step update function for θ is (follow Equations (4) and (6))

$$\begin{aligned} \theta &\leftarrow \theta + \alpha \mathbb{E}_{\pi_\theta}\big[\nabla_\theta \log \pi_\theta(s,a)\big(q_{\pi_\theta}(s,a) - v_{\theta_v}(s)\big)\big] \\ &= \theta + \alpha \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)\big(R' + \gamma v_{\theta_v}(s') - v_{\theta_v}(s)\big)\big], \end{aligned} \tag{7}$$

and one step update function for θ_v is (by applying the mean squared loss function)

$$\begin{aligned} \theta_v &\leftarrow \theta_v - \alpha \mathbb{E}_{\pi_\theta}\big[\partial\big(v_{true} - v_{\theta_v}(s)\big)^2 / \partial \theta_v\big] \\ &= \theta_v - \alpha \mathbb{E}_{\pi_\theta}\big[\partial\big(R' + \gamma v_{\theta_v}(s') - v_{\theta_v}(s)\big)^2 / \partial \theta_v\big]. \end{aligned} \tag{8}$$

In Equation (7) and (8), α is the step size for gradient descent, R' is the actual reward, and s' is the next state from the state s taking the action a.

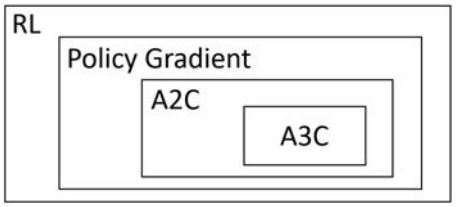

Figure 1: Relationship among the RL Algorithms

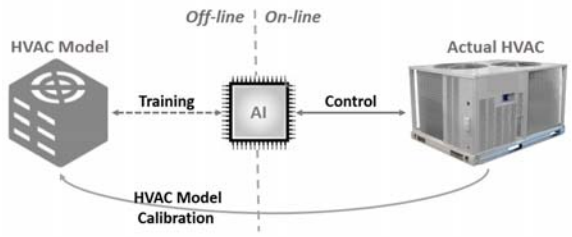

Figure 2: BEM-DRL Control Framework

The relationship among the RL algorithms discussed in this section is shown in Figure 1. The specific RL algorithm used in this study, A3C, is a deep learning extension to A2C. A2C can be categorized as the policy gradient method in RL.

BEM-DRL CONTROL FRAMEWORK

The BEM-DRL optimal control framework for HVAC systems is shown in the Figure 2, which includes four steps:

1. BEM modeling: The building and its HVAC system are firstly modeled using BEM application. The BEM built in building design stage can be reused in this control framework.

2. HVAC model calibration: The BEM needs to be calibrated against real HVAC operation data.

3. DRL agent training: The DRL agent will be trained off-line using the calibrated BEM to learn the optimal control policy. The DRL state, reward and action design will be determined based on the building sensor data availability, control optimization objectives and HVAC system control capability.

4. Actual HVAC system control: The trained DRL agent will control the actual HVAC systems. In this process, the agent continuously refines its control policy by learning from the actual feedbacks from the HVAC system.

This framework will be demonstrated using the Intelligent Workplace (IW) building located in Pittsburgh, PA, USA to control its heating system. In this study, the last step of the framework is not yet implemented. A virtual building model is used instead of the actual building to validate the AI agent's control performance.

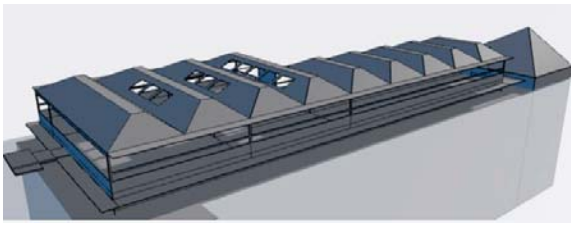

Figure 3: The Geometry Rendering of the IW BEM (rendered by BuildSimHub (BuildSimHub, Inc. 2018))

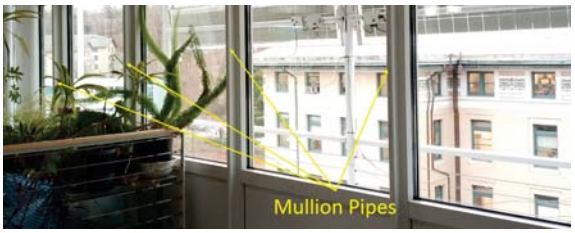

Figure 4: "Mullion" Radiant Heating System in IW

BUILDING MODELING

Overview

The case study building, IW, is a one-level office building built on top of an exiting building in 1997 with complete building automation system installed. It has area around 600 m^2, with about 20 regular occupants and a 30-person classroom. EnergyPlus v8.3 (U.S. Department of Energy 2015) is used to create the BEM. An overview of the geometry of the model is shown in the Figure 3.

Radiant Heating System

IW uses a water-based radiant heating system that is integrated with the mullion of the window frames as shown in the Figure 4. This system has a very slow thermal response. In winter morning, it usually takes more than 2 hours for the indoor air temperature to reach its setpoint.

Current Heating System Control Logic

The hot water for the radiant system is supplied by a district heating plant, as shown in the Figure 5. Steam to water heat exchanger located in the B2 level of the main building supplies the hot water to the system. Currently, the heat exchange is enabled when the outdoor air temperature is below 10 C. The radiant system is also supplied with cold water from the district cooling plant in summer. Valve 2 and valve 3 in the Figure 5 are on/off valves and operate in a synchronized way to switch between heating and cooling mode.

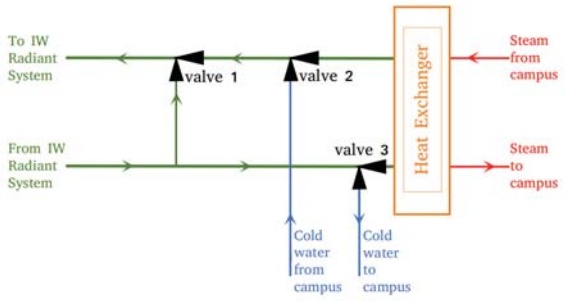

Figure 5: Schematic Diagram of the IW Radiant System Water Flow (valve 1: recirculation value; valve 2: hot/cold water inlet valve; valve 3: hot/cold water outlet valve)

The supply water temperature of the radiant system is controlled by valve 1 in the Figure 5, which changes the proportion between the return water and hot water to the radiant system. Current control logic of the system is to keep supply water flow rate constant and change the supply water temperature. The supply water temperature setpoint is calculated by a PID controller based on the difference between the zones' average indoor air temperature and its setpoint.

MODEL CALIBRATION

Calibration Objective

The calibration objective should be determined based on the requirement of the control problem. In this study, the optimal control aims at reducing heating energy consumption and maintaining acceptable indoor thermal comfort level. Therefore, the BEM of IW will be calibrated for heating energy consumption and indoor average air temperature. Note that the average indoor air temperature is the representation of the indoor thermal response.

This study focuses on the control for the heating season. Only three months measured data from Jan 1st 2017 to Mar 31th 2017 with 5 minutes interval is used for the calibration. The heating energy consumption is a calculated value based on the measured radiant system inlet/outlet water temperature and water mass flow rate.

Bayesian Calibration

Bayesian calibration using the method proposed by Chong et al.(2017) is performed for the IW model. The statistical formulation of the Bayesian calibration in this study is

$$y(x) = \zeta(x) + \varepsilon(x) = \eta(x, c^*) + \varepsilon(x), \qquad (9)$$

where y is the observed building behavior, ζ is the true building behavior, η is the output from the building energy model, x is the observable input parameters, c is the unknown calibration parameters, ε is the observation error.

Table 1: IW Model Calibration Parameters

No.	Parameter)	Range
1	Mullion insulation (Polyisocyanates) thickness	1 mm to 5 mm
2	Internal mass area*	200 to 1000 m^2/zone
3	Mullion radiant surface area†	6.7% to 26.7% of the external wall
4	Infiltration rate	0.01 to 0.30 ACH

Note: *The internal mass is modeled in BEM as 5 cm thick concrete. †The Mullion system of IW is abstracted to the radiant surfaces in BEM.

To adapt the Chong et al.(2017)'s method for the multi-objective calibration, multiple objectives have to be combined into the single objective. In this study, a simple linear method, as shown in the Equation (10), is used to combine multiple objectives (y_n) into one.

$$\mathbf{y} = \mu_1 y_1 + \mu_2 y_2 +, ..., +\mu_n y_n, \; s.t. \sum_1^n \mu_n = 1 \qquad (10)$$

Input Parameters and Calibration Parameters

The input parameters for the BEM calibration generally include real weather conditions, such as outdoor air temperature, humidity, solar radiation, wind speed and wind direction. In addition, actual HVAC system operation parameters should be included based on the control problem requirement. In this study, the control parameter of the optimal control algorithm is the Mullion system supply water temperature setpoint. Therefore, the input parameters should include the real Mullion system supply water temperature setpoint and the steam heat exchanger (as shown in the Figure 5) on/off status.

The calibration parameters are selected by the sensitivity analysis with Morris method (Morris 1991). The selected calibration parameters are listed in the Table 1 with their calibration ranges. The ranges are determined based on experience.

Calibration Results

In this study, there are only two calibration objectives, including indoor average air temperature (y_1) and heating energy consumption (y_2). Different combinations of μ_1 and μ_2 have been tested and the calibration evaluation criteria are the mean bias error (MBE) and cumulative variation of the root mean square error (CVRMSE). It is interesting to find that $\mu_1 = 0, \mu_2 = 1$ gives the best accuracy for both average indoor air temperature and heating energy consumption. This is possibly because the heating

Table 2: Calibration Results: Jan 1st to Mar 31th of 2017 with $\mu_1 = 0, \mu_2 = 1$

Objective	Results
Average Indoor Air Temperature	5-min MBE: 0.52% 5-min CVRMSE: 4.82%
Heating Energy Consumption	Hourly MBE: 0.43% Hourly CVRMSE: 35.96% Daily CVRMSE: 10.46%

Table 3: State: Site Observations (SiteOb)

No.	Site Observations
1	Day of the week
2	Hour of the day
3	Outdoor air temperature (C)
4	Outdoor Air Relative Humidity (%)
5	Wind Speed (m/s)
6	Wind Direction (degree from north)
7	Diffuse Solar Radiation (W/m2)
8	Direct Solar Radiation (W/m2)

energy consumption and average indoor air temperature is highly correlated in this model so capturing the accuracy in heating energy consumption also captures the accuracy in the average indoor air temperature. However, further study should be conducted for the multi-objective calibration using the Bayesian method.

The calibration results are shown in the Table 2. MBE and CVRMSE for the average indoor air temperature is calculated using the 5-minute interval data because air temperature is related to indoor thermal comfort and should be checked very frequently during the MOC of HVAC. It is shown that less than 1% error can be achieved. Heating energy consumption MBE and CVRMSE are calculated for the hourly and daily interval. Even though the CVRMSE for hourly heating energy consumption has relatively worse accuracy, its hourly MBE and daily CVRMSE are still within the acceptable range. As accumulated value is more important than the spot value for the HVAC energy consumption, this accuracy is acceptable.

DRL Training

State, Action and Reward Design

For reinforcement learning, state, action and reward design are critical for learning convergence and control performance. To take advantage of the deep reinforcement learning method, we only use raw observable or controllable parameters for our state, action and reward design with no extra data manipulations.

The state in reinforcement learning is what the DRL agent observes for each control step. In this study, the state consists of site observations (SiteOb) and building observations (BldOb), as shown in the Table 3 and Table 4. The final state observations for the deep reinforcement learning is:

$$state = \big\{\{SiteOb, BldOb\}_t, \{SiteOb, BldOb\}_{t-1}, ..., \{SiteOb, BldOb\}_{t-n}\big\} \tag{11}$$

where t is the current control step, n is the number of the history control steps to be considered. As building usually has slow thermal response, it is important to consider the history observations for the deep reinforcement learning to make the environment roughly a MDP. It should be noted that each item in the state should be normalized to between 0 and 1 for the optimization purpose of the deep neural network. Min-max normalization is used.

Table 4: State: Building Observations (BldOb)

No.	Building Observations
1	IW Steam Heat Exchanger Enable Setpoint (HESSP, C)*
2	IW Average PPD (PPD) †
3	IW Radiant Systems Supply Water Temperature Setpoint (MULSSP, C)
4	IW Average Indoor Air Temperature (IAT, C)
5	IAT Setpoint (IATSSP, C)
6	IW Occupancy Mode Flag (OCCU)◇
7	IW Average Heating Demand Since Last Observation (E_{hvac}, kW)

Note: *The outdoor air temperature setpoint below which the IW steam heat exchanger will be enabled. †PPD is short for Predicted Percentage of Dissatisfied, which is calculated by the BEM engine with assumptions $Clo = 1.0$, $Met = 1.2$ and $v_{air} = 0.137m/s$. ◇The occupancy mode flag determined based on a fixed schedule (the occupancy mode flag is 1 between 7:00 AM and 7:00 PM of weekdays, and between 8:00 AM and 6:00 PM of weekends)

Table 5: DRL Agent Actions

No.	RadSSP	HXOp	No.	RadSSP	HXOp
1	N/A *	0	**...**	...	...
2	20 C	1	**10**	60 C	1
3	25 C	1	**11**	65 C	1

Note: *If the IW steam heat exchanger is turned off, the IW radiant system supply water temperature is no longer needed.

The action is how the DRL agent controls the environment. In this study, the actions are the IW radiant system supply water temperature setpoint (RadSSP) and on/off of

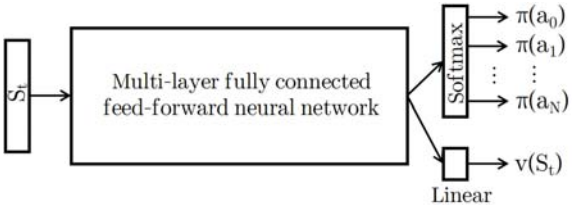

Figure 6: Policy and State-value Function Network Architecture

the IW steam heat exchanger (HXOp). The action space is discrete, as shown in the Table 5.

The reward design determines the control optimization objective. The objective of the control method is to minimize the HVAC energy consumption and maximize thermal comfort. The design of the reward functions may significantly affect the convergence of the deep reinforcement learning and the final control performance. A flexible reward function combining the HVAC energy consumption and the indoor thermal comfort is shown in the Equation (12). This reward design has good convergence property by our experiments.

$$reward = -\begin{cases} \left[\tau * \left([PPD-0.1]^{+} * \rho\right)^2 + \beta * E_{hvac}\right]_0^1 | Occu=1 \\ \left[\tau * [Stpt_{low} - IAT]^{+} * \lambda + \beta * E_{hvac}\right]_0^1 | Occu=0 \end{cases} \quad (12)$$

where $\tau, \beta, \rho, \lambda, Stpt_{low}$ are the tunable hyperparameters. τ and β control the relative importance between the HVAC energy efficiency and the indoor thermal comfort for the optimization; ρ is a scale factor to penalize large PPD value; λ is the penalty level for the indoor air temperature violation during the unoccupied hours, $Stpt_{low}$ is the indoor air temperature penalty threshold. Note all parameters are normalized.

DRL Training Experiment Setup

The neural network architecture for this study is shown in the Figure 6. A3C needs two function approximation neural networks, one for the policy and the other for the state-value function. Therefore, a shared multi-layer feed-forward neural network is used. The output from the shared network is fed into a Softmax layer and a linear layer in parallel, where the Softmax layer outputs the policy and the linear layer outputs the state-value.

The shared network in Figure 6 has 4 hidden layers of which each layer has 512 hidden units with rectifier nonlinearity. RMSProp (Tieleman and Hinton 2012) is used for optimization. The learning rate is fixed to be 0.0001, RMSProp decay factor is 0.90. All gradients during the back-propagation are clipped with their L2 norm ≤ 5.0. 16 A3C agents interacts with the environment in parallel and the total number of interactions is 10 million (about 600K per A3C agent). The history window n in state function (Equation (11)) is 3.

An entropy term is added to the DRL learning update function (Equation (7)) to encourage random exploration of the agent(Mnih et al. 2016). Therefore, the original update function (Equation (7)) becomes

$$\theta \leftarrow \theta + \alpha \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a)(R' + \gamma v_{\theta_v}(s') - v_{\theta_v}(s)) + \kappa \nabla_\theta H(\pi_\theta(s))], \quad (13)$$

where H is the entropy term, and κ is a tunable hyperparameter controlling the exploration level (larger value encourages more exploration). In this study, κ is a piecewise constant with value 1, 0.1, 0.05 and 0.01 for interaction steps before 2 million, 4 million, 6 million and 10 million.

The calibrated BEM of IW is used for DRL training and testing in this study. The DRL agent is trained in heating season (Jan to Mar) using the TMY3 weather data. The trained DRL agent is tested also for heating season but using the actual Jan-Mar weather data of 2017. The BEM simulation time step and the control time step are both 5 minutes. The $Stpt_{low}$ in Equation (12) is the indoor air temperature setpoint calculated by the IW BAS control logic.

Discussion and Results

The DRL performance is usually analyzed by its optimization performance and control performance. The optimization performance is evaluated on the quality of the DRL training convergence (accumulated reward value convergence), and the control performance in this study is evaluated on the heating energy consumption and thermal comfort.

The DRL training in this study can usually converge around 6.5 M interactions, which takes about 10 hours on a Xeon E5 CPU (no GPU is used). However, it is found that for some choices of the hyperparameters, the accumulated reward fluctuates or converges to low level in the DRL training. This is because of the very slow thermal response of the IW due to its radiant system and insufficient heating capacity of the radiant system. As a result, the control actions taken by the DRL agent may not take effect immediately in terms of the environment observation. For example, in winter, it may take more than 1 hour for the IW average indoor air temperature to rise by 1°C after the supply water temperature of the radiant system has been set to the maximum. This phenomenon is known as "delayed reward problem" in DRL. It is found that the careful selection of the reward function hyperparameters and control action repeat can to some level solve this prob-

lem. However, a more structured solution should be found in the future.

The ultimate goal of the optimal control is to maximize the indoor thermal comfort and minimize the heating energy consumption. However, the two goals cannot be fulfilled simultaneously. A good balance must be found between the heating energy consumption and indoor thermal comfort quality, which is determined by the reward function in the DRL. Due to the highly complicated relationships between the thermal comfort and heating energy consumption, it is difficult to know exactly what reward function can lead to the best balance. Therefore, hyperparameters in the reward function (Equation (12)) must be tuned.

Table 6 shows the control performance results of the DRL of some selected experiments. It is interesting to find that the control performance results of different hyperparameters are not necessarily intuitive. We would expect that, bigger β and smaller ρ should lead to lower heating energy consumption and worse indoor thermal comfort. However, cases 4, 5 and 6 in the Table 6 do not respect that. Also, action repeats lower the flexibility of the control algorithm and hence should lead to worse control performance. However, in Table 6 it can be seen that cases with action repeat of 3 generally performs better than cases with action repeat of 1. Such counter-intuitive results are caused by the non-ideal optimization of the DRL training. As DRL is in fact a non-convex optimization problem, finding the global optimal solution (or even the existence of the global optimal solution) is not guaranteed. A small change in the hyperparameters may cause the DRL agent to yield different local optimal solutions. In addition, reward is significantly delayed in this study, which adds more difficulties to the optimization problem. For example, case 5 consumes less heating energy consumption but gives better indoor thermal comfort than case 2, which means the DRL agent in case 2 was stuck in a worse local optimal during the DRL training. Out of the six experiments in the Table 6, case 6 performs comparably the best, which saves 15% of heating energy with only slightly worse indoor thermal comfort quality in the testing model.

Over-fitting is the major problem of almost all machine learning methods. In BEM-DRL, the DRL agent may over-fit to the weather and building thermal behavior patterns of the simulator. Table 6 shows the control performance of the trained DRL agent on the testing model, which uses the same IW BEM but different weather data as the training model. It is shown that the problem of over-fitting to the simulator's weather patterns is not detected. The energy saving and thermal comfort results on the testing model are comparable or even better than the training model. However, further analysis on the problem of over-fitting to the simulator's thermal behavior patterns is required after the trained DRL agent is deployed in the real building.

CONCLUSION AND FUTURE WORK

This study develops a BEM-DRL framework for HVAC optimal control for energy efficiency and thermal comfort. The framework workflow includes building modeling using BEM tools, model calibration and DRL training. The framework is applied to a case study, which controls the supply water temperature of the radiant heating system of an office building in Pennsylvania, USA. Bayesian method is used for the model calibration and less than 1% modeling error is achieved for both indoor average air temperature and heating energy consumption. A3C algorithm is used for DRL training. It is found that selection of the hyperparameters significantly affect the DRL convergence and control performance because of the DRL non-convex optimization nature and the slow thermal response of the case study building. The best trained control policy by DRL can achieve 15% of heating energy saving with similar indoor thermal comfort as the basecase.

Future work will include implementing the trained DRL agent in the real building to test its real control performance. In addition, the energy saving results should be further analyzed to obtain its statistical significance. The "delayed reward" problem will be further studied to a find more structured solution. More case studies should also be conducted for different types of HVAC systems to test the robustness of the framework.

ACKNOWLEDGMENT

The first author would like to thank China Scholarship Council for the financial support of the author's PhD study. We would like to thank Linan Zhang from the Carnegie Mellon University for the help on LaTeX.

REFERENCES

Ascione, Fabrizio, Nicola Bianco, Claudio De Stasio, Gerardo Maria Mauro, and Giuseppe Peter Vanoli. 2016. "Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort." *Energy and Buildings* 111:131–144.

BuildSimHub, Inc. 2018. BuildSimHub.

Chong, Adrian, Khee Poh Lam, Matteo Pozzi, and Junjing Yang. 2017. "Bayesian calibration of building energy models with large datasets." *Energy & Buildings* 154:343–355.

Dalamagkidis, K., D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis. 2007. "Reinforcement learning for energy conservation and comfort in buildings." *Building and Environment* 42 (7): 2686–2698.

Table 6: DRL Control Performance for IW in Heating Season (selected)

#	Hyperparameters		Training Model			Testing Model		
	Action Repeat*	$\tau, \beta, \rho^{\dagger}$	Heating Energy (kWh)	PPD_{mean} (%)	PPD_{std} (%)	Heating Energy (kWh)	PPD_{mean} (%)	PPD_{std} (%)
Basecase	N/A	N/A	45302	10.48	4.48	43709	9.46	5.59
1	1	1.0, 1.5, 20	52806	8.72	4.31	47522	8.23	2.46
2	1	1.0, 2.5, 20	44549	11.58	5.35	39484	11.11	4.53
3	1	1.0, 2.5, 10	40101	16.09	9.46	37238	14.2	8.65
4	3	1.0, 1.5, 20	42255	11.46	4.26	38550	10.63	3.34
5	3	1.0, 2.5, 20	43532	10.63	4.23	39109	10.44	3.75
6	3	1.0, 2.5, 10	42104	11.49	4.24	37131	11.71	3.76

Note: *Action repeat means the DRL agent repeats the same control action for multiple time steps. †ρ is determined by a function $\rho = \frac{1}{PPD_{thres}/100-0.1}$. $\rho = 20$ and $\rho = 10$ correspond to $PPD_{thres} = 15$ and $PPD_{thres} = 20$ respectively, meaning the reward function (Equation (12)) returns the minimum value if the PPD exceeds the PPD_{thres}.

Li, Yuanlong, Yonggang Wen, Kyle Guan, and Dacheng Tao. 2017. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning.

Liu, Simeng, and Gregor P. Henze. 2006. "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis." *Energy and Buildings* 38 (2): 148–161.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. 2016. "Asynchronous Methods for Deep Reinforcement Learning." *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16. New York, NY, USA: JMLR.org, 1928–1937.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. "Playing Atari With Deep Reinforcement Learning." In *NIPS Deep Learning Workshop*.

Morris, Max D. 1991. "Factorial Sampling Plans for Preliminary Computational Experiments." *Technometrics* 33 (2): 161–174.

Sutton, Richard S., and Andrew G. Barto. 2017. *Reinforcement Learning: An Introduction*. Second Edi. Cambridge, MA, USA: MIT Press.

Tieleman, Tijmen, and Geoffrey Hinton. 2012. "Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude." *COURSERA: Neural Networks for Machine Learning* 4:26–31.

U.S. Department of Energy. 2015. EnergyPlus 8.3.0.

Wang, Yuan, Kirubakaran Velswamy, and Biao Huang. 2017. "A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems." *Processes* 5, no. 46.

Wei, Tianshu, Yanzhi Wang, and Qi Zhu. 2017. "Deep Reinforcement Learning for Building HVAC Control." *Proceedings of the 54th Annual Design Automation Conference 2017*. Austin, TX, USA.

Zhang, Zhiang, and Khee Poh Lam. 2017. "An Implementation Framework of Model Predictive Control for HVAC Systems: A Case Study of Energyplus Model-Based Predictive Control." *ASHRAE 2017 Annual Conference*. Long Island, CA, USA.

NOMENCLATURE

t	Control time step
S	Environment's state of RL
A	Control action of the RL agent
R	Reward received by the RL agent
π	Control policy of the RL agent
$\mathbb{E}$	Expected value
γ	Reward discounting factor of RL
v, q, a	State-value, action-value, advantage value
θ	Function parameter
Pr	Probability
d	Stationary distribution
α	Gradient descent step size
H	Entropy of a distribution
y	Observed building behavior
ζ	True building behavior
ε	Building behavior observation error
η	Simulated building behavior
x, c	Calibration input and unknown parameter
μ, τ, β λ, ρ, κ	Tunable hyper-parameter