Building HVAC Scheduling Using Reinforcement Learning via Neural Network Based Model Approximation*

Chi Zhang
Department of Computer Science
University of Southern California
Los Angeles, CA
zhan527@usc.edu

Sanmukh R. Kuppannagari
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA
kuppanna@usc.edu

Rajgopal Kannan
US Army Research Lab-West
Playa Vista, CA
rajgopak@usc.edu

Viktor K. Prasanna
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA
prasanna@usc.edu

ABSTRACT

Buildings sector is one of the major consumers of energy in the United States. The buildings HVAC (Heating, Ventilation, and Air Conditioning) systems, whose functionality is to maintain thermal comfort and indoor air quality (IAQ), account for almost half of the energy consumed by the buildings. Thus, intelligent scheduling of the building HVAC system has the potential for tremendous energy and cost savings while ensuring that the control objectives (thermal comfort, air quality) are satisfied.

Traditionally, rule-based and model-based approaches such as linear-quadratic regulator (LQR) have been used for scheduling HVAC. However, the system complexity of HVAC and the dynamism in the building environment limit the accuracy, efficiency and robustness of such methods. Recently, several works have focused on model-free deep reinforcement learning based techniques such as Deep Q-Network (DQN). Such methods require extensive interactions with the environment. Thus, they are impractical to implement in real systems due to low sample efficiency. Safety-aware exploration is another challenge in real systems since certain actions at particular states may result in catastrophic outcomes.

To address these issues and challenges, we propose a model-based reinforcement learning approach that learns the system dynamics using a neural network. Then, we adopt Model Predictive Control (MPC) using the learned system dynamics to perform control with random-sampling shooting method. To ensure safe exploration, we limit the actions within safe range and the maximum absolute change of actions according to prior knowledge. We evaluate our ideas through simulation using widely adopted EnergyPlus tool on a case study consisting of a two zone data-center. Experiments show that the average deviation of the trajectories sampled from the learned dynamics and the ground truth is below 20%. Compared with baseline approaches, we reduce the total energy consumption by 17.1% ∼ 21.8%. Compared with model-free reinforcement learning approach, we reduce the required number of training steps to converge by 10x.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Hardware** → **Temperature control**.

KEYWORDS

neural network dynamics, model-based reinforcement learning, hvac control, smart buildings, data center cooling, model predictive control

ACM Reference Format:
Chi Zhang, Sanmukh R. Kuppannagari, Rajgopal Kannan, and Viktor K. Prasanna. 2019. Building HVAC Scheduling Using Reinforcement Learning via Neural Network Based Model Approximation. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360322.3360861

1 INTRODUCTION

The energy consumption by buildings consist of 40% of the total energy and 70% of total electricity in the United States [18]. Of the total energy consumption of buildings, the Heating, Ventilation and Air-Conditioning (HVAC) system accounts for 50% while the rest is used for lighting, electrical appliances, electric vehicles, etc. The main objective of the HVAC system is to maintain the indoor temperature and air quality. An intelligent HVAC scheduling system will, additionally, save energy and cost while satisfying the objective. The HVAC system is a nonlinear system and has complex system dynamics with a large number of subsystems including chillers, boilers, heat pumps, pipes, ducts, fans, pumps and heat exchangers [11]. In this paper, we assume the combination of equipment to operate by the HVAC system is fixed and focus on how to set the temperature point for local controllers to reduce the energy

*This work has been sponsored by the U.S. Army Research Office (ARO) under award number W911NF1910362 and the U.S. National Science Foundation (NSF) under award number 1911229.

BuildSys '19, November 13–14, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7005-9/19/11…$15.00
https://doi.org/10.1145/3360322.3360861

consumption or cost while maintaining the thermal comfort at given level.

Traditional approaches for set point scheduling include PID control [28] and rule-based supervisory controllers [26]. The PID controller is a feedback proportionalâĂŞintegralâĂŞderivative controller which works by simply turning on/off the HVAC systems. Some advanced HVAC scheduling systems employ rule-based supervisory controllers given historical operation experience. Such systems require no system modeling and design effort, however, they require an experienced and professional operator to constantly monitor and control the system which increases operational costs. Moreover, these traditional approaches are reactive in nature i.e. they are based on feedback from the system and lack the ability to anticipate how the system evolves. This hinders their energy performance as - (i) the thermal inertia of the building leads to delayed effect of control action requiring more aggressive actions, and (ii) the inability to predict and account for external disturbances such as weather, electricity price and occupancy conditions leads to sub-optimal decision making.

Recent success of deep learning has led to the development of several deep reinforcement learning (DRL) based approaches for HVAC scheduling [15, 27, 29]. These data-driven approaches learn an agent to schedule the HVAC system by interacting with the environment. DRL can be generally divided into model-free approach and model-based approach. In model-free DRL, the agent learns the policy by directly interacting with the environment. The agent explores the environment by extensively trial-and-error. However, for a constrained system such as HVAC which enforces soft constraints of the feasible region of operation i.e. thermal comfort bounds, model-free DRL techniques such as Deep Q-Network [25] require a large amount of operational data (obtained via interactions) to converge (also known as low sample efficiency) which is difficult to gather in a real system.

Thus, the practical alternative is to adopt model-based RL approaches. The model for RL algorithm can be obtained by developing a thermal dynamics model [24]. However, the complexity of the HVAC system and the dynamism of the building environment makes it a daunting task [4]. Thus, an alternative is to use the readily available historical data on HVAC operation and learn a general function approximator (e.g. neural network) for building system dynamics [16]. Planning algorithms such as linear-quadratic regulator (LQR) [3] can be used on the learned dynamic to perform HVAC scheduling to minimize energy consumption with thermal comfort constraints.

In this paper, we develop a model-based reinforcement learning approach for smart building HVAC control by learning the system dynamics using operation data to fit a neural network. Then, we perform Model Predictive Control (MPC) [9] using the learned dynamics with random-sampling shooting method [19]. Compared with model-free approaches, our approach is able to train an accurate model with limited amount of data and achieve good control performance without extensive trial-and-error with the systems. Compared with manually design model, our approach is more general and applicable to various building models since it learns system dynamics automatically from data. Our main contributions are as follows:

- We analyze the fundamental drawbacks of previous model-free DRL-based approaches and emphasize the importance of sample data efficiency in data-driven approaches.
- We propose a model-based DRL approach for building HVAC control that trains the system dynamics with neural networks online. Our approach is both data efficient and self-adaptive online with gradually changing system dynamics (e.g. outdoor temperature).
- Given trained system dynamics, we perform Model Predictive Control (MPC) to produce action for the next step that minimizes the energy cost and the temperature constraints violation collectively with random-sampling shooting method.
- To support real-time inference, we train an auxiliary policy network that imitates the output of MPC.
- We conduct experiments on a two-room data center and show that our approach reduces the total energy consumption by 17.1% $\sim$ 21.8%. Compared with model-free reinforcement learning approach, we reduce the required number of training steps to converge by 10x.

2 RELATED WORK

In [4], the author proposes to estimate the thermal load and use a regulator and a disturbance rejection component to keep the room at comfort temperature. In [12], the author proposes a novel multi-input multi-output (MIMO) to model HVAC system and uses a linear-quadratic regulator (LQR) [3] to optimize control performance and to stabilize the proposed HVAC system. In general, exact modeling of HVAC system dynamics is difficult and several data-driven approaches are proposed recently. In [13], the authors propose linear models for system identification and limit each control variable to a safe range informed by historical data for exploration. Then, the authors minimize a length-L trajectory to obtain the control sequence, execute the action at first step and re-run the optimization. Although they achieve significant performance improvement over baseline controllers, they did not consider distribution shift over time, which requires data aggregation and model fine-tuning. In [27] and [10], the author proposes to use Deep Q-Network [25] to control HVAC systems. In [15], the author proposes a EnergyPlus based research environment for developing reinforcement learning approaches for data-center HVAC control. In [7], the author shows promising results of approximating the Model Predictive Controller using neural networks. In [2], the author proposes to learn system dynamics using Artificial Neural Networks (ANN) and perform model predictive control [9]. The main drawbacks of their approach lies in that they attempt to learn everything offline, which fails to adapt to system distribution shift.

3 BUILDING HVAC SYSTEM MODELING

In this section, we demonstrate a typical single-zone building HVAC system that is extensively studied in [4]. We analyze the system using Control System Equations [1] and generalize it using Partial Observable Markov Decision Process [14]. Then, we emphasize the importance and advantages of data-driven approach for this problem and introduce modern reinforcement learning based control strategies.

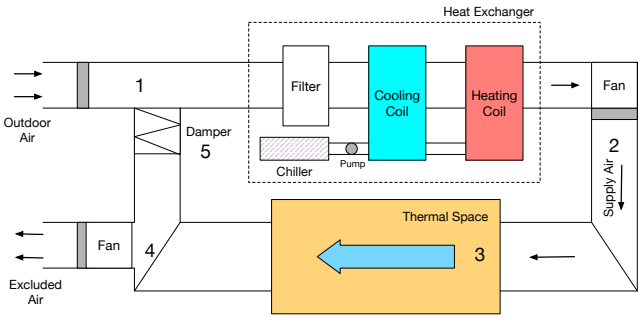


Figure 1: Model of typical single-zone building HVAC system [4]

3.1 Representative System Modeling

We show the representative single-zone system model in Figure 1. It consists of the following components: a heat exchanger; a chiller, which provides chilled water to the heat exchanger; a circulating air fan; the thermal space; connecting ductwork; dampers; and mixing air components [4]. The operating mode of the representative system is as follows [4]:

- First, At position 1 as shown in Figure 1, 25% of the fresh air and 75% of the recirculated air from position 5 is mixed at the flow mixed.
- Second, air mixed at the flow mixer (position 1) enters the heat exchanger, where it is conditioned.
- Third, the conditioned air is moved out of the heat exchanger as shown in position 2. This air is ready to enter the thermal space.
- Fourth, the supply air enters the thermal space in position 3 and offsets the sensible (actual heat) and latent (humidity) heat loads acting upon the system.
- Finally, the air in the thermal space is drawn through a fan as shown in position 4. 75% of the air is recirculated and 25% is exhausted to the outdoor environment.

The control variables in this system are 1) the speed of the variable-air-volume (VAV) fan as shown in position 2. 2) the water flow rate from the chiller to the heat exchanger. The control system equations can be derived from energy conservation principles and are shown in [4]:

$$\begin{aligned}
\dot{T_3} &= \frac{f}{V_s}(T_2 - T_3) - \frac{h_{fg} f}{C_p V_s}(W_s - W_3) + \frac{1}{0.25 C_p V_s}(Q_o - h_{fg} M_o) \\
\dot{W_3} &= \frac{f}{V_s}(W_s - W_3) + \frac{M_o}{\rho V_s} \\
\dot{T_2} &= \frac{f}{V_{he}}(T_3 - T_2) + \frac{0.25 f}{V_{he}}(T_o - T_3) \\
&\quad - \frac{f h_w}{C_p V_{he}}((0.25 W_o + 0.75 W_3) - W_s) - 6000 \frac{\text{gpm}}{p C_p V_{he}}
\end{aligned} \tag{1}$$

where h_w is the enthalpy of liquid water, W_o is the humidity ratio of outdoor air, h_{fg} is the enthalpy of water vapor, V_{he} is the volume of heat exchanger, W_s is the humidity ratio of supply air, W_3 is the humidity ratio of thermal space, C_p is the specific heat of air, T_o is the temperature of outdoor air, M_o is the moisture load, Q_o is the sensible heat load, T_2 is the temperature of supply air, T_3 is the temperature of thermal space, V_s is the volume of thermal space, ρ is the air mass density, f is the volumetric flow rate of air (ft /min) and gpm is the flow rate of chilled water (gal/min). Among them, f and gpm are control variables, T_2, T_3 and W_3 are sensible states, Q_o and M_o are latent/hidden states and the rest are system parameters. In [4], the author proposes a reduced-order observer as an estimate of the latent/hidden/unmeasurable states. Then, a disturbance rejection controller is proposed to solve the linear time-invariant state feedback system. However, there are several drawbacks of this approach:

- The HVAC system is modeled as linear systems due to many perfect systems assumptions [4]. However, they may not be true in real systems.
- The latent states are infeasible to measure at run time.
- The mathematical equations are only applicable to this system dynamics and we need to derive manually when it changes. It is even worse that some real system are too complicated to model using equations.

This leads us to the model-based reinforcement learning approach, where we directly learn system transition model using data. Before that, we introduce the Partial Observable Markov Decision Process (POMDP) that reinforcement learning algorithms solve.

3.2 Partial Observable Markov Decision Process

3.2.1 Notations. We summarize the notations used in this paper as follows:

- $a(t)$: action vector at time step t, which is the control variable f and gpm mentioned in Section 3.1.
- $s(t)$: state vector at time step t, which is $T_o, M_o, Q_o, T_2, T_3, W_3$ in Section 3.1.
- $o(t)$: observation vector at time step t, which is T_2, T_3, W_3 in Section 3.1.
- $p(s(t+1)|s(t), a(t))$: state transition function, which is Equation 1 in Section 3.1.
- $r(t)$: reward at time step t. It can be defined as $-c(t)$, where $c(t)$ is the cost function.
- $c_i(t)$: constraint i at time step t with upper bound $c_{i,max}$ and lower bound $c_{i,min}$. Typical constraints include comfort temperature bound.
- γ: discount factor in reinforcement learning.

3.2.2 Problem Formulation. We can rewrite Equation 1 in discrete time domain and generalize it as follows:

- $s(t+1) = f_{sys}(s(t), a(t))$
- $o(t) = f_{obs}(s(t))$
- $r(t) = f_{out}(o(t), a(t))$
- $c_i(t) = f_{cons_i}(o(t), a(t))$

where $f_{sys}, f_{obs}, f_{out}, f_{cons_i}$ stands for system dynamics, observation emission, reward/cost function, constraint function i, respectively. The objective of POMDP is to maximize discounted reward while satisfying each constraint at each time step:

$$\max \sum_{t=0}^{\infty} \gamma^t r(t), \text{, s.t. } c_{i,min} \le c_i(t) \le c_{i,max}, \forall i = 1, 2, \cdots, N \tag{2}$$

In most building HVAC systems, the cost function is energy consumption or energy cost and constraints are temperature and humidity range. By collecting data from actuators and sensors in building HVAC system, we can fit these functions using general function approximators (e.g. neural networks) without knowing exact complex underlying physics and solve constrained trajectory optimization problem [5].

Note that the state $s(t)$ is not fully measurable in real systems and f_{sys} cannot be directly fitted. Instead, we assume the observation satisfies W-step Markov property and predict the next step observation conditioned on a sliding window of previous W steps observation and actions:

$$\hat{o}(t+1) = f_{obs,sys}(o(t-W+1:t), a(t-W+1:t)) \tag{3}$$

4 REINFORCEMENT LEARNING FOR BUILDING HVAC CONTROL

4.1 Model-Free Approach

In model-free reinforcement learning (MFRL), the agent interacts with the building environment and optimize the policy directly. Since MFRL cannot deal with constraints, reward shaping [17] is required to combine both cost and constraints into a single reward signal through penalty. Following [15], we define our reward function as follows

$$r = r_T + \lambda_P r_P \tag{4}$$

where r_T represents the cost, r_P represents the constraints and λ controls the tradeoff between the cost and the constraints. A careful fine-tuning of λ is required based on different problem emphasis. Data efficiency is the major problem in MFRL since it is generally not possible to sample large amount of data in real systems, which makes the agent convergence extremely slow.

In this paper, we train a HVAC controller in our simulated environment using Proximal Policy Optimization [23] with modified reward defined in Equation 4 as baseline approaches for performance comparison.

4.2 Model-Based Approach

In model-based reinforcement learning (MBRL), the agents learn a system dynamics function by interacting with the systems and use the learned system dynamics to perform trajectory optimization to obtain optimal action sequence. In this section, we elaborate our MBRL approach for building HVAC control.

4.2.1 System Description. We illustrate our overall system diagram in Figure 2 and detailed procedure in Algorithm 1. The MBRL agent consists of four parts: dataset $\mathcal{D}$, neural network dynamics model, model predictive control (MPC) and neural network based imitation policy.

Model Dynamics Representation. We parameterize our model dynamics by $f_{obs,sys}(\cdot;\theta)$ as shown in Equation 3 by a deep neural network, where θ represents the weights. Following [16], we consider learning deterministic dynamics model by fitting $\Delta o = o(t+1) - o(t)$ instead of $o(t+1)$ since this function approximator would be hard to learn if the adjacent observations are similar and the effect of actions is neglected [16]. Advanced stochastic dynamics models including Gaussian Process [20] are candidates for future work.

Data collection. We collect the initial training dataset by executing the default controller (rule-based supervisory or PID) action $a(t)$ and obtain the next observation $o(t+1)$. The dataset $\mathcal{D}$ is a trajectory (execution sequence) of $(o(0), a(0), o(1), a(1), \cdots, o(N-1), a(N-1), o(N))$. Note that this is only the dataset for training the initial dynamics model. Later on, we will add on-policy execution data into $\mathcal{D}$ such that the learned dynamics model can adapt to the potential missing or changes of dynamics distribution.

Data preprocessing. The neural network based dynamics model takes previous W time step observations $o(t-W+1:t)$ and actions $a(t-W+1:t)$ and output the next observation $o(t+1)$ as shown in Figure 2. In building HVAC control, observations can be temperature, humidity ratio, power, etc. These measurements have various range and the weights of the losses will be different if we feed the raw value directly to train the neural network model. Thus, we subtract the mean of the observation/action and divide by the standard deviation as shown in Equation 5

$$x' = \frac{x - \overline{x}}{\sigma(x)} \tag{5}$$

where x stands for observation or action.

Training dynamics model. We train the dynamics model by minimizing Mean Square Error (MSE) between predicted delta observation and ground truth delta observation as follows:

$$\begin{aligned}\mathcal{E}(\theta) = &\frac{1}{|\mathcal{D}|} \sum_{\substack{o(t-W+1:t)\in\mathcal{D} \\ a(t-W+1:t)\in\mathcal{D}}} \frac{1}{2} || o(t+1) \\ &- f_{obs,sys}(o(t-W+1:t), a(t-W+1:t); \theta) ||^2\end{aligned} \tag{6}$$

We perform stochastic gradient descent [21] on Equation 6 for M epochs. Selecting M is a little bit tricky. Large M may cause $f_{obs,sys}$ to overfit to the current distribution and fail to adapt when new data appended to the dataset. Small M may cause $f_{obs,sys}$ to underfit and deteriorate the performance of Model Predictive Control.

Model Predictive Control. With learned dynamics model, we perform constrained trajectory optimization of horizon H as follows:

$$\begin{aligned}A_t^H = &\arg\max_{A_t^H} \sum_{t'=t}^{t+H-1} r(\hat{o}(t'), a(t')) : \hat{o}(t) = o(t), \\ &\hat{o}(t'+1) = f_{obs,sys}(o(t'-W+1:t'), a(t'-W+1:t'))\end{aligned} \tag{7}$$

In model predictive control (MPC) [9], we only take the first action Equation 7 returns and rerun constrained trajectory optimization for the next time step. We consider random-sampling shooting method [19] to perform MPC, in which K random action sequences are generated and evaluated by the objective and the constraints using the learned dynamics. Then, we select the one with maximum reward that satisfy the constraints or with minimum violations.

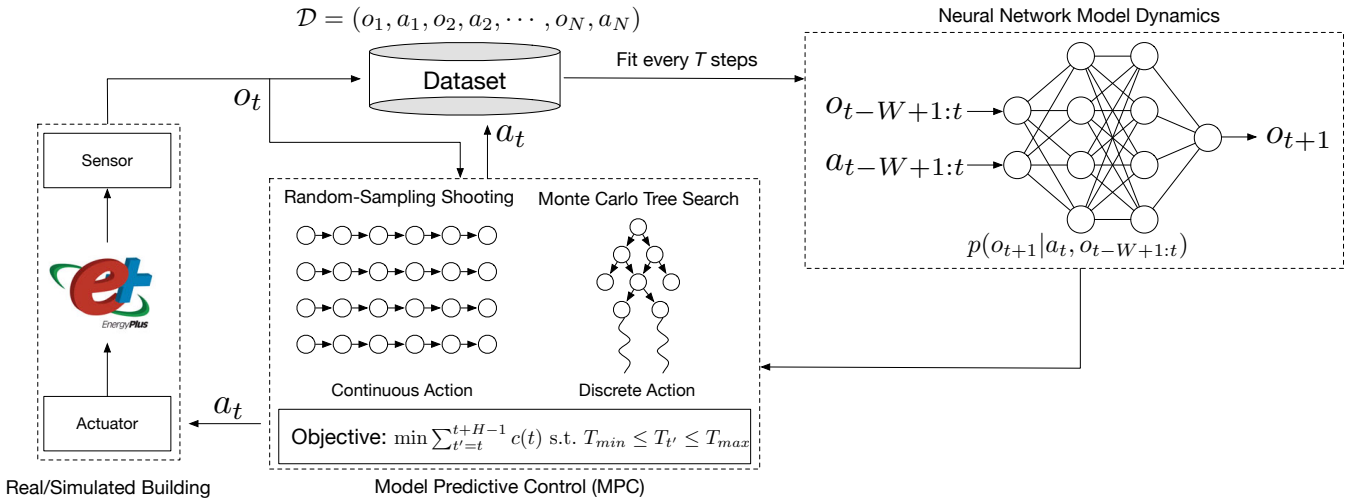


Figure 2: System overview of our model-based reinforcement learning for building HVAC control

```
Gather dataset 𝒟 using default policy;
Randomly initialize model parameter f_{obs,sys}(·;θ);
while True do
    Fit f_{obs,sys}(·;θ) using 𝒟 on Equation 6 by performing M epochs stochastic gradient descent;
    for i = t : t+T do
        Obtain building observation o(t) from sensors;
        Obtain historical observations o(t−W+1 : t−1) from 𝒟;
        Solve optimization problem defined in Equation 7 and obtain action sequence A_t^H;
        Execute the first action a(t) returned from A_t^H;
        Append (o(t), a(t)) to 𝒟;
    end
end
```

Algorithm 1: Model-based Reinforcement Learning for HVAC Control

Imitating MPC using Neural Networks. The main issue of the MPC algorithm is the poor runtime performance and it is infeasible to perform real-time control. Thus, we adopt the idea of DAGGER [22] by training a neural network $f_{imit}(\cdot;\phi)$ that clones the output of model predictive controller with on-policy data aggregation. To achieve this, we need another dataset that stores the observation and action pair returned from MPC **offline**. Then, we minimize the following objective by stochastic gradient descent [21]:

$$\min_{\phi} \frac{1}{2} ||f_{imit}(o(t-W+1:t);\phi) - a(t)||^2 \tag{8}$$

The detailed procedure is shown in Algorithm 2. A neural network based policy has extremely fast inference speed for real-time control. However, it may increase constraints violation rate, which is not ideal if it affects system security.

```
Gather dataset 𝒟 using default policy;
Initialize empty dataset 𝒟′ for observation-action pair;
Randomly initialize model parameter f_{obs,sys}(·;θ);
Randomly initialize policy parameter f_{imit}(·;φ);
while True do
    Fit f_{obs,sys}(·;θ) using 𝒟 on Equation 6 by performing stochastic gradient descent M epochs;
    Fit f_{imit}(·;φ) using 𝒟′ on Equation 8 by performing stochastic gradient descent M epochs;
    for i = t : t+T do
        Obtain building observation o(t) from sensors;
        Obtain historical observations o(t−W+1 : t−1) from 𝒟;
        Compute a(t) = f_{imit}(o(t−W+1 : t);φ) and execute a(t);
        Append (o(t), a(t)) to 𝒟;
        Solve optimization problem defined in Equation 7 and obtain action sequence A_t^H offline;
        Append the first action and observation pair (o(t−W+1 : t), a_{mpc}(t)) to 𝒟′;
    end
end
```

Algorithm 2: Model-based Reinforcement Learning for HVAC Control with Neural Network Policy

4.2.2 *Discussions and Notes.*

- **Dataset replacement policy**: We simply apply First-In-First-Out (FIFO) policy to replace old data. The reason is that $f_{obs,sys}$ needs to adapt to the new system dynamics distribution as the system progresses. In this sense, our approach is **online learning** and is advantageous to model-free approach.

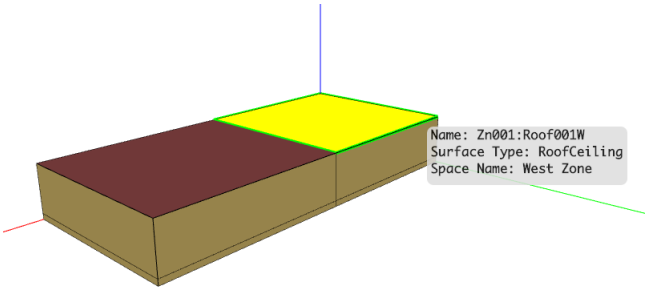


Figure 3: Case study: two room data center

- **Safety-aware exploration**: Exploration plays a crucial role in reinforcement learning. In model-free reinforcement learning, the agents try novel actions and obtain rewards signal from these actions in order to update policy network. However, certain actions are forbidden in real systems due to security. In this case, a model is necessary for the agent to foresee the outcome.

5 CASE STUDY

As a case study, we evaluate our model-based reinforcement learning approach on a two-room data center proposed in [15]. The testbed is based on OpenAI Gym [6] and EnergyPlus [8] and open sourced at https://github.com/IBM/rl-testbed-for-energyplus.

5.1 System Modeling

5.1.1 Overall Description. The target system contains two zones (east zone and west zone), where the thermal load is IT Equipment (ITE) such as servers as shown in Figure 3. Each zone has a dedicated HVAC system similar to Figure 1 with the following components: outdoor air system (OA System), variable volume fan (VAV Fan), direct evaporative cooler (DEC), indirect evaporative cooler (IEC), direct expansion cooling coil (DX CC) and chilled water cooling coil (CW CC) [15]. For each zone, the temperature for all the components are specified by a common setpoint. The air volume supplied to each zone is also adjusted by the VAV Fan.

5.1.2 POMDP Formulation.

Observations. The observation vector contains:

- T_{out}: outdoor air temperature
- T_{west}: west zone air temperature
- T_{east}: east zone air temperature
- P_{ite}: IT equipment (ITE) electric demand power
- P_{hvac}: HVAC electric demand power

Raw Actions. The action vector contains:

- TS_{west}: west zone setpoint temperature
- TS_{east}: east zone setpoint temperature
- F_{west}: west Zone supply fan air mass flow rate
- F_{east}: east Zone supply fan air mass flow rate

Safety-Aware Exploration and Control Strategy. According to our prior knowledge of the building HVAC system, a good temperature setpoint scheduling should fluctuate around the target temperature within safety bounds. Moreover, the maximum absolute changes in setpoint temperature and supply fan air mass flow rate must be limited to ensure hardware security. Following [13], we redefine our action space as:

$$a(t) = \text{clip}(\Delta \times z(t) + a(t-1), a_{min}, a_{max}) \tag{9}$$

where $a(t)$, $a(t-1)$ is the raw action at t and $t-1$. a_{min} and a_{max} are safe action bounds. Δ is the maximum action change. $z(t)$ is normalized action within $[-1, 1]$.

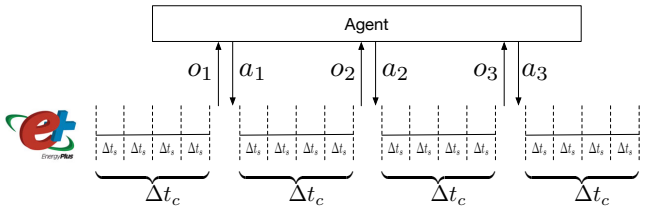


Figure 4: Building Control Sequence

Objective. Minimize the total power consumption P_{total}, where $P_{total} = P_{ite} + P_{hvac}$.

Constraints. The west and east zone temperature is maintained within certain bounds, which is defined as

$$T_{min} \leq T_{west} \leq T_{max}, \quad T_{min} \leq T_{east} \leq T_{max} \tag{10}$$

where T_{min} and T_{max} is the minimum and maximum temperature.

Reward Function. We adopt the reward function defined in [15] to train agents using model-free RL approaches e.g. PPO [23] and Model Predictive Control. It incorporates total power consumption and temperature violation as defined in Equation 4 where

$$r_T = -\sum_{i=\substack{east\\west}} (e^{(-\lambda_1(T_i - T_C)^2)} + \lambda_2([T_i - T_{min}]_+ + [T_{max} - T_i]_+)) \tag{11}$$

$$r_P = -P_{total} \tag{12}$$

This reward function encourages the agents to maintain temperature as close to T_C as possible while minimizing total power consumption. In our experiments, we set $\lambda_1 = 0.5$, $\lambda_2 = 0.1$, $\lambda_P = 10^{-5}$.

5.1.3 Building Control Sequences. We show the building control sequences in Figure 4. There are two types of timesteps during the simulation. Δt_s refers to the internal simulation timestep in EnergyPlus [8]. Its length varies dynamically ranging from 1 minute/timestep to zone timestep (15 minutes/timestep in this case) to balance simulation precision and speed. Δt_c refers to the control interval, which is set to 15 minutes/timestep. During each control interval, the environment sends the **average** observation of all simulation steps to the agent and the agent send the action back to the simulation environment. The same action is **repeated** during each simulation step within each control interval.

5.2 Simulation Setup

5.2.1 Parameter Settings. We show the parameter settings of our model-based RL in Table 1. Notice that the maximum air temperature is set to target temperature since ITE always heat the air.

Table 1: Parameter Settings in Model-based RL

Size of historical data	6240/65 days
Batch Size	128
Dataset size $\|\mathcal{D}\|$	11520/120 days
# of random action sequences K	8192
$\Delta TS_{west,,east}/\Delta F_{west,east}$	1°C/1
$TS_{west,east}$ lower bound/upper bound	13.5°C/23.5°C
$F_{west,east}$ lower bound/upper bound	2.5/10.0
Train/Validation split ratio	0.8/0.2
Comfortable Zone	23.5° ± 1.5°C

Table 2: Amplitude of IT Equipment Simulation Data

Time Period	0:00-6:00	6:00-8:00	8:00-18:00	18:00-24:00
Normalized Amplitude	0.5	0.75	1.00	0.80

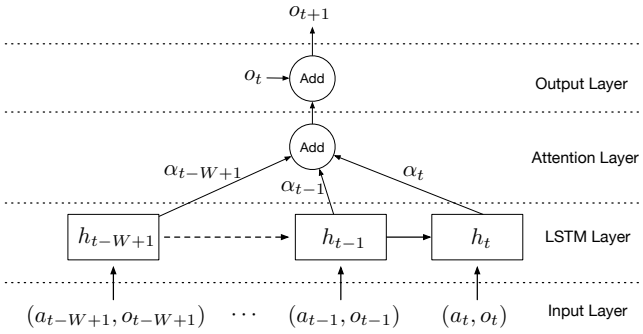


Figure 5: Long short-term memory (LSTM) with attention based system dynamics

5.2.2 *Data Acquisition.*

Weather Data. We use historical weather data bundled with EnergyPlus [8] for our experiments. They are collected and published by the World Meteorological Organization from the following locations:

- **SF:** San Francisco IntâĂŹl Airport, CA, USA
- **Golden:** National Renewable Energy Laboratory at Golden, CO, USA
- **Chicago:** Chicago-OâĂŹHare IntâĂŹl Airport, IL, USA
- **Sterling:** Sterling-Washington Dulles IntâĂŹl Airport, VA, USA

IT Equipment Data. The data center server load is simulated by white noise of various amplitude within different time period of a day as shown in Table 2.

Baseline Controllers. We evaluate the performance of EnergyPlus builtin controllers [8] and Proximal Policy Optimization (PPO) [23] based controllers as baselines.

5.2.3 *Neural Network Model Dynamics Architecture.*

We show the neural network based model dynamics architecture in Figure 5. The idea of the model dynamics architecture is adapted from [30]. It contains Input Layer, LSTM Layer, Attention Layer and Output Layer. The Input Layer takes in observations and actions within previous W steps. The LSTM Layer is used to extract time series features. The Attention Layer is used to combine those features with automatically learnable weights $\alpha_{t-W+1}, \cdots, \alpha_t$. The Output Layer adds $o(t)$ and produce next step observation $o(t+1)$.

Table 3: H-step Deviation Percentage of Various Window Lengths and Cities. $H = 96$

		Window Length			
		5	10	15	20
City	SF	0.37	1.08	0.19	0.19
	Golden	0.15	0.19	0.29	0.14
	Chicago	0.17	0.68	0.20	0.15
	Sterling	0.48	0.39	0.34	0.15

5.3 Performance Metric

5.3.1 *Learned System Dynamics.*

We evaluate the performance of learned system dynamics using ***H*-step deviation percentage**. Given action sequence $a(t-W+1:t+H)$ and observation sequence $o(t-W+1:t+H)$, we predict $o(t+1:t+H)$ using open-loop prediction as:

$$\begin{aligned}\hat{o}(t+h) =& f_{obs,sys}(\hat{o}(t-W+h:t+h), a(t-W+h:t+h)) \\ & , h = 1, 2, \cdots, H \\ \hat{o}(i) =& o(i), i = t-W+1, \cdots, t\end{aligned} \tag{13}$$

Then, the ***H*-step deviation percentage** is given as:

$$d_h = \frac{1}{H}\sum_{h=1}^{H} || \frac{o(t+h) - \hat{o}(t+h)}{o(t+h)} || \tag{14}$$

5.3.2 *Controller.*

We evaluate **energy efficiency** of building HVAC controllers by **Daily Average Power Consumption** and **temperature requirements** by **Daily Temperature Violation Rate (TVR)**. The daily average power consumption includes ITE and HVAC power. The daily TVR is defined as the ratio between the number of steps that temperature violates the constraints and the total number of steps within a day.

5.4 Evaluation of Learned Dynamics

We collect ground truth observation sequence of the HVAC system by executing random actions sequences. Then, we perform ***H*-step open loop prediction** defined in Equation 13 and measure the ***H*-step deviation percentage** defined in Equation 14 of various window length and cities in Table 3. We set H to be 96, which amounts to a day. We also show the open-loop predictions vs. ground truth curve for west zone temperature in Figure 6.

We observe that larger window length yields more accurate predictions in general as shown in Table 3. In Figure 6, the west temperature prediction of Chicago with window length 10 even diverges; it may lead to catastrophic outcomes if this predicted temperature is used for optimization. Thus, we set window length to be 20 in all later experiments.

5.5 Evaluation of Design Decision

We perform the evaluation of various design decision by training our MBRL agent in two-room data center environment and report

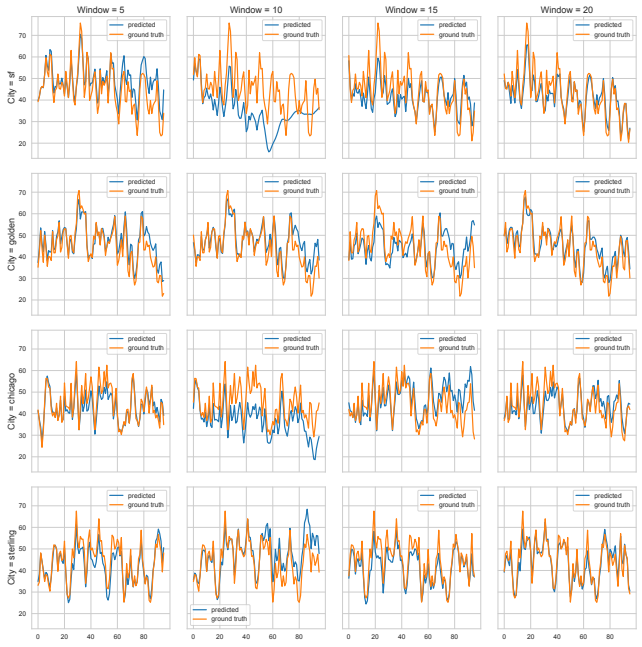


Figure 6: West zone temperature (°C) predictions vs. ground truth for random action open-loop predictions

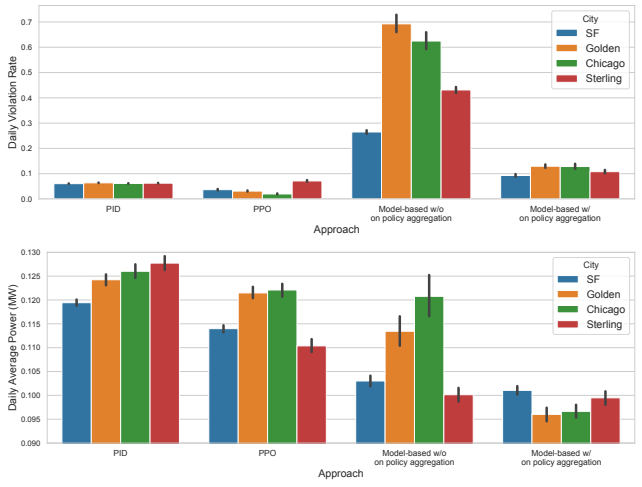


Figure 7: Daily Violation Rate and Average Power Consumption of various algorithms

the Daily Average Power Consumption and Daily Temperature Violation Rate.

5.5.1 Training Epochs. It refers to the number of epochs M to fit model dynamics using current dataset in Algorithm 1 and Algorithm 2. The model may be underfit and fail to predict future observations accurately if M is too small. On the contrary, the model may overfit to the current dataset and hard to adapt when the data distribution changes over time if M is too large. We vary training epochs by 30, 60, 90 and 120 and show the control performance in Figure 10. It is shown that the performance of training epochs 30 works best.

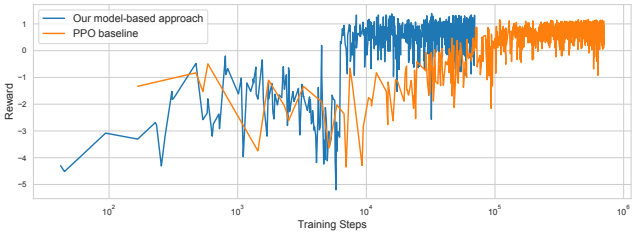


Figure 8: Reward vs. Training steps of model-based and PPO

5.5.2 MPC Horizon. The MPC horizon refers to the number of steps to look ahead when performing model predictive control (MPC). Small MPC horizon results in more greedy actions that may fail to overcome the inertia of thermal dynamics. Large MPC horizon may produce worse actions since the prediction errors aggregate as the horizon becomes larger. We vary the MPC horizons by 5, 10, 15 and 20 and show the control performance in Figure 11. Results show that the controller works best when MPC horizon equals 5 steps.

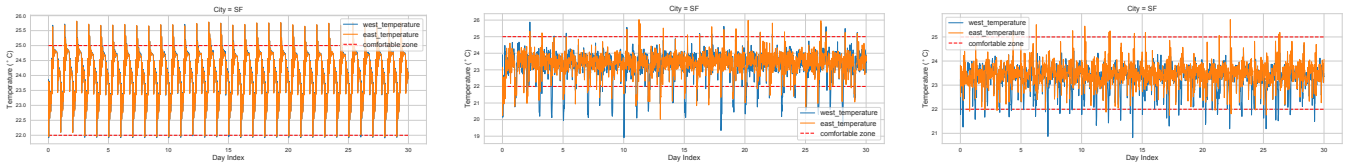
5.5.3 On Policy Frequency. The on policy frequency refers to the number of days the agent uses the current system dynamics to perform model predictive control. It refers to T in Algorithm 1 and Algorithm 2. Smaller T results in more system dynamics training iterations with the risk of overfitting to the current data distribution. Large T results in fewer system dynamics training iterations with the risk of failing to adapt to the new data distribution. We vary the on policy frequency by 3, 7, 10 and 14 days and show the control performance in Figure 12. Results show that the controller works best when on policy frequency equals 7 days.

5.5.4 Imitation Learning. The main issue of model predictive control is the slow processing speed that fails for real-time control with finer control intervals. Thus, we train a neural network policy that mimics the output of MPC for real time control. We show the control performance of both MPC and neural network policies in Figure 13. We observe that the neural network policy works worse because the historical data collected from MPC may not cover all future control scenarios and the neural network fail to generalize.

5.6 Model Predictive Control using Learned Dynamics on Benchmarks

We evaluate our learned model-based controller using the best configuration found in Section 5.5. As comparison, we evaluate the performance of model-free approach (PPO) and builtin controller on the same environment.

5.6.1 Effectiveness of our approach in satisfying temperature requirements. We show one month controlled temperature curve of various algorithms in Figure 9. We observe that both model-based and model-free RL manages to maintain the west and east zone temperature around 23.5°C.



(a) Builtin control (b) Model-based Reinforcement Learning with Neural Network Dynamics (c) Model-free Reinforcement Learning (PPO)

Figure 9: One month test temperature curve of various algorithms

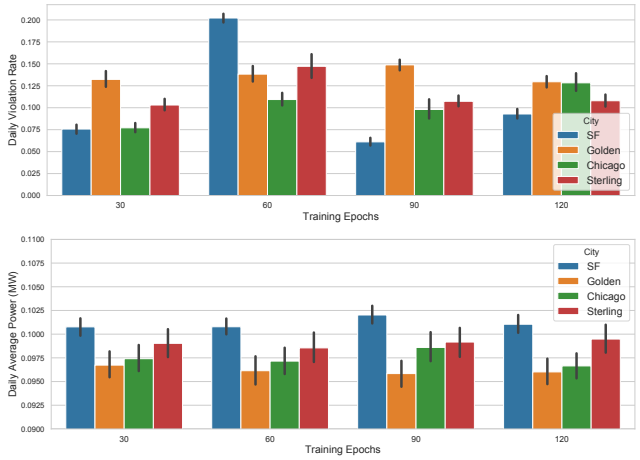


Figure 10: Daily Violation Rate and Average Power Consumption of various dynamics model training epochs

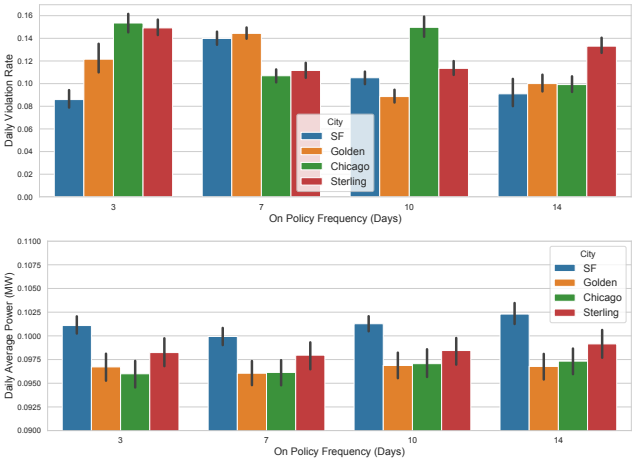


Figure 12: Daily Violation Rate and Average Power Consumption of various on policy steps

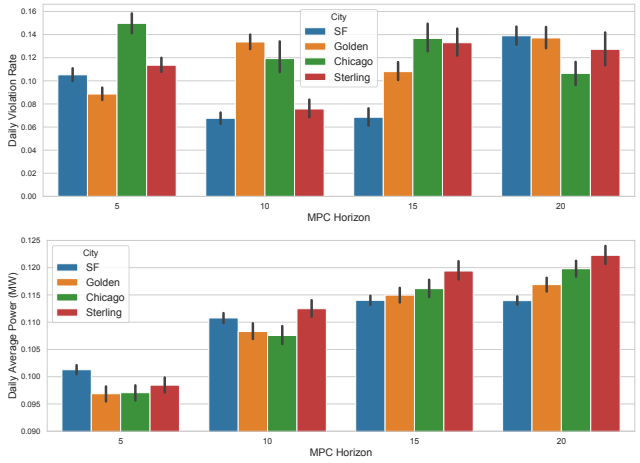


Figure 11: Daily Violation Rate and Average Power Consumption of various MPC horizons

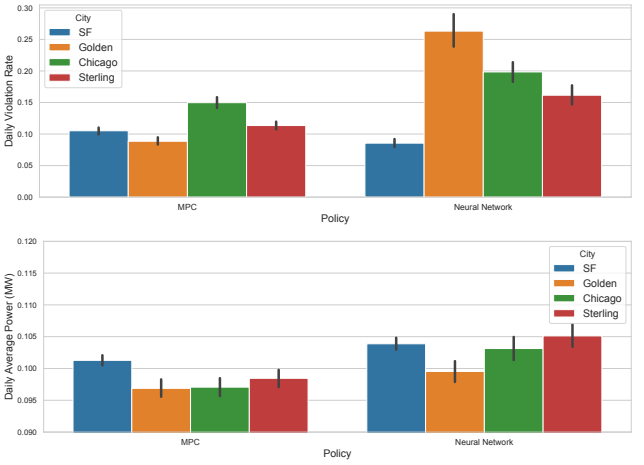


Figure 13: Daily Violation Rate and Average Power Consumption of MPC vs. Imitation Learning

5.6.2 Performance Comparison. We show the daily violation rate and daily average power in Figure 7. Compared with model-free approach, our model-based approach achieves 17.1% ∼ 21.8% power reduction with slightly increased violation rate. Also, the on-policy data aggregation plays critical role in addressing observation distribution shifting as shown by the violation reduction.

5.6.3 Computation Speed Analysis. As the experiments suggest, the MPC takes more than 30 seconds to finish while the neural

network policy takes less than 1 second for inference. It indicates the advantages of neural network policy in high frequency control. However, possible failure modes need to be prevented by proper interference due to the weak robustness of neural network policy.

5.6.4 Convergence Analysis. Although, model-free RL approach achieves similar control performance with model-based approach, it requires tremendous amount of trial-and-error with the actual environments, which is not possible in real systems. We show the reward vs. training steps of model-based approach and PPO in Figure 8. We notice that model-free approaches requires approximately 10x more training steps to converge to the same performance level as model-based approach.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a model-based reinforcement learning approach for building HVAC scheduling via neural network based model approximation. We first learn the system dynamics using neural network by collecting data through interactions with the system. Then, we use the learned system dynamics to perform model predictive control using random-sampling shooting method. To overcome system distribution shift such as outside temperature and IT equipment load schedules, we retrain the dynamics using on-policy data aggregation. Experiments show that our approach achieves significant improvement of power reduction compared with baseline controllers. Compared with model-free reinforcement learning approach (PPO), our approach improves the sample efficiency by 10x.

Future work includes conducting experiments with more complex systems containing larger observation and action space, or even probabilistic system dynamics. It is also useful to experiment with time-variant system objective and constraints and see how our model-based approach advantages over model-free approaches.

REFERENCES

[1] 2019. Control Systems/State-Space Equations. https://en.wikibooks.org/wiki/Control_Systems/State-Space_Equations

[2] Abdul Afram, Farrokh Janabi-Sharifi, Alan S. Fung, and Kaamran Raahemifar. 2017. Artificial Neural Network (ANN) based Model Predictive Control (MPC) and Optimization of HVAC Systems: A State of the Art Review and Case Study of a Residential HVAC System. *Energy and Buildings* 141 (02 2017). https://doi.org/10.1016/j.enbuild.2017.02.012

[3] S. Ahmad and M. O. Tokhi. 2011. Linear Quadratic Regulator (LQR) approach for lifting and stabilizing of two-wheeled wheelchair. In *2011 4th International Conference on Mechatronics (ICOM)*. 1–6. https://doi.org/10.1109/ICOM.2011.5937119

[4] B. Arguello-Serrano and M. Velez-Reyes. 1999. Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation. *IEEE Transactions on Control Systems Technology* 7, 1 (Jan 1999), 56–63. https://doi.org/10.1109/87.736752

[5] JOHN T. BETTS and WILLIAM P. HUFFMAN. 1993. Path-constrained trajectory optimization using sparse sequential quadratic programming. *Journal of Guidance, Control, and Dynamics* 16, 1 (1993), 59–68. https://doi.org/10.2514/3.11428 arXiv:https://doi.org/10.2514/3.11428

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *CoRR* abs/1606.01540 (2016). arXiv:1606.01540 http://arxiv.org/abs/1606.01540

[7] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari. 2018. Approximating Explicit Model Predictive Control Using Constrained Neural Networks. In *2018 Annual American Control Conference (ACC)*. 1520–1527. https://doi.org/10.23919/ACC.2018.8431275

[8] Drury B. Crawley, Curtis O. Pedersen, Linda K. Lawrie, and Frederick C. Winkelmann. 2000. EnergyPlus: Energy Simulation Program. *ASHRAE Journal* 42 (2000), 49–56.

[9] C. Ekaputri and A. Syaichu-Rohman. 2012. Implementation model predictive control (MPC) algorithm-3 for inverted pendulum. In *2012 IEEE Control and System Graduate Research Colloquium*. 116–122. https://doi.org/10.1109/ICSGRC.2012.6287146

[10] Guanyu Gao, Jie Li, and Yonggang Wen. 2019. Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep Reinforcement Learning. arXiv:arXiv:1901.04693

[11] Reid Hart. 2016. Introduction to Commercial Building HVAC Systems and Energy Code Requirements. https://www.energycodes.gov/sites/default/files/becu/HVAC_Systems_Presentation_Slides.pdf.

[12] Chang-Soon Kang, Jong-Il Park, Mignon Park, and Jaeho Baek. 2014. Novel Modeling and Control Strategies for a HVAC System Including Carbon Dioxide Control. *Energies* 7 (06 2014), 3599–3617. https://doi.org/10.3390/en7063599

[13] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. 2018. Data center cooling using model-predictive control. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 3814–3823. http://papers.nips.cc/paper/7638-data-center-cooling-using-model-predictive-control.pdf

[14] Michael L. Littman. 2009. A tutorial on partially observable Markov decision processes.

[15] Takao Moriyama, Giovanni De Magistris, Michiaki Tatsubori, Tu-Hoa Pham, Asim Munawar, and Ryuki Tachibana. 2018. Reinforcement Learning Testbed for Power-Consumption Optimization. *CoRR* abs/1808.10427 (2018). arXiv:1808.10427 http://arxiv.org/abs/1808.10427

[16] Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. 2017. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. *CoRR* abs/1708.02596 (2017). arXiv:1708.02596 http://arxiv.org/abs/1708.02596

[17] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 278–287. http://dl.acm.org/citation.cfm?id=645528.657613

[18] U.S. Department of Energy. 2011. Buildings Energy Data Book. https://ieer.org/resource/energy-issues/2011-buildings-energy-data-book/.

[19] Anil Rao. 2010. A Survey of Numerical Methods for Optimal Control. *Advances in the Astronautical Sciences* 135 (01 2010).

[20] Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[21] H. Robbins and S. Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22 (1951), 400–407.

[22] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2010. No-Regret Reductions for Imitation Learning and Structured Prediction. *CoRR* abs/1011.0686 (2010). arXiv:1011.0686 http://arxiv.org/abs/1011.0686

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[24] Jian Sun and T Agami Reddy. 2005. Optimal control of building HVAC&R systems using complete simulation-based sequential quadratic programming (CSB-SQP). *Building and Environment* 40, 5 (5 2005), 657–669. https://doi.org/10.1016/j.buildenv.2004.08.011

[25] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. *CoRR* abs/1509.06461 (2015). arXiv:1509.06461 http://arxiv.org/abs/1509.06461

[26] Shengwei Wang and Zhenjun Ma. 2008. Supervisory and Optimal Control of Building HVAC Systems: A Review. *HVAC&R Research* 14, 1 (2008), 3–32. https://doi.org/10.1080/10789669.2008.10390991 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/10789669.2008.10390991

[27] T. Wei, Yanzhi Wang, and Q. Zhu. 2017. Deep reinforcement learning for building HVAC control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1145/3061639.3062224

[28] Ya-Gang Wang, Zhi-Gang Shi, and Wen-Jian Cai. 2001. PID autotuner and its application in HVAC systems. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, Vol. 3. 2192–2196 vol.3. https://doi.org/10.1109/ACC.2001.946075

[29] Zhiang Zhang and Khee Poh Lam. 2018. Practical Implementation and Evaluation of Deep Reinforcement Learning Control for a Radiant Heating System. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys '18)*. ACM, New York, NY, USA, 148–157. https://doi.org/10.1145/3276774.3276775

[30] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *ACL*.