Building Demand Response Control Through Constrained Reinforcement Learning with Linear Policies [⋆,⋆⋆]

Jerson Sanchez and Jie Cai

Abstract

Recent advancements in model-free control strategies, particularly reinforcement learning (RL), have enabled more practical and scalable solutions for controlling building energy systems. These strategies rely solely on data, eliminating the need for complex models of building dynamics during control decision making, the development of which is expensive involving significant engineering efforts. Conventional unconstrained RL controllers typically manage indoor comfort by incorporating a penalty for comfort violations into the reward function. This penalty function approach leads to control performance very sensitive to the penalty factor setting. A low comfort penalty factor can result in significant violations of comfort constraints while a high penalty factor tends to degrade economic performance. To address this issue, the present study presents a constrained RL-based control strategy for building demand response that explicitly learns a constraint value function from

[⋆]This material is based upon work supported by the National Science Foundation under Grant No. 2238381.

[⋆⋆]Jerson Sanchez is with the School of Aerospace and Mechanical Engineering, University of Oklahoma, Norman, OK, 73072. Jie Cai is with Ray W. Herrick Laboratories, Purdue University, West Lafayette, IN, 47907. (`cai40@purdue.edu`).

operation data. This study considers both linear mapping and deep neural networks for value and policy function approximation to evaluate their training stability and control performance in terms of economic return and constraint satisfaction. Simulation tests of the proposed strategy, as well as baseline model predictive controllers (MPC) and unconstrained RL strategies, demonstrate that the constrained RL approach could achieve utility cost savings of up to 16.1%, comparable to those achieved with MPC baselines, while minimizing constraint violations. In contrast, the unconstrained RL controllers either lead to high utility costs or significant constraint violations, depending on the penalty factor settings. The constrained RL strategy with linear policy and value functions shows more stable training and offers 4% additional cost savings with reduced constraint violations compared to constrained RL controllers with neural networks.
Keywords: Constrained Reinforcement Learning, Demand response, HVAC, Model-free control,

1. INTRODUCTION

Building electrical loads account for 75% of the electricity consumption in the United States [1]. Technologies and solutions for efficient and sustainable building operations have witnessed growing attention in the past few years, with heating, ventilation and air conditioning (HVAC) demand response being one of them. Demand response plays an important role in improving electric power system reliability against uncertainties in renewable generation, peak demand, asset availability, and other grid contingencies [2, 3]. The concept of demand response, originally rooted in the power utility sector, has found its way into the building automation industry as a means to enhance energy resilience, reduce peak load demands, and mitigate the environmental impact of energy consumption. It provides a mechanism for building systems to adapt in real-time to changing energy supply and demand, allowing them to intelligently and autonomously adjust their energy consumption patterns.

Model predictive control (MPC) is a extensively used technique in improving energy efficiency and optimizing performance of building thermal systems. Researchers have explored the application of MPC in a range of applications involving HVAC and building control systems for energy cost reduction and occupant comfort optimization [4, 5, 6]. An extensive and detailed review of MPC applications in buildings can be found in [7]. In the context of demand response, MPC has been researched to reduce peak energy demand under a real time pricing (RTP) regime for building climate control and proved to be effective in reducing overall electricity costs [8]. MPC was also used in demand response to optimize the battery energy storage system and HVAC system schedules to minimize the annual electricity cost with an EnergyPlus building model [9]. Vedullapalli et al. compared different MPC formulations for commercial buildings under time-of-use rates. The proposed formulations were shown to effectively reduce energy costs and increase load-shifting capacity under demand response events, while maintaining low peak demands [10]. In order to determine the maximum potential savings with MPC for commercial buildings under demand response, the effects of look-ahead horizon, timing of peak demand, and peak demand target reset schemes have been studied. It was found that a 24-hour look-ahead horizon can achieve up to 78% of the cost savings of a single-shot monthly optimization [11]. While

MPC offers a powerful framework that can explicitly consider constraints during control decision making, a decent process model is needed to achieve satisfactory control performance, the development of which would require significant engineering costs, especially for complex systems such as buildings. The high upfront costs associated with model development represent a major barrier to the widespread adoption of MPC in building automation industry.

Reinforcement learning (RL) offers an alternative, model-free control approach and numerous studies have researched the application of RL techniques for building controls [12]. The earliest works in HVAC system control with RL techniques used Q-learning for optimized operation of thermal energy storage in commercial buildings [13]. This study found that the RL controller achieved 8.1% of cost savings under TOU rates and 13.9% when real-time pricing is present. However, the control actions led to rebounds of total energy use of 3.5% and 6.2%, respectively, for the two rate structures. While Q-learning showed the capability to learn a working policy, its application was found to be restricted to small-scale control problems since the size of the Q-table increases considerably as the action and observation space dimensions grow. The control problem becomes intractable quickly. Besides, the required training data for achieving an acceptable control policy was found to be significant, another major obstacle for practical applications of the algorithm [14, 15].

With the rapid increase in computing power and the development of new algorithms, such as deep Q-networks (DQN) and policy optimization, better performances have been achieved even for applications in the control of complex systems such as building demand response. Early efforts sought to use deep RL to minimize the electricity cost of the HVAC system using DQN algorithms. However, the study failed to consider time varying temperature bounds with occupancy and the algorithm required hundreds of months of training data to converge [16]. In another study, DQN was used to minimize building energy use while maintaining CO_2 concentration levels under an acceptable threshold [17]. To accelerate DQN learning, Jiang et al. applied an action processor to leverage previously known information from rule-based controllers to reduce training time [18]. Better performance was shown using algorithms similar to or derived from DQN such as deep deterministic policy gradient (DDPG) and asynchronous advantage actor critic (A3C). DDPG was used to optimize energy cost through continuous control of a multi-zone residential building under a simulated retail price sequence composed of two price signals. Comparison of the proposed strategy with a ruled-based algorithm showed energy use cost savings of 15 %. This study also showed the capabilities of the algorithm to generalize to building with different physical characteristics and to different retail price signals [19]. In another study, DDPG was used in a smart home energy management system (HEMS)[20] to minimize energy use cost, achieving 8% to 15% savings. These studies considered fixed comfort temperature bounds, hence the effectiveness of DDPG in handling time varying comfort constraints is unknown. A3C methods were applied for control of multi-zone small and medium size commercial buildings (SMCB) subject to demand charges and their performance was compared with heuristic and MPC controllers showing energy cost savings of 6% [21]. This study showed that policy training using cloud computing is a cost-effective solution, making intelligent DR control affordable and accessible to SMCBs. DQN strategies have also been implemented and demonstrated in real buildings, e.g., a DQN strategy pre-trained with a simulation model was deployed in a residential building demonstrating 10% to 20% cost savings potential [22].

Policy optimization algorithms, unlike value-based methods such as DQN, directly learn a policy from operation data and have also been studied in the context of building controls. A trust region policy optimization (TRPO) controller was proposed for appliance scheduling in residential buildings. The algorithm generated an effective control policy for the appliances although it did not consider occupancy-based, time-varying comfort requirements [23]. A similar study used a multi-agent TRPO algorithm for an autonomous HEMS. The algorithm was trained to optimize the scheduling of appliances considering the dynamic retail price and behind-the-meter solar power generation, and showed better performance than previously studied policy optimization algorithms [24]. In reference [25], a two-stage RL training framework was proposed integrating evolutionary strategies and proximal policy optimization (PPO) to address a grid-interactive building control problem. Pure PPO algorithms were also tested under different demand response scenarios using a building simulation tool (EnergyPlus) showing energy reduction of up to 22% and peak demand reduction of up to 50% [26]. In a different application, PPO was used to find optimal indoor airflow to minimize exposure of the building occupants to pathogens such as COVID-19 [27]. The study highlighted the advantages of using a model-free approach against computationally expensive CFD models. The studies above used unconstrained RL that cannot directly handle constraints, e.g., capacity and indoor comfort constraints in building control applications, during control decision making. In unconstrained RL implementations, constraints are often addressed heuristically, e.g., by adding a penalty for constraint violations in the reward function. However, the control performance is sensitive to the arbitrarily set penalty factor. In building controls, it is difficult to put a price tag on thermal discomfort as individuals would have different perceptions and tolerances thereof. The local utility rate structures could further complicate the situation as the financial benefits can highly depend on the energy rates as well as the peak-to-off-peak ratio.

Constrained RL has arisen as a promising framework to address the challenges of control problems involving operational constraints and has seen applications in various fields. Schmoeller et al. [28] proposed a constrained hierarchical RL (HRL) framework for robot navigation using OpenAI Safety Gym benchmarks in a simulation study. The HRL model consisted of a high-level control policy and a low-level controller. The high-level agent selects target positions to maximize long-term rewards, while the low-level agent controls the robot to reach the identified target locations. A safety layer monitors and overrides unsafe low-level control actions to maintain safe operation. In the energy management field, constrained RL has been proposed for electric vehicle (EV) charging controls. In reference [29], the EV charging/discharging scheduling problem in a demand response context was formulated as a constrained Markov decision process (CMDP) and a constrained policy optimization algorithm was applied to obtain a policy that minimizes the charging cost and satisfies the charging demand. The control performance was compared with different baseline strategies including heuristic controls, DQN, DDPG, MPC, and a theoretical performance upper limit. Test results show that the constrained policy optimization algorithm outperformed most of the baselines by 1 to 2 orders of magnitude on satisfying the charging demand constraints. In another field, a flight decision optimization problem for an unmanned aerial vehicle (UAV) aiming to minimize search time while restricting the number of actions taken by the UAV was formulated as a CMDP [30]. This study applied a UAV tracking scheme based on Gaussian process regression to estimate reference points and used it jointly with multi-agent Q-learning (MQL) to make flight decisions. These studies showed that constrained RL offers an effective framework to tackle control problems with operational constraints.

1.1. Contributions of this work

This study reports a first attempt of applying constrained RL to address the complex building demand responsive control problem, to the authors' best knowledge. We first report the challenges of applying unconstrained RL algorithms to building dmand response control – (1) sensitivity to the choice of the penalty factor, and (2) difficulties in benchmarking these algorithms with other techniques such as MPC-based strategies without guaranteed comfort constraint satisfaction. To overcome the challenges, a constrained RL-based strategy is proposed for building demand response, with both neural network and linear approximates of the policy and value functions. Numerical test results in comparison with baseline unconstrained RL strategies as well as MPC are presented to demonstrate the performance improvements achieved with the proposed controller in terms of utility cost reduction and constraint satisfaction. The simulation results highlight that linear policies can work as well or better than neural network-based policies in terms of training convergence and overall control performance for building demand response.

This paper is structured as follows. First, the numerical building model used as the emulator that interacts with the RL agent is presented in Section 2. MPC formulations for the demand response problem are presented and the associated MPC baseline strategies are defined in Section 3. Sections 4 and 5 introduce the general RL concept along with the state-of-the-art algorithms for both unconstrained and constrained policy optimization. Section 6 reports the simulation test results for the proposed constrained RL strategies as well as the conventional RL and MPC baselines. Concluding remarks are provided in Section 7.

2. Building System Model

This section introduces the building dynamic model used for control testing and MPC synthesis. We need to emphasize that the dynamic model is only utilized as a simulation test bed – an environment that the RL agent interacts with. In the control decision making process, the RL agent (both unconstrained and constrained versions) does not have prior knowledge about the system dynamics and solely relies on the control interactions with the test bed to learn a control policy. However, in the MPC implementation, the same model is assumed for the plant and control synthesis; therefore, the presented MPC performance represents the theoretical upper bound and mainly serves the benchmarking purpose. A discrete-time state-space model is used to reproduce building thermal dynamics, based on a thermal network approach [31]. The model uses the cooling rate as the control input and outputs the zone temperature, in the following form:

$$x^{t+1} = \mathbf{A}x^t + \mathbf{B_w}w^t + \mathbf{B_u}Q_z^t, \tag{1}$$

$$T_z^{t+1} = \mathbf{C}x^{t+1}, \tag{2}$$

where x^t is the state vector containing all nodal temperatures of a thermal network, Q_z^t is the average cooling rate within each time step, T_z^t is the zone temperature, and w^t is the disturbance vector comprised of outdoor temperature, internal heat gains, and solar radiation. In real-building applications, disturbances can present dynamic changes which can be captured by the on-policy RL algorithms presented in this study. The state-space matrices $\mathbf{A}$, $\mathbf{B_w}$, $\mathbf{B_u}$ and $\mathbf{C}$ are dependent on the thermal resistances and capacitances of the thermal network. The model adopts a thermal network approach with the network shown in Fig. 1, which consists of two main wall branches: the floor branch (Flr) that represents most of the thermal storage in the building construction; the external wall (Ext) that bridges the indoor space to the ambient. The radiative internal heat gain ($Q_{gain,rad}$) is applied to the floor and external wall with a uniform heat flux while the convective internal heat gain ($Q_{gain,conv}$) interacts with the air node directly. A facade branch is used to capture the dynamics of the facade cavity temperature ($T_{fac,space}$). The facade branch also captures fast couplings between the indoor, facade and outdoor spaces including infiltration and heat transfer through windows. However, the case study building is a typical commercial office space with negligible infiltration. The solar radiation transmitted through the windows into the indoor space ($Q_{sol,trans}$) and the facade space ($Q_{sol,trans,fac}$) are also modeled where the window transmittance is estimated in the training process. This particular thermal network was obtained after rigorous identifiability analysis following the procedure detailed in [31], which uses norms of the Fisher information matrix to evaluate the informativeness of training data relative to a given model structure. More complex model structures (e.g., with more RC branches) may provide better match to experimental data but suffer from poor structural identifiability. Low-order models come with better identifiability but may not capture all the dynamics. The thermal network structure adopted here achieves a balance between accuracy and identifiability.

The model parameters were estimated from two weeks' worth of operation data collected in an office building in West Lafayette, IN. The estimated model was validated over a longer period of time covering different seasons and operating conditions. The training data includes building operation data from the building management system, weather data collected in a weather station on the building rooftop, and sub-circuit power meters that measure the plug loads and lighting power usage. Further details of the training methodology and the identified parameter values can be found in [31, 32]. The resultant building load model uses the cooling rate as input and predicts the zone temperature.

The model is used as a simulation test bed to evaluate the various control strategies. The same model is adopted for the MPC implementation, while in RL-based control, the system dynamics are not known to the agent and the control policy is learned from the recorded interactions between the agent and the emulator model that include control commands sent to the building and the measurable observations. The power used by the HVAC system can be estimated from the cooling rate by

$$P^t = \frac{Q_z^t}{COP}, \tag{3}$$

where the coefficient of performance (COP) assumes a constant in this study. Assuming a constant COP is a simplifying assumption used in the simulation test bed. But boundary condition-dependent COP can be easily handled by all control strategies, both MPC and RL-based, considered in this study. For example, MPC strategies were shown to be effective in capturing temperature-dependent COP in control decision making [33, 34]. Since ambient and zone temperatures are observable state variables in the RL strategies, their influence on the COP can also be captured via learning.

Eq. (1) to Eq. (3) define the environment which the agent interacts with. The agent applies a temperature setpoint and a set of measurable observations are available to the agent after execution of the set-point. We consider a supervisory level controller to determine the zone temperature set point, while a perfect local controller is assumed to steer the zone temperature to track set point. The set point changes are mostly incremental, so HVAC dynamics are believed to be insignificant compared to the envelope dynamics. In this study, only a cooling scenario is considered although the methods can be directly applied for heating season analysis.

3. Model Predictive Control

MPC optimizes system performance by using a dynamic model to predict future behavior over a specified time horizon. Key components include a dynamic model, a cost function, and constraints. At each time step, an optimization problem is solved to find optimal control inputs, considering

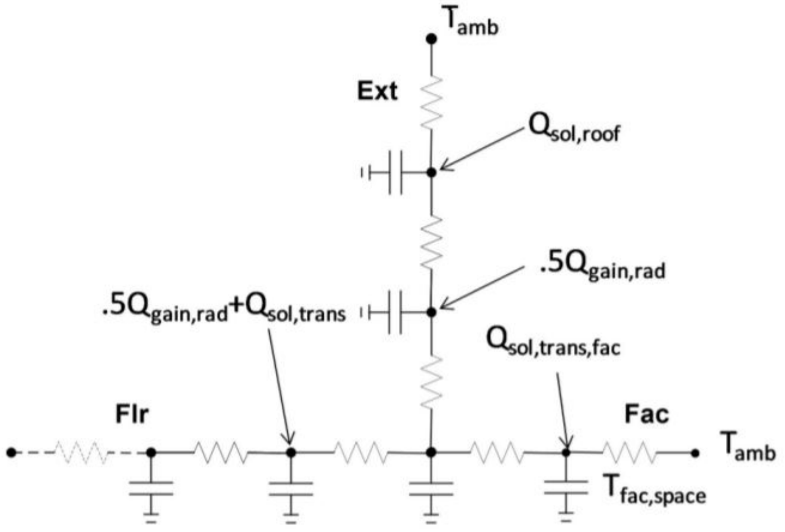

Figure 1: Thermal network model

the dynamic model and constraints over a moving horizon. MPC is advantageous for handling complex systems, managing constraints, and adapting to changes in real-time. For building demand response applications, an MPC uses the forecasts of the outdoor temperature, solar radiation, and internal heat gains to identify the optimal control (zone temperature set-point) trajectory within the look-ahead time horizon (e.g., 24 hours in this work). Only the control decision of the first time step is implemented, while this control decision procedure is repeated when next decision time arrives. Two control objectives are considered for benchmarking purposes.

3.1. Energy Minimization Control (E-Min)

The first baseline strategy represents the current practice for maximum energy efficiency. This strategy tends to maintain the zone temperature as close to the upper limit of the comfort temperature zone as possible to minimize the HVAC energy consumption of each time step, with the following cost function:

$$W_1 = P^t. \tag{4}$$

The energy minimization strategy represents a greedy control policy over the HVAC energy use with only one step ahead prediction. A number of operational constraints should be respected in control decision making. Upper and lower limits are imposed for the zone temperature in order to meet the indoor comfort requirements:

$$T^t_{z,min} \leq T^t_z \leq T^t_{z,max}, \tag{5}$$

where $T^t_{z,min}$ and $T^t_{z,max}$ are the lower and upper bounds of the comfort band. These temperature limits can vary with the occupancy status: when the building is occupied, a tighter temperature constraint is needed to ensure comfort, while during unoccupied hours, relaxed temperature bounds can be used to achieve energy or cost savings. The cooling rate at each time step should be bounded by the cooling capacity Q_T, which is assumed to be time-invariant in this study:

$$0 \leq Q^t_z \leq Q_T. \tag{6}$$

3.2. Utility Cost Minimization Control (U-Min)

The second baseline strategy minimizes the electric utility cost under time-of-use rates over a look-ahead time horizon (e.g., 24 hours in the case study) and represents the current practice for predictive demand responsive control. This strategy can be implemented by solving an LP with a cost function in the following form:

$$W_2 = \sum_{t=t_i}^{t_f} (u^t \cdot P^t), \tag{7}$$

where u^t is the retail energy rate (\$/kWh) that may change with time of the day, t_i and t_f are the first and last time steps of the prediction horizon. The constraints discussed for the E-min strategy are also present in this cost-minimizing strategy, over the whole look-ahead time horizon. E-min seeks to minimize the cumulative energy consumption of a HVAC system and U-min seeks to minimize the total electric utility cost.

4. Unconstrained Reinforcement Learning Controller

This section introduces the formulation and setup of the unconstrained RL controller for building demand response, which serves as a second benchmark to assess the performance of the proposed constrained RL control strategy.

4.1. Building Demand Response as a Markov Decision Process

Traditional RL is based on the Markov decision process (MDP) composed of states, actions, and transition probabilities. In an MDP, any future state is only dependent on the current state and action. The agent's goal in MDPs is to find a policy that maps actions from states such that an objective is maximized. In the context of the building demand response control, the basic elements of the MDP are defined as follows.

4.1.1. States

The states (s_t) are the observations available to the agent obtained through interactions with the environment. At every 15-min time step, a set of observations is measured and provided to the agent, which includes the current zone temperature T_z^t, as well as the past six-step zone temperatures

$T_z^{t-1}, ..., T_z^{t-6}$; the reason of including the previous zone temperature measurements is discussed in Section 4.3. The observation set also contains the hour of the day h^t, the measured power P^t required to achieve the current temperature set-point, and the 24-hour forecasts of the outdoor temperature $T_{out}^t, ..., T_{out}^{t+95}$ as well as the global horizontal solar radiation $q_{sol}^t, ..., q_{sol}^{t+95}$.

$$s_t = \left\{T_z^t, \dots, T_z^{t-6}, h^t, T_{out}^t, \dots, T_{out}^{t+95}, q_{sol}^t, \dots, q_{sol}^{t+95}, P^t\right\}.$$

4.1.2. Actions

The actions (a_t) are the control inputs that the agent takes given a state s_t. The action taken by the agent is the zone temperature set point T_{sp}^{t+1} for the next decision step

$$a_t = T_{sp}^{t+1}.$$

The action space is a discrete space with p values equally spaced in a predetermined temperature range:

$$A = \{\underbrace{T_{z,LB}, \dots, T_{z,UB}}_{\text{p available actions}}\}.$$

4.1.3. Rewards

The reward (r_t) is a signal that measures how good the state-action pair is. The goal of the demand response RL agent is to maximize the reward (negative of the cost) under a time-of-use tariff while maintaining the zone temperature within the comfort bounds. The reward associated with the energy cost is defined as:

$$r_{cost}^t = -u_t P^t, \tag{8}$$

. Conventional (unconstrained) RL controllers cannot explicitly handle control constraints. In almost all previous building control applications, the comfort constraints were addressed as a soft penalty cost added to the energy cost with a prescribed weighting factor. In this study, the reward associated with the comfort violation penalty is considered in the following form:

$$r^t_{com} = -\psi\big(\max(T^t_{com,lb} - T^t_z, 0) + \max(T^t_z - T^t_{com,ub}, 0)\big), \tag{9}$$

where ψ is the constraint violation penalty factor and $T^t_{com,lb}$ and $T^t_{com,ub}$ are the upper and lower zone temperature comfort bounds. Comfort violation is defined as the cumulative zone temperature excursion outside the defined lower and upper comfort temperature bounds. The occupants' thermal comfort perception is not a focus of this study. The comfort penalty is proportional to the cumulative temperature excursions out of the comfort band. The overall reward for the unconstrained RL case is $r_t = r^t_{cost} + r^t_{com}$.

4.2. Reinforcement Learning Basic Concepts

This section briefly describes the key concepts for RL algorithms while details thereof can be found in [35].

4.2.1. Policy

A policy represents a map that takes the observed states as inputs and outputs a set of probabilities if the action space is discrete, or the mean and standard deviation pair if the action space is continuous. In this study, the policy is formulated as:

$$\pi(s_t, a_t) = P(a_t|s_t), \tag{10}$$

where $P(a_t|s_t)$ is the probability of taking action a_t given a state s_t.

4.2.2. Return

A trajectory of an episode consists of the collected states, actions, and rewards under the current policy:

$$\tau = \{s_0, a_0, r_0, s_1, a_1, r_1 . . . s_T, a_T, r_T\}, \tag{11}$$

where T is the number of time steps in the trajectory. The return of a trajectory is defined as the discounted cumulative reward of the episode:

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t, \tag{12}$$

where γ is the discount factor which determines the importance of future actions in decision making of the current step.

4.2.3. Value Function

The value function of a state is defined as the expected discounted cumulative reward collected during an episode τ following a policy π given a state s_t:

$$V^{\pi}(s_t) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{l=0}^{T} \gamma^k r_{t+l+1} | s_t\right]. \tag{13}$$

4.2.4. Action Value Function

The action value function of a state and action is defined as the expected discounted cumulative reward collected for an episode following a policy π given a state and the action taken in that state:

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{l=0}^{T} \gamma^k r_{t+l+1} | s_t, a_t\right]. \tag{14}$$

Parameterized functions are used to represent the state value functions, action value functions, and policies to solve the problem of dramatically increased computation burden that is commonly seen in tabular RL techniques for control applications:

$$V^{\pi}(s_t) \approx V^{\pi}_{\phi}(s_t), \tag{15}$$

$$Q^{\pi}(s_t, a_t) \approx Q^{\pi}_{\phi_q}(s_t, a_t), \tag{16}$$

$$\pi(s_t, a_t) \approx \pi_{\theta}(s_t, a_t), \tag{17}$$

where ϕ, ϕ_q and θ are the parameters of the respective functions. While several types of parameterization can be used to represent these functions, this study considers deep neural networks and linear maps.

4.2.5. Advantage Function

An advantage function $A^{\pi}(s_t, a_t)$ quantifies the benefit of taking a specific action a_t over choosing the action according to the policy $\pi(\cdot|s_t)$:

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t). \tag{18}$$

A positive advantage $A^{\pi}(s_t, a_t) > 0$ indicates that taking an action a_t while being in state s_t is better than the expected action dictated by policy π. Hence, the advantage function guides the learning process to encourage actions that lead to improved overall performance.

4.3. Partial Observability of the Environment States

In an ideal case, the states of a dynamic system are all measurable during interactions with the environment where the control problem can be formulated as an MDP. In reality, however, not all states are available to the agent, e.g., wall internal temperatures which characterize building thermal dynamics, and only one or a few of the states are observable, e.g., zone temperatures. Control of such systems can be addressed as a partially observable Markov decision process (POMDP) [36] or as a regular MDP by including historical measurements of the observable outputs and inputs in the state vector. The latter is analogous to the state-space realization of a high-order linear system using the high-order time derivatives of the inputs and outputs as the state variables. This MDP formulation has been used for RL implementations for building HVAC controls [18] and is also adopted in this study.

4.4. Policy Gradient Methods

Value-based methods such as deep Q-learning showed remarkable performance in solving popular games such as Atari [37] and has also been studied for HVAC system controls. These methods learn the value function and derive a greedy policy that chooses the action with the highest action value. In recent years, policy gradient methods have shown the capability of solving complex problems such as robotic controls [38]. These methods directly learn the policy instead of using the action value function which affords better performance. For a discrete action space, the policy outputs the probability of taking each action given a state.

$$\pi_\theta(s_t, a_t) = P(a_t|s_t). \tag{19}$$

Then policy gradient methods seek to identify a policy that maximizes the return as:

$$\max_\theta \quad J(\pi_\theta) = \quad \mathbb{E}_{\tau\sim\pi_\theta}\Big[R(\tau)\Big], \tag{20}$$

which can be solved by a classic gradient descent method:

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta J(\pi_{\theta_k}), \tag{21}$$

where α is the learning rate and k is the update step. $\nabla_{\theta_k} J(\theta_k)$ has been shown to be in the following form:

$$\nabla_\theta J(\pi_{\theta_k}) = \mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k}\Phi_t\right], \tag{22}$$

where Φ_t can be chosen between the return $R(\tau)$, the action value function $Q^\pi(s_t, a_t)$, the reward-to-go defined as $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$, or the advantage $A^\pi(s_t, a_t)$. Expectations are estimated by a sample mean over trajectories and time steps in this study. If the agent collects N trajectories of T steps each, the expected value of the objective function in Eq. (22) can be estimated as follows:

$$\frac{1}{NT}\sum_{N}\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k}\Phi_t. \tag{23}$$

4.4.1. Generalized Advantage Estimation

The large number of samples required and unstable learning become a challenge for policy gradient methods. Ref. [39] recognized that a correct estimation of the advantages is the key to reducing variance, and proposed to use general advantage estimation (GAE), a method to reduce the variance of the estimated policy gradient at the cost of some bias. The advantages are estimated using GAE as follows:

$$\hat{A}_t^{\pi,GAE} = \sum_{l=0}^{T}(\gamma\lambda_{GAE})^l \delta_{t+l}^{V^\pi}, \tag{24}$$

$$\delta_{t+l}^{V^\pi} = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t), \tag{25}$$

where $\delta_{t+l}^{V^\pi}$ is the discounted temporal difference (TD) residual which represents the difference between predicted and actual value function $V^\pi(s_t)$. The variance and bias trade-off is controlled by two parameters λ_{GAE} and

γ. Most of the studies that implement a policy optimization algorithm used GAE to estimate the advantages. However, the ideal λ_{GAE} value for different applications may differ. The present study found that a value of $\lambda_{GAE} = 0.52$ would provide a reasonable estimation of the advantages for building control applications.

4.4.2. Trust Region Policy Optimization (TRPO)

One of the simplest policy optimization algorithms is vanilla policy gradient (VPG). This algorithm takes m trajectories with the current policy and uses them to estimate the advantages and policy gradient. Then it uses gradient descent steps to update the parameterized policy and update the value function parameters at the end of each step using the mean squared error (MSE) [40]. While this method is straightforward to implement, it suffers from a proclivity to fall in local optimum and has poor sample efficiency and high variance. To address this issue, TRPO [41] seeks to maximize a surrogate objective function subject to a constraint that limits the size of a policy update during each iteration measured by the KL divergence:

$$\max_{\theta} \quad \mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}\hat{A}^{\pi_{\theta_k}}(s,a)\right] \tag{26}$$

$$\text{s.t.} \quad \mathbb{E}_{\tau\sim\pi_\theta}\left[D_{KL}(\pi_\theta(\cdot|s)||\pi_{\theta_k}(\cdot|s))\right] \leq \delta, \tag{27}$$

where $\hat{A}^{\pi_{\theta_k}}$ is the advantage that can be calculated using the GAE method [39] and δ is the upper bound imposed on the KL divergence between two consecutive policy updates. This problem is usually simplified using a linear approximation of the objective function and a quadratic approximation of the KL divergence constraint. TRPO starts collecting trajectories under the current policy π_{θ_k}. With these trajectories, it estimates the rewards-to-go, advantages and value function for the current state, based on which the gradient is calculated and the policy parameters are updated. Finally, it fits the value function parameters using the rewards-to-go as target values for training the value function network, minimizing the mean squared error (MSE) between the predicted values and the actual rewards to go. This step ensures that the parameterized value function network accurately estimates the rewards to go for each state.While TRPO overcomes some of the challenges of VPG algorithms, there are also some drawbacks with TRPO such as high computational burden. While this can be improved by using approximations or iterative methods, errors may be introduced.

4.4.3. Proximal Policy Optimization (PPO)

PPO, the second tested policy optimization technique, seeks to solve the same problem presented in the TRPO case in a simpler manner. Two versions of PPO algorithms were proposed in the original work [42], while PPO-clip is more broadly used. PPO-clip uses a clip function to limit the incentive of policy change to stabilize training as follows:

$$\max \quad \mathbb{E}_{\tau \sim \pi_{\theta_k}} L(s, a, \theta_k, \theta), \tag{28}$$

where

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \hat{A}^{\pi_{\theta_k}}(s, a), h\left(\epsilon, \hat{A}^{\pi_{\theta_k}}(s, a)\right) \right), \tag{29}$$

and the clip-range ϵ is a hyper-parameter that determines how far the new policy candidate is allowed to deviate from the old one and the function h is defined as:

$$h\left(\epsilon, \hat{A}^{\pi_{\theta_k}}\right) = \begin{cases} (1+\epsilon)\hat{A}^{\pi_{\theta_k}}, & \hat{A}^{\pi_{\theta_k}} > 0 \\ (1-\epsilon)\hat{A}^{\pi_{\theta_k}}, & \hat{A}^{\pi_{\theta_k}} < 0. \end{cases} \tag{30}$$

PPO starts in a similar manner as TRPO, collecting trajectories under the current policy and estimating the rewards-to-go, advantages and state value function. Note that both TRPO and PPO can use a parameterized network to approximate the value function $V_\beta(s)$ and compute advantages at each timestep. Next, it uses Eq. (28) to calculate the policy network parameters update and lastly it fits the value function similarly with TRPO. PPO is an effective policy gradient method and avoids some of the drawbacks of TRPO since it does not rely on approximations or computationally expensive procedures. However, the performance is sensitive to the choice of the clip-range hyper-parameter ϵ. An improper setting of this hyper-parameter can result in unstable training and the policy can get stuck in local optimum quickly for high clip ranges.

5. Constrained Reinforcement Learning Controller

This section presents the proposed constrained RL controller for building demand response. It is built on top of traditional RL techniques and the major differences are highlighted.

5.1. Constrained Markov Decision Process (CMDP)

A CMDP is an extension of an MDP with constraints added to the formulation. These constraints limit the set of feasible policies for the MDP. A CMDP is defined as a set of states, actions, transition probabilities and cost functions. The cost function is the only additional element compared to unconstrained MDP, which measures the level of constraint violations.

5.2. Cost

The cost for a time step (c_t) quantifies the violations of the constraints that the system must satisfy. In the context of the building HVAC system control, it is defined in the same manner as the comfort reward r_{com} for the unconstrained scenario:

$$c_t = -\phi\big(\max(T^t_{com,lb} - T^t_z, 0) + \max(T^t_z - T^t_{com,ub}, 0)\big). \tag{31}$$

5.3. Constrained Reinforcement Learning Key Concepts

A CMDP extends an MDP by introducing several additional elements, all associated with the cost function for constraint violations.

5.3.1. Cost Return

The cost return of a trajectory is defined as the discounted cumulative constraint violation for the episode:

$$C(\tau) = \sum_{t=0}^{T} \gamma^t c_t. \tag{32}$$

5.3.2. Cost Value Function

The cost value function of a state is defined as the expected discounted cumulative cost during an episode τ following a policy π given a state s_t.

$$V^\pi_C(s_t) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{l=0}^{T} \gamma^k c_{t+l+1} | s_t\right]. \tag{33}$$

5.3.3. Cost Action Value Function

The cost action value function of a state is defined as the expected discounted cumulative cost during an episode τ following a policy π given a state s_t and action a_t.

$$Q^\pi_C(s_t, a_t) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{l=0}^{T} \gamma^k c_{t+l+1} | s_t, a_t\right]. \tag{34}$$

These functions are also parameterized as:

$$V_C^\pi(s_t) \approx V_{C,\omega}^\pi(s_t), \tag{35}$$

$$Q_C^\pi(s_t, a_t) \approx Q_{C,\omega_q}^\pi(s_t, a_t), \tag{36}$$

where ω and ω_q are the parameters of the value function and action value function, respectively.

5.3.4. Cost Advantage Function

The cost advantage function is defined in a similar way as the reward advantage function:

$$A_C^\pi(s_t, a_t) = Q_C^\pi(s_t, a_t) - V_C^\pi(s_t). \tag{37}$$

5.4. Constrained Policy Optimization

An inherent challenge of non-constrained RL algorithms (e.g., TPRO and PPO) is the inability to address constraints, while for most control applications, operational constraints exist to ensure safe operation and quality services. Constrained RL seeks to maximize the original merit function but restricts the set of feasible policies so that a discounted constraint violation return is upper-bounded by a pre-set threshold d [43]. Among the proposed algorithms in the literature, constrained policy optimization (CPO) [44] has been widely studied, which seeks to solve the optimization problem:

$$\max_\theta \quad \mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}\hat{A}^{\pi_{\theta_k}}(s,a)\right] \tag{38}$$

$$\text{s.t.} \quad J_C(\pi_{\theta_k}) + \frac{1}{1-\gamma}\mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\hat{A}_C^{\pi_{\theta_k}}(s,a)\right] \leq d \tag{39}$$

$$\mathbb{E}_{\tau\sim\pi_\theta}\left[D_{KL}(\pi_{\theta_k}||\pi_{\theta_k})\right] \leq \delta. \tag{40}$$

The constraint advantage and return are defined in a similar manner to those of the reward function. A positive slack variable d is introduced to stabilize training. CPO uses linear approximations of the objective shown in Eq. (38). The first constraint shown in Eq. (39) and the KL divergence constraint shown in Eq. (40) are also linearized when solving the optimization problem.

CPO starts collecting trajectories using the currrent policy. Then it estimates the gradients for the reward and constraint functions as well as the total costs $J_C(\pi_k)$. When the problem is feasible it solves the linearized version of the constrained optimization problem shown in Eq. (38) to Eq. (39). Due to approximation errors, the approximate solution may not be feasible for the original problem; for such cases, a different optimization problem that only reduces the constraint violation is used to obtain the update step. Lastly it performs backtracking line search to ensure the new parameters satisfy the constraints and result in a positive policy improvement. CPO uses two parameterized value networks, for the constraint and reward value functions, respectively. As a final step it fits the reward and constraint value functions $V_\phi^{\pi_k}$ and $V_{C,\omega}^{\pi_k}$ using estimations of these value functions collected during the update steps.

6. Case Study Results

Simulation tests were conducted to assess the performance of the constrained RL algorithm in comparison with the different baseline strategies. This section first introduces the key features of the case study building and the operation requirements which include the comfort bounds and demand response pricing structure. Secondly, it shows the results obtained with the unconstrained RL algorithms followed by the results obtained by the constrained RL counterpart. A parametric study of the penalty factor in the unconstrained RL controller and the constraint violation upper bound in the constrained RL case is also presented to illustrate the sensitivity of the algorithms to these parameters.

6.1. Simulation Case Study Description

A simulation testbed developed for a single-zone office space with a total floor area of approximately 100 m^2 is used for performance assessment of the control strategies. The considered zone is part of a larger building with four identical office spaces all serving graduate students. Each zone is equipped with a dedicated AHU-VAV system. Cross-zone thermal coupling is negligible due to the high-grade insulation used in the separating walls. The office space is attached to a south-facing double façade with a gap of 1 m. The simulation testbed incorporates a thermal network model calibrated with field data [32]. TOU retail energy rates were used in the simulations obtained from El Paso Electric Co. [45] and are shown in Table 1. A lower energy rate of \$0.07/kWh is involved during non-peak hours while electricity is charged at a much higher rate of \$0.22/kWh during on-peak hours. The zone temperature bounds change with the occupancy of the building, with 9AM to 6PM being the occupied period and the rest of the day being unoccupied. During occupied hours the upper and lower temperature bounds are 21.5°C and 23.5°C, while during unoccupied hours are 20.5°C and 24.5°C, respectively.

6.2. MPC Baselines Results

The MPC baselines used a look-ahead horizon of 24 hours with a decision implemented for every 15-min time step. The MPC baselines were formu-

Table 1: Summer time of use tariff

Electricity price ($/kWh)	**Hours**
0.222	12:00 to 18:00
0.077	Rest of day

lated as a convex program using the CVX package in MATLAB [46] and solved using Gurobi [47]. The MPC strategies serve as benchmarks, assuming identical control and plant models. Therefore, the test results represent the theoretical optimal performance that can possibly be achieved. Table 2

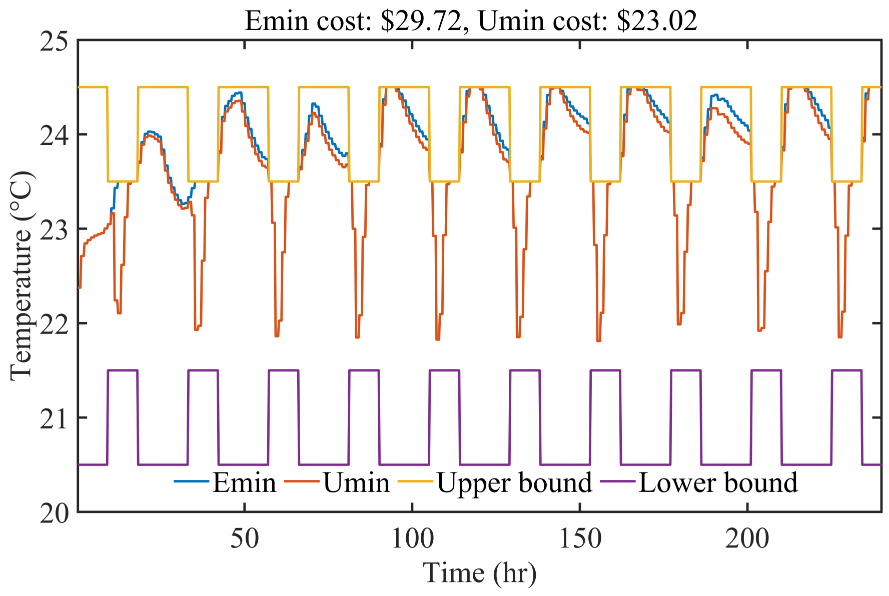

Figure 2: Zone temperature under MPC baselines.

shows the cost break-downs for the MPC, unconstrained and constrained RL strategies. Fig. 2 and Fig. 3 present the simulation results of the two MPC baseline strategies. The energy minimization baseline (E-min) maintains the

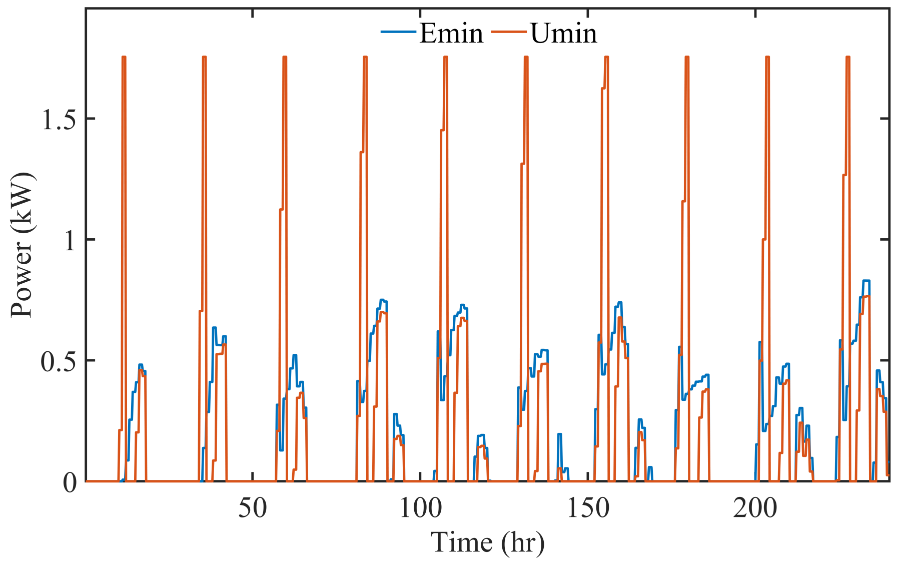

Figure 3: Power profile under MPC baselines.

zone temperature at the upper bound when mechanical cooling is called for, resulting in minimum energy usage. During unoccupied hours the temperature floats and the HVAC system remains off. The utility-cost-minimizing (U-min) strategy engages a pre-cooling action before each on-peak period. The pre-cooling action maintains a lower zone temperature prior to on-peak hours so that "cooling" energy is stored in the building thermal mass; during on-peak hours, the zone temperature is adjusted upwards to allow the stored cooling energy to be released, resulting in shifting of building electricity use to low-cost hours to reduce the overall utility cost. While this strategy provides utility cost savings, it increases the total energy used by the HVAC system. Compared to strategy E-min, the cost-minimizing MPC strategy achieves cost savings of 22.5%, with a total energy rebound of 4.6%. This represents the best economic performance that can possibly be obtained under the same prediction horizon setting without any comfort constraint violation. Actual control performance would be worse due to potential control-plant model mismatches.

6.3. Unconstrained RL Baseline Results

The RL strategies were trained using two years of simulation data following an on-policy scheme. The baseline, unconstrained RL simulations were conducted using the Stable Baselines 3 OpenAI library [48]. The algorithms in this library were customized for the purposes of this study.

6.3.1. Simulation Setup

Episodic training was utilized with each episode or trajectory consisting of 1 days (96 steps). TRPO and PPO with different constraint violation penalty factors, i.e. , $\psi = 0.01$, $\psi = 0.1$, $\psi = 1$ and $\psi = 10$, were chosen as RL baselines to illustrate the effect of the weighting factor on the control performance.

Policy Network: Policy networks use the observation set as input, 2 hidden layers of 64 neurons and an output layer with 17 possible action logits. After each layer a hyperbolic tangent (tanh) activation function is used. A final softmax layer is used to predict the probability of choosing an action given an observation set.

Value Function Network: Value function networks use the observation set as input, 2 hidden layers of 64 neurons, and output the estimated value function. After each layer, a hyperbolic tangent (tanh) activation function is used.

6.3.2. Simulation Results

Fig. 4 to Fig. 7 present the simulation test results under the two unconstrained RL strategies subject to different constraint violation penalty

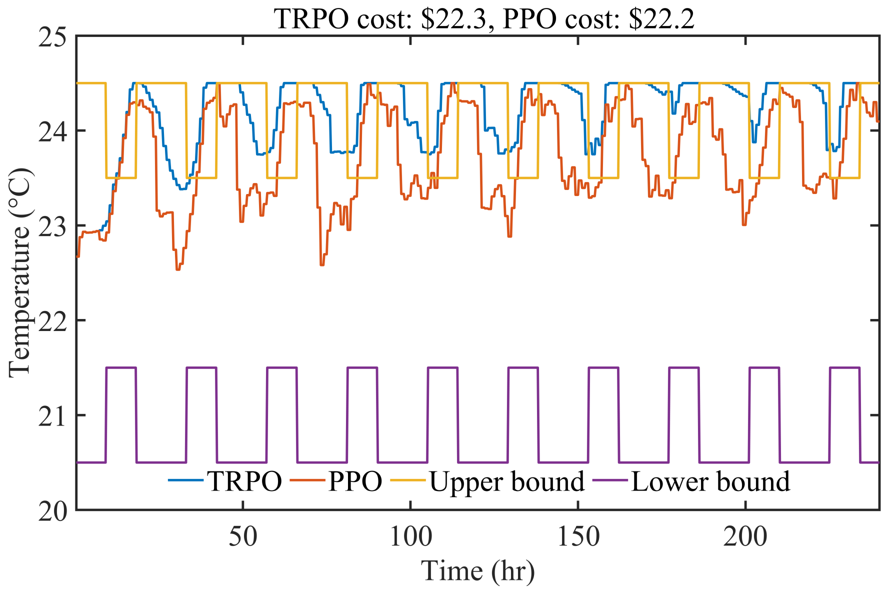

Figure 4: Zone temperature for unconstrained RL strategies with $\psi = 0.01$.

factors. As expected, the unconstrained RL strategy with the lowest comfort penalty factors leads to the lowest energy cost, with cost savings up to 26.2% relative to E-min, but at the expense of significant comfort issues. The RL strategies with high comfort penalty factors are able to regulate the zone temperature within the comfort zone but lead to high HVAC energy costs (savings of only less than 3% relative to E-min). The high-comfort-penalty RL strategies engage very mild pre-cooling actions for load shifting for most cases, which is the major cause of the high energy cost as can be seen in Table 2. It is evident from these results that performance achievable with the unconstrained RL strategies is highly dependent on the comfort penalty factor setting. This also makes the comparison with the MPC benchmarks challenging.

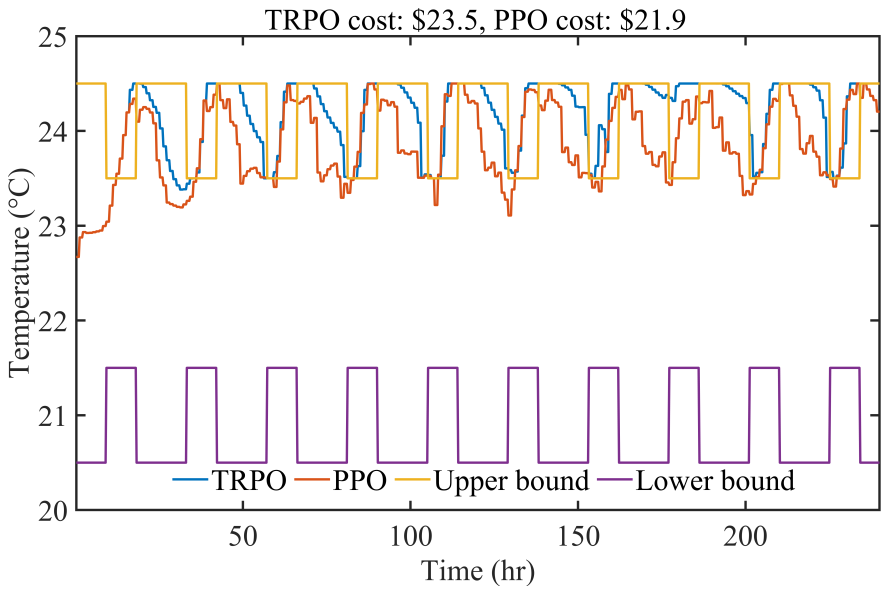

Figure 5: Zone temperature for unconstrained RL strategies with $\psi = 0.1$.

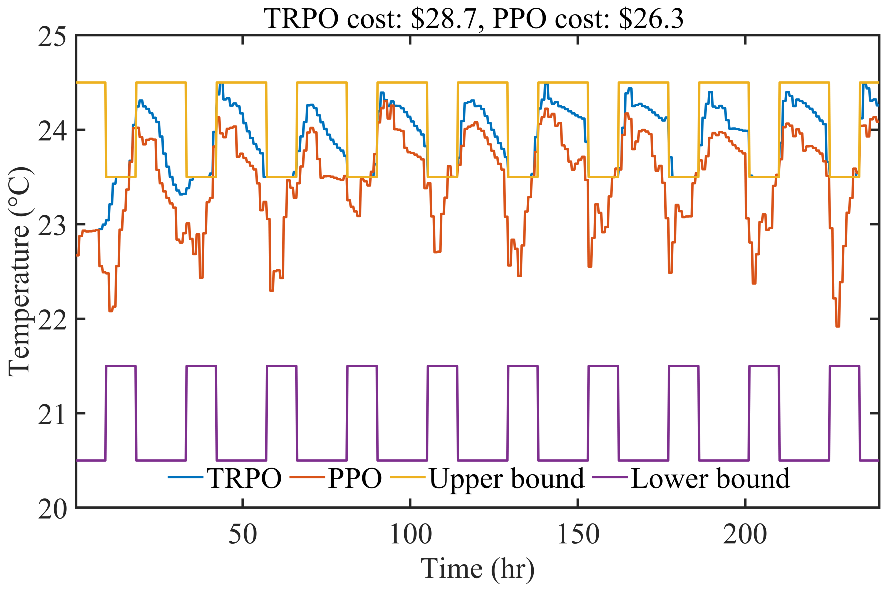

Figure 6: Zone temperature for unconstrained RL strategies with $\psi = 1$.

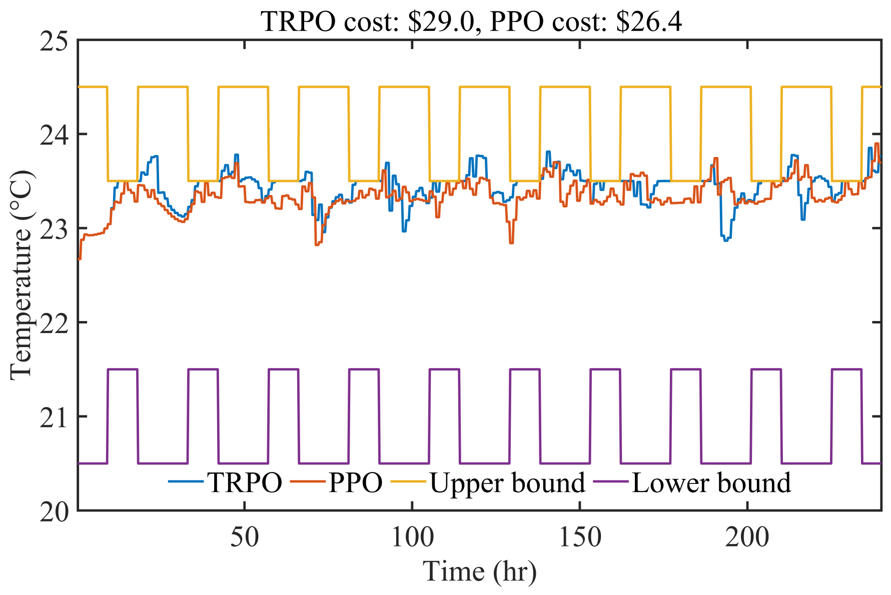

Figure 7: Zone temperature for unconstrained RL strategies with $\psi = 10$.

6.3.3. Parametric Study of Penalty Factor

The simulations reported in the previous subsection consider penalty factors that are one order of magnitude apart. A high-granularity analysis of the effect of the penalty factor on performance is needed to find the optimal setting. Table 2 shows that better balances between cost and constraint violations may be found with penalties between 0.1 and 1. For this purpose, ten penalty factors equally spaced between $\psi = 0.1$ and $\psi = 1$ were simulated and the average performance over 10 random seeds was evaluated.

Fig. 8 shows the performance of the PPO algorithm under different penalty factors. The cost performance of CPO is shown in yellow dashed line and the comfort violation performance is shown in green dashed line, as benchmarks. Detailed analysis of the CPO algorithm performance will be covered in Section 6.4. The red line shows the comfort violation average while the blue line shows the average electricity use cost for the PPO cases. It can be seen that unconstrained RL with penalty factors of $\psi \leq 0.4$ outperforms the CPO strategy in terms of cost savings. However, these cases have a comfort violation of at least 30 °C-step which is approximately 15 times the violations of the CPO case. On the other hand, none of these cases could outperform the CPO control strategy in terms of constraint satisfaction. As it will be shown in the next section, constrained RL with linear mapping performed better than any of the PPO cases presented in this subsection. These results confirm that CPO-based strategies can successfully enforce constraints subject to a minor tolerance, without any need for fine-tuning of a penalty factor.

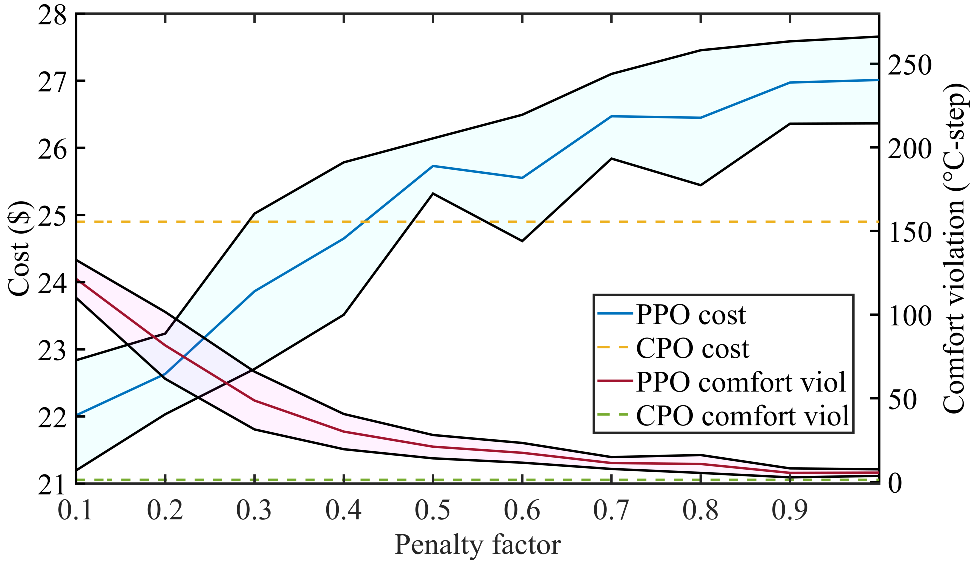

Figure 8: PPO performance under different penalty factors.

6.4. Constrained RL Results

The constrained RL strategy was trained using two years of simulation data following an on-policy scheme. The CPO algorithm was implemented using PyTorch [49].

6.4.1. Simulation Setup

Episodic training was utilized with each episode or trajectory consisting of 1 days (96 steps). The algorithm was trained on 10 different random seeds to assess the average performance. The policy and value networks used in the unconstrained and constrained case share the same architecture. In the constrained case, an additional value function network is used for the cost value function.

6.4.2. Simulation Results

Fig. 9 shows the average learning curve for the constrained RL algorithm, while the average constraint function value is shown in Fig. 10. All learning curves presented, are 60-episode average curves. A slight decrease in reward value and high variances can be seen after episode 500. The use of complex neural networks may have caused this performance degradation as more data is used for training. Another reason for this defect can be attributed to approximation errors in the policy optimization where the obtained approximate solution may be infeasible for the original problem. When the infeasibility issue happens, the update will solely seek to reduce constraint violation, causing performance degradation. To address the performance degradation issue, linear policy and value functions were proposed and evaluated with major results presented in Section 6.5. Two representative control solutions out of those associated with the 10 different seeds are discussed below, corresponding to the best- and worst-performing strategies. Fig. 11 and Fig. 12 show the hourly control behaviors for one of the best-performing seeds. Deep pre-cooling actions can be seen under this scenario, resulting in potential savings of 20.25% and minimal constraint violations. On the other hand, a less-aggressive (worst-performing) strategy shown in Fig. 13 and Fig. 14 results in limited pre-cooling with potential savings of 11.13% and slightly higher constraint violation. However, the violations are still considerably lower than those seen in the unconstrained RL case. The total average cost and constraint violations are shown in Table 2. The potential average savings are 12.4% with an average constraint violation of 4.24 °C-step. For all seed averages, it can be seen in Fig. 10 that the controllers succeed in enforcing the constraint satisfaction within the set limit. Note that the constraint violations can be further reduced by lowering the slack (d) setting, but a positive slack is needed for stable training. These results clearly demonstrate the superior performance of constrained RL over the unconstrained counterparts for predictive control of building flexible loads while ensuring indoor comfort.

6.5. CPO with Linear Policy Maps

While in the literature, neural networks are widely used to parameterize the policies of RL agents, some studies have found that for many control applications, linear policy maps show more robust performance, especially for systems with dynamics close to linear in the range of operation [50]. In addition, policy and value functions with linear maps are cheap to evaluate and thus, function approximation is not needed, which is the main cause of the

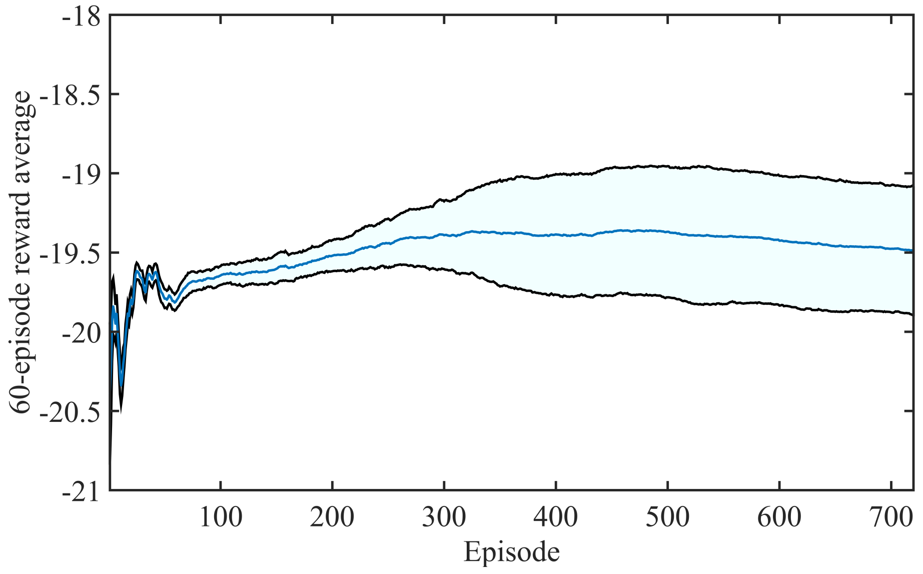

Figure 9: CPO learning curve for rewards.

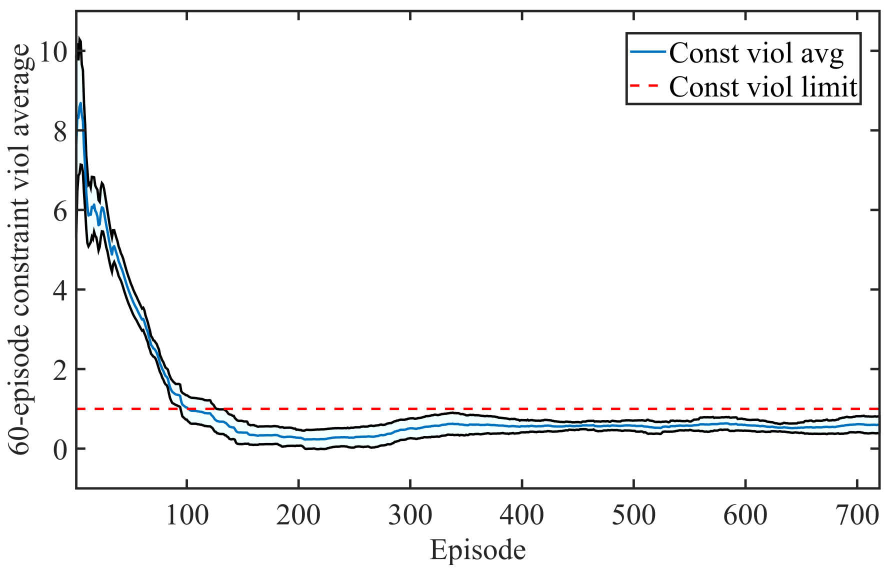

Figure 10: CPO learning curve for constraint violations.

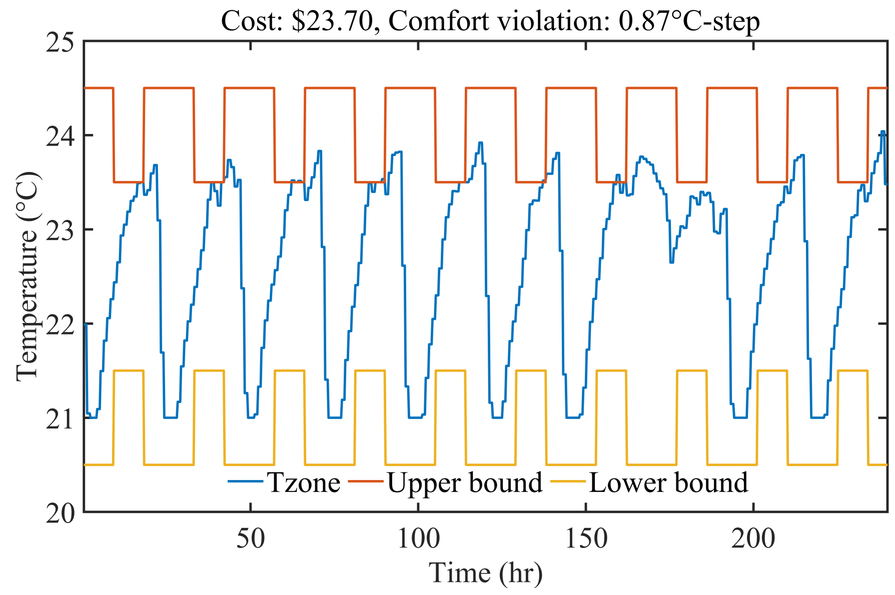

Figure 11: Zone temperature for best CPO control strategy.

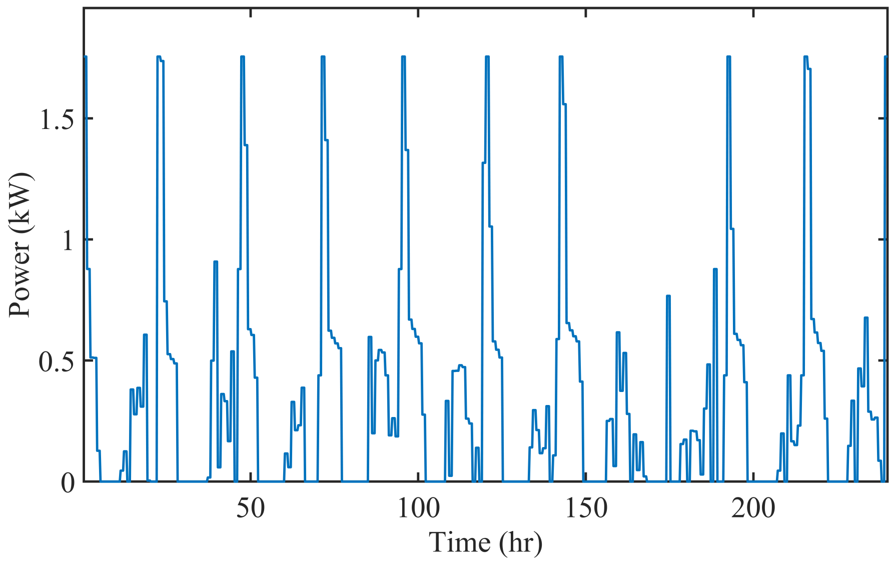

Figure 12: Power profile for best CPO control strategy.

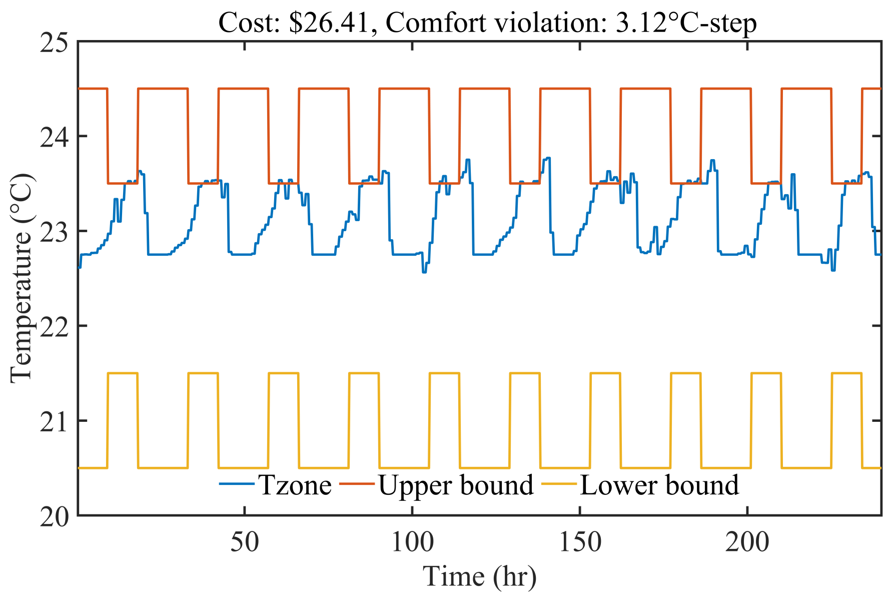

Figure 13: Zone temperature for worst CPO control strategy.

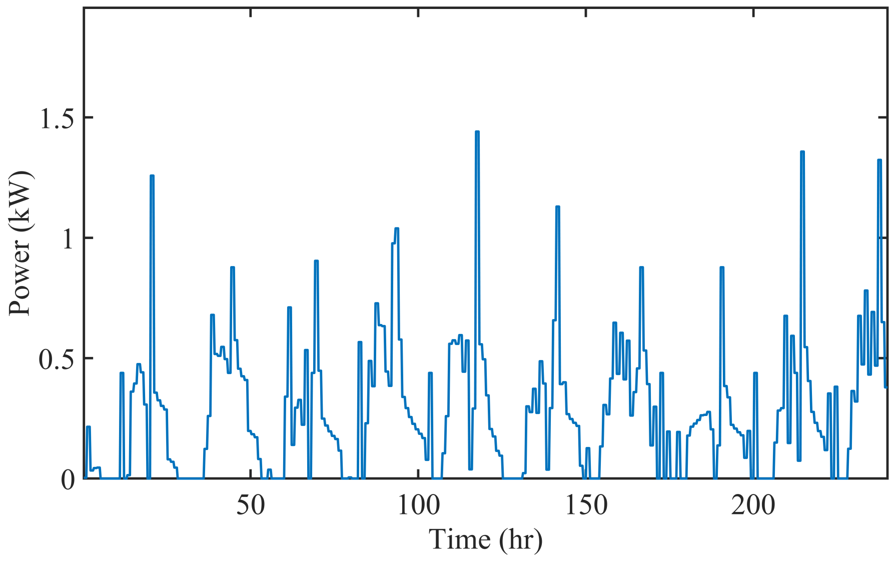

Figure 14: Power profile for worst CPO control strategy.

performance degradation seen in Fig. 9 starting at episode 300. This section seeks to empirically test the performance of the CPO algorithm using linear maps. The policy, value function, and cost value function are represented by linear maps, hence activation functions are not needed.

6.5.1. Simulation Results

Fig. 15 shows the average learning curve for the CPO with linear maps, while the constraint function value is shown in Fig. 16. It can be seen that CPO with linear mapping yields more stable training and much reduced variance. The performance also improves monotonically as more training data is used. Fig. 17 and Fig. 18 show one of the best-performing control results. This strategy yields potential savings of 22.7% which are slightly higher than the utility minimization MPC strategy; this is caused by minor constraint violations which have an accumulative value of 2 °C-step during the whole month. The COP with linear mappings results in a much better average performance with average potential cost savings of 16.1% (30% better than the neural network-based CPO strategy), and constraint violation of only 1.3 °C-step in 30 days (75% reduction compared to the neural network-based CPO).

6.6. Parametric Study of Constraint Upper Limit

In theory, if the slack term is set to zero and exact solutions are sought, then the control strategy obtained should strictly enforce constraints. However, since the CPO implementation solves an approximate instead of the original policy optimization problem, the obtained solution may not be feasible for the original problem, leading to possible constraint violations. When the solution is infeasible, the control tends to minimize the constraint viola-

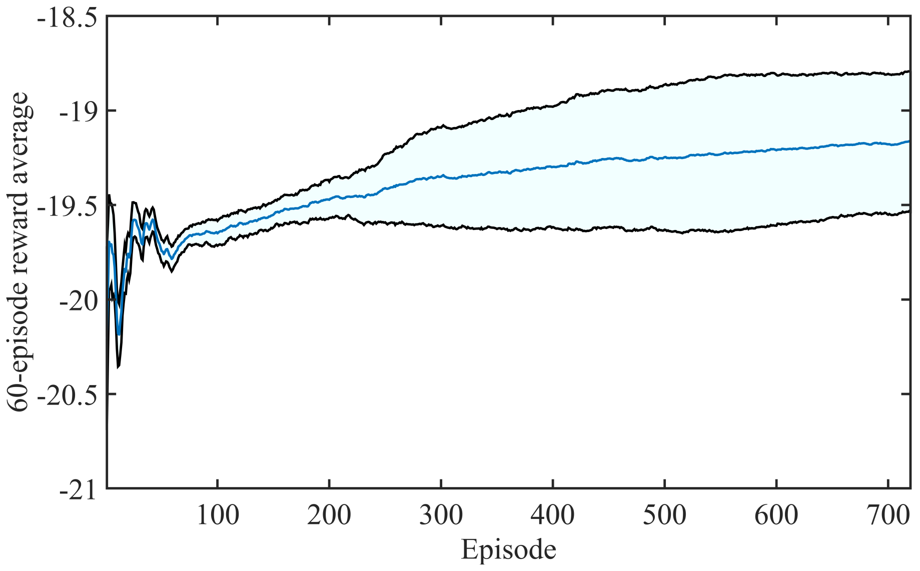

Figure 15: CPO algorithm learning curve with linear mapping.

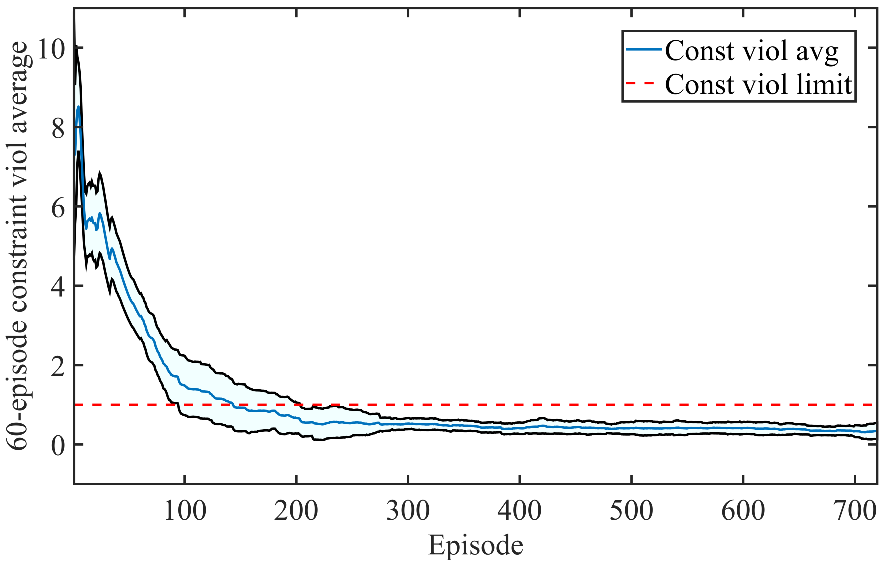

Figure 16: CPO algorithm constraint violation learning curve with linear mapping.

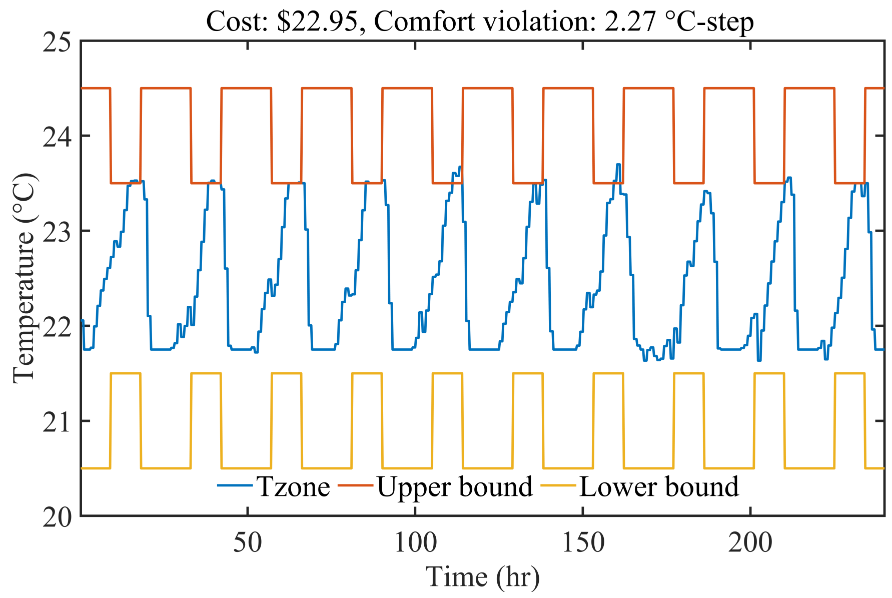

Figure 17: Zone temperature for best CPO control strategy with linear mapping.

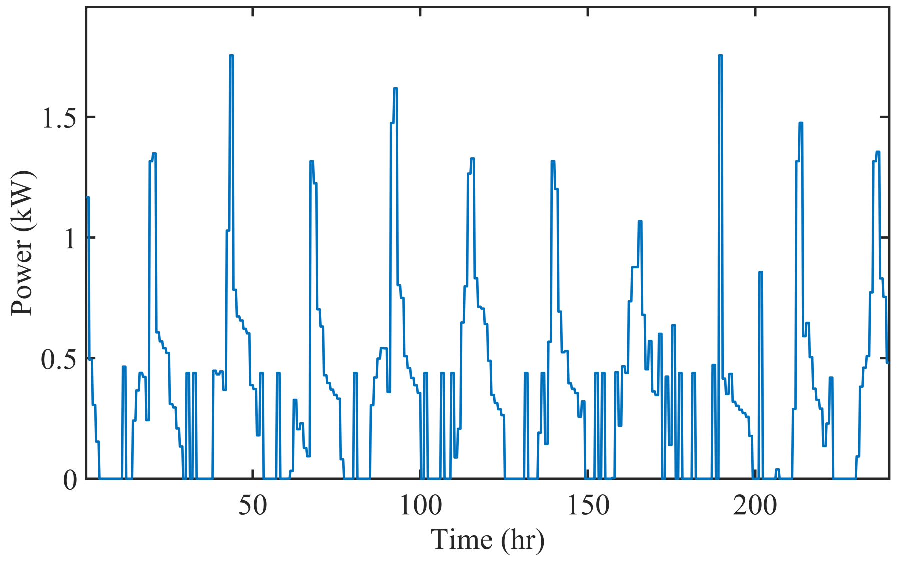

Figure 18: Power for best CPO control strategy with linear mapping.

tions instead of the actual reward, which may result in sub-optimal performance cost-wise. To evaluate the performance of the constrained RL algorithm under different constraint violation slack settings, a parametric study was carried out with the slack parameter d assuming different values from 0 to 5 and under 10 different random seeds.

Fig. 19 shows the cost and constraint violation averages under different constraint violation slack settings. When this setting approaches zero, the training becomes unstable since the agent is only seeking to reduce the constraint violation without any improvement in the cost. While a slack term of zero results in unstable training, small values such as $d = 0.1$ proved to result in performances that are comparable to the case study performed in Section 6.5. It can also be seen that with the highest value tested, i.e. $d = 5$, the controller still outperforms the unconstrained RL strategy with much lower constraint violations while achieving similar electricity usage costs. However, the slack term should remain low since increasing it will allow more constraint violations. In conclusion, the algorithm only requires the slack parameter to be slightly higher than zero to perform well, and even higher settings that allow for more constraint violations produce strategies outperforming the unconstrained cases.

7. Conclusions

This work presented a constrained RL-based control strategy for demand responsive control of building thermal loads. The performance of the proposed strategy was assessed through comparisons with unconstrained RL baselines with different comfort penalty factors and with MPC benchmarks. The constrained RL strategy with linear mapping showed much improved

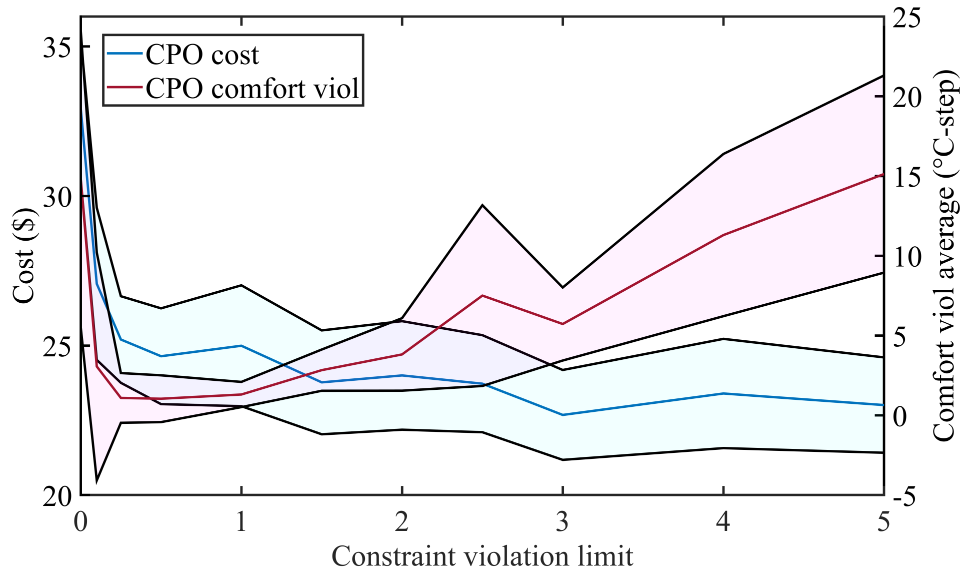

Figure 19: CPO performance under different constraint violation limits.

performance with more stable training and lower variance compared to neural network-based constrained RL. The test results show that the constrained RL strategy with linear mapping was able to reduce the electricity cost by 16.1%, relative to the energy-minimizing MPC controller, with very minor temperature excursions out of the comfort zone, while the unconstrained RL strategies led to either high energy costs or significant constraint violations, depending on the comfort penalty settings. The constrained RL controller achieved utility cost reduction close to 71% of that achieved with the MPC utility minimization control benchmark, demonstrating its superior performance over unconstrained RL techniques for building demand response.

Two major observations can be made about the implementation of the proposed algorithm. First, selection of appropriate functional forms for the value and policy functions is crucial to achieve satisfactory performance and is

Table 2: Cost and Power Break-down

	Energy use (kWh)	**Energy cost ($)**	**Cost savings (%)**	**Comfort violation (°C-step)**
Emin	184.1	29.7	-	-
Umin	192.7	23.0	22.5	-
$\mathbf{TRPO_{0.01}}$	168.2	23.2	21.8	238
$\mathbf{TRPO_{0.1}}$	168.60	23.1	22.2	231.2
$\mathbf{TRPO_{1}}$	186.6	28.6	3.7	3
$\mathbf{TRPO_{10}}$	193	28.8	3	0.1
$\mathbf{PPO_{0.01}}$	185.2	21.9	26.2	155.4
$\mathbf{PPO_{0.1}}$	190.2	22.0	25.9	121.4
$\mathbf{PPO_{1}}$	204.9	27.0	9.0	5.5
$\mathbf{PPO_{10}}$	217	27.2	8.4	0.36
CPO	220.2	26.0	12.3	4.2
CPOLinear	240.1	24.9	16.1	1.3

highly dependent on the specific applications. It was shown that using linear mapping instead of deep neural networks for the policy and value functions could yield better results in terms of electricity cost and total constraint violations – the strategy with linear mapping achieved 130% of the cost savings obtained with the neural network-based strategy while reducing constraint violations by 75%. Secondly, the CPO algorithm was shown to be sensitive to the violation slack d settings, especially when it approaches zero. This is due to the increased likelihood of obtaining approximate solutions that are infeasible for the original problem, triggering update steps solely to reduce constraint violations instead of the reward. A small but positive slack term is effective in enforcing the constraints and results in performances better than the unconstrained counterparts.

7.1. Future work

Further development is necessary to allow practical implementations of these algorithms. One of the challenges of the current design is that the training process can be time consuming and requires a large amount of training data – simulation tests show that more than 10 months worth of training data is needed to obtain reasonable control policies. Transfer learning is a promising approach to overcome this challenge that allows the use of trained building policies in other similar buildings. It has been proven to be effective in reducing training time and improving learning speed even when buildings have different physical characteristics, such as layout, materials, weather conditions, and even type of HVAC equipment [51]. Imitation learning also shows good potential of reducing training data requirement by providing a better starting point for the constrained RL algorithm by leveraging existing operational data [52]. The integration of these techniques with constrained RL algorithms will be addressed in future work.

References

[1] Eia, use of electricity explained. `https://www.eia.gov/energyexplained`.

[2] US Department of Energy. Benefits of demand response in electricity markets and recommendations for achieving them., 2006.

[3] Qinglong Meng, Yang Li, Xiaoxiao Ren, Chengyan Xiong, Wenqiang Wang, and Jiewei You. A demand-response method to balance electric power-grids via hvac systems using active energy-storage: Simulation and on-site experiment. *Energy Reports*, 7:762–777, 2021.

[4] Jerson Sanchez Zhimin Jiang and Jie Cai. Modelling and mitigating lifetime impact of building demand responsive control of heating, ventilation and air-conditioning systems. *Journal of Building Performance Simulation*, 15(6):771–787, 2022.

[5] Ye Yao and Divyanshu Kumar Shekhar. State of the art review on model predictive control mpc in heating ventilation and air-conditioning hvac field. *Building and Environment*, 200:107952, 2021.

[6] Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[7] Ján Drgoňa, Javier Arroyo, Iago Cupeiro Figueroa, David Blum, Krzysztof Arendt, Donghun Kim, Enric Perarnau Ollé, Juraj Oravec, Michael Wetter, Draguna L. Vrabie, and Lieve Helsen. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50:190–232, 2020.

[8] Frauke Oldewurtel, Andreas Ulbig, Alessandra Parisio, Göran Andersson, and Manfred Morari. Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1927–1932, 2010.

[9] Divya Tejaswini Vedullapalli, Ramtin Hadidi, and Bill Schroeder. Com-

bined hvac and battery scheduling for demand response in a building. *IEEE Transactions on Industry Applications*, 55(6):7008–7014, 2019.

[10] Olivier Van Cutsem, Maher Kayal, David Blum, and Marco Pritoni. Comparison of mpc formulations for building control under commercial time-of-use tariffs. In *2019 IEEE Milan PowerTech*, pages 1–6, 2019.

[11] Donghun Kim Jie Cai, James E. Braun and Jianghai Hu. General approaches for determining the savings potential of optimal control for cooling in commercial buildings having both energy and demand charges. *Science and Technology for the Built Environment*, 22(6):733–750, 2016.

[12] Mengjie Han, Ross May, Xingxing Zhang, Xinru Wang, Song Pan, Da Yan, Yuan Jin, and Liguo Xu. A review of reinforcement learning methodologies for controlling occupant comfort in buildings. *Sustainable Cities and Society*, 51:101748, 2019.

[13] Gregor P. Henze and Jobst Schoenmann. Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC&R Research*, 9(3):259–275, 2003.

[14] Simeng Liu and Gregor Henze. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory –part 1: Theoretical foundation. *Energy and Buildings*, 38:142–147, 02 2006.

[15] Simeng Liu and Gregor Henze. Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory.

Journal of Solar Energy Engineering-transactions of The Asme - J SOL ENERGY ENG, 129, 05 2007.

[16] Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017.

[17] Ki Uhn Ahn and Cheol Soo Park. Application of deep q-networks for model-free optimal control balancing between different hvac systems. *Science and Technology for the Built Environment*, 26(1):61–74, 2020.

[18] Zhanhong Jiang, Michael J. Risbeck, Vish Ramamurti, Sugumar Murugesan, Jaume Amores, Chenlu Zhang, Young M. Lee, and Kirk H. Drees. Building hvac control with reinforcement learning for reduction of energy cost and demand charge. *Energy and Buildings*, 239:110833, 2021.

[19] Yan Du, Helia Zandi, Olivera Kotevska, Kuldeep Kurte, Jeffery Munk, Kadir Amasyali, Evan Mckee, and Fangxing Li. Intelligent multi-zone residential hvac control strategy based on deep reinforcement learning. *Applied Energy*, 281:116117, 2021.

[20] Liang Yu, Weiwei Xie, Di Xie, Yulong Zou, Dengyin Zhang, Zhixin Sun, Linghua Zhang, Yue Zhang, and Tao Jiang. Deep reinforcement learning for smart home energy management. *IEEE Internet of Things Journal*, 7(4):2751–2762, 2020.

[21] Xiangyu Zhang, Dave Biagioni, Mengmeng Cai, Peter Graf, and Saifur Rahman. An edge-cloud integrated solution for buildings demand

response using reinforcement learning. *IEEE Transactions on Smart Grid*, PP:1–1, 08 2020.

[22] Kuldeep Kurte, Jeffrey Munk, Olivera Kotevska, Kadir Amasyali, Robert Smith, Evan McKee, Yan Du, Borui Cui, Teja Kuruganti, and Helia Zandi. Evaluating the adaptability of reinforcement learning based hvac control for residential houses. *Sustainability*, 12(18), 2020.

[23] Hepeng Li, Zhiqiang Wan, and Haibo He. Real-time residential demand response. *IEEE Transactions on Smart Grid*, 11(5):4144–4154, 2020.

[24] Kuthsav Thattai, Jayashri Ravishankar, and Chaojie Li. Consumer-centric home energy management system using trust region policy optimization- based multi-agent deep reinforcement learning. In *2023 IEEE Belgrade PowerTech*, pages 1–6, 2023.

[25] Xiangyu Zhang, Rohit Chintala, Andrey Bernstein, Peter Graf, and Xin Jin. Grid-interactive multi-zone building control using reinforcement learning with global-local policy search, 2020.

[26] Donald Azuatalam, Wee-Lih Lee, Frits de Nijs, and Ariel Liebman. Reinforcement learning for whole-building hvac control and demand response. *Energy and AI*, 2:100020, 2020.

[27] Ashkan Haji Hosseinloo, Saleh Nabi, Anette Hosoi, and Munther A. Dahleh. Data-driven control of covid-19 in buildings: A reinforcement-learning approach. *IEEE Transactions on Automation Science and Engineering*, pages 1–0, 2023.

[28] Felippe Schmoeller Roza, Hassan Rasheed, Karsten Roscher, Xiangyu Ning, and Stephan Günnemann. Safe robot navigation using constrained hierarchical reinforcement learning. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 737–742, 2022.

[29] Hepeng Li, Zhiqiang Wan, and Haibo He. Constrained ev charging scheduling based on safe deep reinforcement learning. *IEEE Transactions on Smart Grid*, 11(3):2427–2439, 2020.

[30] Yu-Jia Chen, Deng-Kai Chang, and Cheng Zhang. Autonomous tracking using a swarm of uavs: A constrained multi-agent reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, 69(11):13702–13717, 2020.

[31] Jie Cai and James Braun. An inverse hygrothermal model for multi-zone buildings. *Journal of Building Performance Simulation*, 9(5):510–528, 2016.

[32] Jie Cai. *A low cost multi-agent control approach for building energy system management*. PhD thesis, Purdue University, 2015.

[33] Jie Cai, James E. Braun, Donghun Kim, and Jianghai Hu and. General approaches for determining the savings potential of optimal control for cooling in commercial buildings having both energy and demand charges. *Science and Technology for the Built Environment*, 22(6):733–750, 2016.

[34] Jie Cai, Hao Zhang, Donghun Kim, James E. Braun, and Jianghai Hu. Convex optimization-based control of sustainable communities with on-

site photovoltaic (pv) and batteries. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1007–1012, 2017.

[35] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[36] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[38] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225, 2006.

[39] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[40] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

[41] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and

Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[43] *Constrained Markov Decision Processes.* 1999.

[44] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *CoRR*, abs/1705.10528, 2017.

[45] El Paso Electric. Small general service rate. `https://www.epelectric.com/customers/rates-and-regulations/business-rates-and-information/texas-rate-tariffs-rules-and-regulations/texas-rate-tariffs`, November 2021.

[46] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[47] Gurobi solver. Online at `https://www.gurobi.com/`.

[48] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[49] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward

Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[50] Benjamin Recht. A tour of reinforcement learning: The view from continuous control, 2018.

[51] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. One for many. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. ACM, nov 2020.

[52] Sourav Dey, Thibault Marzullo, Xiangyu Zhang, and Gregor Henze. Reinforcement learning building control approach harnessing imitation learning. *Energy and AI*, 14:100255, 2023.