Optimal Control of District Cooling Energy Plant with Reinforcement Learning and MPC

Zhong Guo*
PhD candidate
Department of Mechanical Engineering
University of Florida
Gainesville, Florida 32601
Email: zhong.guo@ufl.edu

Aditya Chaudhari
Postdoctoral Researcher
Department of Mechanical Engineering
University of Florida
Gainesville, Florida 32601
Email:ad.chaudhari@ufl.edu

Austin R. Coffman
PhD
Department of Mechanical Engineering
University of Florida
Gainesville, Florida 32601
Email: bubbaroney@ufl.edu

Prabir Barooah
Professor
Department of Electronics and Electrical Engineering
Indian Institute of Technology (Guwahati)
Guwahati, Assam, 781039
India
Email: pbarooah@iitg.ac.in

We consider the problem of optimal control of district cooling energy plants (DCEPs) consisting of multiple chillers, a cooling tower, and a thermal energy storage (TES), in the presence of time-varying electricity price. A straightforward application of model predictive control (MPC) requires solving a challenging mixed-integer nonlinear program (MINLP) because of the on/off of chillers and the complexity of the DCEP model. Reinforcement learning (RL) is an attractive alternative since its real-time control computation is much simpler. But designing an RL controller is challenging due to myriad design choices and computationally intensive training.

In this paper, we propose an RL controller and an MPC controller for minimizing the electricity cost of a DCEP, and compare them via simulations. The two controllers are designed to be comparable in terms of objective and information requirements. The RL controller uses a novel Q-learning algorithm that is based on least-squares policy iteration. We describe the design choices for the RL controller, including the choice of state space and basis functions, that are found to be effective. The proposed MPC controller does not need a mixed integer solver for implementation, but only a nonlinear program (NLP) solver. A rule-based baseline controller is also proposed to aid in comparison. Simulation results show that the proposed RL and MPC controllers achieve similar savings over the baseline controller, about 17%.

*Corresponding author. The research reported here has been partially supported by the NSF through award 1934322 (CMMI) and 2122313 (ECCS).

1 Introduction

In the U.S., 75% of the electricity is consumed by buildings, and a large part of that is due to heating, ventilation, and air conditioning (HVAC) systems [1]. In university campuses and large hotels, a large portion of the HVAC's share of electricity is consumed by District Cooling Energy Plants (DCEPs), especially in hot and humid climates. A DCEP produces and supplies chilled water to a group of buildings it serves (hence the moniker "district"), and the air handling units in those buildings use the chilled water to cool and dehumidify air before supplying it to building interiors. Figure 1 shows a schematic of such a plant, which consists of multiple chillers that produce chilled water, a cooling tower that rejects the heat extracted from chillers to the environment, and a thermal energy storage system (TES) for storing chilled water. Chillers - the most electricity-intensive equipment in the DCEP - can produce more chilled water than buildings' needs when the electricity price is low. The extra chilled water is stored in the TES, and then used during periods of high electricity price to reduce the total electricity cost. The District Cooling Energy Plants are also called central plants or chiller plants.

DCEPs are traditionally operated with rule-based control algorithms that use heuristics to reduce electricity cost while meeting the load, such as "chiller priority", "storage priority", and additional control sequencing for the cooling tower operation [2–8]. But making the best use of the chillers and the TES to keep the electricity cost at the minimum requires non-trivial decision making due to the discrete nature of some control commands, such as chiller on/off actuation, and highly nonlinear dynamics of the equipment in DCEPs. A growing body of work has proposed algorithms for optimal real-time control of DCEPs. Both Model Predictive Control (MPC) [9–17] and Reinforcement Learning (RL) [18–26] have been studied.

For MPC, a direct implementation requires solving a high dimension mixed-integer linear program (MINLP) that is quite challenging to solve. Various substitutive approaches are thus used, which can be categorized into two groups: NLP approximations [9–12] and MILP approximations [13–17]. NLP approximations generally leave the discrete commands for some predetermined control logic and only deal with continuous control commands, which may limit the potential of their savings. MILP approximations mostly adopt a linear DCEP model so that the problem is tractable, though solving large MILPs is also challenging.

An alternative to MPC is Reinforcement Learning (RL): an umbrella term for a set of tools used to approximate an optimal policy using data collected from a physical system, or more frequently, its simulation. Despite a burdensome design and learning phase, real-time control is simpler since control computation is an evaluation of a state-feedback policy. However, designing an RL controller for a DCEP is quite challenging. The performance of an RL controller depends on many design choices and training an RL controller is computationally onerous.

In this paper, we propose an RL controller and an MPC controller for a DCEP, and compare their performances with that of a rule-based baseline (BL) controller through simulations. All three controllers are designed to minimize total energy cost while meeting the required cooling load. The main source of flexibility is the TES, which allows a well-designed controller to charge the TES in periods of low electricity price. The proposed RL controller is based on a new learning algorithm that is inspired by the "convex Q-learning" proposed in recent work [27] and the classical least squares policy iteration (LSPI) algorithm [28]. Basis functions are carefully designed to reduce the computational burden in training the RL controller. The proposed MPC controller solves a two-fold non-linear program (NLP) that is transformed from the original MINLP via heuristics. Hence the MPC controller is "stand-in" for a true optimal controller and provides a sub-optimal solution to the original MINLP. The baseline controller that is used for comparison is designed to utilize the TES and time-varying electricity prices (to the extent possible with heuristics) to reduce energy costs. The RL controller and baseline controller have the same information about electricity price: the current price and a backward moving average.

The objective behind this work is to compare the performance of the two complementary approaches, MPC and RL, for the optimal control of all the principal actuators in a DCEP. The two controllers are designed to be comparable, in terms of objective and information requirements. We are not aware of many works that have performed such a comparison; the only exceptions are [25,26], but the decision-making is limited to a TES or temperature setpoints. Since both RL and MPC approaches have merits and weaknesses, designing a controller with one approach and showing it performs well leaves open the question: would the other have performed better? This paper takes a first step in addressing such questions. To aid in this comparison, both the controllers are designed to be approximations of the same intractable infinite horizon optimal control problem. Due to the large difference in the respective approaches (MPC and RL), it is not possible to ensure exact parallels for an"apples–to–apples" comparison. But the design problems for RL and MPC controllers have been formulated to be similar to the possible extent.

Simulation results show that both the controllers, RL and MPC, lead to significant and similar cost savings (16-18%) over a rule-based baseline controller. These values are comparable to that of MPC controllers with mixed-integer formulation reported in the literature, which vary from 10% to 17% [13–17]. The cooling load tracking performance is similar between them. The real-time computation burden of the RL controller is trivial compared to that of the MPC controller, but the RL controller leads to higher chiller switches (from off to on and vice versa). However, the MPC controller enjoys the advantage of error-free forecasts in the simulations, something the RL controller does not.

The rest of the manuscript is organized as follows. The contribution of the paper over the related literature is discussed in detail in Section 1.1. Section 2 describes the District Cooling Energy Plant and its simulation model as well as the control problem. Section 3 describes the proposed RL controller, Section 4 presents the proposed MPC controller,

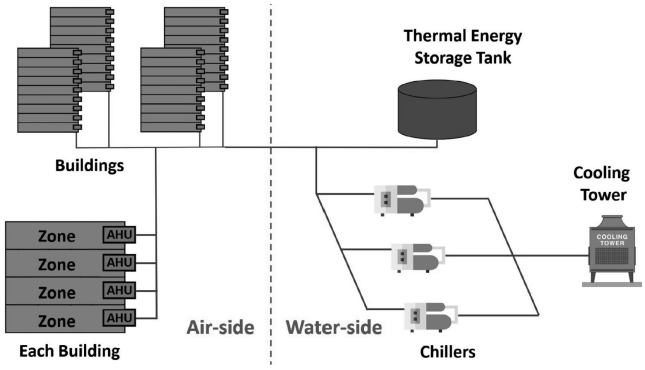


Fig. 1: Layout of District Cooling Energy Plant.

and Section 5 describes the baseline controller. Section 6 provides a simulation evaluation of the controllers. Section 7 provides an "under-the-hood" view of the design choices for the RL controller. Section 8 concludes the paper.

1.1 Literature Review and Contributions

1.1.1 Prior work on RL for DCEP

There is a large and growing body of work in this area, e.g. [18–26]. Most of these papers limit the problem to controlling part of a DCEP. For instance, the DCEPs considered in [18–21, 23] do not have a TES. Refs. [18–22] optimize only the chilled water loop but not the cooling water loop (at the cooling tower), while [24] only optimize the cooling water loop. The reported energy savings are in the 10-20% range over rule-based baseline controllers; e.g. 15.7% in [23], 11.5% in [18] and around 17% in [21].

The ref. [25] considers a complete DCEP, but the control command computed by the RL agent is limited to TES charging and discharging. It is not clear what control law is used to decide chiller commands and cooling water loop setpoints. The work [26] also considers a complete DCEP, with two chillers, a TES, and a large building with an air handling unit. The RL controller is tasked with commanding only the zone temperature setpoint and TES charging/discharging flowrate whilst the control of the chillers or the cooling tower is not considered. Besides, trajectories of external inputs, e.g., outside air temperature and electricity price, are the same for all training days in [26]. Another similarity of [25, 26] with this paper is that these references compare the performance of RL with that of a model-based predictive control.

1.1.2 Prior work on MPC for DCEP

The works that are closest to us in terms of problem setting are [13–15], which all reported MILP relaxation-based MPC schemes to optimally operate a DCEP with TES in presence of time-varying electricity prices. The paper [13] reports an energy cost savings with MPC of about 10% over a baseline strategy that uses a common heuristic (charge TES all night) with some decisions made by optimization. In [14], around 15% savings over the currently installed rule-based controller is achieved in a real DCEP. The study [15] reported a cost savings of 17% over "without load shifting" with the help of the TES in a week-long simulation. The paper [16] also proposes an MILP relaxation-based MPC scheme for controlling a DCEP and approximately 10% savings in electricity cost over a baseline controller over a one-day long simulation is reported. But the DCEP model in [16] ignores the effect of weather condition on plant efficiency, and the baseline controller is not purely rule-based; it makes TES and chiller decisions based on a greedy search. The recent paper [17] deserves special mention since it reports an experimental demonstration of MPC applied to a large DCEP; the control objective being the manipulation of demand to help with renewable integration and decarbonization. It too uses an MILP relaxation. The decision variables include plant mode (combination of chillers on) and TES operation, but cooling water loop decisions are left to legacy rule-based controllers.

There is another body of work applying MPC to the control a DCEP, such as [9–12]. But they either ignore the on/off nature of the chiller control [9, 10] or reformulate the problem using some heuristics [11, 12] so that the underlying optimization problem is naturally an NLP.

1.2 Contribution over Priori Arts

1.2.1 Contribution over "RL for DCEP" literature:

Unlike most prior works on RL for DCEPs that only deal with a part of DCEP [18–24], the control commands in this work consist of all the available commands (five in total) of both the chilled and cooling water loops in a full DCEP. To the best of our knowledge, no prior work has used RL to command both the water loops and a TES. Second, unlike some of the closely related work such as [26], we treat external inputs such as weather and electricity price as RL states, making the proposed RL controller applicable for any time-varying disturbances that can be measured in real time. Otherwise the controller is likely to work well only for disturbances seen during training. Third, the proposed RL controller commands the on/off status of chillers directly rather than the chilled/cooling water temperature setpoints [19, 21, 23] or zone temperature setpoints [26], which eliminates the need for another control system to translate those setpoints to chiller commands. Fourth, all the works cited above rely on discretizing the state and/or action spaces in order to use the classical tabular learning algorithms with the exception of [22]. The size of the table will become prohibitively large if the number of states and control commands becomes large and a fine-resolution discretization is used. Training a such controller and using it in real time, which will require searching over this table, will become computationally challenging. That is perhaps why only a small number of inputs are chosen as control commands in prior work even though several more setpoints can be manipulated in a real DCEP. Although [22] considers continuous states, its proposed method only controls part of a DCEP with simplified linear plant models, which may significantly limit its potential of cost savings in reality. In contrast, the RL controller proposed in this paper is for a DCEP model consisting of highly nonlinear equations, and the states and actions are

kept as continuous except for the one command that is naturally discrete (number of chillers that are on).

While there is an extensive literature on learning algorithms and on designing RL controllers, the design of an RL controller for practically relevant applications with non-trivial dynamics is quite challenging. RL's performance depends on myriad design choices, not only on the stage cost/reward, function approximation architecture and basis functions, learning algorithm and method of exploration, but also on the choice of the state space itself. A second challenge is that training an RL controller is computationally intensive and brute force training is beyond the computational means of most researchers. For instance, The hardware cost for a single AlphaGo Zero system in 2017 by DeepMind has been quoted to be around $25 million [29]. Careful selection of the design choices mentioned above is thus required, which leads to the third challenge: if a particular set of design choices leads to a policy that does not perform well, there is no principled method to look for improvement. Although RL is being extensively studied in the control community, most works demonstrate their algorithms on plants with simple dynamics with a small number of states and inputs; e.g. [30, 31]. The model for a DCEP used in this paper, arguably still simple compared to available simulation models (e.g. [32]), is quite complex: it has 8 states, 5 control inputs, 3 disturbance inputs, and requires solving an optimization problem to compute the next state given the current state, control, and disturbance.

1.2.2 Contribution over "MPC for DCEP" literature:

The MPC controller proposed here uses a combination of relaxation and heuristics to avoid the MINLP formulation. In contrast to [13–17], the MPC controller does not use a MILP relaxation. The controller does compute discrete decisions (number chillers to be on, TES charge/discharge) directly, but it does so by using NLP solvers in conjunction with heuristics. The cost saving obtained is similar to those reported in earlier works that use MILP relaxation. Comparing other NLP formulations [9–12], our MPC controller determines the on/off actuation of equipments and TES charging/discharging operation directly.

Closed-loop simulations are provided for all three controllers, RL, MPC, and baseline, to assess the trade-offs among these controllers, especially between the model-based MPC controller and the "model-free" RL controller.

1.2.3 Contribution over a preliminary version:

The RL controller described here was presented in a preliminary version of this paper [33]. There are three improvements. Firstly, an MPC controller, which is not presented in [33], was designed, evaluated, and compared with our RL controller. Therefore, the optimality of our control with RL is better assessed. Another difference is that the baseline controller described here is improved over that in [33] so that the frequency of on/off switching of chillers is reduced. Lastly, a much more thorough discussion of the RL controller design choices and their observed impacts are included here than in [33]. Given the main challenge with designing RL controllers for complex physical systems discussed above, namely, "what knob to tweak when it doesn't work?", we believe this information will be valuable to other researchers.

2 System description and control problem

The DCEP contains a TES, multiple chillers and chilled water pumps, a cooling tower and cooling water pumps, and finally a collection of buildings that uses the chilled water to provide air conditioning; see Figure 2. The heat load from the buildings is absorbed by the cold chilled water supplied by the DCEP, and thus the return chilled water temperature is warmer. This part of the water is called *load water*, and the related variables are denoted by superscript lw for "load water". The *chilled water loop* (subscript chw) removes this heat and transmits it to the *cooling water loop* (subscript cw). The cooling water loop absorbs this heat and sends it to the cooling tower, where this heat is then rejected to the ambient. The cooling tower cools down the cooling water returned from the chiller by passing both the sprayed cooling water and ambient air through a fill. During this process, a small amount of water spray will evaporate into the air, removing heat from the cooling water. The cooling water loss due to its evaporation is replenished by fresh water, thus we assume the supply water flow rate equals to the return water flow rate at the cooling tower. A fan or a set of fans is used to maintain the ambient airflow at the cooling tower. Connected to the chilled water loop is a TES tank that stores water (subscript tw). The total volume of the water in the TES tank is constant, but a thermocline separates two volumes: cold water that is supplied by the chiller (subscript twc for "tank water, cold") and warm water returned from the load (subscript tww for "tank water, warm").

2.1 DCEP dynamics

Assuming time is discretized with a sampling period t_s with a counter $k = 0, 1, \cdots$ denoting the time step. With the consideration of hardware limits and ease of implementation, the control commands are chosen as follows:

1. $\dot{m}_k^{\mathrm{lw}}$, the chilled water flowrate going through the cooling coil, to ensure the required cooling load is met.
2. $\dot{m}^{\mathrm{tw}}$, charging/discharging flowrate of the TES, to take advantage of load shifting.
3. n^{ch}, the number of active chillers to ensure the amount of chilled water required is met and the coldness of the chilled water is maintained.
4. $\dot{m}^{\mathrm{cw}}$, the flowrate of cooling water going through the condenser of chillers to absorb the heat from the chilled water loop.
5. $\dot{m}^{\mathrm{oa}}$, the flowrate of ambient air that cools down the cooling water to maintain its temperature within the desired range.

Therefore, the control command u_k is:

$$u_k := [\dot{m}_k^{\mathrm{lw}}, \dot{m}_k^{\mathrm{tw}}, n_k^{\mathrm{ch}}, \dot{m}_k^{\mathrm{cw}}, \dot{m}_k^{\mathrm{oa}}]^T \in \mathsf{U}. \tag{1}$$

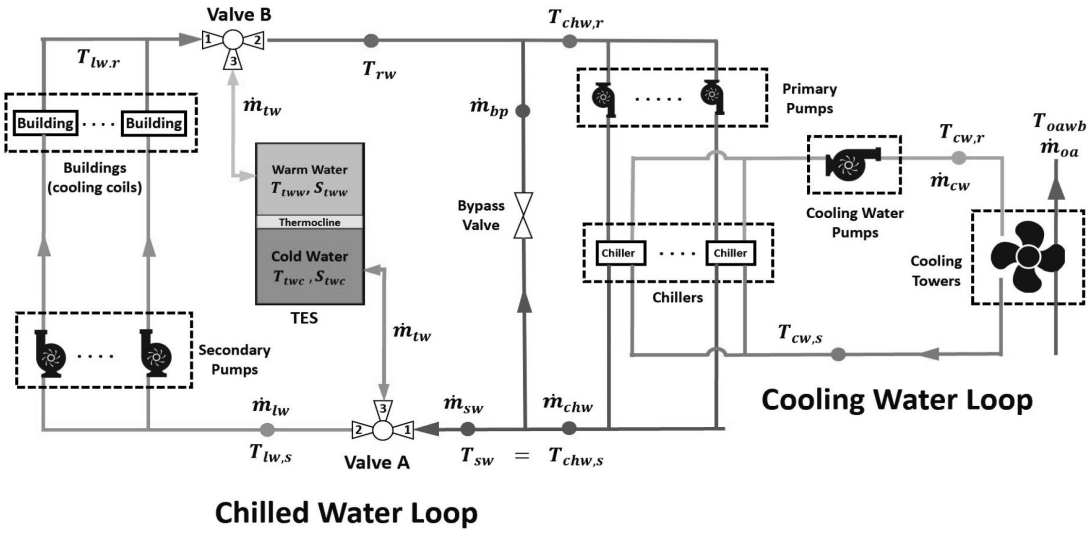


Fig. 2: Detailed description of District Cooling Energy Plant.

Each of these variables can be independently chosen as setpoints since lower level PI-control loops maintain them. There are limits to these setpoints, which determine the admissible input set $\mathsf{U} \subset \{0, \dots \dot{m}^{\text{ch}}_{\max}\} \times \mathsf{R}^4$:

$$\mathsf{U} \triangleq \{0, \dots \dot{m}^{\text{ch}}_{\max}\} \times [\dot{m}^{\text{lw}}_{\min}, \dot{m}^{\text{lw}}_{\max}] \times [\dot{m}^{\text{tw}}_{\min}, \dot{m}^{\text{tw}}_{\max}] \dots \times [\dot{m}^{\text{cw}}_{\min}, \dot{m}^{\text{cw}}_{\max}] \times [\dot{m}^{\text{oa}}_{\min}, \dot{m}^{\text{oa}}_{\max}] \subset \mathsf{R}^5. \quad (2)$$

Since the TES can be charged and discharged, we declare $\dot{m}^{\text{tw}} > 0$ for charging and $\dot{m}^{\text{tw}} < 0$ for discharging as a convention.

The state of the DCEP x^p is:

$$x^p_k \triangleq [T^{\text{lw,r}}_k, S^{\text{tww}}_k, S^{\text{twc}}_k, T^{\text{twc}}_k, T^{\text{tww}}_k, T^{\text{chw,s}}_k, T^{\text{cw,r}}_k, T^{\text{cw,s}}_k]^T, \quad (3)$$

where $S^{\text{tww}}, S^{\text{twc}}$ are the fractions of the warm water and cold water in the TES tank, $S^{\text{tww}} + S^{\text{twc}} = 1$. The other state variables are temperatures at various locations - supply (subscript "s") and return (subscript "r") - in each water loop: load water, cooling water, tank water, and chiller; see Figure 2 for details. All the plant state variables x^p can be measured with sensors. The superscript "p" of x emphasizes that x^p is the state of the "plant", not the state in the reinforcement learning method that will be introduced in Section 3.3.

The plant state x^p is affected by exogenous disturbances $w^p_k := [T^{\text{oawb}}_k, \dot{q}^{\text{L,ref}}_k]^T \in \mathsf{R}^2$, where $\dot{q}^{\text{L,ref}}_k$ is the required cooling load, the rate at which heat needs to be removed from buildings, and T^{oawb}_k is the ambient wet bulb temperature. The disturbance w^p_k cannot be ignored, e.g., ambient wet-bulb temperature plays a critical role in cooling tower dynamics.

The control command and disturbances affect the state through a highly nonlinear dynamic model:

$$x^p_{k+1} = f(x^p_k, u_k, w^p_k), \quad (4)$$

that is described in detail in [34]. The dynamics (4) are implicit: there is no explicit function $f(\cdot)$ that can be evaluated to obtain x_{k+1}. The reason is that all the heat exchangers (in the building cooling coils, in each chiller, and in the cooling tower) have limited capacities. Depending on the cooling load, number of active chillers, and the outdoor air wet-bulb temperature, one of the heat exchangers might saturate. Meaning, it will then only deliver as much exchange as its capacity, less than what is desired due to the load. Which heat exchanger will saturate first depends on the current state and disturbance and control in a complex manner. Hence, some form of iterative computation is required to simulate the dynamics, e.g., the method developed in [35]. A generalized way to perform the iterative update to account for the limits of heat exchange capacities is by solving a constrained optimization problem, which is the method used in this work. The method is described in detail in [34].

Here we provide an outline for use in the sequel. First, define the decision variable z_k as:

$$z_k \triangleq (x^p_{k+1})^T, \dot{q}^{\text{L}}_k, \dot{q}^{\text{ch}}_k, \dot{q}^{\text{ct}}_k{}^T, \quad (5)$$

where x^p is defined in (3), $\dot{q}^{\text{L}}$ is the cooling load met by the DCEP, $\dot{q}^{\text{ch}}$ and $\dot{q}^{\text{ct}}$ are the cooling provided by chillers and cooling towers. Then the value of z_k is computed by solving the following optimization problem:

$$z^*_k = \arg \min_{z_k \in \Omega(x^p_k, w^p_k, u_k)} r_1 \| \dot{q}^{\text{L}}_k - \dot{q}^{\text{L,ref}}_k \| + r_2 \| T^{\text{chw,s}}_{k+1} - T^{\text{chw,s}}_{\text{set}} \| + r_3 \| T^{\text{cw,s}}_{k+1} - T^{\text{cw,s}}_{\text{set}} \|, \quad (6)$$

where $T^{\text{chw,s}}_{\text{set}}$ and $T^{\text{cw,s}}_{\text{set}}$ are pre-specified setpoints that reflect DCEP nominal working conditions and r_1, r_2, and r_3 are positive design choices, with $r_1 \gg r_2, r_3$ to promote load tracking. The set $\Omega(x^p_k, w^p_k, u_k)$ is defined by the dynamics and constraints of the DCEP system, including the dynamics of

the various heat exchangers and the TES, and the capacity limits of the heat exchangers in the buildings' air handling units, chillers, and the cooling tower. Please refer to [34] for the derivation of $\Omega(x_k^p, w_k^p, u_k)$. When the required cooling lo $\dot{q}_k^{\mathrm{L,ref}}$ is within the capacity of all the heat exchangers, then the solution to (6) yields $\dot{q}_k^{\mathrm{L}} = \dot{q}_k^{\mathrm{L,ref}}$. When the required load exceeds the capacity of the DCEP, then (6) will lead to a solution that trades off maintaining nominal setpoints and meeting the cooling load, while respecting the limits of the heat exchangers. The solution leads to the next state x_{k+1}^p (as the first component of z_k^*), and thus (6) implicitly defines the model $f(\cdot)$. In this paper, we use CasADi/IPOPT [36, 37] to solve (6) for simulating the plant.

2.2 Electrical demand and electricity cost

In the DCEP considered, the only energy used is electricity. The relationship between the thermal quantities and the electricity consumption in chillers and cooling tower are complex. We model the chillers power consumption P^{ch} as [38]:

$$P_k^{\mathrm{ch}} = \left(\frac{T_k^{\mathrm{cw,s}}}{T_k^{\mathrm{chw,s}}} - 1\right)\dot{q}_k^{\mathrm{ch}} - \beta_1 + \beta_2 T_k^{\mathrm{cw,s}} - \beta_3 \frac{T_k^{\mathrm{cw,s}}}{T_k^{\mathrm{chw,s}}}. \quad (7)$$

Power consumption of water pumps is modeled using the black-box model in [13]:

$$P_k^{\mathrm{chw,pump}} = \alpha_1 \ln(1 + \alpha_2 \dot{m}_k^{\mathrm{chw}}) + \alpha_3 \dot{m}_k^{\mathrm{chw}} + \alpha_4, \quad (8)$$

$$P_k^{\mathrm{cw,pump}} = \gamma_1 \ln(1 + \gamma_2 \dot{m}_k^{\mathrm{cw}}) + \gamma_3 \dot{m}_k^{\mathrm{cw}} + \gamma_4. \quad (9)$$

Finally, the electrical power consumption of the cooling tower mainly comes from its fan and is modeled as [39]:

$$P_k^{\mathrm{ct}} = \lambda(\dot{m}_k^{\mathrm{oa}})^3. \quad (10)$$

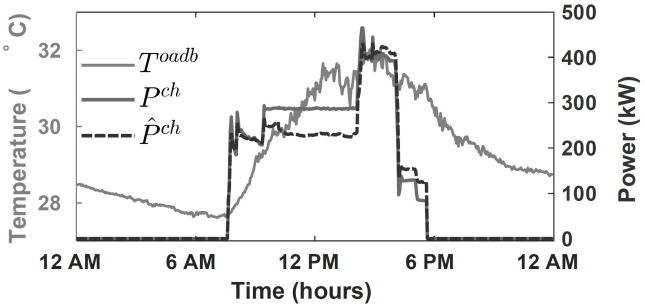
The constants $\alpha_i, \beta_i, \gamma_i$, and λ are empirical parameters. The total electric power consumption of the DCEP is:

$$P_k^{\mathrm{tot}} = P_k^{\mathrm{ch}} + P_k^{\mathrm{ct}} + P_k^{\mathrm{chw,pump}} + P_k^{\mathrm{cw,pump}}. \quad (11)$$

2.3 Model calibration and validation

The parameters of the simulation model in Section 2.1 and electrical demand model in Section 2.2 are calibrated using data from the energy management system in United World College (UWC) of South East Asia Tampines Campus in Singapore, shown in Figure 3b. The data is publicly available in [40], and details of the data are discussed in [41]. There are three chillers and nine cooling towers in the DCEP. The data from chiller one and cooling tower one are used for model calibration. We use 80% of data for model identification and 20% of data for verification. The out-of-sample prediction results for the total electrical demand are shown in Figure 3. Comparison between data and prediction for other variables are not shown in the interest of space.

(a) Schematic of the UWC Tampines Campus, from [40].



(b) Chiller power measurement P^{ch} and prediction $\hat{P}^{\mathrm{ch}}$.

Fig. 3: (Top) Map of the campus with a DCEP whose data is used for model calibration, and (Bottom) Out of sample prediction for P^{ch} using the calibrated model (7). T^{oadb} is the ambient dry-bulb temperature.

2.4 The (ideal) control problem

The electricity cost incurred during the k-th time step is:

$$c_k^{\mathrm{E}} := t_s \rho_k P_k^{\mathrm{tot}}, \quad (12)$$

where P_k^{tot} is the total electric power consumed in k and is defined in (11). The goal of operating the DCEP to minimize electricity cost while meeting the required cooling load $\dot{q}_k^{\mathrm{L,ref}}$ can be posed as the following infinite horizon optimal control problem.

$$\min_{\{u_k\}_{k=0}^{\infty}} \sum_{k=0}^{\infty} c_k^{\mathrm{E}}, \quad (13)$$

$$\text{s.t.} \quad x_{k+1}^p = f(x_k^p, u_k, w_k^p),\ x_0^p = x,$$
$$x_k^p \in \mathsf{X}^p(w_k^p), \quad u_k \in \mathsf{U}(x_k^p, w_k) \quad (14)$$

$$\dot{q}_k^{\mathrm{L}}(x_k^p, u_k) = \dot{q}_k^{\mathrm{L,ref}}, \quad (15)$$

where ρ_k ($\frac{\mathrm{USD}}{\mathrm{kWh}}$) is the electricity price. The state x_k^p, input u_k, and disturbance w_k^p of the DCEP are defined in Section 2.1; $\dot{q}_k^{\mathrm{L}}$ ("L" stands for "load") represents the actual cooling load met by the DCEP, which is a function of x_k^p and u_k. The bounds for x_k^p and u_k are $\mathsf{X}^p(w_k)$ and $\mathsf{U}(x_k^p, w_k)$. The reason

these sets are dependent on the state or disturbance can be found in the description of the dynamic model of the plant in [34].

Even when truncated to a finite planning horizon considered, Problem (13) is an MINLP due to n_k^{ch} being an integer and the nonlinear dynamics (4). In the sequel, we propose two controllers to solve approximations of this idealized problem.

3 RL basics and proposed RL controller

3.1 RL basics

For the following construction, let x represent the state with state space X and u the input with input space $\mathsf{U}(x)$. Now consider the following infinite horizon discounted optimal control problem:

$$J^*(\bar{x}) = \min_{\mathbf{U}} \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k), \quad x_0 = \bar{x}, \qquad (16)$$
$$\text{s.t.} \quad x_{k+1} = F(x_k, u_k), \; u_k \in \mathsf{U}(x_k),$$

where $\mathbf{U} \triangleq \{u_0, \ldots,\}$ $c : \mathsf{X} \times \mathsf{U} \to \mathsf{R}^{\geq 0}$ is the stage cost, $\gamma \in (0, 1)$ is the discount factor, $F(\cdot, \cdot)$ defines the dynamics, and $J^* : \mathsf{X} \to \mathsf{R}^+$ is the optimal value function. The goal of the RL framework is to learn an approximate optimal policy $\phi : \mathsf{X} \to \mathsf{U}$ for the problem (16) without requiring explicit knowledge of the model $F(\cdot, \cdot)$. The learning process is based on the Q function. Given a policy ϕ for the problem (16), the Q function associated with this policy is defined as

$$Q_\phi(x, u) = \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k), \quad x_0 = x, \quad u_0 = u, \qquad (17)$$

where for $k \geq 0$ we have $x_{k+1} = F(x_k, u_k)$ and for $k \geq 1$ we have $u_k = \phi(x_k)$. A well-known fact is that the optimal policy satisfies [42]:

$$\phi^*(x) = \arg\min_{u \in \mathsf{U}(x)} Q^*(x, u), \quad \text{for all} \quad x \in \mathsf{X}, \qquad (18)$$

where $Q^* \triangleq Q_{\phi^*}$ is the Q function for the optimal policy. Further, for any policy ϕ the Q function satisfies the following fixed point relation:

$$Q_\phi(x, u) = c(x, u) + \gamma Q_\phi\left(x^+, \phi(x^+)\right), \qquad (19)$$

for all $u \in \mathsf{U}(x)$, $x \in \mathsf{X}$, and $x^+ = F(x, u)$. The above relation is termed here as the fixed-policy Bellman equation. If the optimal Q-function can be learned, the optimal control command u_k^* is computed from the Q-function as:

$$u_k^* := \phi^*(x_k) = \arg\min_{u \in \mathsf{U}(x_k)} Q^*(x_k, u), \qquad (20)$$

3.2 Proposed RL algorithm

The proposed learning algorithm has two parts: policy evaluation and policy improvement. First, in policy evaluation, a parametric approximation to the fixed policy Q-function is learned by constructing a residual term from (19) as an error to minimize. Second, in policy improvement, the learned approximation is used to define a new policy based on (18). For policy evaluation, suppose for a policy ϕ the Q function is approximated as:

$$Q_\phi(x, u) \approx Q_\phi^\theta(x, u) \qquad (21)$$

where $Q_\phi^\theta(\cdot, \cdot)$ is the function approximator (e.g., a neural network) and $\theta \in \mathsf{R}^d$ is the parameter vector (e.g., weights of the network). To fit the approximator, suppose that the system is simulated for T_{sim} time so that T_{sim} tuples of (x_k, u_k, x_{k+1}) are collected to produce T_{sim} values of:

$$d_k(\theta) = c(x_k, u_k) + \gamma Q_\phi^\theta(x_{k+1}, \phi(x_{k+1})) - Q_\phi^\theta(x_k, u_k), \qquad (22)$$

which is the temporal difference error for the approximator. We then obtain θ^* by solving the following optimization problem:

$$\theta^* \triangleq \arg\min_{\theta} \|D(\theta)\|_2 + \alpha \|\theta - \bar{\theta}\|_2, \qquad (23)$$
$$\text{s.t.} \quad Q_\phi^\theta \geq 0$$

where $D(\theta) \triangleq [d_0(\theta), \ldots d_{\mathsf{T}_{\text{sim}}-1}(\theta)]$. The term $\|\theta - \bar{\theta}\|_2$ is a regularizer and α is a gain. The values of $\bar{\theta}$ and α are specified in step 3) of Algorithm 1. The non-negativity constraint on the approximate Q-function is imposed since the Q-function is a discounted sum of non-negative terms (17). How it is enforced is described in Section 3.3.3. The solution to (23) results in $Q_\phi^{\theta^*}$, which is an approximation to Q_ϕ. The quantity $Q_\phi^{\theta^*}$ can be used to obtain an improved policy, denoted ϕ^+, through

$$\phi^+(x) = \arg\min_{u \in \mathsf{U}(x)} Q_\phi^{\theta^*}(x, u), \quad \text{for all} \quad x \in \mathsf{X}. \qquad (24)$$

This process of policy evaluation (23) and policy improvement (24) are repeated. This iterative procedure is described formally in Algorithm 1, with N_{pol} denoting the number of policy improvements.

This algorithm is inspired by: (i) the Batch Convex-Q learning algorithm found in [27, Section III] and (ii) the least squares policy evaluation (LSPI) algorithm [28]. The approach here is simpler than the batch optimization problem that underlies the algorithm in [27, section III], which has an objective function that itself contains an optimization problem. In comparison to [28] we include a regularization term that is inspired by proximal methods in optimization that aids convergence, and a constraint to ensure the learned Q-function is non-negative.

Algorithm 1: Data Driven Policy Iteration: Batch mode and off-policy

Result: An approximate optimal policy $\phi^{N_{pol}}(x)$.
Input: T_{sim}, θ^0, N_{pol}, $\beta > 1$
for $j = 0, \ldots, N_{pol} - 1$ **do**

1) Follow an exploration stategy and obtain input sequence $\{u_k^j\}_{k=0}^{T_{sim}-1}$, initial state x_0^j, and state sequence $\{x_k^j\}_{k=1}^{T_{sim}}$.

2) For $k = 1, \ldots, T_{sim}$, obtain: $\phi^j(x_k) = \arg\min_{u \in U(x_k^j)} Q_\phi^{\theta^j}(x_k^j, u)$.

3) Set $\bar{\theta} = \theta^j$ and $\alpha = \frac{j}{\beta}$ appearing in (23).

4) Use the samples $\{u_k^j\}_{k=0}^{T_{sim}-1}$, $\{x_k^j\}_{k=0}^{T_{sim}}$, and $\{\phi^j(x_k)\}_{k=1}^{T_{sim}}$ to construct and solve (23) for θ^*.

5) Set $\theta^{j+1} = \theta^*$.

end

3.3 Proposed RL controller for DCEP

We now specify the ingredients required to apply Algorithm 1 to obtain an RL controller (i.e., a state feedback policy) for the DCEP from simulation data. Namely, (1) the state description, (2) the cost function design, (3) the approximation architecture, and (4) the exploration strategy. Parts (1), (2), and (3) refer to the setup of the optimal control problem that the RL algorithm is attempting to approximately solve. Part (4) refers to the selection of how the state/input space is explored (step 1 in Algorithm 1).

3.3.1 State space description

In RL, the construction of the state space is an important feature, and the state is not necessarily the same as the plant state. To define the state space for RL, we first denote w_k as the vector of exogenous variables:

$$w_k = [(w_k^p)^T, \rho_k, \bar{\rho}_k] \in \mathbb{R}^4. \tag{25}$$

where $\bar{\rho}_k = \frac{1}{\tau}\sum_{t=k-\tau}^{k} \rho_t$ is a backwards moving average of the electricity price. The expanded state for RL is:

$$x_k \triangleq [x_k^p, w_k]^T \in \mathcal{X} \triangleq \subset \mathbb{R}^{12}. \tag{26}$$

Note that with the state defined by (26),*a state feedback policy is implementable* since all entries of x_k can be measured with commercially available sensors (e.g., outside wet-bulb temperature, T_{oawb}), or estimated from measurements (e.g., the thermal load from buildings, $\dot{q}^{L,ref}$), or known via real-time communication (e.g., the electricity prices, ρ_k and $\bar{\rho}_k$).

3.3.2 Design of stage cost

The design of the stage cost is also an important aspect of RL. We wish to obtain a policy that tracks the load $\dot{q}_k^{L,ref}$ whilst spending a minimal amount of money, as described in section 2.4. Therefore we choose:

$$c(x_k, u_k) \triangleq c_k^E + \kappa \left\| \dot{q}_k^L - \dot{q}_k^{L,ref} \right\|^2, \tag{27}$$

where κ is a design parameter; $\kappa \gg 1$ will prefer load tracking over energy cost.

3.3.3 Approximation architecture

We choose the following linear-in-the-parameter approximation of the Q function:

$$Q_\phi^\theta(x, u) = \sum_{\ell=1}^{d} \psi_\ell(x, u)\theta_\ell, \tag{28}$$

where $\psi_\ell(x, u)$ are nonlinear basis functions and $\theta \in \mathbb{R}^d$ is the parameter vector. We elect a quadratic basis, so that each $\psi_\ell(x, u)$ is of the form xu, x^2, or u^2. Specifically, a subset of all possible quadratic terms is chosen as the basis. More on this subset is provided in Section 7. We can equivalently express the approximation (28) as:

$$Q_\phi^\theta(x, u) = [x, u]P_\theta[x, u]^T, \tag{29}$$

for appropriately chosen P_θ. In this form, it is straightforward to enforce the constraint in (23) by enforcing the convex constraint $P_\theta \succeq 0$.

3.3.4 Exploration strategy

Exploration refers to how the state/input sequences appearing in step 1) of Algorithm 1 are simulated. We utilize a modified $\varepsilon-$greedy exploration scheme. At time step k of iteration j, we obtain the input u_k^j from one of three methods: (i) by using the policy in step 2) of Algorithm 1, (ii) electing uniformly random feasible inputs, and (iii) using a rule-based baseline controller (described in Section 5). The states are obtained sequentially through simulation, starting from state x_0^j for each j. The choice to use either of the three controllers is determined by the probability mass function $\nu_{exp}^j \in \mathbb{R}^3$, which depends on the iteration index of the policy iteration loop:

$$\nu_{exp}^j = \begin{cases} [0, 0.1, 0.9] & \text{for } j \leq 5. \\ [0.5, 0.25, 0.25] & \text{for } j > 5. \end{cases} \tag{30}$$

The entries correspond to the probability of using the corresponding control strategy, which appears in the (i)-(iii) order as just introduced. The rationale for this choice is that the BL controller provides "reasonable" state input examples for the RL algorithm in the early learning iterations to steer the parameter values in the correct direction. After this early learning phase, weight is shifted towards the current working policy to force the learning algorithm to update the parameter vector in response to its actions.

3.3.5 Training settings

The policy evaluation problem (23) during training is solved using CVX [43]. The simulation model (4) to generate state updates, which requires solving a non-convex NLP, is solved using CasADi and IPOPT [36, 37].

The parameters used for RL training are $\gamma = 0.97$, $d = 36$, $\kappa = 500$, $\beta = 100$, $T_{\text{sim}} = 432$ and $N_{\text{pol}} = 50$. The parameter τ for the backward moving average filter on the electricity price is chosen to represent 4 hours. The choice of the 36 basis functions is a bit involved; they are discussed in Section 7. Because a simulation time step, k to $k+1$, corresponds to a time interval of 10 minutes, $T_{\text{sim}} = 432$ corresponds to 3 days. The controller was trained with weather and load data for the three days Oct. 10-12, 2011, from the Singapore UWC campus dataset described in Section 2.3. The electricity price data used for training was taken as a scaled version of the locational marginal price from PJM [44] for the three days Aug. 30 - Sept. 1, 2021.

3.4 Real time implementation

Once the RL controller is trained, it computes the control command u_k in real-time as:

$$u_k := \phi^*(x_k) = \arg\min_{u \in \mathsf{U}(x_k)} Q_\phi^{\hat{\theta}}(x_k, u), \tag{31}$$

where $\hat{\theta}$ is the parameter vector learned in Algorithm 1. This $\hat{\theta}$ needs not to be $\theta^{N_{\text{pol}}}$ but the one with the best closed-loop performance, which is explained later in Section 6.2.

Due to the non-convexity of the set $\mathsf{U}(x_k)$ and the integer nature of n_k^{ch}, the problem (31) is non-convex and integer-valued. We solve it using exhaustive search: for each possible value of n_k^{ch}, we solve the corresponding continuous variable non-linear program using CasADi/IPOPT [36, 37], and then choose the minimum out of $(n_{\max}^{\text{ch}}+1)$ solutions by direct search. Direct search is feasible because $n_{\max}^{\text{ch}}$ for DCEPs is a small number in practice ($n_{\max}^{\text{ch}} = 7$ in our simulated example).

4 Proposed Model Predictive Controller

Recall that a straightforward translation of (13) to MPC will require solving the following problem at every time index k (here we only describe the one at $k = 0$ to avoid cumbersome notation):

$$\begin{aligned} \min_{\{u_k\}_{k=0}^{T^{\text{plan}}-1}} \quad & \sum_{k=0}^{T^{\text{plan}}-1} c_k^{\text{E}}, \\ \text{s.t.} \quad & x_{k+1}^p = f(x_k^p, u_k, w_k^p),\ x_0^p = x, \\ & x_k^p \in \mathsf{X}^p(w_k^p), \quad u_k \in \mathsf{U}(x_k^p, w_k) \\ & \dot{q}_k^{\text{L}}(x_k^p, u_k) = \dot{q}_k^{\text{L,ref}}, \end{aligned} \tag{32}$$

where c_k^{E} is defined in (12), and T^{plan} is the planning horizon. Even for a moderate planning horizon T^{plan} the optimization problem (32) will be a large MINLP. We now describe an algorithm that uses a dynamic model of the DCEP to approximately solve (32) without needing to solve an MINLP or even an MILP. This algorithm, which we call *MBOC*, for *Model Based (sub) Optimal Controller*, is then used to implement MPC by repeatedly applying it in a receding horizon setting as new forecasts of external disturbances become available.

The first challenge we have to overcome is not related to the mixed-integer nature of the problem but is related to the complex nature of the dynamics. Recall from Section 2.1 that the dynamic model, i.e., the function f in the equality constraint $x_{k+1} = f(\cdot)$ in (4) is not available in explicit form; rather the state is propagated in the simulation by solving an optimization problem. Without an explicit form for the function $f(\cdot)$, modern software tools that reduce the drudgery in nonlinear programming, namely numerical solvers with automatic differentiation, cannot be used.

We address this challenge by substituting the implicit equality constraint $x_{k+1}^p = f(x_k^p, u_k, w_k^p)$ in (32) with the underlying constraints $\Omega_k(\cdot)$ in (6), and add the objective of (6) to the objective of (32). The modified problem becomes:

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=0}^{T^{\text{plan}}-1} c_k^{\text{E}} + r_1 \| \dot{q}_k^{\text{L}} - \dot{q}_k^{\text{L,ref}} \|^2 + r_2 \| T_{k+1}^{\text{chw,s}} - T_{\text{set}}^{\text{chw,s}} \|^2 \\ & + r_3 \| T_{k+1}^{\text{cw,s}} - T_{\text{set}}^{\text{cw,s}} \|^2, \\ \text{s.t.} \quad & x_{k+1}^p \in \Omega_k(x_k^p, u_k, w_k^p),\ x_0^p = x, \\ & x_k^p \in \mathsf{X}^p(w_k^p), \quad u_k \in \mathsf{U}(x_k^p, w_k). \end{aligned} \tag{33}$$

Since the input n_k^{ch} takes integer value in the set $\{0, 1, \ldots n_{\max}^{\text{ch}}\}$, the problem (33) is still a high-dimensional MINLP.

The proposed algorithm to approximately solve (33) without using an MINLP solver or an MILP relaxation consists of three steps. These are listed below in brief, with more details provided subsequently.

1. The integer variable $n^{\text{ch}} \in [0, 1, \ldots n_{\max}^{\text{ch}}]$ is relaxed to a continuous one $n^{\text{ch},c} \in [0, n_{\max}^{\text{ch}}]$. The relaxed problem, an NLP, is solved using an NLP solver to obtain a locally optimal solution. In this paper, we use IPOPT (through CasADi) to solve this relaxed NLP.
2. The continuous solution $\{n_k^{\text{ch},c}\}_{k=0}^{T^{\text{plan}}-1} \in \mathbb{R}^{T^{\text{plan}}}$, resulting from Step 1, is processed by using Algorithms 2 and 3 to produce a transformed solution that is integer-valued, which is denoted by $\{n_k^{\text{ch},d}\}_{k=0}^{T^{\text{plan}}-1}$.
3. In Problem (33), the input $\{n_k^{\text{ch}}\}_{k=0}^{T^{\text{plan}}-1} = \{n_k^{\text{ch},d}\}_{k=0}^{T^{\text{plan}}-1}$ is fixed at the values obtained in Step 2, and the resulting NLP is solved again. The resulting solution is called the model-based sub-optimal solution (MBOC).

In the sequel, we will refer to a vector with non-negative integer components, $x \in \mathbb{Z}^n$, as an n-length *discrete signal*. For a discrete signal $x \in \mathbb{Z}^n$, the number of switches, N_{switch},

is defined as the number of times two consecutive entries differ: $N_{\text{switch}} := \sum_{i=1}^{n-1} I(x_i - x_{i+1})$, where $I(\cdot)$ is the indicator function: $I(0) = 0$ and $I(y) = 1$ for $y \neq 0$.

The continuous relaxation in Step 1 is inspired by branch and bound algorithms for solving MINLPs, since such a relaxation is the first step in branch and bound algorithms. However, a simple round-off-based method to convert the continuous variable $n^{\text{ch},c}$ to a discrete one leads to a high number of oscillations in the solution. This corresponds to frequent turning on and off of one or more chillers, which is detrimental to them.

Step 2 converts the continuous solution from Step 1 to a discrete signal, and involves multiple steps in itself. The first step is Algorithm 2, which filters the signal $n^{\text{ch},c}$ with a modified moving average filter with a two-hour window (corresponding to 12 samples with a 10-minute sampling period) and then rounding up the filtered value to the nearest integer. Thus by operating the moving average filter on $n^{\text{ch},c}$ one obtains a discrete signal for the chiller command $n^{\text{ch},f} = \text{moving\_average\_round}(n^{\text{ch},c})$.

Algorithm 2: $\mathbf{x_d} = \text{moving\_average\_round}(\mathbf{x})$

Input: Signal $\mathbf{x} \in \mathbb{Z}^n$, $w \in \mathbb{Z}^+$ (window length)

```
for i=1:w
    x_d[i] = ⌈mean(x[1 : i + w/2])⌉
end
for i = w/2 + 1 : n − w/2
    x_d[i] = ⌈mean(x[i − w/2 : i + w/2])⌉
end
for i = n − w/2 + 1 : n
    x_d[i] = ⌈mean(x[i − w/2 : end])⌉
end
```

Output: Discrete signal $\mathbf{x}_d$

The rounding moving average filter typically does not reduce the switching frequency sufficiently. This is why an additional step, Algorithm 3, described below, is used to operate on this signal and produce the output $n^{\text{ch},d} := \text{reduce\_switching}(n^{\text{ch},f})$ that has fewer switches.

The need for Step 3 is that the chiller command $\{n^{\text{ch},d}\}$ at the end of the second step, together with other variables in the solution vector from Step 1, may violate some constraints of the optimization problem (33). Even if $\{x^p_{k+1}\}$ and $\{n^{\text{ch},d}\}$ are feasible, the resulting control commands may not track the cooling load adequately. Step 3 ensures a feasible solution and improves tracking.

Forecasts: Implementation requires the availability of the forecasts of disturbance w^p_k, i.e., cooling load reference and electricity price, over the next planning horizon. There is a large literature on estimating and/or forecasting loads for buildings and for real-time electricity prices; see [45–47] and references therein. The forecast of T^{oawb}_k is available from National Weather Service [48]. We therefore assume the forecasts of the three disturbance signals, $\dot{q}^{\text{L,ref}}_k$, T^{oawb}_k, and ρ_k, are available to the MPC controller at each k.

Algorithm 3: $\mathbf{x}_{rs} = \text{reduce\_switching}(\mathbf{x})$

Input: Discrete signal $\mathbf{x} \in \mathbb{Z}^n$ and $w \in \mathbb{Z}^+$ (window length)

1: Obtain indices of the entries of $\mathbf{x}$ that are not to be changed, **index_freezed**, as follows:

```
Initialize index_freezed = zeros(n,1) # Array of dimension n with all entries zero
for i = 1 : n
    if N_switch(x[i − w : i]) = 0
        index_freezed[i] ← 1
    end
```

2: Initialize $\mathbf{x}_{rs}$: $\mathbf{x}_{rs}[i] = \mathbf{x}[i]$ for each i such that **index_freezed**[i]=1.

3: For each i in **index_freezed** which is 0:

Find all the consecutive 0 entries till the next 1. Let these indices be I^i_s, and define $y_i = \lceil mean(\mathbf{x}[I^i_s]) \rceil$.
Set $\mathbf{x}_{rs}[j] \leftarrow y_i$ for every $i \in I^i_s$.
Set **index_freezed**$[I^i_s] \leftarrow 1$

end

Output: $\mathbf{x}_{rs}$

5 Rule-based Baseline Controller

In order to evaluate the performances of the RL and MPC controllers, we will compare them to a rule-based baseline controller (BL). The proposed baseline controller is designed to utilize the TES and time-varying electricity prices (to the extent possible with heuristics) to reduce energy costs. The RL controller and baseline controller have the same information about the price: the current price ρ_k and a backward moving average $\bar{\rho}_k$. At each timestep k, the baseline controller determines the control command $u_k = [\dot{m}^{\text{lw}}_k, \dot{m}^{\text{tw}}_k, n^{\text{ch}}_k, \dot{m}^{\text{cw}}_k, \dot{m}^{\text{oa}}_k]^T$ following the procedure shown in Figure 4. The flowcharts are elaborated in [34] and briefly explained in Section 5.1 and 5.2. The subscript "sat" indicates the variable is saturated at its upper or lower bounds; the numerical values of the bounds used in simulations are shown in Table 1. For estimating the outputs under nominal conditions and the time-dependent bounds, please refer to [34].

5.1 For chilled water loop

1. At time step k, n^{ch}_k, $\dot{m}^{\text{lw}}_k$ and $\dot{m}^{\text{tw}}_k$ are initialized to n^{ch}_{k-1}, $\dot{m}^{\text{lw}}_{k-1}$ and $\dot{m}^{\text{tw}}_{k-1}$.
2. The BL controller increases or decreases $\dot{m}^{\text{tw}}_k$ by a fixed amount (10 kg/sec.) if ρ_k is 5% lower or higher than $\bar{\rho}_k$ in order to take advantage of time-varying electricity price.
3. The BL controller estimates $T^{\text{lw,r}}_{k+1}$, $T^{\text{chw,s}}_{k+1}$, S^{twc}_{k+1} under the assumption of $\dot{m}^{\text{bp}} = 0$ and $\dot{q}^{\text{ch}}_k = n^{\text{ch}}_k \dot{q}^{\text{ch}}_{\text{indv}}$. If $T^{\text{lw,r}}_{k+1}$, $T^{\text{chw,s}}_{k+1}$, S^{twc}_{k+1} are within their bounds, the current control command for the chilled water loop is executed. Other-

Table 1: Simulation Parameters

Para	Unit	value	Para	Unit	value
t_s	min	10	$\frac{t_s T^{\text{plan}}}{60}$	hours	24
$\frac{\tau t_s}{60}$	hours	4	$\frac{w t_s}{60}$	hours	2
$n^{\text{ch}}_{\max}$	N/A	7	$\dot{m}^{\text{tw}}_{\max/\min}$	$\frac{\text{kg}}{\text{sec}}$	30/-30
$\dot{m}^{\text{lw}}_{\max/\min}$	$\frac{\text{kg}}{\text{sec}}$	350/20	$\dot{m}^{\text{cw}}_{\max/\min}$	$\frac{\text{kg}}{\text{sec}}$	300/20
$S^{\text{twc/ tww}}_{\max}$	N/A	0.95	$S^{\text{twc/ tww}}_{\min}$	N/A	0.05
$\dot{m}^{\text{indv}}$	$\frac{\text{kg}}{\text{sec}}$	50	$\dot{q}^{\text{ch}}_{\text{indv}}$	kW	1046
$T^{\text{lw,r}}_{\max/\min}$	$^\circ$C	15/5	$T^{\text{chw,s}}_{\max/\min}$	$^\circ$C	10/5
$T^{\text{cw,r}}_{\max/\min}$	$^\circ$C	40/25	$\dot{m}^{\text{oa}}_{\max/\min}$	$\frac{\text{kg}}{\text{sec}}$	1.25/1
$T^{\text{cw,s}}_{\text{set}}$	$^\circ$C	29	$T^{\text{chw,s}}_{\text{set}}$	$^\circ$C	7

wise, the controller repeatedly increases/decreases $\dot{m}^{\text{lw}}_k$ and $\dot{m}^{\text{tw}}_k$ by a fixed amount (10 kg/sec), and n^{ch}_k by 1 until $T^{\text{lw,r}}_{k+1}$, $T^{\text{chw,s}}_{k+1}$, and S^{twc}_{k+1} are within their bounds. Since $\dot{m}^{\text{lw}}_k + \dot{m}^{\text{tw}}_k$ determines the minimum required n^{ch}_k, the final n^{ch}_k is readjusted to meet the minimum required n^{ch}_k.

5.2 For cooling water loop

1. $\dot{m}^{\text{cw}}_k$ and $\dot{m}^{\text{oa}}_k$ are initialized to $\dot{m}^{\text{cw}}_{k-1}$ and $\dot{m}^{\text{oa}}_{k-1}$.
2. The BL controller estimates $T^{\text{cw,r}}_{k+1}$ by assuming a fixed fraction of electric power consumed by chillers is added into the cooling water loop. This fraction is to be estimated from historical data. If $T^{\text{cw,r}}_{k+1}$ is above/below its bound, $\dot{m}^{\text{cw}}_k$ is increased/decreased by a fixed amount (20 kg/sec) repeatedly until $T^{\text{cw,r}}_{k+1}$ is within its bound.
3. Once $\dot{m}^{\text{cw}}_k$ is determined, the capacity of cooling tower $\dot{q}^{\text{ct}}_{\text{UB},k}$ and the required cooling $\dot{q}^{\text{ct}}_{\text{set},k}$ that cools down $T^{\text{cw,r}}_k$ to $T^{\text{cw,s}}_{\text{set}}$ is computed. If $\dot{q}^{\text{ct}}_{\text{set},k} \le \dot{q}^{\text{ct}}_{\text{UB},k} \le 1.1\dot{q}^{\text{ct}}_{\text{set},k}$, then the current control command for the cooling water loop is executed. If $\dot{q}^{\text{ct}}_{\text{UB},k} < \dot{q}^{\text{ct}}_{\text{set},k}$ or $\dot{q}^{\text{ct}}_{\text{UB},k} > 1.1\dot{q}^{\text{ct}}_{\text{set},k}$, $\dot{m}^{\text{oa}}_k$ is increased or decreased by a fixed amount (0.05 kg/sec.). Since $\dot{q}^{\text{ct}}_{\text{UB},k}$ depends on the ambient wet-bulb temperature T^{oawb}_k (illustrated in [34]), there can be a case that $\dot{q}^{\text{ct}}_{\text{UB},k}$ cannot satisfy $\dot{q}^{\text{ct}}_{\text{set},k} \le \dot{q}^{\text{ct}}_{\text{UB},k} \le 1.1\dot{q}^{\text{ct}}_{\text{set},k}$ even when $\dot{m}^{\text{oa}}_k$ is already at its bound. In this case, $\dot{m}^{\text{cw}}_k$ is varied by a fixed amount (20 kg/sec) repeatedly until $T^{\text{cw,r}}_{k+1}$ and $\dot{q}^{\text{ct}}_{\text{UB},k}$ are within their bounds.

6 Performance evaluation

6.1 Simulation setup

Simulations for closed-loop control with RL, MPC, and baseline controllers are performed for the week of Sept. 6-12, 2021, which we refer to as the *testing week* in the sequel. The weather data for the testing week is obtained from the Singapore data set described in Section 2.3. The real-time electricity price used is a scaled version of PJM's locational margin price for the same week [44]. Other relevant simulation parameters are located in Table 1. There is no plant-model mismatch in the MPC simulations. In particular, since the forecasts of disturbance signals are available in practice (see the discussion at the end of Section 4), in the simulations the MPC controller is provided with error-free forecasts in the interest of simplicity.

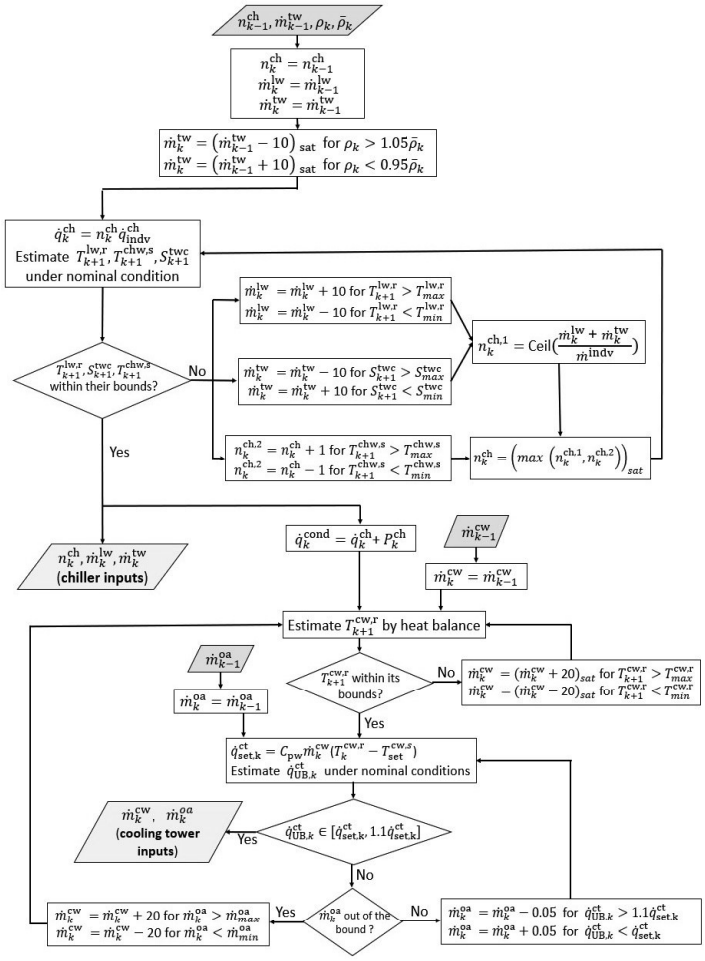


Fig. 4: Baseline Controller

.

We emphasize that the closed-loop results with the RL controller presented here are "out-of-sample" results, meaning the external disturbance w_k (weather, cooling load, and electricity price) used in the closed-loop simulations are different from those used in training the RL controller.

Four performance metrics are used to compare the three controllers. The first is the energy cost incurred. The second is the Root Mean Square Error (RMSE) in tracking the cooling load reference:

$$e_{RMSE} := \left(\frac{1}{N_{\text{sim}} - 1} \sum_{k=1}^{N_{\text{sim}}} (\dot{q}^{\text{L,ref}}_k - \dot{q}^{\text{L}}_k)^2 \right)^{\frac{1}{2}}, \qquad (34)$$

where N_{sim} is the duration for which closed-loop simulations are carried out, which in this paper is 1008 (corresponding to a week: $7 \times 24 \times 6$). The third is the number of chiller

switches over the simulation period:

$$n^{\text{ch}}_{\text{switch}} := \sum_{k=1}^{N_{\text{sim}}-1} |n^{\text{ch}}_{k+1} - n^{\text{ch}}_{k}|. \tag{35}$$

Fast cycling decreases the life expectancy of a chiller greatly. The fourth is control computational time during closed-loop simulations.

6.2 Numerical Results and Discussion

A summary of performance comparisons from the simulations is shown in Table 2. All three controllers meet the cooling load adequately (more on this later), and both the RL and MPC controllers reduce energy cost over the baseline by about the same amount (16.8% for RL vs. 17.8% for MPC). These savings are comparable with those reported in the literature for MPC with MILP relaxation and RL.

In terms of tracking the reference load, both RL and MPC again perform similarly while the baseline controller performs the best in terms of the standard deviation of tracking error; see Figure 5 and Table 2. The worst tracking RMSE is 61 kW, which is a small fraction of the mean load (1313 kW). Thus the tracking performance is considered acceptable for all three controllers. The fact that the baseline performs the best in tracking the cooling load is perhaps not surprising since it is designed primarily to meet the required load and keep chiller switching low, with energy cost a secondary consideration.

In terms of chiller switches, the RL controller performs the worst; see Table 2. This is not surprising because no cost was assigned to higher switching in its design. The MPC performs the best in this metric, again most likely since keeping switching frequency low was an explicit consideration in its design. Ironically, this feature was introduced into the MPC controller after an initial design attempt without it, which led to a high switching frequency.

In terms of real-time computation cost, the baseline performs the best, which is not surprising since no optimization is involved. The RL controller has two orders of magnitude lower computation cost compared to MPC. The computation time for all controllers is well within the time budget since control commands are updated every 10 minutes.

Deeper look: Simulations are done for a week, but the plots below show only two days to avoid clutter. The cost savings by RL and MPC controller come from their ability to use the TES to shift the peak electric demand to periods of low price better than that of the baseline controller; see Figure 6. The MPC controller has the knowledge of electricity price along the whole planning horizon, and thus achieves the most savings. The cause for the cost-saving differences between BL and RL controllers is that the RL controller learns the variation in the electricity price well, or at least better than the BL controller. This can be seen in Figure 7. The RL controller always discharges the TES (S^{twc} drops) during the peak electricity price while the baseline controller sometimes cannot do so because the volume of cold water is already at its minimum bound. The BL controller discharges the TES as soon as the electricity price rises, which may result in insufficient cold water stored in the TES when the electricity price reaches its maximum. While both the RL and BL controllers are forced to use the same price information (current and a backward moving average), the rule-based logic in the baseline controller cannot use that information as effectively as RL.

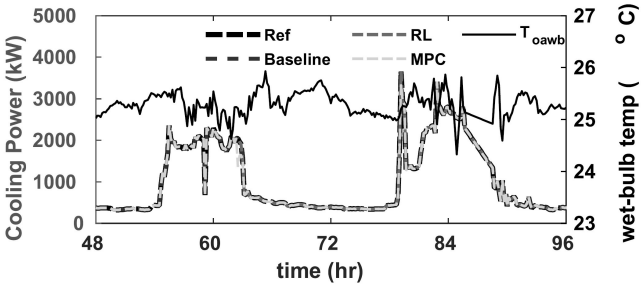


Fig. 5: Load tracking performances of the MPC, RL, and BL controllers: The "Ref" is cooling required $\dot{q}_k^{\text{L,ref}}$.

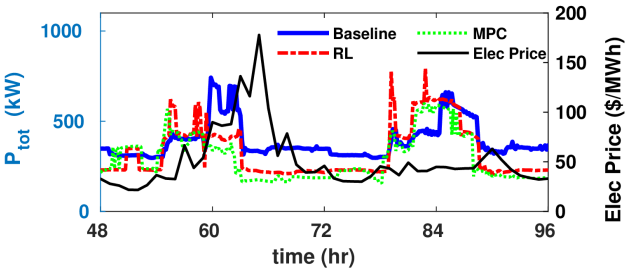


Fig. 6: Power consumption vs. real-time electricity price for the MPC, RL, and BL controllers.

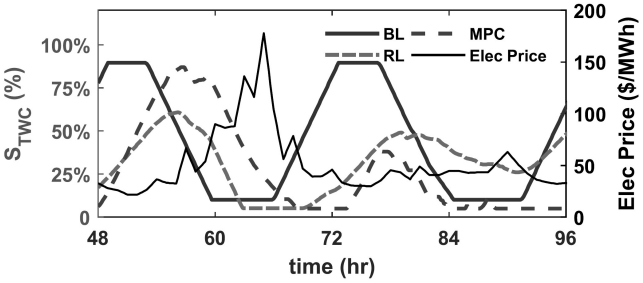


Fig. 7: TES cold water volume vs. real-time electricity price for the MPC, RL, and BL controllers.

An alternate view of this behavior can be obtained by looking at the times when the chillers are turned on and off,

Table 2: Comparison of RL, MPC, and baseline controllers (for a week-long simulation).

	Total cost ($)	e_{RMSE} (kW)	No. of switches	Control computation time (sec, $\mu \pm \sigma$)
Baseline	3308	4.14e-4	45	8.9e-5 ± 3.9e-4
RL	2752	1.85	114	0.32 ± 0.01
MPC	2719	61.38	65	27.33 ± 5.99

since using chillers costs much more electricity than using the TES, which only needs a few pumps. We can see from Figure 8 that all controllers shift their peak electricity demand to the times when electricity is cheap. But the rule-based logic of the BL controller is not able to line up electric demand with the low price as well as the RL and MPC controllers do.

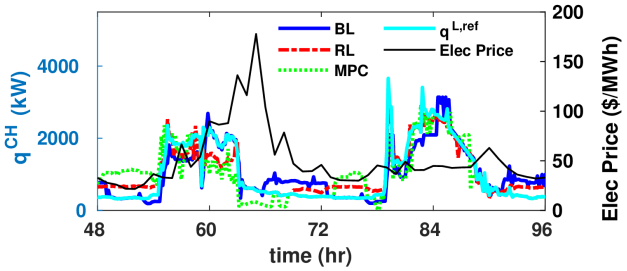


Fig. 8: Required cooling load vs. real-time electricity price for the MPC, RL, and BL controllers.

Another benefit of the RL controller is that it typically activates fewer chillers than the BL controller, though the cost of running active chillers is not incorporated in the cost function; see Figure 9. This effect may increase the life expectancy of the DCEP.

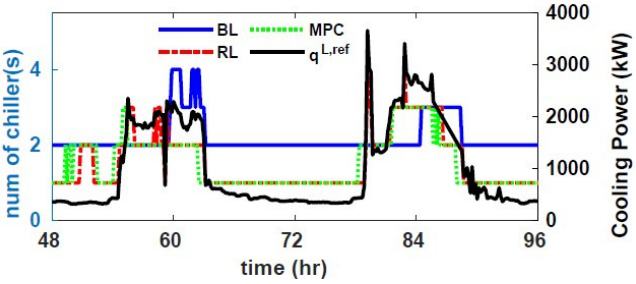


Fig. 9: Number of active chillers vs. real-time electricity price for the MPC, RL, and BL controllers.

7 Under the hood of the RL controller

More insights about why the learned policy works under various conditions can be obtained by taking a closer look at the design choices made for the RL controller. All these choices were the result of considerable trial and error.

Choice of basis functions The choice of basis to approximate the Q-function is essential to the success of the RL controller. It defines the approximate Q-function, and consequently the policy (31). Redundant basis functions can lead to overfitting, which causes poor out-of-sample performance of the policy. We avoid this effect by selecting a reduced quadratic basis, which are the 36 unique non-zero entries shown in Figure 10. Another advantage of reducing the number of basis functions is that it reduces the number of parameters to learn, as training effort increases dramatically with the number of parameters to learn.

The choices for the basis were based on physical intuition about the DCEP. First, basis functions can be simplified by dropping redundant states. One example is S^{tww}. Since S^{twc} and S^{tww} are dual terms: $S^{\mathrm{twc}} + S^{\mathrm{tww}} = 1$, so one of them can be dropped. Considering that the S^{twc} reflects the amount of cooling saved in the TES, we dropped S^{tww}. Another example is the term T^{tww}, which is dropped since it is bounded by $T^{\mathrm{lw,r}}$ which is already included in the basis function. Second, if two terms have a strong causal or dependent relationship, e.g., $\dot{m}^{\mathrm{lw}}$ and $T^{\mathrm{lw,r}}$ then the corresponding quadratic term $\dot{m}^{\mathrm{lw}} T^{\mathrm{lw,r}}$ should be selected as an element of the basis. Third, if two terms have minimal causal or dependent relationship, e.g., $\dot{m}^{\mathrm{oa}}$ and $\dot{m}^{\mathrm{tw}}$ (they are from different equipment and water loops), then the corresponding quadratic term $\dot{m}^{\mathrm{oa}} \dot{m}^{\mathrm{tw}}$ should not be selected as an element of the basis.

Choice of States Exogenous disturbances have to be included into the RL states to make the controller work under various cooling load, electricity price, and weather trajectories *that are distinct from what is seen during training*. Without this feature, the RL controller will not be applicable in the field.

Convergence of the learning algorithm: The learning algorithm appears to converge in training, meaning, $|\theta_k - \theta_{k-1}|$ is seen to reduce as the number of training epochs k increases; see Figure 11. This convergence should not be confused with convergence to any meaningful optimal policy. The policy learned in the 40th iteration can be a better-performing controller than the policy obtained in 50th iteration. We believe the proximal gradient type method used in learning helps in the parameters not diverging, but due to the same reason it may prevent the parameters from converging to a far away optima. This trade-off is necessary: our initial attempts without the damping proximal term were not successful in learning anything useful. As a result, after

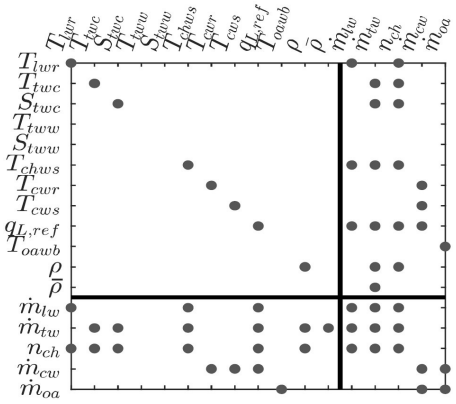


Fig. 10: Sparsity pattern of the matrix P_θ appearing in (29).

a few policy improvement iterations, every new policy obtained had to be tested by running a closed-loop simulation to assess its performance. The best performing one was selected as "the RL controller", which happened to be the 26th one.

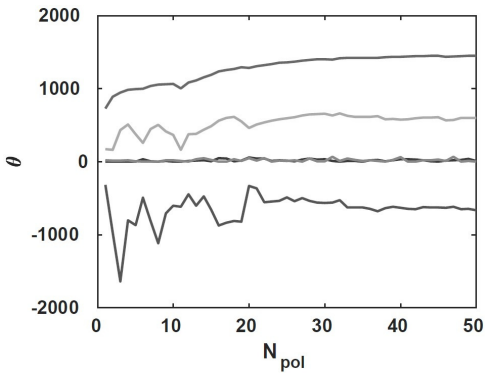


Fig. 11: Values of θ vs. policy iteration index. Only five θ's are shown to avoid clutter.

Numerical considerations for training: Training of the RL controller is an iterative task that requires trying many various configurations of the parameters appearing in Table 1. In particular, we found the following considerations useful.

1. If the value of κ is too small, the controller will not learn to track the load $\dot{q}_k^{L,ref}$. On the other hand, if κ is too large the controller will not save energy cost. The chosen κ in Section 3.3.5 is determined by trial-and-error.
2. The condition number of (23) significantly affects the performance of Algorithm 1. However, the relative magnitudes of state and input values are very different, for example, $\dot{q}^L \in [300, 4000]$ (kW) and $S^{TWC} \in [0.05, 0.95]$, which makes the condition number of (23) extremely large. Therefore, we normalize all magnitudes of state and input values with their average values. With appropriate scaling of the states/inputs, we reduced the magnitude of the condition number from 1×10^{20} to 1×10^{3}.

8 Conclusion

The proposed MPC and RL controllers are able to reduce energy cost significantly, around 17% in a week-long simulation, over the rule-based baseline controller. Apart from the dramatically lower real-time computationally cost of the RL controller compared to the MPC, load tracking and energy cost-saving performances of the two controllers are similar. This similarity in performance is somewhat surprising. Though both controllers are designed to be approximations of the same intractable infinite horizon problem, there are nonetheless significant differences between them, especially the information the controllers have access to and the objectives they are designed to minimize. It should be noted that the MPC controller has a crucial advantage over the RL controller in our simulations: the RL controller has to implicitly learn to forecast disturbances while the MPC controller is provided with error-free forecasts. How much will MPC's performance degrade in practice due to inevitable plant-model mismatch is an open question.

Existing work on RL and on MPC tend to lie in their own silos, with comparisons between them for the same application being rare. This paper contributes to such comparisons for a particular application: control of DCEPs. Much more remains to be done, such as an examination of robustness to uncertainties.

There are several other avenues for future work. One is to explore non-linear bases, such as neural networks, for designing an RL controller. Another is to augment the state space with additional signals, especially with forecasts, which might improve performance. Of course, such augmentation will also increase the cost and complexity of training the policy. Another avenue for improvement in the RL controller is to reduce the number of chiller switches. In this paper, all the chillers are considered to be the same. An area of improvement is to extend heterogeneous chillers with distinct performance curves, for both RL and MPC. On the MPC front, an MILP relaxation is a direction to pursue in the future.

References

[1] U.S. Energy Information Administration, 2012. Commercial buildings energy consumption survey (CBECS): Overview of commercial buildings, 2012. Tech. rep., Energy information administration, Department of Energy, U.S. Govt., December.

[2] Pacific Gas and Electric Company, 1997. Thermal energy storage strategies for commercial HVAC systems. Application note.

[3] Hydeman, M., and Zhou, G., 2007. "Optimizing chilled water plant control". *ASHRAE journal,* **49**, 6, p. 45=54.

[4] Teleke, S., Baran, M. E., Bhattacharya, S., and Huang, A. Q., 2010. "Rule-based control of battery energy storage for dispatching intermittent renewable

sources". *IEEE Transactions on Sustainable Energy,* **1**(3), pp. 117–124.

[5] Tam, A., Ziviani, D., Braun, J., and Jain, N., 2018. "A generalized rule-based control strategy for thermal energy storage in residential buildings". In 5th International Conference on High Performance Buildings.

[6] Pinamonti, M., Prada, A., and Baggio, P., 2020. "Rule-based control strategy to increase photovoltaic self-consumption of a modulating heat pump using water storages and building mass activation". *Energies,* **13**(23).

[7] Lee, K.-H., Joo, M.-C., and Baek, N.-C., 2015. "Experimental evaluation of simple thermal storage control strategies in low-energy solar houses to reduce electricity consumption during grid on-peak periods". *Energies,* **8**(9), pp. 9344–9364.

[8] Schibuola, L., Scarpa, M., and Tambani, C., 2015. "Demand response management by means of heat pumps controlled via real time pricing". *Energy and Buildings,* **90**, pp. 15–28.

[9] Ma, Y., Kelman, A., Daly, A., and Borrelli, F., 2012. "Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments". *IEEE Control Systems Magazine,* **32**(1), feb., pp. 44 –64.

[10] Cole, W. J., Edgar, T. F., and Novoselac, A., 2012. "Use of model predictive control to enhance the flexibility of thermal energy storage cooling systems". In 2012 American Control Conference (ACC), pp. 2788–2793.

[11] Touretzky, C. R., and Baldea, M., 2014. "Integrating scheduling and control for economic MPC of buildings with energy storage". *Journal of Process Control,* **24**(8), pp. 1292–1300. Economic nonlinear model predictive control.

[12] Zabala, L., Febres, J., Sterling, R., López, S., and Keane, M., 2020. "Virtual testbed for model predictive control development in district cooling systems". *Renewable and Sustainable Energy Reviews,* **129**, p. 109920.

[13] Risbeck, M. J., Maravelias, C. T., Rawlings, J. B., and Turney, R. D., 2017. "A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment". *Energy and Buildings,* **142**, pp. 220–235.

[14] Rawlings, J. B., Patel, N. R., Risbeck, M. J., Maravelias, C. T., Wenzel, M. J., and Turney, R. D., 2018. "Economic MPC and real-time decision making with application to large-scale HVAC energy systems". *Computers & Chemical Engineering,* **114**, pp. 89–98.

[15] Patel, N. R., Risbeck, M. J., Rawlings, J. B., Maravelias, C. T., Wenzel, M. J., and Turney, R. D., 2018. "A case study of economic optimization of HVAC systems based on the Stanford University campus airside and waterside systems". In 5th International High Performance Buildings Conference.

[16] Deng, K., Sun, Y., Li, S., Lu, Y., Brouwer, J., Mehta, P. G., Zhou, M., and Chakraborty, A., 2015. "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming". *IEEE Transactions on Automation Science and Engineering,* **12**(2), pp. 565–579.

[17] Kim, D., Wang, Z., Brugger, J., Blum, D., Wetter, M., Hong, T., and Piette, M. A., 2022. "Site demonstration and performance evaluation of mpc for a large chiller plant with tes for renewable energy integration and grid decarbonization". *Applied Energy,* **321**, p. 119343.

[18] Manoharan, P., Venkat, M. P., Nagarathinam, S., and Vasan, A., 2021. "Learn to chill: Intelligent chiller scheduling using meta-learning and deep reinforcement learning". In Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '21, Association for Computing Machinery, p. 21–30.

[19] Qiu, S., Li, Z., Li, Z., and Zhang, X., 2020. "Model-free optimal chiller loading method based on q-learning". *Science and Technology for the Built Environment,* **26**(8), pp. 1100–1116.

[20] Qiu, S., Li, Z., Fan, D., He, R., Dai, X., and Li, Z., 2022. "Chilled water temperature resetting using model-free reinforcement learning: Engineering application". *Energy and Buildings,* **255**, p. 111694.

[21] Nagarathinam, S., Menon, V., Vasan, A., and Sivasubramaniam, A., 2020. "Marco-multi-agent reinforcement learning based control of building hvac systems". In Proceedings of the Eleventh ACM International Conference on Future Energy Systems, e-Energy '20, Association for Computing Machinery, p. 57–67.

[22] Campos, G., El-Farra, N. H., and Palazoglu, A., 2022. "Soft actor-critic deep reinforcement learning with hybrid mixed-integer actions for demand responsive scheduling of energy systems". *Industrial And Engineering Chemistry Research*.

[23] Ahn, K. U., and Park, C. S., 2020. "Application of deep q-networks for model-free optimal control balancing between different hvac systems". *Science and Technology for the Built Environment,* **26**(1), pp. 61–74.

[24] Qiu, S., Li, Z., Li, Z., Li, J., Long, S., and Li, X., 2020. "Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation". *Energy and Buildings,* **218**, p. 110055.

[25] Henze, G. P., and Schoenmann, J., 2003. "Evaluation of reinforcement learning control for thermal energy storage systems". *HVAC&R Research,* **9**(3), pp. 259–275.

[26] Liu, S., and Henze, G. P., 2007. "Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory". *ASME Journal of Solar Energy Engineering,* **129**, May, p. 215–225.

[27] Lu, F., Mehta, P. G., Meyn, S. P., and Neu, G., 2021. "Convex Q-Learning". In 2021 American Control Conference (ACC), IEEE, pp. 4749–4756.

[28] Lagoudakis, M. G., and Parr, R., 2003. "Least-squares policy iteration". *The Journal of Machine Learning Research,* **4**, pp. 1107–1149.

[29] Gibney, E., 2017. "Self-taught AI is best yet at strategy game Go". *Nature*.

[30] Banjac, G., and Lygeros, J., 2019. "A data-driven policy iteration scheme based on linear programming". In 2019 IEEE 58th Conference on Decision and Control (CDC), IEEE, pp. 816–821.

[31] Luo, B., Liu, D., Wu, H.-N., Wang, D., and Lewis, F. L., 2017. "Policy gradient adaptive dynamic programming for data-based optimal control". *IEEE Transactions on Cybernetics,* **47**(10), pp. 3341–3354.

[32] Fan, C., Hinkelman, K., Fu, Y., Zuo, W., Huang, S., Shi, C., Mamaghani, N., Faulkner, C., and Zhou, X., 2021. "Open-source modelica models for the control performance simulation of chiller plants with water-side economizer". *Applied Energy,* **299**, p. 117337.

[33] Guo, Z., Coffman, A. R., and Barooah, P., 2022. "Reinforcement learning for optimal control of a district cooling energy plant". In American Control Conference (ACC), pp. 3329–3334.

[34] Guo, Z., Chaudhari, A., Coffman, A. R., and Barooah, P., 2023. "Optimal control of District Cooling Energy Plant with reinforcement learning and MPC". *arXiv preprint 2310.03814*.

[35] Yu, F., and Chan, K., 2008. "Optimization of water-cooled chiller system with load-based speed control". *Applied Energy,* **85**(10), pp. 931–950.

[36] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., 2019. "CasADi: a software framework for nonlinear optimization and optimal control". *Mathematical Programming Computation,* **11**(1), Mar, pp. 1–36.

[37] Wächter, A., and Biegler, L. T., 2006. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical Programming,* **106**(1), Mar, pp. 25–57.

[38] American Society of Heating Refrigerating and Air Conditioning Engineers, 2002. *ASHRAE Guideline14-2002 for Measurement of Energy and Demand Savings*. ASHRAE, Atlanta, GA, ch. ANNEX E.3, p. 151.

[39] Braun, J. E., and Diderrich, G. T., 1990. "Near-optimal control of cooling towers for chilled-water systems". *ASHRAE Transactions (American Society of Heating, Refrigerating and Air-Conditioning Engineers),* **96:2**, Jan.

[40] Miller, C., 2014. united-world-colledge-open-data. https://github.com/buds-lab/united-world-college-open-data. [Online].

[41] Miller, C., Nagy, Z., and Schlueter, A., 2014. "A seed dataset for a public, temporal data repository for energy informatics research on commercial building performance". In Proceeding Of 3rd Conf. on Future Energy Business and Energy Informatics, Rotterdam, Netherlands.

[42] Sutton, R., and Barto, A., 2018. *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, Cambridge, MA.

[43] Grant, M., and Boyd, S., 2011. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, Feb.

[44] PJM data miner, 2022. PJM Interconnection Real-Time Hourly LMPs. https://www.pjm.com/markets-and-operations/etools/data-miner-2. [Online, accessed 2022-10-02].

[45] Braun, J. E., and Chaturvedia, N., 2002. "An inverse gray-box model for transient building load prediction". *HVAC&R Research,* **8**, pp. 73–99.

[46] Guo, Z., Coffman, A. R., Munk, J., Im, P., Kuruganti, T., and Barooah, P., 2021. "Aggregation and data driven identification of building thermal dynamic model and unmeasured disturbance". *Energy and Buildings,* **231**, January, p. 110500: 9 pages.

[47] Oldewurtel, F., Ulbig, A., Parisio, A., Andersson, G., and Morari, M., 2010. "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), pp. 1927–1932.

[48] National Weather Service and NOAA, 2022. https://www.weather.gov/. [Online, accessed 2022-11-01].