INVESTIGATION OF REINFORCEMENT LEARNING FOR BUILDING THERMAL MASS CONTROL

Simeng Liu and Gregor P. Henze, Ph.D., P.E.
University of Nebraska – Lincoln, Architectural Engineering
1110 South 67th Street, Peter Kiewit Institute, Omaha, Nebraska 68182-0681 U.S.A.

ABSTRACT

This paper describes a simulation-based investigation of machine-learning control for the supervisory control of building thermal mass. Model-free reinforcement learning control is investigated for the operation of electrically driven chilled water systems in heavy-mass commercial buildings. The reinforcement learning controller learns to precool the building at night before the onset of occupancy based on the feedback it receives from past control actions. The learning agent interacts with its environment by commanding the global zone temperature setpoints and extracts cues about the environment solely based on the reinforcement feedback it receives, which in this study is the monetary cost of each control action. No prediction or system model is required. Over time and by exploring the environment, the reinforcement learning controller establishes a statistical summary of plant operation, which is continuously updated as operation continues. The controller learns to account for the time-dependent cost of electricity, the availability of passive thermal storage inventory, and weather conditions. This study revealed that learning control is a feasible methodology to find a near-optimal setpoint profile for exploiting the passive building thermal storage capacity. The freedom from a building model makes it especially attractive in real-time control problems, and theoretically it can reach the “true” optimum eventually, no matter what building it is dealing with, if only the environment could be sampled for an infinite period of time. The analysis showed that the learning controller is affected by the dimension of the action and state space, the utility rate differentials between on- and off-peak, learning rate and several other factors. Moreover, learning speed is relatively low when dealing with problems with large state space and action space.

INTRODUCTION

The motivation of this research stems from a research project funded by the U.S. Department of Energy that investigates predictive optimal control of active and passive building thermal storage inventory. The goal of this project is to develop a model-based supervisory building controller to exploit the potential of both active and passive building thermal storage capacity. In the first two years of this study, both numerical analysis and laboratory experiment have demonstrated cost saving potential when applying optimal control to the utilization of active and passive building thermal storage inventory. However, modeling complexity and inaccuracy of this model-based approach have been revealed as well. Meanwhile, numerous application of reinforcement learning control to engineering problems provides a new direction to tackle this control problem in a more efficient manner. To this end, an investigation has been carried out to determine the feasibility and merits of learning control applied to the control of building active and passive thermal storage inventory. As an initial step, a simulation environment in Simulink was developed to investigate the feasibility of applying reinforcement learning control to passive building thermal storage only.

REVIEW OF PAST WORK

Previous studies on building thermal mass utilization demonstrated the potential of peak cooling load reduction and associated electrical demand. However, cost savings vary widely among the published case studies (Rabl and Norford 1991; Conniff 1991; Morris et al. 1994; Keeney and Braun 1997). In a simulation study presented by Braun (1990), cost savings for a design day varied from 0-35% depending on system type and utility rate. Andresen and Brandemuehl (1992) demonstrated energy and cost savings potential by precooling the building structure, calling attention to the importance of the mass of furnishing which significantly affects the precooling strategy. Braun et al. (2001) developed a tool to evaluate different precooling strategies by comparing the HVAC utility costs in each application. Simulation studies have been carried out for selected locations, climates, and utility rate structures. A comparison showed cost savings varying from 40% at best to zero or even excess costs for some less favorable cases. In a review article on load control using building thermal mass, Braun (2003) concluded that the savings potential is very sensitive to the utility rates, building and plant characteristics, and weather conditions and occupancy schedule. The greatest cost savings were realized for the case of heavy construction, good part-load characteristics and low ambient temperature which enabled free cooling during night ventilation.

A simulation study was carried out by Henze et al. (2004a) to investigate the combined usage of both active and passive building thermal storage inventory. The term 'active' refers to a thermal energy storage (TES) system, which is either based on chilled water or ice, and 'passive' refers to the building mass itself including structure and furniture. The analysis uses a model-based approach to find the optimal global zone air temperature setpoint and TES operating commands. Substantial utilization of both storage media was observed using a perfect building model. However, in real-time applications, a mismatch between building model and the actual building can never be completely avoided. In a companion study by Liu and Henze (2004) on the impact of model accuracy on the quality of the optimization system, it was found that a mismatch of internal heat gain and building construction will significantly affect the optimal zone air temperature setpoint. It is desired to have a model calibration procedure in place to prevent large deviations of the model, but this will increase the computational cost and may also not be practical for real-time control. Meanwhile, even though the predictive optimal control always looks "forward" by relying on short-term forecasts, the knowledge gained from the past has been ignored. Consequently, the reinforcement learning control paradigm provides an alternative approach that eliminates the need for a model and is built entirely upon experience while avoiding the disadvantages of model-based control such as modeling complexity, inaccuracy, and need for prediction.

Although techniques of machine learning have been widely applied in many industries, the concept of learning control appears still new in the area of HVAC control. Kretchmar et al. (2001) employed reinforcement learning assisted by artificial neural networks to learn to improve multiple-input multiple-output (MIMO) control performance of a heating system within a stable environment guaranteed by robust control. Henze and Dodier (1997) investigated learning control of a grid-independent photovoltaic system consisting of a collector, battery storage, and a load. Q-learning, a model-free reinforcement learning algorithm was applied to optimize control performance of the system. Simulation analysis compared the performance between the conventional PV-priority control and the optimal control. Better performance was found by applying the reinforcement learning to optimize the operation of the system. Henze and Schoenmann (2003) applied reinforcement learning control to the optimization of active thermal energy storage systems. Though reinforcement learning control proved sensitive to the selection of state variables, level of discretization, and learning rate, it effectively learns a difficult task of controlling thermal energy storage and displays good performance. The cost savings compare favorably with conventional cool storage control strategies, but do not reach the level of predictive optimal control.

INTRODUCTION TO LEARNING CONTROL

Reinforcement learning control stems from the development of two different disciplines, which are psychology and optimal control (Sutton and Barto 1998). It is defined as a process in which an agent learns and improves its behavior by trail-and-error interactions with a dynamic environment to achieve a long-term goal. Algorithms have been well applied to solve sequential decision making problems.

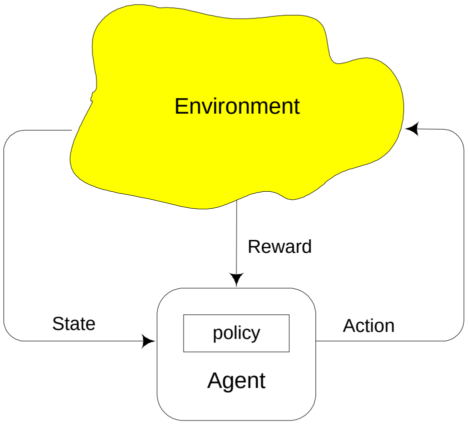


Figure 1: Schematic of sequential decision making problems

Figure 1 presents a schematic of a typical sequential decision making problem. At each time step or stage *t*, the agent will execute an action a_t selected from action space *A* according to a policy; the environment will then be transmitted from state s_t to s_{t+1}, along with a feedback signal r_t, which is defined as reward or reinforcement. The goal of the sequential decision making problem is to find an optimal policy that maximizes the accumulated rewards in the future starting from a particular state

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \qquad (1)$$

A policy is the agent's selection of actions in a given state, i.e., it is a mapping between state and actions, $\pi : S \rightarrow A$. In Eq.1, γ is introduced as a discounting factor, which is used to weight the future rewards. One of the most widely applied approaches to solve sequential decision making problems is dynamic programming (Bellman 1957), which assumes there is an explicit model of the environment available. The transition of the environment state is defined as transition probability function

$$P_{ss'}^{a} = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$

The expression of the transition in terms of a probability density function allows for random effects to be considered. Similarly, the instant reward is also defined as a function of current state, current action and next state

$$r_{ss'}^{a} = E(r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s').$$

In dynamic programming, the policy is usually defined as a probability function *π(s,a)* of taking action *a* when in state *s*. Given a policy *π* and certain state, the *state value function* is defined as

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}\{R_t \mid s_t = s\} \\ &= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}. \end{aligned} \tag{2}$$

Similarly, we define the value of taking action *a* in state *s* according to a policy as

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}\{R_t \mid s_t = s, a_t = a\} \\ &= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\} \end{aligned} \tag{3}$$

The goal of the optimization is to find the optimal policy to maximize the state value function. Bellman's principle of optimality states that whatever the initial state and action are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. Bellman's optimality equation can be written as

$$V^{*}(s) = \max_{a} \sum_{s'} P_{ss'}^{a}[r_{ss'}^{a} + \gamma V^{*}(s')] \tag{4}$$

Based on the recursive nature of dynamic programming, reinforcement learning algorithms were developed to deal with problems when there is no explicit model available of the environment. The only access of the agent to the information about the environment is via the direct interactions with the environment including perception of the state and the reward. The value function is progressively estimated by continuous sampling values associated with a particular policy. This so-called Monte Carlo method overcomes the necessity of the transition probability function $P_{ss'}^{a}$. The real power of reinforcement learning lies in the fact that the agent does not have to wait until the terminal cost is incurred to adjust its policy (Kaelbling et al. 1996). This is realized by the *temporal difference* (TD) method introduced by Sutton (Sutton 1988), in which the value function for a particular state is updated based on the previous estimation, immediate reward and the estimated value of the next state. The simplest TD method, known as *TD(0)*, is defined as:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{5}$$

where α is defined as learning rate.

The Q-Learning Algorithm

Developed by Watkins (Watkins 1992), the *Q*-Learning algorithm is one of the most widely applied reinforcement learning algorithms. The quality value $Q(s_t, a_t)$ associated with a state-action pair is introduced, which can be considered an estimate of the cumulative rewards by performing action *a* in state *s*. Bellman's optimality equation can be formulated as

$$Q(s,a) = \sum_{s'} P_{ss'}^{a}[r_{ss'}^{a} + \gamma \max_{a'} Q(s',a')] \tag{6}$$

The Q function can be learned by the algorithm $TD(0)$, which samples the transition probability through repeated actions. Given the state *s* the agent just visited, the selected action *a*, the next state s', and the reinforcement $r(s,a,s')$, the value function can be updated according to:

$$\begin{aligned} Q(s,a) &\leftarrow Q(s,a) + \\ &\alpha\left[r(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \end{aligned} \tag{7}$$

An important issue for reinforcement learning algorithm is the trade-off between exploration and exploitation. The online sampling algorithm relies on repeatedly visiting every state-action pair to update the associated Q-value by means of exploration process. On the other hand, to control effectively, the agent should pick an action for control, e.g., the action with the highest *Q* value, which is known as a greedy exploitation policy. Algorithms have been introduced to balance exploration and exploitation. One of simplest approaches is called *ε*-greedy method, in which, instead of being greedy all of the time, the agent takes non-greedy exploratory actions with a probability of *ε*. Another category of methods is called *softmax* action selection methods, among which, Gibbs or Boltzmann distribution is one of the most popular methods. It defines the rule of choosing an action with probability:

$$P(s,a) = \frac{e^{Q(s,a)/\tau}}{\sum_{b=1}^{n} e^{Q(s,b)/\tau}}.$$

The positive parameter τ is called temperature, the lower the value of the temperature, the higher the probability becomes that an action with a high *Q*-value will be chosen. When $\tau \to 0$, this becomes the greedy policy.

SIMULATION ANALYSIS

The goal of this analysis is to design a controller which learns to control the zone air temperature setpoint to reduce operating costs by utilizing the building passive thermal storage capacity. The *Q*-Learning algorithm will be applied to govern the model-free learning procedure, and subsequently the optimal policy will be compared with the result of model-based optimal control, which is found by direct search algorithm.

Development of simulation environment

A commercial building and associated energy system model was developed in the Simulink environment of Matlab to model the dynamic thermal response of the building and energy consumption of the HVAC system.
Figure 7 depicts the structure of the simulation environment. The first schematic provides an overview of the models. The simulation environment is made up of four major groups of components. The first one consists of the external and internal heat gain models including modules processing weather data, solar radiation, and building internal heat gain. The second group is the building envelope modules. Using state space modeling, the transient heat transfer through each construction element of the building is calculated using a second-order lumped capacitance model. Previous research shows that a second-order lumped capacitance model with one internal and one external capacitance as well as three resistors can adequately approximate the thermal response of building construction (Gouda et al., 2003). The third group shows HVAC components modules, which includes a VAV terminal box with reheat coil, an air handling unit including an economizer, a cooling coil, and a circulation fan, and finally a simple plant module including an electrical chiller, a cooling tower and a chilled water pump. The three groups of modules are finally linked together into the fourth group, the thermal and humidity balance function block in the fourth schematic, in which the building zone air temperature is updated according to the following equation

$$C_z \frac{dT_z}{dt} = \sum Q_{in} + \sum_{i=1}^{N_{surfaces}} h_i A_i (T_{si} - T_z) + \dot{m}_{\inf} c_p (T_\infty - T_Z) + \dot{m}_{sys} c_p (T_\infty - T_Z)$$

where, T_z, T_∞, T_s are zone air temperature, ambient air temperature and supply air temperature; C_z is the room air thermal capacity; $\sum Q_{in}$ is the convective internal heat gain; h_i, A_i, T_{si} are the interior convection heat transfer coefficient, surface area and temperature of the internal surface of the envelope of the building; $\dot{m}_{\inf}, \dot{m}_{sys}$ are the mass flow rate of infiltration and supply air, respectively.
Similarly, the zone air humidity is updated according the following function:

$$M_z \frac{dg_z}{dt} = \dot{m}_{\inf} (g_\infty - g_Z) + \dot{m}_{sys} (g_s - T_Z) + Q_L / h_{fg}$$

where M_z is the room air mass; g_z, g_∞, g_s are moisture content of zone air, external air and supply air; Q_L, h_{fg} are the latent heat gain to the room space and standard latent heat of vaporization of water in air respectively. The model has been validated with an EnergyPlus model, there is about a 7~8% discrepancy in the load profile between the EnergyPlus and Simulink models, which is considered acceptable.

Simulation parameters

For convenience, the building is modeled as a simple box model with a total area of 1200 m^2, and only one thermal zone is considered. Actually, past research (Liu and Henze 2004) shows that the simplification of building geometry and zoning only marginally affects the optimization. Typical meteorological weather data (TMY2) for Omaha, Nebraska are selected and lighting power densities typical of office buildings are assumed. The utility rate structure and occupancy schedule are synchronized in order to simplify the problem as shown in Figure 2.

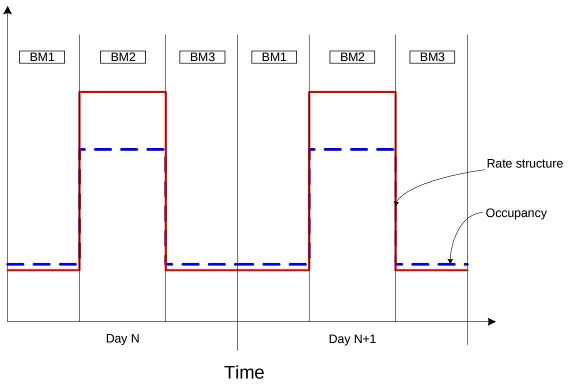


Figure 2: Profiles of the building modes

A new term, *building mode*, is introduced to facilitate the problem representation. Building mode describes the characteristic of rate structure and occupancy of building at certain period of time. Figure 2 presents an example with three building modes, in which, building mode one (BM1) is defined as the time period from midnight to the occurrence of occupancy and on-peak utility rate, which is 7:00 AM in our study; BM2 is the time period covering all the hours with occupancy and on-peak utility rate (7:00 AM to 17:00 PM); and BM3 covers the remaining hours of the day, which has no occupancy and off-peak utility rate. The purpose of the building modes is to simplify the optimization problem. Instead of setting the zone air setpoint hourly, setpoints within same building mode can be set as the same. As a result, the number of optimization variables is reduced from the hours of a day (24) to the total number of building modes. We can define more than three building modes in each day. Meanwhile, zone air temperatures are subject to the constraints that they lie in the range of 15°C to 30°C for unoccupied period and 20°C to 24°C for occupied period.

Implementation of Reinforcement Learning Control

Now we are able to formulate the problem in the framework of reinforcement learning. Each building mode is considered a stage in the sequential decision making problem; the action variable is the setpoint of global zone air temperature, which will be discretized between the temperature constraints corresponding to the building mode. We define the problem as an infinite horizon control problem. Since our goal is to minimize the accumulated operating cost in the long-term, the instant reward is defined as the cost of operating during the building mode multiplied by negative one. There are many ways to select the state variables, which should completely describe all of the relevant information of the control problem and its environment. As an initial step of the research, we simply define each building mode as the only state variable.

A Simulink block for the *Q*-Learning controller is implemented into the simulation environment and the calling sequence of the *Q*-Learning controller is presented in Figure 8. The *Q*-value of each action-state pair is stored in a *Q*-table. At the beginning of each building mode, a temperature setpoint among the available options for the current state will be selected according to either the softmax or ε-greedy exploration method; the setpoint will then be applied by the simulation model and simulated, and at the end of this building mode (beginning of next mode), a cost will be generated, which will be fed back along with the information on the new state into the learning controller. This information will be used to update the *Q*-table.

In our simulation study, we assume that the capacity of the HVAC system is always able to reach the control setpoint, but this may not be true in an actual application. In the case where the HVAC system does not have enough capacity, the selected action cannot lead to the expected state transition. However, even in this case, as long as the state is fully observable and the actual state can be ascertained, the Q-table is still going to be updated properly, and then the optimal policy will be identified eventually.

RESULTS

Base case and model-based optimal control case

In order to evaluate the performance of the reinforcement learning controller, its results are compared to a base case, which applies typical nighttime setback control of zone air temperature setpoint, and a model-based optimal control case. The nighttime setback allows the setpoint of zone air to float to its upper bound (30°C for unoccupied period, 24°C for occupied period). The model-based optimal control is achieved by applying a direct search algorithm (Nelder-Mead algorithm) to find the optimal setpoint profile. The total operating cost for the simulation period (one day in this study) is selected as the objective function; optimization variables are the setpoints of each building mode subject to constraints. Figure 3 presents a simple diagram of the model-based optimal control approach.

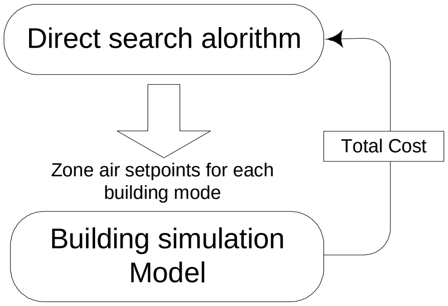

Direct search alorithm
Zone air setpoints for each building mode
Total Cost
Building simulation Model

Figure 3: Schematic of model-based optimal control

Since there is not deviation in this study between building model and actual building as well as between modeled weather and actual weather, the model-based optimal control constitutes a truly optimal scenario. Five cases of model-based optimizations have been carried out with the number of building modes varying from 3 to 24. The arrangement of building modes and the optimal setpoint is presented in Table 5. Two utility rate structures have been tested; the first one has an on-peak utility rate of $0.25/kWh and an off-peak utility rate as $0.05/kWh; the second rate structure has the same off-peak rate as the first one, but the on-peak rate is increased to $0.50/kWh, which is supposed to exaggerate the incentive for load shifting. The result of model-based optimal control and base case under two utility rate schemes are summarized in Table 5. Although the optimal setpoint values are different given different building mode and utility rate, the general pattern of precooling can be found among all the simulation cases. The cost of one day's simulations of optimal control cases and base case are summarized in Table 1.

Table 1: Cost of base case and optimal control cases

	Utility ratio	Base case	3 BM	6 BM	9 BM	12 BM	24 BM
Cost [$]	$0.25/$0.05	$64	$55	$56	$56	$55	$57
	$0.50/$0.05	$127	$100	$99	$101	$99	$102
Saving [%]	$0.25/$0.05	-	14%	13%	13%	14%	11%
	$0.50/$0.05	-	21%	22%	21%	22%	20%

Results of reinforcement learning control

Different reinforcement learning control scenarios have been investigated by varying the action space discretization and the number of building modes, which represents the dimension of state space. The size of the *Q*-table is defined by the dimension of action space multiplied by the dimension of the state space. Two schemes of action space were investigated in the preliminary study (Table 2).

Table 2: Action space

Action space	Off-peak building modes	On-peak building modes
I	[18,21,24]	
II	[16,19,22,25,28]	[20,21,22,23,24]

Action space I has three values ([18,21,24]), and it is same for on-peak and off-peak periods. The action space II has five action values, but it is different to on-peak period and off-peak period. For on-peak periods, it is [20,21,22,23,24], and [16,19,22,25,28] for off-peak period. This is because the upper and lower bound of zone air temperature is different for on and off-peak periods.

In all of the simulation cases, the *Q*-table is initialized with zeros for all entries, and the weather data for a July day is repeated during the simulation to simplify the training procedure. The maximum number of training days is 6,000. The optimal results for the simulated cases are summarized in Table 3 and Table 4.

Table 3: Learning control results for utility cost ratio of 5/1

Utility ratio: $0.25/$0.05

Action space	3 building mode		6 building mode		9 building mode	
	a=0.5	a=0.3	a=0.5	a=0.3	a=0.5	a=0.3
I	24	21	18	24	18	18
					24	21
			18	21	24	18/24
	24	24	24	24	24	24
					24	24
			24	24	24	24
	24	24	24	21	24	18/21
					21	21
			21	24	24	21
II	28	16/25	22	19	22	19/28
					22	19
			19	19/28	22/28	25
	23	23	22/24	24	21	22
					22/24	24
			22	24	24	21
	25	25	19	16	19	19
					16	22
			22/25	22	22/25/28	22

Table 4: Learning control results for utility cost ratio of 10/1

Utility ratio: $0.50/$0.05

Action space	3 building mode		6 building mode		9 building mode	
	a=0.5	a=0.3	a=0.5	a=0.3	a=0.5	a=0.3
I	18	21	18	18	21	24
					24	18
			21	18	24	21
	24	24	24	24	21	21
					24	24
			24	24	24	24
	24	24	21	18/21	21	24
					21	18
			24	18	21	21
II	16	22	16/28	28	28	22
					19	19
			25	19/25	22/25	22
	24	23	23	22	24	23/24
					23/24	23
			24	23	23/24	24
	28	22	28	19/28	28	19/28
					25	19/25
			25	28	22	16/22

From these tables it can be deduced, the performance of the *Q*-controller is affected by many factors including dimensions of the action space and state space, on/off peak utility ratio and learning rate etc. Generally speaking, the cases with higher on/off peak utility ratio generate better results. The best result is obtained for the case of three building modes under utility rate ratio of 10/1. The learning controller found the optimal policy within 2,000 training days, and its result is very close to the one found by model-based optimal control as shown in Figure 4.

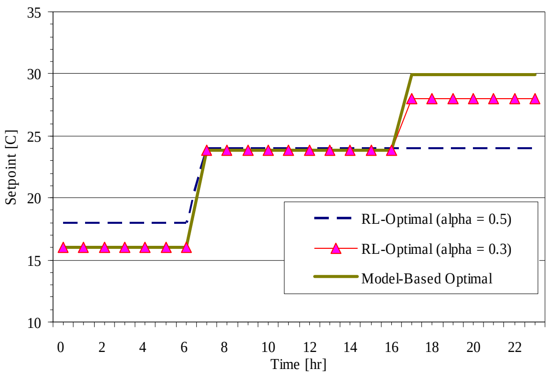


Figure 4: Comparison of learning control with model-based optimal control (BM = 3, utility ratio = 10/1)

Among the cases with lower utility ratio, the one with 6 building modes, action space-I and learning rate $\alpha = 0.5$ found the best optimal result (Figure 5). This gives the implication that less dimension of state space does not necessarily lead to the faster and better learning. However, it is important to identify the proper state space, by which the problem can be well represented and the optimal control policy then will be found.

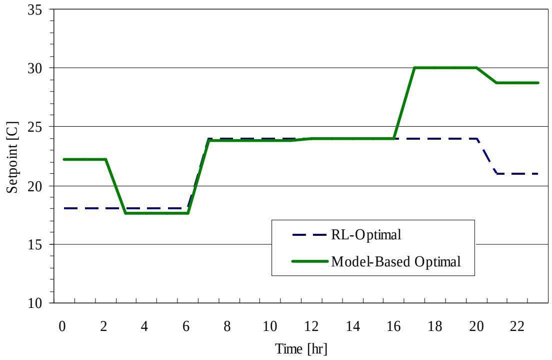


Figure 5: Comparison of learning control with model-based optimal control (BM = 3, utility ratio = 10/1, $\alpha = 0.5$)

In Table 3 and Table 4, it is noticeable that some optimal values of certain cases have two or more values; this indicates that the *Q*-values of two or more action-state pairs are tied or very close to each other after the simulation ended (6,000 days of training). In the example presented in Figure 6, at state 1 (building mode 1), two actions have the same *Q*-value, which are higher than any other values. This implies that Q-controller currently considers that taking either of these two actions would produce the same return. This occurs more frequently when higher dimensions of both state and action spaces are applied. Simply saying, with the increase of the dimension of the *Q*-table, more training is needed to let the controller find the optimal policy.

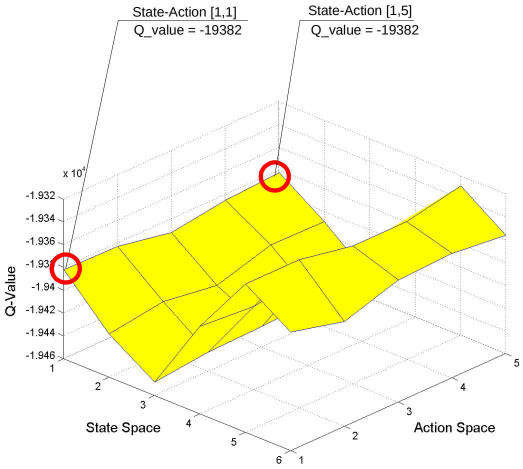


Figure 6: Learning control results with tied *Q*-value

The phenomenon above brings the important issue of the learning controller to attention, which can be called speed of learning. As mentioned before, the maximum training days is set at 6,000 in this preliminary study because of limited computation resources. However, it is found that in some cases the learning controller cannot find the optimal policy in the given training period, such as the example given above, which leads to the tied of values of certain action-state pairs in the *Q*-table. Even for the cases that finally found the optimal result, the computation time is also relatively longer than the predictive optimal control cases. This result is to be expected because the learning controller does not have an explicit building model available but can only improve incrementally from one decision to the next.

CONCLUSIONS AND FUTURE WORK

This simulation study shows that reinforcement learning control is a feasible methodology to find a near-optimal setpoint profile for exploiting the passive building thermal storage capacity. The freedom from a building model makes it especially attractive in real-time control problems, and theoretically it can reach the "true" optimum eventually, no matter what building it is dealing with, if only the environment could be sampled infinitely often. On the other hand, analysis shows that the optimal policy of learning controller is affected by the dimension of the action and state space, the utility rate differentials between on- and off-peak, learning rate and several other factors. Learning speed is relatively low when dealing with problems with large state space and action space. Future work is needed to address these problems to exploit the power of reinforcement learning control.

It was found particularly important to select the proper action space and state space for the learning controller. In this study, the state variable is simply defined as the building mode. It is desirable to define state variables, which describe the thermal history of the building and ambient conditions. However, it is important to realize that there is a downside associated with expanding the state space. Adding new state variables, such as ambient condition, can make the state space representation more appropriate, i.e., it can represent the environment closer. On the other hand, any state variable that is added but that plays no role will prolong the learning process since explorative actions within irrelevant states do not contribute to learning the cost function.

The resolution of the discretization of the continuous action space also needs to be analyzed further. Instead of arbitrarily choosing 3 or 5 discrete action value, an optimal size of the action space needs to be identified. One may argue that the on/off peak utility ratio used in this study is not very realistic (5/1 or 10/1). It was attempted to exaggerate the load-shifting incentive from the utility rate structure in this simulation analysis. However, the real cost for each building mode may not necessarily be the instant reward for the learning controller. Post-processing of the real cost may be useful to make the learning controller learn faster. It is point of future research to identify a modified reward signal instead of using the real cost.

One disadvantage of the reinforcement learning controller is the speed of learning. Compared with model-based optimal control, learning control needs much more training data to find the optimum, which may not be practical in the perspective of computational resources when large state and action spaces are involved. One alternative way to overcome this problem is to implement a set of neural networks to replace the functions of the *Q*-table. It is also expected to accelerate the learning process of the controller.

NOMENCLUTURE

t	discrete time step or stage
s_t	state at *t*
a_t	action at *t*
R_t	cumulative discounted reward (or return) following *t*
$P_{ss'}^{a}$	probability of transition from state *s* to state *s'* under action *a*
$r_{ss'}^{a}$	expected instant reward on transition from state *s* to state *s'* under action *a*
$\pi(s,a)$	probability of taking action *a* in state *s* under the policy π
$V^{\pi}(s)$	value of state *s* under policy π
$Q^{\pi}(s,a)$	value of taking action *a* in state *s* under policy π
$V^{*}(s)$	value of state *s* under optimal policy
α	leaning rate
γ	discount factor

REFERENCES

[1] Andresen, I. and Brandemuehl, M. J. (1992) “Heat storage in building thermal mass: a parametric study.” ASHRAE Transactions 98 (1).

[2] Bellman, R.E. (1957). Dynamic Programming. Princeton University Press, Princeton.

[3] Braun, J. E. (1990) “Reducing energy costs and peak electrical demand through optimal control of building thermal mass.” ASHRAE Transactions 96 (2): 876-888.

[4] Braun, J. E., Montgomery, K.W. and Chaturvedi, N. (2001) “Evaluating the performance of building thermal mass control strategies,” Int. J. of HVAC&R Research, 7 (4): 403-428.

[5] Braun, J. E. (2003) “Load Control Using Building Thermal Mass”, Journal of Solar Energy Engineering, Vol. 125, No. 3, pp. 292-301, American Society of Mechanical Engineers, New York, New York.

[6] Conniff, J. P. (1991) “Strategies for reducing peak air-conditioning loads by using heat storage in the building structure.” ASHRAE Transactions 97 (1): 704-709.

[7] Gouda M.M.; Underwood C.P.; Danaher S. (2003) "Modelling the robustness properties of HVAC plant under feedback control." Building Services Engineering Research and Technology, 1 December 2003, Vol. 24, No. 4, pp. 271-280.

[8] Henze, G. P., Dodier, R. H. and Krarti, M. (1997) “Development of a Predictive Optimal Controller for Thermal Energy Storage Systems.” International Journal of HVAC&R Research, Vol. 3, No. 3, pp. 233-264.

[9] Henze, G.P. and Schoenmann, J. (2003) “Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems.” International Journal of HVAC&R Research, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Atlanta, Georgia.

[10] Henze, G.P., Felsmann, C. and Knabe, G. (2004) “Evaluation of Optimal Control for Active and Passive Building Thermal Storage.” International Journal of Thermal Sciences, February 2004.

[11] Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996) "Reinforcement Learning: A Survey", Journal of Artificial Intelligence Research, 4: 237-285.

[12] Keeney, K.R. and Braun, J.E. (1996) “A simplified method for determining optimal cooling controls strategies for thermal storage in building mass.” Int. J. of HVAC&R Research, 2 (1): 59-78.

[13] Keeney, K. R. and Braun, J. E. (1997) “Application of building precooling to reduce peak cooling requirements.” ASHRAE Transactions 103 (1): 463-469.

[14] Kretchmar, R.M., Young, P.M., Anderson, C.W., Hittle, D.C., Anderson, M. L., Delnero, C. C. (2001) “Robust Reinforcement Learning Control with Static and Dynamic Stability”. *International Journal of Robust and Nonlinear Control.* no. 11, pp. 1469-1500.

[15] Liu, S. and Henze, G.P. (2004) “Impact of Modeling Accuracy on Predictive Optimal Control of Active and Passive Building Thermal Storage Inventory." ASHRAE Transactions, Technical Paper No. 4683. Vol. 110, Part 1.

[16] Morris, F.B., Braun, J. E. and Treado, S. J. (1994) “Experimental and simulated performance of opti-

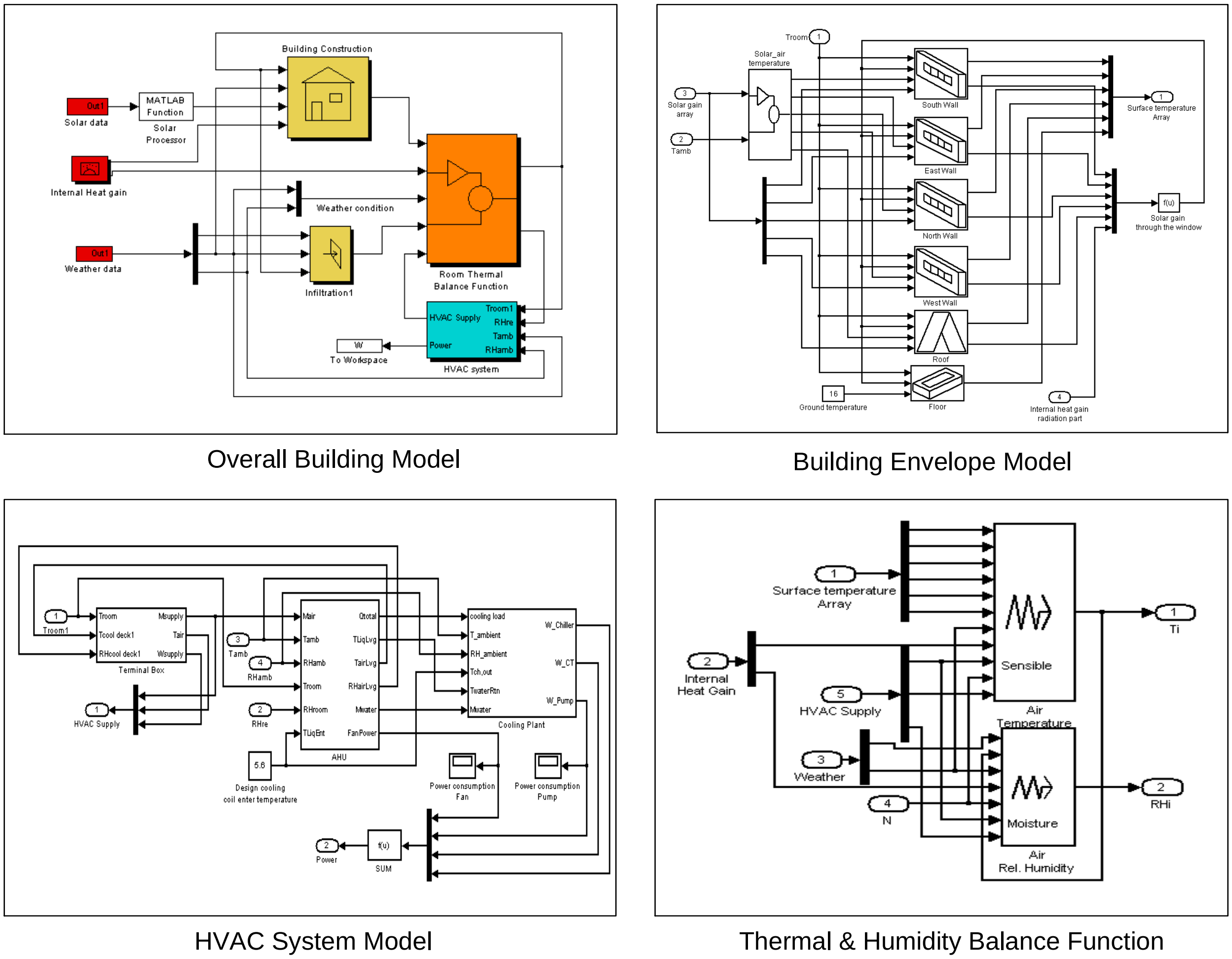
mal control of building thermal storage." ASHRAE Transactions 100 (1): 402-414.

[17] Rabl, A. and Norford, L.K. (1991) "Peak load reduction by preconditioning buildings at night," International Journal of Energy Research 15: 781-798.

[18] Sutton, R.S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning 3*: 9-44.

[19] Sutton, R.S. and Barto, A.G. (1998) Reinforcement learning: An introduction. MIT Press, Cambridge, MA.

[20] Watkins, C. and Dayan, P. (1992) "Q-learning", Machine learning, 8 279-292.



Overall Building Model

Building Envelope Model

HVAC System Model

Thermal & Humidity Balance Function

Figure 7: Schematic of the simulation environment

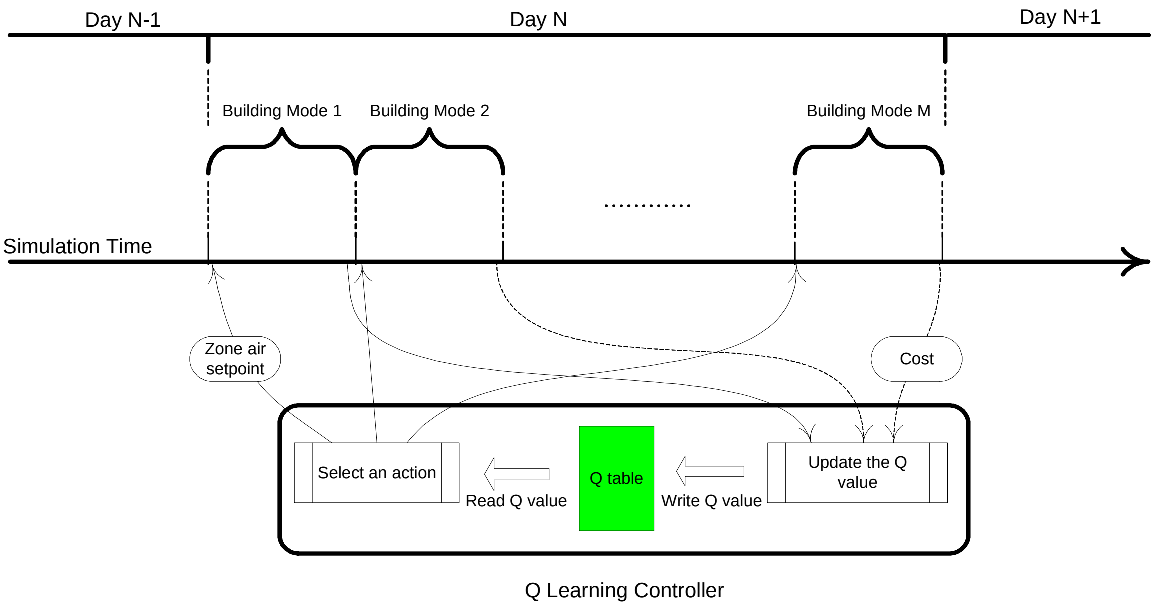


Figure 8: Schematic of calling sequence of the learning controller

Table 5: Summary of results for model-based optimal control

	3 building mode			6 building mode			9 building mode			12 building mode			24 building mode		
Time	BM	T_sp1	T_sp2	BM	T_sp1	T_sp2	BM	T_sp1	T_sp2	BM	T_sp1	T_sp2	BM	T_sp1	T_sp2
0:00	1	15.44	15.00	1	20.48	16.61	1	20.47	18.54	1	21.89	15.00	1	21.52	20.53
1:00													2	20.07	18.97
2:00							2	17.79	15.00	2	18.70	15.05	3	18.84	17.20
3:00				2	15.03	15.00							4	17.00	15.80
4:00							3	15.39	15.00	3	15.69	15.00	5	16.06	15.10
5:00													6	22.00	18.00
6:00										4	15.00	15.00	7	22.49	15.26
7:00	2	24.00	24.00	3	24.00	24.00	4	24.00	24.00	5	24.00	24.00	8	24.00	24.00
8:00													9	24.00	24.00
9:00										6	24.00	24.00	10	24.00	24.00
10:00							5	24.00	24.00				11	24.00	24.00
11:00													12	24.00	24.00
12:00				4	24.00	24.00				7	24.00	24.00	13	24.00	24.00
13:00													14	24.00	24.00
14:00							6	24.00	24.00				15	24.00	24.00
15:00										8	24.00	24.00	16	24.00	24.00
16:00													17	24.00	24.00
17:00	3	28.80	17.89	5	29.00	27.80	7	27.20	26.20	9	26.80	26.57	18	25.20	25.60
18:00										10	28.10	28.74	19	28.20	27.80
19:00							8	28.00	28.10				20	25.10	28.80
20:00										11	27.10	27.55	21	28.00	27.50
21:00				6	29.78	29.10	9	27.60	27.80				22	30.00	29.00
22:00										12	29.00	25.13	23	26.00	29.50
23:00													24	28.00	28.50

T_sp1 denote the optimal setpoint under On/Off peak utility ratio at $0.25/$0.05

T_sp2 denote the optimal setpoint under On/Off peak utility ratio at $0.5/$0.05