Available online at www.sciencedirect.com
ScienceDirect
Building and Environment 42 (2007) 2686–2698
BUILDING AND ENVIRONMENT
www.elsevier.com/locate/buildenv

Reinforcement learning for energy conservation and comfort in buildings

K. Dalamagkidis[a], D. Kolokotsa[b,*], K. Kalaitzakis[c], G.S. Stavrakakis[c]

[a]Computer Science and Engineering Department, University of South Florida, Tampa, FL, USA
[b]Technical Educational Institute of Crete, Department of Natural Resources and Environment, Chania, Crete,Greece
[c]Technical University of Crete, Department of Chania, Crete, Greece

Received 13 March 2006; received in revised form 12 June 2006; accepted 5 July 2006

Abstract

This paper deals with the issue of achieving comfort in buildings with minimal energy consumption. Specifically a reinforcement learning controller is developed and simulated using the Matlab/Simulink environment. The reinforcement learning signal used is a function of the thermal comfort of the building occupants, the indoor air quality and the energy consumption. This controller is then compared with a traditional on/off controller, as well as a Fuzzy-PD controller. The results show that, even after a couple of simulated years of training, the reinforcement learning controller has equivalent or better performance when compared to the other controllers.

Keywords: Energy management; Indoor environment; Reinforcement learning; Adaptive control; Energy efficiency

1. Introduction

Although under investigation for decades, comfort remains an open issue. The international standards are re-evaluated to include the new adaptive comfort standard, while the applicability of the PMV index [1] in naturally ventilated buildings is under scrutiny. At the same time the effort for energy conservation that begun in the 1970s, along with the CO_2 emission reduction requirements renders the use of energy management systems in buildings imperative. During the last decades the applications of automatic control have profited from the use of artificial intelligence—neural networks, fuzzy logic and more recently reinforcement learning. The technique of reinforcement learning is of particular interest. In contrast to other techniques of adaptive control, a reinforcement learning agent does not know what the correct answer is, instead it receives only an indication of the ''correctness'' of its response.

The objective of this work is to design an adaptive controller that will take into account user preferences, in order to achieve energy conservation and user comfort. User comfort entails three distinct goals, thermal comfort, indoor air quality and adequate illuminance.

Although visual comfort is usually considered a significant part of user comfort, it was decided that it was beyond the scope of this paper. This is mainly due to the fact that including visual comfort would require additional states and actions which will increase the complexity of the controller and the training time. Additionally because of the small interdependence between the parameters that affect visual and thermal comfort, it is estimated that the former can be accommodated with the use of an independent, dedicated controller. This controller would also be able to take into consideration the local phenomena like glare that affect visual comfort, which is difficult to take into account in a more general space controller.

In order to assess user comfort the controller should be capable of using any of the comfort measures like the PMV, or the Adaptive Comfort Standard [15–19]. This is achieved by incorporating the comfort measure into a penalty signal that the controller is trying to minimize.

*Corresponding author. Tel.: +30 28210 23017; fax: +30 28210 23003.
E-mail address: kolokotsa@chania.teicrete.gr (D. Kolokotsa).

0360-1323/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.buildenv.2006.07.010

Nomenclature

a	action
e	eligibility trace
J	cost function
R	return
s	state
T_{in}	indoor air temperature
T_{out}	outdoor air temperature
V	state value
w_1	weight of thermal comfort penalty in reinforcement
w_2	weight of energy penalty in reinforcement signal
w_3	weight of indoor air quality penalty in reinforcement
W	weight vector
α	learning rate
δ	TD error
λ	eligibility trace decay parameter
γ	discount parameter
μ	forgetting factor
ϕ	feature vector

Additionally, the controller developed is also able to use direct feedback from the user. The latter was modelled as an intermittent discomfort signal since we assumed that people will report only when they feel discomfort.

In order to put as few restrictions to the application of the controller as possible, we assume that an accurate model of the environment is not available. The only information that is available to the controllers is simple sensor measurements like indoor temperature, humidity and a user response to the current environmental conditions inside the building.

2. State of the art on artificial intelligence on building indoor environment and energy management

There are three main areas of activity in building controller development, namely neural networks, fuzzy systems, predictive control and their combinations [2–5]. Many of the proposed controllers incorporate provisions for occupant thermal comfort and almost all seek to maximize the building's energy efficiency, either directly or indirectly. In [6] a controller consisting of two separate fuzzy systems was designed, with significant energy conservation when compared to a simple on/off controller. The first part determines the comfort zone based on current conditions and a user dependent model, while the second one provides the control. Fuzzy on/off and fuzzy-PID controllers were tested in [7] for the temperature control of an environmental chamber. Both controllers performed better than their non fuzzy counterparts. A fuzzy controller was also chosen by Dounis and Manolakis [8] for thermal comfort regulation. The controller uses the PMV index and the ambient temperature in order to control heating, cooling and natural ventilation (by means of a window). In [9] the authors developed and tested a family of fuzzy controllers, namely a fuzzy PID, a fuzzy PD and an adaptive fuzzy PD. The controllers were used to regulate thermal and visual comfort as well as air quality inside a building. The inputs used, were the PMV index, the CO_2 concentration and the illuminance level.

On the other hand Ben-Nakhi and Mahmoud [10] developed and evaluated a family of six neural networks. They were used to determine the time of the end of thermostat setback in an office building so that by the arrival of the employees the conditions inside were back to normal. The neurobat project developed by Morel et al. [11] uses neural networks for predictive control. In this project neural networks predict outdoor conditions which are then fed to another neural network that forecasts the building behavior.

Neural-fuzzy systems have also been studied. In [12] Egilegor et al. tested a fuzzy-PI controller with and without neural adaptation. The system was used to control heating and cooling within the PMV comfort zone. Although compared to an On/Off controller, the fuzzy-PI performed better, neural adaptation did not offer significant improvement. Karatasou et al. [13] studied feed forward neural networks for modelling energy use and predicting hourly load profiles. Yamada et al. [14] developed a controller that uses neural networks, fuzzy systems and predictive control. This controller is used to improve energy saving in air conditioning systems. Specifically it predicts outdoor conditions (air temperature and solar radiation) as well as the number of occupants. These predictions are subsequently used to estimate building performance (air temperature, wall temperature and heat load) in order to determine the heat sources' optimal start/stop times, optimal night purge time during summer and minimum outdoor air intake. In addition to that the controller aims at maintaining indoor conditions within a comfort zone, which is determined by the PMV index [15–19].

It is noteworthy that even when the controller design aimed solely at achieving thermal comfort, the results also showed reduced energy consumption. Almost all the controllers used the PMV index as thermal comfort measure.

3. Reinforcement learning

In this section the main terms used in reinforcement learning are discussed. These terms are actions, states, policy, reward function, value function, return, discounting, episodic and continual tasks, backup and the Markov property [20].

Actions refer to the decision that the agent will be called to make and can be as low-level as the voltage applied to a motor unit or as high-level as where the agent should focus its attention on. *State* on the other hand refers to the available information that is pertinent to the agent's decision making. State can be comprised of any kind of information ranging from sensor signals to symbolic characteristics of the environment.

Policy defines the way a reinforcement learning agent behaves. It provides a mapping between the situations the agent can find itself in (states) and the action it should take. Policies can be deterministic by specifying which action should be taken under each state or stochastic when for example instead of a specific action, probabilities of choosing several actions are given. Judging from the above, the reinforcement learning problem becomes the problem of determining the optimal policy, the policy that will collect the maximum reward in the long run.

The reward function describes the expected reward of being in a certain state or choosing a certain action while being in a specific state. The reward function can be said to be "short-sighted" as it looks only one step ahead. In contrast to the reward function, the value function defines the total amount of reward that the agent should expect to receive in the long-term by being in a specific state or by choosing an action while being in a specific state.

Value functions are very important in determining the optimal policy. Specifically when an exact model of the environment is available the agent can determine which action will result in the best successor state. The best successor state is defined as that with the largest value. Alternatively in problems were a precise model of the environment is not available, the state-action value is used instead since it provides the means to selecting the actions.

A return is the actual reward received by an agent while following a certain policy. The return may refer to the total reward received or to the reward received after a small amount of time. The return can be used to update the value function because it is in fact an estimate of the value function taken from interaction with the environment. In many situations the reinforcement learning problem may continue indefinitely and the return may reach infinity.

In order to overcome this problem and ensure the boundedness of the return, *discounting* is used. Discounting assigns greater weight to immediate rewards and less to very distant ones. The discounted return can be written as:

$$R_t = r_t + \gamma r_{t+1} + \cdots + \gamma^k r_{t+k}. \quad (1)$$

The γ parameter is known as the discount factor and takes values in the [0 1] interval. The smaller the value the less we care about long-term rewards.

All reinforcement learning problems can be divided into two categories: *episodic and continual*. Episodic problems are problems that have one or more terminal states. When the agent reaches one of these states the episode finishes and the state is reset to its initial setting. An example of an episodic problem is a game of chess, where each episode refers to a single game and the terminal states refer to board positions where either player has won or the game is tied. Continual problems, on the contrary, never terminate but continue indefinitely. An example of a continual problem is a process control problem. Backups refer to the way value function updating occurs. Specifically it refers to which values are used for updating the current value function. For example when using a one-step backup, the agent looks only one step ahead, that is it uses the value function of only the next state (or state action) to update the value of the current state (or state-action).

The Markov property is an important property regarding the state signal, since the applicability and performance of many reinforcement learning algorithms depends on it. In order for a state signal to have the Markov property, the environment dynamics must depend only on the current state and chosen action, thus enabling us to predict the next state and its expected reward only using currently available information and not the entire history up to the current situation. Even when a state signal does not have the Markov property, it is desirable that it represents a good approximation of a Markov signal, because in a different case the reinforcement learning system's performance will be poor. A reinforcement learning task that satisfies the Markov property is called a Markov Decision Process (MDP).

3.1. Reinforced learning methods

There are three major categories of reinforcement learning methods, each with its application scope, advantages and disadvantages. These are Dynamic Programming (DP) methods, Monte-Carlo (MC) methods and Temporal Difference (TD) methods. In this application Temporal Difference (TD) learning methods are selected as they combine the advantages of MC and DP methods. TD learning methods do not require a model of the environment and are able to learn from interaction with the environment on a step by step basis. In order to make an update of the state value function, TD methods only require the observed reward and an estimate of the value of the next state. TD methods, under certain assumptions, have been proven to converge to an optimal policy and it is also true that in several applications they have been found to converge faster than MC methods [20]. Also TD algorithms are possible to be used in order to perform multi-step backup. This is expected to increase the speed of the algorithm since from a single experience, several states visited in the past, will be updated. In order to achieve multi-step backups online, eligibility traces are used. Eligibility traces can be seen from two viewpoints, the forward and the backward [20]. The forward view is more theoretically oriented and it states that eligibility traces represent how far ahead and which states should we look in order to determine the current best action. The backward view is oriented towards implementation and it states that eligibility traces represent a memory of which state

(or state-action) values are "eligible" for updating due to the currently received reward. One method to implement eligibility traces is the complex backups. Complex backups refer to backups that average in any way two or more n-step backups. In order to facilitate implementation of these complex backups the TD(λ) algorithm was developed. The λ constant is a parameter that defines the weighting of each backup. The return used is defined by

$$R_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} R_t^n. \tag{2}$$

In order to implement the TD(λ) algorithm we need to keep track of the eligibility trace of each state $e(s)$, in addition to its value. Care should be taken though to the way the eligibility traces are updated, because now the actions should also be taken into account. The following schema is proposed:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & s \neq s_t, \\ \gamma\lambda e_{t-1}(s) & s = s_t \quad a = a_t, \\ 0 & s = s_t \quad a \neq a_t. \end{cases} \tag{3}$$

Using the Temporal Difference learning method (TD (λ) algorithm) we get the following update rule for the state value V and the weight vector W:

$$\begin{aligned} V_{t+1}(s_t) &= V_t(s_t) + a\delta e_{t+1}(s_t), \\ W_{t+1} &= W_t + a(r_t + \gamma\phi^{\mathrm{T}}(x_{t+1})W_t - \phi^{\mathrm{T}}(x_t)W_t)e_{t+1}, \end{aligned} \tag{4}$$

where $e_{t+1} = \gamma\lambda e_t + \phi(x)$.

The least-squares problem as presented in Eq. (3) has the following objective function:

$$J = \left\| \sum_{i=1}^{\infty} AW - \sum_{i=1}^{\infty} b \right\|^2. \tag{5}$$

The parameters A, b are defined as

$$\begin{aligned} A &= e_t(\phi^{\mathrm{T}}(x_t) - \gamma\phi^{\mathrm{T}}(x_{t+1})), \\ b &= e_t r_t. \end{aligned} \tag{6}$$

Getting the least-squares estimate of W requires a computation that involves the whole sequence of states, actions and rewards and has both computational and memory demands. In several applications of control adaptive filtering and system identification the recursive least-squares algorithm (RLS) is used instead. The RLS algorithm updates the weight vector every time a training sample is available. The weight update rule as adapted for TD(λ) reinforcement learning by Xu et al. [22] is given by the following equations:

$$K_{t+1} = \frac{P_t e_t}{(\mu + \phi^{\mathrm{T}}(x_t) - \gamma\phi^{\mathrm{T}}(x_{t+1})P_t e_t)}, \tag{7}$$

$$W_{t+1} = W_t + K_{t+1}(r_t - (\phi^{\mathrm{T}}(x_t) - \gamma\phi^{\mathrm{T}}(x_{t+1}))W_t), \tag{8}$$

$$\begin{aligned} P_{t+1} = \frac{1}{\mu}\{P_t - P_t e_t[1 + (\phi^{\mathrm{T}}(x_t) \\ - \gamma\phi^{\mathrm{T}}(x_{t+1}))P_t e_t]^{-1}(\phi^{\mathrm{T}}(x_t) - \gamma\phi^{\mathrm{T}}(x_{t+1}))P_t\}, \end{aligned} \tag{9}$$

where $P_0 = \delta \mathrm{I}$.

There are four tunable parameters in the RLS algorithm δ, μ, γ and λ for the eligibility trace update. The δ parameter is used for the initialization of the P matrix and it has been shown that it can influence the convergence speed of the algorithm. The μ parameter is known from adaptive filtering as the forgetting factor and for the standard RLS-TD(λ) should be equal to one. The γ and λ parameters are the same as those used in the TD(λ). It should be noted that no learning rate parameter is essential.

4. The linear reinforcement learning controller (LRLC)

The use of conventional fuzzy controllers although simple and computationally undemanding presents a serious problem regarding the fuzzy rule base generation. Usually the rule base is designed using expert knowledge and trial-and-error techniques which are time consuming and do not provide an update scheme when the environment changes. Reinforcement learning controllers on the other hand choose actions based on a reward function or look-up table. This reward is based on the controller's previous experience [20,21].

The controller was developed using linear function approximation of the state-action value function. The feature vector is constructed using radial basis functions (RBFs). RBFs were preferred to other ways of feature coding, because they provide continuous valued features using a simple and intuitive functional form. For example it is easy to determine the parameters of any number of RBFs that will evenly cover a given range. Besides that, there exist algorithms that can adjust the RBF parameters using supervised training. This feature can lead to better approximation but it was not used in this work because the resulting nonlinearities could lead to convergence problems. In some cases it has been found that RBFs with adaptive centres may leave parts of the space under-represented.

The LRLC approximates the state-action value function as the product of a weight vector and a feature vector. The controller decides on the appropriate action by constructing the feature vector for every possible action under the current state. Then it multiplies all these vectors with the weight vector and chooses the action with the largest expected value. It is therefore obvious that the controller performance depends on the accuracy of the weight vector. The weight vector is constructed at first randomly and consequently it is updated every time a new reward is received.

4.1. The architecture of the RLLC

Since the value matrix contains all possible combinations of states and actions, a large number of inputs and outputs and/or a fine partitioning of the input space will result in a very large value matrix. On the other hand if the

partitioning of the input space is very coarse the controller will be unable to identify different conditions, thus behaving inadequately. Therefore special care should be given during the controller design phase so that redundant partitioning is avoided.

The controller is designed to use any number of inputs. During testing two environments were provided, one that uses the indoor and outdoor air temperatures as well as the relative humidity and CO_2 concentrations as inputs and another that does not use relative humidity. These input variables were chosen since they are usually readily available (with a possible exception of the CO_2 concentration) and do provide the controller with sufficient information about its environment.

The control variables are three. The first refers to the operating status of the heat pump and has seven possible settings, off and high, medium and low for heating and cooling, respectively. The second is the air ventilation subsystem that can operate in three different modes, off, low and high. Finally the third is window control that can take one of the four following states: closed, slightly open, open, wide open. The resulting 84 possible actions are generated by the combinations of the above variables.

The use of a heat pump model for both heating and cooling was preferred because of several advantages it offers without restricting the controller's applicability. Even if a building has separate heating and cooling systems they can be adequately simulated by the heat pump model since the simultaneous operation of both systems would be inefficient and therefore avoided by the controller anyway. The use of one variable for both cooling and heating leads to fewer possible actions that the LRLC needs to try and learn. Further reduction of the possible actions can be achieved by restricting window usage during the operation of the air conditioning system. This reduction could lead to as few as 30 actions at the expense of further complicating the controller design. Nevertheless, in the long term, the controller should learn what the best course of action is without the need of designer imposed restrictions. The use of conventional heating bodies instead of a heat pump can be treated similarly although some modifications are essential. The control variable will have only five states: off, heating, cool low, cool medium and cool high. Although the reward assignment spreads over a number of time steps, the response delay of such a heating system can be quite large resulting in poor behaviour. To overcome this problem we can add one more state referring to the current state of the heating bodies. This additional state will allow the controller to "forecast" the effect of turning on or off the heating.

5. Reinforcement signal design

In order for reinforcement learning to take place, a proper reinforcement signal is necessary. For the problem at hand the reinforcement signal should be a function of the energy consumption and the user satisfaction level. User satisfaction is further divided into thermal comfort and satisfaction with the indoor air quality. The reinforcement signal is modelled as a variable that can take any value in the interval [−1 0]. This variable is in effect a penalty that is higher (closer to −1) during high energy consumption and/or user discomfort. An estimate of current energy consumption can be obtained from the operational characteristics of the heating and cooling devices and their current operating settings. Estimating user satisfaction on the other hand is quite difficult using available measurements. Although we can have a user response signal, this signal may not be used directly since it is irregular and therefore does not convey information regarding user satisfaction levels at every time step.

To overcome this problem an Adaptive Occupant Satisfaction Simulator (AOSS) is developed.

The AOSS associates current environmental conditions inside and outside the building with user satisfaction level, so that for any given set of conditions AOSS' output will be 1 for user dissatisfaction and 0 for user satisfaction. Every time a signal from the user simulator is available, the AOSS is updated to incorporate the new information. In real building applications this information could be stored in a Building Energy Management System database, so that when the user enters his or her office the environmental control will be suited to his or her preferences. For the modelling of the indoor air quality a sigmoid of the CO_2 concentration is used that gives close to 0 values when the CO_2 concentration is less than 780 ppm and values close to 1 when the CO_2 rises above 950 ppm. The final reinforcement signal is given by the following equation:

$$\begin{aligned} r.s. = & -w_1(\text{thermal\_comfort\_penalty}) \\ & -w_2(\text{energy\_penalty})- \\ & -w_3(\text{indoor\_air\_quality\_penalty}) \end{aligned} \tag{10}$$

$$\text{thermal\_comfort\_penalty} = \frac{\sum_{t=0}^{k}\text{AOSS signal}}{k}, \tag{11}$$

$$\text{energy\_penalty} = \frac{\sum_{t=0}^{k}\text{energy consumption}}{\max\text{energy consumption}}, \tag{12}$$

$$\text{indoor\_air\_quality\_penalty} = \sum_{t=0}^{k}\frac{1}{1+\exp[-0.06(\text{CO}_2\text{ concentration-870})]}\Big/k. \tag{13}$$

The w_i variables are constants that represent the importance of each element, namely user satisfaction and energy conservation. Each constitute of the penalty is averaged over the time period between controller re-evaluation so that the final reinforcement signal will be representative of the whole period.

In Eq. (11) the AOSS signal is used as measure of user dissatisfaction. The AOSS signal, as it has already been described, originates from the direct feedback of the building occupants. Of course using the AOSS is not always possible,

especially when a large number of people share the same space. In such situations the use of the Fanger or the Adaptive Comfort Standard (ACS) model are preferable [15–19]. In order for the Fanger model to be used, the AOSS signal needs to be replaced with a value depicting discomfort, namely the PPD. Since the PPD takes values in the interval [0,1] 0 corresponds to all user feeling comfortable and 1 all users being in discomfort, it is more suitable for direct use than the PMV index. Equivalently if the ACS model is in use, the AOSS signal should be replaced by a measure of the distance of current indoor temperature to the comfortable one or by a binary feature showing if current indoor temperature is inside the comfort zone or not.

Regardless of the comfort measure used, the controller will try to optimize by reducing the PPD or minimizing the time that the ACS is showing environmental conditions outside the comfort zone.

6. Results

6.1. Controller testing

In order to evaluate the performance of the controller a suitable testing environment is required. This environment should incorporate a model of the building and one of the user responses. The building was simulated using the SIBIL application [23]. Since user input is essential for the operation and evaluation of the controllers an add-in, called herein user simulator, is developed. This user simulator models user response based on current PMV conditions inside the building and the following tunable parameters:

- Preference—Depending on climatic conditions and cultural background a user may prefer colder or warmer conditions, namely higher or lower PMV values.
- Sensitivity—This parameter describes how far the PMV index can drift from the optimal value before the user senses discomfort.

Table 1
The CO_2 concentration statistics for the On/Off and the Fuzzy PD controller

	On/Off	Fuzzy PD	
	Building A and B	Building A	Building B
Min	485	489	400
Mean	823	787	658
Max	1099	1098	935

Table 2
Training parameters of an one-year LRLC simulation

w_1: 0.80	λ: 0.5
w_2: 0.05	γ: 0.9
w_3: 0.25	μ: 1

Table 3
Reinforcement learning parameters used for training a LRLC over a period of 4 years

	1st year	2nd year	3rd year	4th year
w_1	0.80	0.80	0.80	0.80
w_2	0.01	0.01	0.01	0.01
w_3	0.20	0.20	0.20	0.27
λ	0.50	0.50	0.50	0.50
γ	0.95	0.95	0.95	0.90
μ	1.00	1.00	1.00	1.00
ε	0.075	0.025	0.000	0.000

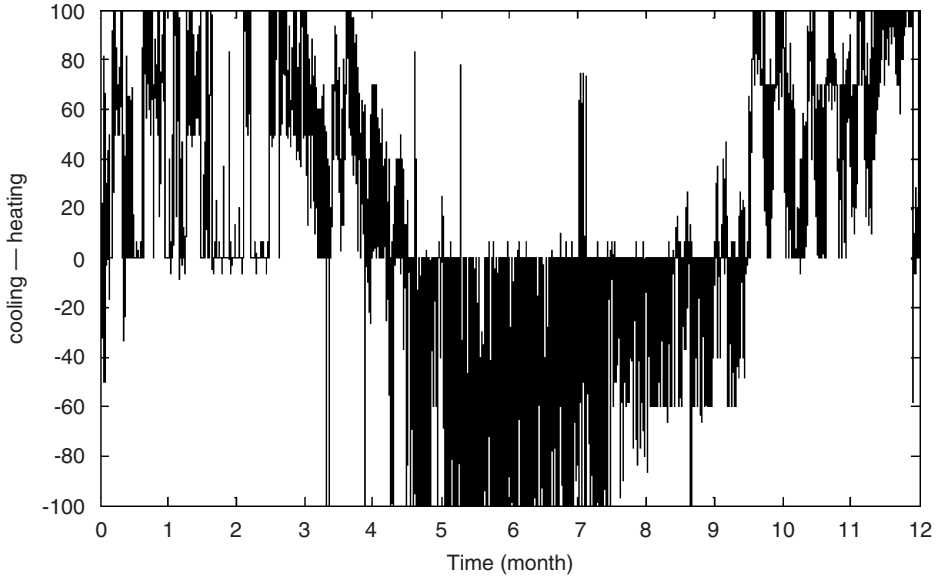

Fig. 1. Heat pump response of the LRLC during its first year of simulated training. The response is averaged over an one hour period. This controller utilizes only three inputs (T_{in}, T_{out} and $[CO_2]$) and is applied in a building with no insulation.

- Interest—This takes into account that each user will report his discomfort to the controller with different frequency.

Table 4
Simulation results of the LRLC over a period of 4 years

	1st year	2nd year	3rd year	4th year
Average PPD (%)	12.1	12.4	12.8	12.0
Annual energy consumption (MWh)	9.39	7.13	5.83	4.85
Min [CO_2] (ppm)	385	385	389	394
Mean [CO_2] (ppm)	464	462	450	539
Max [CO_2] (ppm)	1658	860	860	1697

There is also a facility to save user preferences along with current controller knowledge. If there is a need, the controller can be reset in order to start learning from scratch. User response is modelled as a two state signal. One state denotes that the user feels discomfort, while the other provides no real information since it may denote that the current conditions are satisfactory or that although the opposite is valid the user did not report it. Two buildings are modelled in the Sibil application. Building A has an area of 15 m^2, one window (1 m^2) and the walls are made from 21 cm of concrete and a 2.5 cm insulating layer of foamed polystyrene. Building B has an area of 14 m^2 and one window (2 m^2) facing in a different direction. The walls are made of 21 cm of concrete but without insulation.

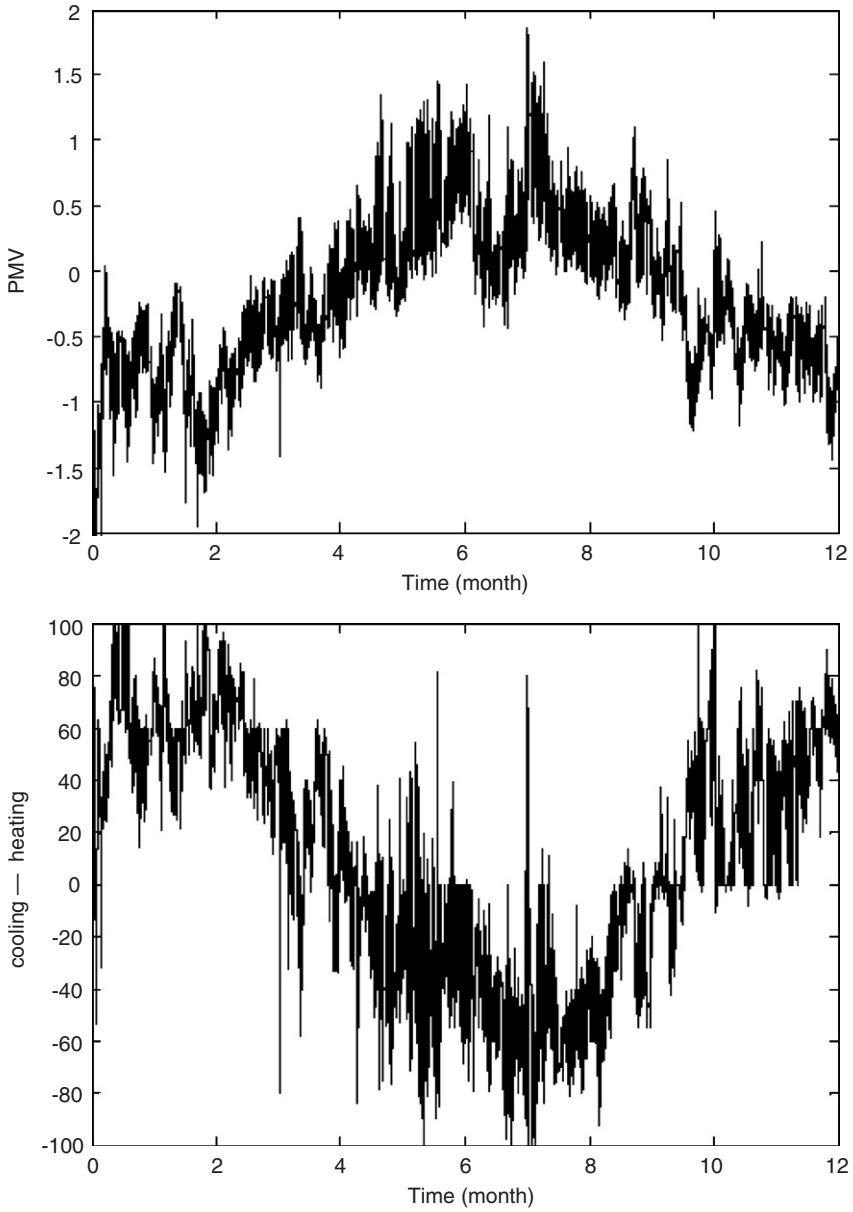

Fig. 2. 1st year of LRLC training. The controller utilizes four inputs (*T*in, *T*out, month and [CO_2]) and is applied in an insulated building. The heat pumpresponse is averaged over a period of 2 h.

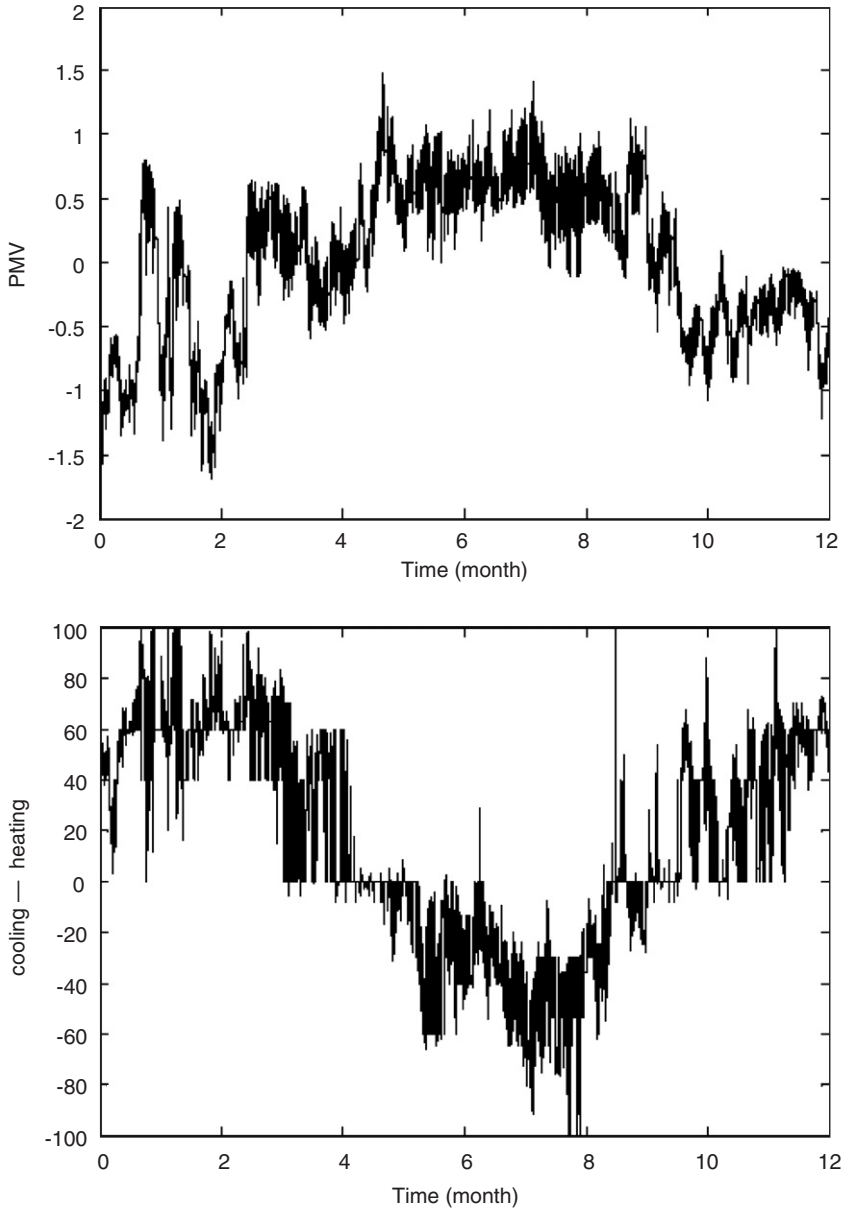

Fig. 3. Second year of LRLC training. The controller utilizes four inputs (Tin, Tout, month and [CO_2]) and is applied in an insulated building. The heat pumpresponse is averaged over a period of 2 h.

For comparison purposes all controllers are based on the PMV model and the average PPD index is used as a measure of thermal comfort.

6.2. Reference controllers

Two reference controllers are used. The first is an On/Off controller that uses the PMV index. This controller only operates the heating and cooling. Specifically it turns on the appropriate device when the PMV index moves outside the [−0.8 0.8] region and turns it off when the PMV moves inside the [−0.5 0.5] region. The ventilator unit is always on in the low setting to prevent large increase in CO_2 concentration levels. Since this controller operates based on the real PMV value it is expected that it will achieve low average PPD. In a real building application we should expect that the PMV index is only estimated or even that the controller will operate based on temperature. In any case it is doubtful that we can expect less energy consumption or smaller average PPD.

The second controller is a fuzzy-PD controller that is described in detail in [9]. This controller operates besides heating and cooling, the window and the ventilator.

The annual energy consumption of the On/Off controller in building A is about 4.77 MWh and for building B is 8.65 MWh. The corresponding consumptions for the fuzzy-PD controller are 3.28 and 5.83 MWh, respectively. During this year the On/Off controller achieved an annual average PPD of 13.4% and 16.7% while the fuzzy controller had 16.5% and 24.5% for the first and second building,

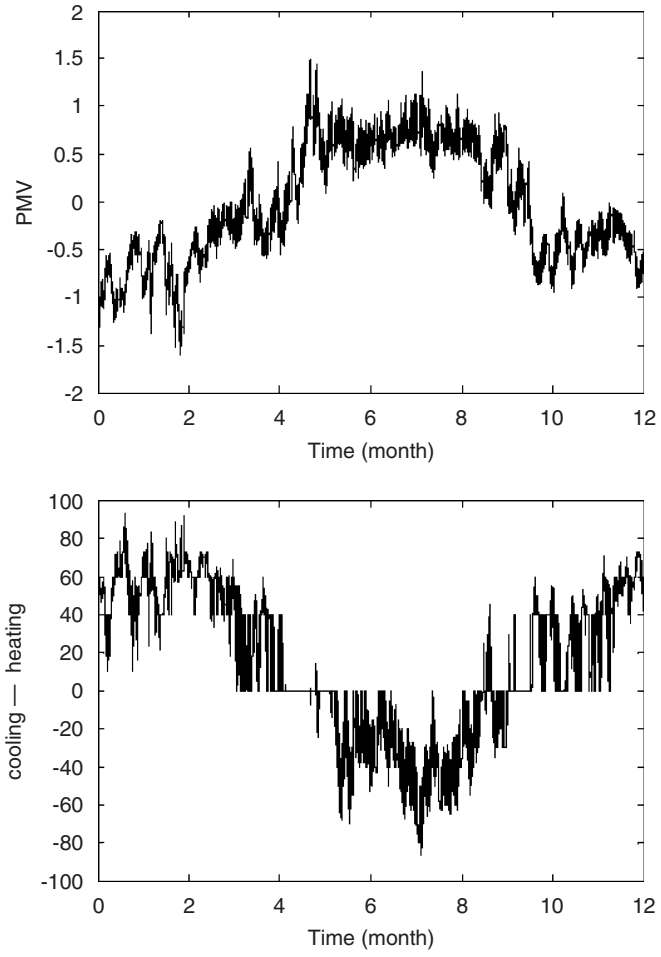

Fig. 4. Third year of LRLC training. The controller utilizes four inputs (Tin, Tout, month and [CO_2]) and is applied in an insulated building. The heat pump response is averaged over a period of 2 h.

respectively. The maximum, minimum and mean CO_2 concentrations are summarized in Table 1.

6.3. LRLC testing

6.3.1. Test configurations

The LRLC is tested using three different controller configurations:

A. Four inputs (T_{in}, T_{out}, RH, [CO_2]).
B. Four inputs (T_{in}, T_{out}, Time, [CO_2]).
C. Three inputs (T_{in}, T_{out}, [CO_2]).

For each of these configurations two different feature vectors were tested. The first feature vector consists of the values of all the state vectors, augmented by the action vector. The components of the action vector represent heating, cooling, ventilation and window opening as values in the [0,1] range. The second feature vector comes from multiplying the first vector with the action vector. The smaller feature vector used was of size 34 and the largest of size 232.

The controller is tested for varying periods of time, radial basis functions (RBFs) and parameter values $\varepsilon, \gamma, \lambda$. After testing, the eligibility trace decay parameter was chosen to be 0.5. This value is consistent with what we expected, that is the actions taken up to 30 or 40 min ago should influence the current reward. The discount factors γ used are between 0.8 and 0.95 since higher values are not recommended from the bibliography and smaller values (0.3–0.6) exhibited inefficient behaviour. The inadequate behaviour for low discount factors can be attributed to the fact that the controller probably found ways to increase immediate rewards, while simultaneously losing access to better long-term rewards. It is possible that even by using small discount factors the controller will eventually converge to a good policy.

The forgetting factor μ was chosen to be one at all cases since even a small change (0.99) caused bad behaviour by the controller. In general large feature vectors exhibited better performance as it was expected. Using feature vectors that combined states and actions resulted in increased performance since these feature vectors where able to capture some of the nonlinear relationships between states and/or actions.

6.3.2. LRLC performance

The LRLC controller exhibits adequate training speeds. It is noteworthy that even during the first year the controller is able to quickly develop a policy that although is far from optimal, it contains only few clearly wrong actions. Applying an LRLC of the C configuration in building B we took the response depicted in Fig. 1. This figure shows the controller's heat pump response with the exploratory actions eliminated. It is apparent that the controller quickly found that a good action during winter is to turn on heating and cooling during summer. It should be noted that the ε parameter was only 2% and that the exploratory actions where not completely random but chosen as one setting higher or lower than the calculated optimal. The annual energy consumption was 6.95 MWh and the average PPD 31.5%. The high PPD is due to the fact that the controller begins with no knowledge of its environment and therefore makes a lot of mistakes especially in the beginning. This is evident from the fact that the average PPD of the last 6 months is only 25.5% while the average PPD of the first 3 months is more than 60%. The rest of the training parameters are summarized in Table 2.

The evolution of training is described in the following paragraph where the results from four single-year simulations are discussed. The controller tested corresponds to configuration B and is simulated using the building A definition. The parameters used during these simulations are cited in Table 3. The results are summarized in Table 4.

The results show that the annual average PPD does not change significantly with time but the energy consumption is reduced significantly from year to year by about 25% each time. At the same time the CO_2 concentrations vary

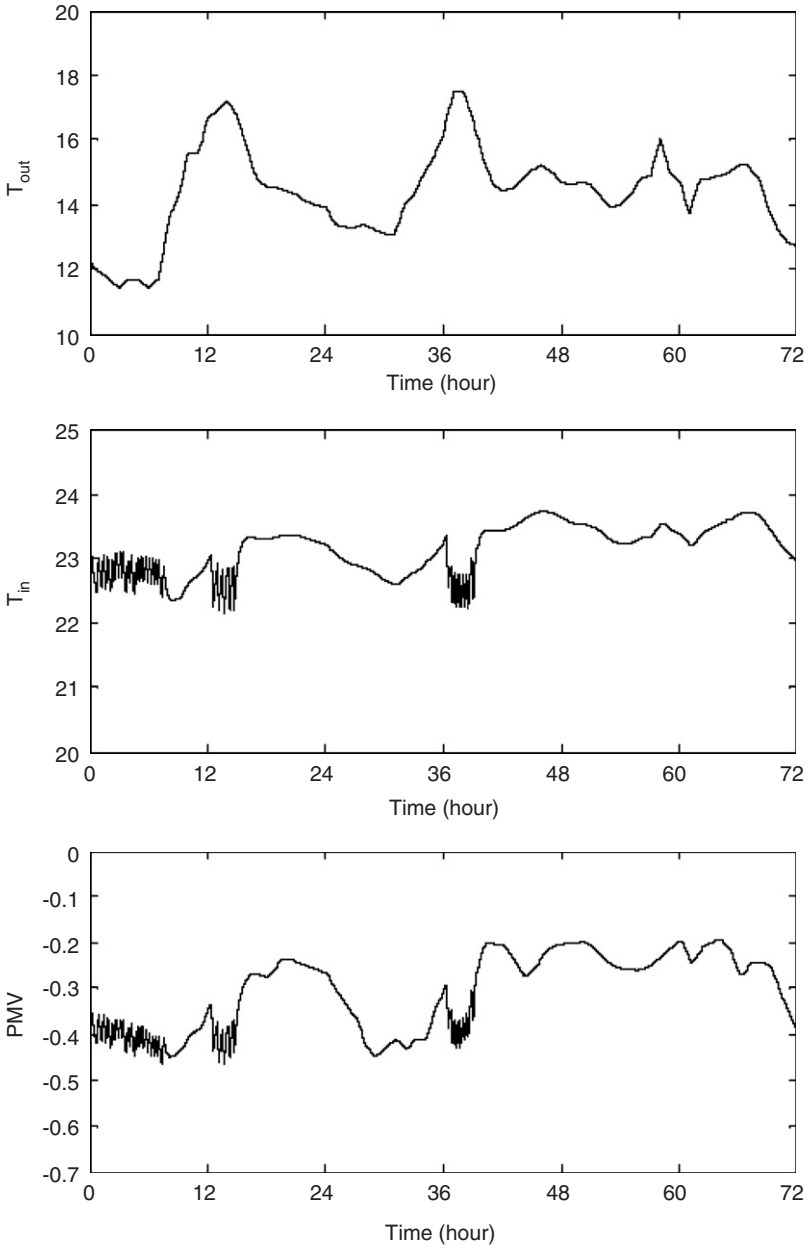

Fig. 5. Temperature and PMV variations for a 3-day winter period of a trained LRLC.

within acceptable ranges after the first year, despite the fact that the CO_2 weight on the reinforcement signal is very small. It is noteworthy that even during the second year the CO_2 concentration is above 800 ppm for less than an hour in a whole year. During the fourth and last year we increased the energy weight on the reinforcement signal and decreased the discounting factor. This had as an effect a decrease in the annual energy consumption and an increase of the CO_2 concentrations. The latter can be attributed to the fact that the CO_2 concentrations are reversely analogous to the energy consumption. In order to conserve energy, the agent needs to reduce heat losses, reduces air exchanges with the outdoor environment and as a result the CO_2 concentrations increase. Fig. 2 shows the controller heat pump response for the first year and the corresponding PMV. Although a pattern is visible, there is a large number of random actions where the controller continuously switches from heating to cooling regardless of the season. This is due to the fact that the controller has no experience yet and because the ε value is 7.5% which means that the controller takes random actions quite frequently. Fig. 3 shows the response of the controller during the second year of simulation. Now the ε value is smaller (2.5%) and the controller choices are based mostly on experience. Correspondingly the variations of the PMV index are smaller. Fig. 4 corresponds to the third year simulated. This time the controller uses the greedy algorithm. It is obvious that it has learned not to use

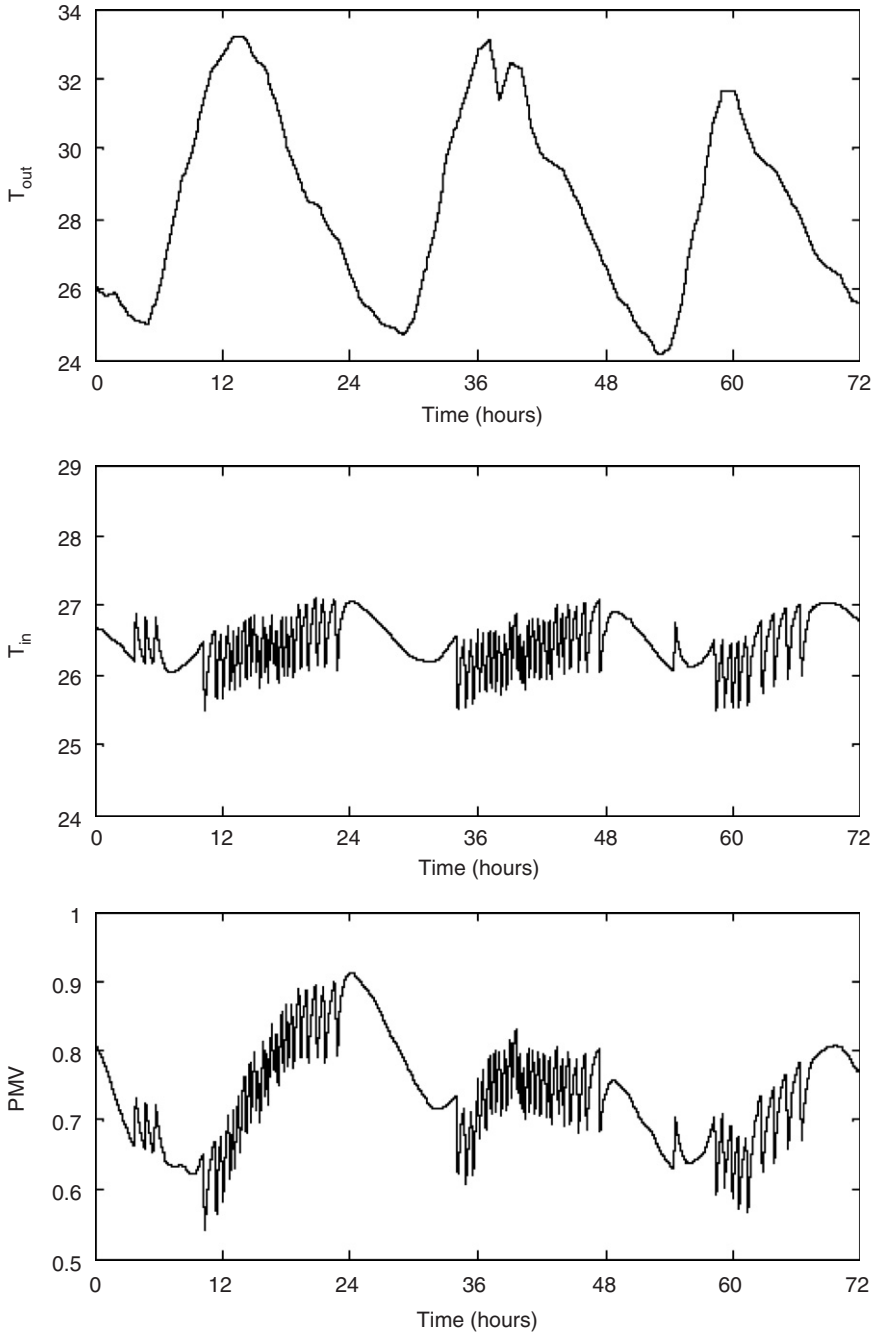

Fig. 6. Temperature and PMV variations for a 3-day summer period of a trained LRLC.

cooling during the winter months and although it occasionally chooses to turn heating during summer the performance is greatly improved. The three figures described above provide only a very rough view of the controller's response. In order to better visualize the controller's true response, Fig. 5 shows the variations of indoor and outdoor temperature and the PMV for a period of three days in winter and Fig. 6 for a corresponding period during summer. The data used are from the third year of simulation. During the winter three day period the controller kept the indoor temperature at a mean value of 23.1 °C. The width of variation for the same period was 1.5 °C for the indoor temperature and 6.1 °C for the outdoor. The worst PMV value is −0.47 and the average is −0.31. Equivalently for the summer period the controller kept the temperature 26.5 °C with a variation width of 1.6 °C, while the outdoor temperature had a variation width of 9.1 °C. The worst PMV value is 0.91 with a mean of 0.73. The variations in indoor temperature during noontime occur due to the fact that the controller switches the cooling between low, medium and high in order to keep the temperature from rising while at the same time maintaining low energy consumption. It should be noted that despite the fact that during the last year the greedy

algorithm is used, the controller still learns and improves its performance by updating its value function.

7. Conclusions

After training a LRLC for a simulated period of 4 years, we achieved the results summarized in Table 5 which also provides the corresponding results of the On/Off and Fuzzy-PD controllers. It is evident from this table that the LRLC has achieved a performance close to that of the other two controllers. The larger energy consumption is attributed to the fact that the controller is still changing its policy and to the fact that the user comfort carries a larger weight thus outperforming the other two controllers in that measure. It should be noted that although the PPD index is used to assess user comfort, the results would be similar with any other comfort measure. The energy consumption depends only on how far the comfort penalty allows the controller to go without using the heat pump.

It is significant that the LRLC has achieved this performance while still making errors. Even during the fourth year the agent may turn heating in summer or cooling during winter. Further training should eliminate these wrong decisions and result in even better energy conservation. This is expected since even after a period of 4 simulated years the controller still makes significant improvements. The errors that still occur are in part attributable to the exploration and in part to the fact that due to the nature of the learning algorithm, the controller needs multiple visits to each state in order to determine its real value and develop a robust policy. Significant improvement may also be achieved by using different feature vectors or by providing the controller with more information about its environment.

The main benefit from the use of reinforcement learning in BEMS is that the controller continually learns and improves on its policy. Specifically it is possible to create pretrained controllers with a general knowledge of the building. These controllers will then gradually adapt to optimize their behaviour with respect to the specific characteristic of the building/space they are used. Additionally this controller could adapt to changes of the building characteristics stemming for example from equipment ageing or replacement, leaks, etc. which can not be taken into account with other controller designs.

An issue that involves the application of reinforcement learning controllers in BEMS is that of sufficient exploration. It is true that taking random actions even during a small fraction of the time is unacceptable in a real building. Even when the choice is between near-optimal actions we should expect temporary increases in user dissatisfaction and an increase in the total energy consumption (2–3% for $\varepsilon = 0.02$). As a result it is necessary to exhaustively train the controller before installation and allow very little controlled exploratory actions or no exploration at all.

Table 5
Comparison of LRLC, On/Off and fuzzy-PD controllers

	LRLC	On/Off	Fuzzy PD
Average PPD (%)	12.1	13.4	16.5
Annual energy consumption (MWh)	4.85	4.77	3.28

References

[1] Fanger PO. Thermal comfort analysis and applications in environmental engineering. New York: Mc Graw Hill; 1970.

[2] Dounis AI, Santamouris M, Lefas CC, Manolakis DE. Thermal Comfort degradation by a visual comfort fuzzy reasoning machine under natural ventilation. Journal of Applied Energy 1994;48:115–30.

[3] Dounis AI, Santamouris M, Lefas CC, Argiriou A. Design of a fuzzy set environment comfort system. Energy and Buildings 1995;22:81–7.

[4] Bruant M, Dounis AI, Guarracino G, Michel P, Santamouris M. Indoor air quality control by a fuzzy reasoning machine in naturally ventilated buildings. Journal of Applied Energy 1996;53.

[5] Clarke JA, Cockroft J, Conner S, Hand JW, Kelly NJ, Moore R, et al. Simulation-assisted control in building energy management systems. Energy and Buildings 2002;34(9):933–40.

[6] Hamdi M, Lachiver G. A fuzzy control system based on the human sensation of thermal comfort, Fuzzy Systems Proceedings, 1998. IEEE world congress on computational intelligence. The 1998 IEEE international conference, vol. 1, 4–9 May 1998, pp. 487–92.

[7] Salgado P, Cunha JB, Couto C. A computer-based fuzzy temperature controller for environmental chambers. Industrial electronics, 1997. ISIE '97. Proceedings of the IEEE international symposium, vol. 3, 7–11 July 1997, pp. 1151–56.

[8] Dounis AI, Manolakis DE. Design of a fuzzy system for living space thermal-comfort regulation. Applied Energy 2001;69(2):119–44.

[9] Kolokotsa D, Tsiavos D, Stavrakakis GS, Kalaitzakis K, Antonidakis E. Advanced fuzzy logic controllers design and evaluation for buildings' occupants thermal—visual comfort and indoor air quality satisfaction. Energy and Buildings 2001;33:531–43.

[10] Ben-Nakhi AE, Mahmoud MA. Energy conservation in buildings through efficient A/C control using neural networks. Applied Energy 2001;73:5–23.

[11] Morel N, Bauer M, El-Khoury M, Krauss J. Neurobat, a predictive and adaptive heating control system using artificial neural networks. International Journal of Solar Energy 2001;21:161–201.

[12] Egilegor B, Uribe JP, Arregi G, Pradilla E, Susperregi L. A fuzzy control adapted by a neural network to maintain a dwelling within thermal comfort. Proceedings of Building Simulation 1997;2:87–94.

[13] Karatasou S, Santamouris M, Geros V. Modeling and predicting building's energy use with artificial neural networks: methods and results. Energy and Buildings 2005, in press.

[14] Yamada F, Yonezawa K, Sugarawa S, Nishimura N. Development of air-conditioning control algorithm for building energy-saving. IEEE international conference on control applications, Hawai, USA, 1999.

[15] Olesen BW, Parsons KC. Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730. Energy and Buildings 2002;34:537–48.

[16] Memarzadeh F, Manning A. Thermal comfort, uniformity, and ventilation effectiveness in patient rooms: performance assessment using ventilation indices. ASHRAE transactions, Symposia 2000.

[17] Jones BW. Capabilities and limitations of thermal models for use in thermal comfort standards. Energy and Buildings 2000;34:653–9.

[18] Humphreys MA, Nicol JF. The validity of ISO-PMV for predicting comfort votes in every-day thermal environments. Energy and Buildings 2002;34:667–84.

[19] Nicol JF, Humphreys MA. Adaptive thermal comfort and sustainable thermal standards for buildings. Energy and Buildings 2002;34:563–72.
[20] Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge, MA: MIT Press; 1998.
[21] Reynolds SI. Reinforcement learning with exploitation, in School of Computer Science. Birmingham: The University of Birmingham; 2002.
[22] Xu X, He HG, Hu D. Efficient reinforcement learning using recursive least-squares methods. Journal of Artificial Research 2002: 259–92.
[23] Eftaxias G, Sutherland G, Santamouris M. A building simulation toolbox for MATLAB/SIMULINK, Installation guide and user manual, Group Building Environmental Studies, Department of Applied Physics, University of Athens; 1999.