Air Conditioner Control Learning Users' Sensations Based on Reinforcement Learning and Its Scalability Improvement

Noritaka Shigei, Yohei Yamaguchi, Hiromi Miyajima

***Abstract*—This study proposes the air conditioner (AC) control system based on users' sensations. The purpose of the system is to improve low-performance ACs in terms of energy efficiency and comfortableness performance. The system consists of wireless sensor nodes and user nodes such as PCs and smartphones, and it is applicable to already installed ACs. The users enter their sensation such as *cold*, *good*, *a little hot* and *very hot* through the user node, and the system determines the appropriate control according to the users' sensations. The control signal is transmitted via an equipped IR remote controller. The appropriate control policy is determined based on Q-Learning, which is a reinforcement learning method. In this study, we propose several types of methods and investigate effective methods in terms of energy efficiency and comfortableness performance. For multiple users, several methods for integrating users' sensations are presented. These make our proposed system applicable to a large number of users. Further, in order to reduce the energy consumption and the number of users' inputs, several types of reward functions are presented. In addition to the reward functions proposed in [8], in this paper, we propose a new reward function, which improves the performance for the large number of users. In the simulation, five types of methods including a new proposal in this paper, are evaluated in terms of the time needed for providing a comfortable environment and the energy consumption. We clarify the effective methods among them and the tendency of the scalability against the number of users.**

***Index Terms*—air conditioning, reinforcement learning, wireless sensor network, sensory scale, integration of sensations**

I. Introduction

IN our modern life, the air conditioning system is an essential system. The system needs to be energy efficient and to provide a comfortable room temperature environment[1]. Its state-of-the-art systems, which are generally expensive, equip many high-precision sensors and can achieve high energy efficiency and high comfortableness. Due to budgetary constraints, many facilities often have to use old systems or introduce systems with low introducing cost, which are of low-performance in the control ability. However, such systems are of low energy efficiency and cannot always reasonably provide a comfortable environment. Further, for a large room with many workers, the difference between the ideal performance and the actual one would be large. Because the low-performance systems generally do not equip sufficient number of sensors for sensing the whole room.

Thanks to the recent advances in wireless sensor network (WSN) technologies, WSNs have been effectively employed in various fields such as industry[2] and home automation[3]. The sensing capability of WSN can be utilized for improving the old or low-performance systems. Since wireless sensor devices become much cheaper year by year, it is also a realistic approach. Further, recent years, mobile devices such as tablet PCs and smartphones are to be found everywhere. By incorporating them into WSN system, a powerful system can be realized with a reasonable cost.

For realizing intelligent control systems, reinforcement learning (RL) is a promising technique. RL does not need any teacher signal and it can obtain an appropriate behavior pattern by trial and error according to rewards given from the environment[6], [7]. Therefore, RL can be applied to problems with unknown environment. In RL, in order to obtain an appropriate behavior pattern (policy) for the environment defined by the target problem, the agent repeats trial and error and updates its policy according to some reward obtained from the environment (see Fig.1). RLs have been applied to room environment conditioning. In [4], for air conditioning, PI controller has been combined with RL. The purpose of the control is the minimization of the control error. In [5], for lighting system, an intelligent control system has been proposed by actor-critic algorithm, which is one of RLs. The system controls the lighting system according to the presented user's sensation such as brighter and darker. This approach is gentle to humans and attractive in the case where the control has to be determined according to many users.

In this study, we propose the AC control system based on users' sensations. The purpose is to realize the control system that improves the low-performance AC in terms of energy efficiency and comfortableness. The system consists of wireless sensor nodes and user nodes such as PCs and smartphones, and it is applicable to already installed ACs. The users enter their sensation such as *cold*, *good*, *a little hot* and *very hot* through the user node, and the system determines the appropriate control according to the users' sensations. The appropriate control policy is determined based on Q-Learning, which is one of RLs. For multiple users, several methods for integrating users' sensations are presented. Further, in order to reduce the energy consumption and the number of users' inputs, several types of reward functions are presented. In addition to the reward functions proposed in [8], in this paper, we propose a new reward function, which improves the performance for the large number of users. In the simulation, five types of methods including a new proposal in this paper, are evaluated in terms of the time needed for providing a comfortable environment

This work was supported by JSPS KAKENHI Grant Number 26330108.

N. Shigei and H. Miyajima are with the Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima, Japan, and Y. Yamaguchi was with the Graduate School of Science and Engineering, Kagoshima University. e-mail: k9630866@kadai.jp, shigei@eee.kagoshima-u.ac.jp, miya@eee.kagoshima-u.ac.jp

(Advance online publication: 10 July 2015)

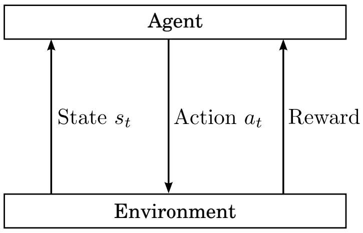


Fig. 1. The concept of reinforcement learning.

TABLE I
THE USED PARAMETERS IN THE ROOM TEMPERATURE MODEL.

Param.	The used values
AC op. mode (C, W)	(0,0), (1,1), (1,1.6), (1,3), (2,1), (2,1.6), (2,3), (3,1), (3,1.6), (3,3)
α	0.1
β	10.0
$c(0)$	T_o
$c(1)$	27.0
$c(2)$	23.0
$c(3)$	18.0

and the energy consumption. We clarify the effective methods among them and the tendency of the scalability against the number of users.

II. Model of Room Environment

In this section, a model of the temperature in a room environment with an air conditioner is described. Let $C > 0$ and $W > 0$ be the cooling strength of the air conditioner and the fan speed, respectively. Let $T(t,d)$ be the air temperature at time t and at distance d away from the air conditioner. Then, we define the relation of $T(t,d)$, C, W and d, based on the RC electric circuit model. The temperature $T(t,d)$ is defined as follows:

$$T(t,d) = T(t-1,d) + T_{\text{drp}} \cdot \left(1 - \exp\left(\frac{-1}{\tau(d,C,W)}\right)\right), \tag{1}$$

$$T_{\text{drp}} = -T(t-1,d) + \alpha \cdot T_o + c(C), \text{ and} \tag{2}$$

$$\tau(d,C,W) = \min\left(\frac{\beta \cdot d}{C \cdot W}, \ \beta \cdot d\right) \tag{3}$$

where T_o is the outer air temperature, α and β are constant numbers, and $c(C)$ is a decreasing function with C. The factor T_{drp} determines whether the temperature $T(t,d)$ raises or falls. If T_{drp} is positive, the temperature increases. Otherwise, it decreases. The coefficient α is the influence rate of T_o and $0 \le \alpha \le 1.0$. The function $c(C)$ determines the lower (upper) limit of the temperature in the case of a rise (fall) in temperature. The parameters used in this paper is shown in Table I.

Fig.2 shows the changes of the temperature $T(t,d)$ for $d = 2$ and the different C and W. In the figure, it is observed that, as C and W increase, the convergence temperature decreases. Fig.3 shows the changes of the temperature $T(t,d)$ for $C = 3$, $W = 2$ and the different d. In the figure, it is observed that, the convergence temperatures are same but the convergence speed becomes faster with the distance d.

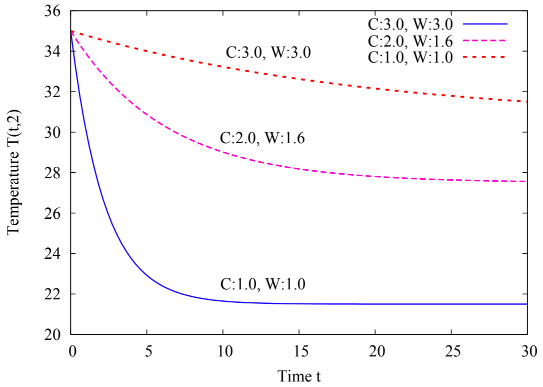


Fig. 2. Changes in the room temperature when changing C and W.

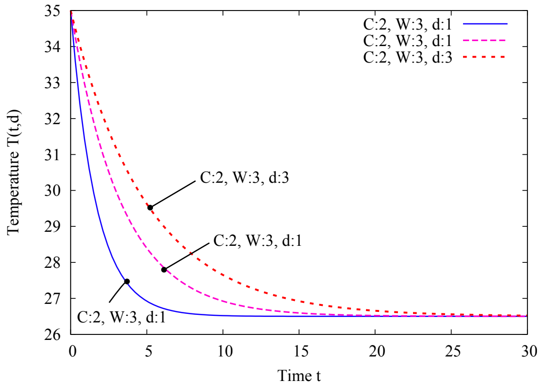


Fig. 3. Changes in the room temperature when changing d

.

III. Air Conditioning Control System based on Sensuous Instruction

A. System Configuration

In this section, we explain our air conditioning system based on sensuous instruction. Our system consists of an air conditioner (AC), two types of sensor devices, user nodes such as personal computer (PC), tablet and smartphone (see Fig.4). The AC equips an infrared remote controller. The first type of sensor device is installed around the receiver of the AC so as to receive the infrared signal from the remote

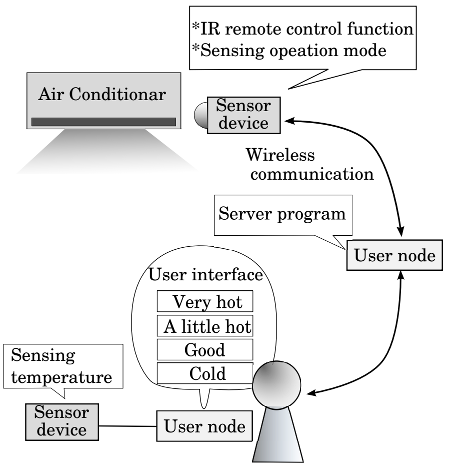


Fig. 4. Air conditioning control system.

controller. The sensor device sniffs the infrared signal and monitors the operation mode of the AC. The sensed information on the operation mode is sent to the server program run on one of the user nodes by wireless communication. Further, the sensor device also equips the infrared LED, which is used for controlling the AC according to the command from the server program. The second type of the sensor device is installed around the users and each sensor monitors the air temperature around the user. The sensed temperature is also sent to the server program by wireless communication. The primary role of the user nodes is the interface between the user and the system. Through the user interface application run on the node, the user enters his/her sensation such as *cold*, *good*, *a little hot*, *very hot*. The entered sensation is also sent to the server program. The server program is run on one of the user nodes. The program collects the information from the sensor groups and the user nodes, calculates the action for control according to the collected information, and controls the AC by emitting the infrared signal from the sensor device.

B. Q-Learning for AC Control Based on User's Sensation

We apply Q-learning[6], which is one of reinforcement learning methods, to air conditioning control based on user's sensory scale. In reinforcement learning, in order to obtain an appropriate behavior pattern (policy) for the environment defined by the target problem, the agent repeats trial and error and updates its policy according to some reward obtained from the environment (see Fig.1). The advantage of reinforcement learning is to not need any teacher signal, which indicates the ideal action at each state. Therefore, it is applicable to problems with unknown environments. Profit Sharing (PS)[7] and Q-learning are well-known reinforcement learning methods. Compared with PS, it is know that, Q-learning has a higher ability to obtain an optimal policy.

In this subsection, we present the basic form of our Q-learning algorithm for controlling our AC system. In section IV, some key components in the basic form are customized, and five types of the algorithm are presented. Let N be the number of users. Let S be the set of states, in which each member corresponds to the user sensation. S is a key component defined in the next section. Further, the determination of the state involves the integration of N users' sensations, which is also described in the next section. Let A be the set of actions, in which each member corresponds to the AC operation mode (C, W), that is, $A = \{(0,0), (C,W) | C \in \{1,2,3\}, W \in \{1, 1.6, 3\}\}$. Let $Q(s,a)$ be the Q-value of the pair of state s and action a, which indicates how much worth is the action a at the state s. As the learning progresses, the Q-value of an appropriate pair of state and action increases. At each state s, the action with higher Q-value is taken with a higher probability compared with other actions. We assume that the goal of air conditioning is to keep the satisfaction (*good*) of at least M_{S} users (e.g. two-thirds of N users like $\lceil \frac{2}{3}N \rceil$) for consecutive T_{G} time steps. The algorithm is given as follows:

ALGORITHM Q-Learning for AC Control
Initialization:

Let l be the current epoch number, and set $l \leftarrow 0$.
Let t be the current time, and set $t \leftarrow 0$.
For all $s \in S$ and $a \in A$, $Q(s,a) \leftarrow 0$.

Step 1 (State observation): Determine the current state s_0 according to the N user inputs.
Step 2 (Taking action): For each action $a \in A$, calculate the probability $\pi(s_t, a)$ as follows:

$$\pi(s_t, a) = \frac{\exp(Q(s_t, a_t)/T_{\mathrm{b}})}{\sum_{a' \in A} \exp(Q(s_t, a')/T_{\mathrm{b}})} \tag{4}$$

$$T_{\mathrm{b}} = T_{\mathrm{b0}} \cdot \left(\frac{T_{\mathrm{b1}}}{T_{\mathrm{b0}}}\right)^{\frac{l}{L_{\max}}} \tag{5}$$

where $L_{\max}$ is the number of maximum epochs, T_{b} is the temperature for Boltzmann selection, and T_{init} and T_{fin} are maximum and minimum temperatures of T_{b} respectively.

Probabilistically take an action $a_t \in A$ according to the probabilities $\pi(s_t, a)$ $(a \in A)$.
Step 3 (Reward acquisition and state observation): Let r_t be the acquired reward, which is a key component described in section IV. Determine the next state s_{t+1} according to N users' inputs.
Step 4 (Updating Q value):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \delta_t, \tag{6}$$

where

$$\delta_t = r_t + \gamma \cdot \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t), \tag{7}$$

α is the learning rate and γ is a constant such that $0 \leq \gamma \leq 1$.
Step 5 (Judgment of goal achievement): If the goal is reached, M_{S} users among N users feel *good* for consecutive T_{G} time steps, go to the next step. Otherwise, set $t \leftarrow t + 1$ and go to Step 2.
Step 6 (Judgment of termination): Set $l \leftarrow l + 1$. If $l = L_{\max}$, then terminate the algorithm. Otherwise, for the next epoch, set $T(0, d)$, T_{o} and the users' sensation scales, which depend on the environment. Set $t \leftarrow 0$ and go to Step 1. □

In section IV, the set of states S, the integration of users' sensations and the reward r_t are described in detail.

IV. Integration of Users' Sensations and Reward Function

In this section, we explain the set of states S, the integration of users' sensations and how to give the reward. The naive design of the state set is to represent all the possible combinations of the users' sensations. However, this design needs the number of states that is proportional to the exponential of the number of users N. As N increases, the memory size needed by the naive design exponentially increases. Therefore, in this section, we presents three types of integration methods of users' sensations. Further, how to give the reward is an important issue in reinforcement learning. Firstly, we present the reward function using not only the users' sensations and but also the changes of room temperature obtained from the sensor. Then, we present the reward functions using only the users' sensations, which do not need sensing the room temperature. In addition to the reward function proposed in [8], in this paper, we also propose a new reward function, which improves the performance for the large number of users.

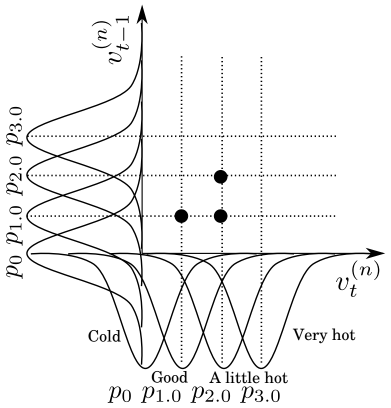

$v_{t-1}^{(n)}$
p_0 $p_{1.0}$ $p_{2.0}$ $p_{3.0}$
$v_t^{(n)}$
Cold
Very hot
Good
A little hot
p_0 $p_{1.0}$ $p_{2.0}$ $p_{3.0}$

Fig. 5. The probability density function arrangement for SI-L and an input example for 3 users.

A. Naive Design of State Set

The state is coded as a tuple of $N+1$ elements $(s^{(1)}, s^{(2)}, \cdots, s^{(N)}, s^{\Delta\mathrm{T}})$. Each of N elements corresponds to a user sensation, that is, for each $n \in \{1, 2, \cdots, N\}$, $s^{(n)} \in \mathcal{S}$, where $\mathcal{S} = \{$*cold*, *good*, *a little hot*, *very hot*$\}$ is the set of user's sensations. The rest one element $s^{\Delta\mathrm{T}}$ represents the state of temperature change, which is determined according to the sensing data of a sensor node. Let $\mathit{\Delta T} = T(t, d_{\mathrm{sn}}) - T(t-1, d_{\mathrm{sn}})$ be the change of the temperature, where d_{sn} is the distance of the sensor node from AC. The element $s^{\Delta\mathrm{T}}$ is defined as follows:

$$s^{\Delta\mathrm{T}} = \begin{cases} \mathit{Down} & ;\ \mathit{\Delta T} < -0.5 \\ \mathit{Unchanged} & ;\ |\mathit{\Delta T}| \leq 0.5 \\ \mathit{Up} & ;\ \mathit{\Delta T} > 0.5. \end{cases} \tag{8}$$

Then, the number of states is $|\mathcal{S}|^N \cdot 3 = 4^N \cdot 3$.

B. Integration of Users' Sensations

Let us consider to integrate the users' sensations into the collective sensation. We present three types of sensation integration (SI) methods: SI based on Majority Vote (SI-MV), SI based on Averaging (SI-A) and SI based on Likelihood (SI-L).

The first method, SI-MV, integrates N users' sensations into the collective sensation s_{c} as follows:

$$s_{\mathrm{c}} = \operatorname*{argmax}_{s \in \mathcal{S}} U(s, t), \tag{9}$$

where $U(s, t)$ is the number of users whose sensations are s at time t, $\mathcal{S} = \{$*cold*, *good*, *a little hot*, *very hot*$\}$ is the set of user's sensations, and if $U(s, t)$s for plural sensations s tie in majority vote then the function argmax returns the sensation in order of *cold*, *good* and *a little hot*. The state set for SI-MV is $S = \{(s_{\mathrm{c}}, s^{\Delta\mathrm{T}}) | s_{\mathrm{c}} \in \mathcal{S}, s^{\Delta\mathrm{T}} \in \{\mathit{Down},\ \mathit{Unchanged},\ \mathit{Up}\}\}$. Then, the number of states for SI-MV is $|\mathcal{S}| \cdot 3 = 12$ for any number N.

The second method, SI-A, represents each user sensation as a scalar value, calculates the average of the scalar values and returns the integrated one from 12 possible sensations whose segmentation is finer than the one of the original sensation with 4 segments. The mapping from the user sensation $s^{(n)} \in \mathcal{S}$ to a scalar value $v(s^{(n)}) \in \mathcal{V}$ is as follows:

$$v(s^{(n)}) = \begin{cases} 0 & ;\ s^{(n)} = \mathit{cold} \\ 1.0 & ;\ s^{(n)} = \mathit{good} \\ 2.0 & ;\ s^{(n)} = \mathit{a\ little\ hot} \\ 3.0 & ;\ s^{(n)} = \mathit{very\ hot}, \end{cases} \tag{10}$$

where $\mathcal{V} = \{0, 1.0, 2.0, 3.0\}$. The averaging value $\bar{v} = \frac{1}{N}\sum_{n=1}^{N} v(s^{(n)})$ is mapped to the collective value $v^{\mathrm{c}} \in \mathcal{V}^{\mathrm{c}}$ as follows:

$$v^{\mathrm{c}} = \operatorname*{argmax}_{v \in \mathcal{V}^{\mathrm{c}}} |v - \bar{v}|, \tag{11}$$

where $\mathcal{V}^{\mathrm{c}} = \{0.25,\ 0.5,\ 0.75,\ 1.0,\ 1.25,\ 1.5,\ 1.75,\ 2.0,\ 2.25, 2.5,\ 2.75,\ 3.0\}$. The state set for SI-MV is $S = \{(v^{\mathrm{c}}, s^{\Delta\mathrm{T}}) | v^{\mathrm{c}} \in \mathcal{V}^{\mathrm{c}}, s^{\Delta\mathrm{T}} \in \{\mathit{Down},\ \mathit{Unchanged},\ \mathit{Up}\}\}$. Then, the number of states for SI-A is $|\mathcal{V}^{\mathrm{c}}| \cdot 3 = 36$ for any N.

The last method, SI-L, determines the current state based on the likelihood on the users' sensations at time t and $t-1$. Likewise SI-A, SI-L represents the user sensation as a scalar value. Unlike SI-MV and SI-A, SI-L takes into account not only the user sensations at the current time t but also the one at the previous time $t-1$. With this feature, SI-L does not use the information on the temperature change.

Let $s_t^{(n)}$ and $s_{t-1}^{(n)}$ be the n-th user's sensations at time t and $t-1$, respectively. Let $v_t^{(n)}$ and $v_{t-1}^{(n)}$ be the converted scalar values of the n-th user's sensations at time t and $t-1$, respectively. The conversion is done by using Eq.(10). Then, the set of states is $S = \{(v_{\mathrm{cur}}^{\mathrm{c}}, v_{\mathrm{prev}}^{\mathrm{c}}) | v_{\mathrm{cur}}^{\mathrm{c}}, v_{\mathrm{prev}}^{\mathrm{c}} \in \mathcal{V}\}$, where $v_{\mathrm{cur}}^{\mathrm{c}}$ and $v_{\mathrm{prev}}^{\mathrm{c}}$ correspond to the collective values at time t and $t-1$, respectively. The number of states for SI-L is $|\mathcal{V}|^2 = 16$.

In the following, the determination of the current state for SI-L is described in detail. The probability density function p_v of the user sensation corresponding to the converted scalar value $v \in \mathcal{V}$ is defined as follows:

$$p_v(v^{(n)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(v^{(n)} - v)}{2\sigma^2}\right), \tag{12}$$

where $\sigma^2 = 1$ is the variance of the distribution. Given the N users' sensation scalar values at time t and $t-1$, $v_t^{(1)}$, $v_t^{(2)}$, $\cdots$, $v_t^{(N)}$, $v_{t-1}^{(1)}$, $v_{t-1}^{(2)}$, $\cdots$, $v_{t-1}^{(N)}$, then the likelihood of each state $(v_{\mathrm{cur}}, v_{\mathrm{prev}}) \in S$ is calculated as follows:

$$L(v_t^{(1)}, \cdots, v_t^{(N)}, v_{t-1}^{(1)}, \cdots, v_{t-1}^{(N)}; v_{\mathrm{cur}}; v_{\mathrm{prev}}) = \prod_{n=1}^{N} p_{v_{\mathrm{cur}}}(v_t^{(n)}) \cdot p_{v_{\mathrm{prev}}}(v_{t-1}^{(n)}). \tag{13}$$

The collective state $s^{\mathrm{c}} = (v_{\mathrm{cur}}^{\mathrm{c}}, v_{\mathrm{prev}}^{\mathrm{c}}) \in S$ is calculated as follows:

$$s^{\mathrm{c}} = \operatorname*{argmax}_{(v_{\mathrm{cur}}, v_{\mathrm{prev}}) \in S} L(v_t^{(1)}, \cdots, v_t^{(N)}, v_{t-1}^{(1)}, \cdots, v_{t-1}^{(N)}; v_{\mathrm{cur}}; v_{\mathrm{prev}}). \tag{14}$$

C. Reward Function

We consider four types of reward functions. The functions take into account the following reward functions and penalties.

- $R_{\mathrm{L}} > 0$: Reward for goal achievement.

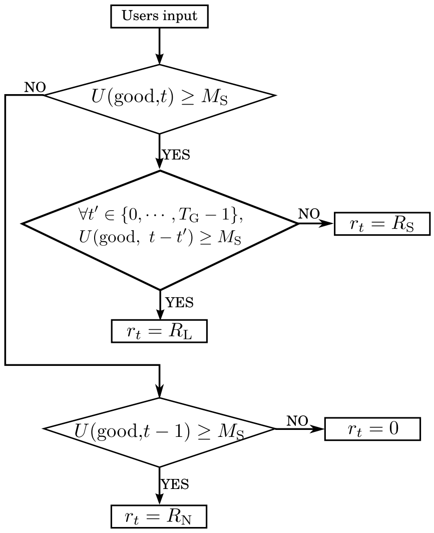

Users input
NO
$U(\text{good},t) \geq M_\mathrm{S}$
YES
$\forall t' \in \{0, \cdots, T_\mathrm{G}-1\}$,
$U(\text{good},\ t-t') \geq M_\mathrm{S}$
NO
$r_t = R_\mathrm{S}$
YES
$r_t = R_\mathrm{L}$
$U(\text{good},t-1) \geq M_\mathrm{S}$
NO
$r_t = 0$
YES
$r_t = R_\mathrm{N}$

Fig. 6. The definition of Excess-Cooling-Unaware Reward Function (ECU-RF).

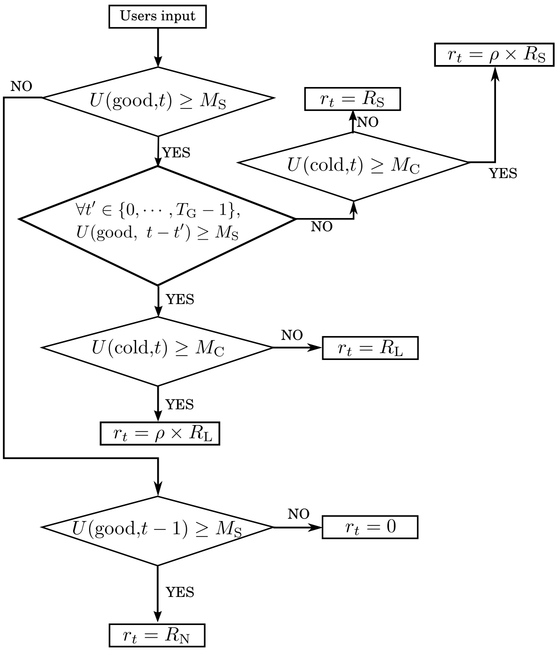

Users input
$r_t = \rho \times R_\mathrm{S}$
NO
$U(\text{good},t) \geq M_\mathrm{S}$
$r_t = R_\mathrm{S}$
NO
YES
$U(\text{cold},t) \geq M_\mathrm{C}$
YES
$\forall t' \in \{0, \cdots, T_\mathrm{G}-1\}$,
$U(\text{good},\ t-t') \geq M_\mathrm{S}$
NO
YES
$U(\text{cold},t) \geq M_\mathrm{C}$
NO
$r_t = R_\mathrm{L}$
YES
$r_t = \rho \times R_\mathrm{L}$
$U(\text{good},t-1) \geq M_\mathrm{S}$
NO
$r_t = 0$
YES
$r_t = R_\mathrm{N}$

Fig. 7. The definition of Soft-Penalty-for-excess-Cooling Reward Function (SPC-RF).

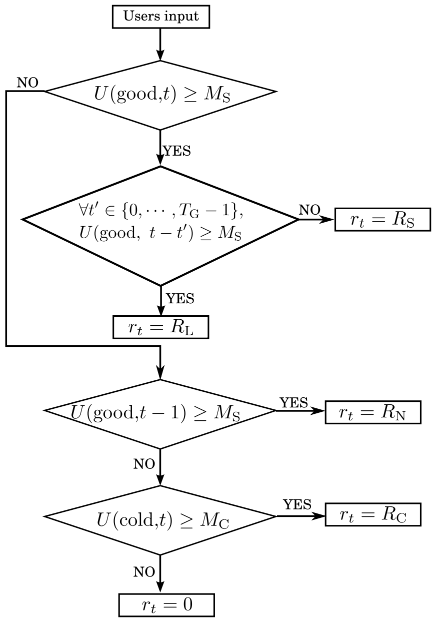

Users input
NO
$U(\text{good},t) \geq M_\mathrm{S}$
YES
$\forall t' \in \{0, \cdots, T_\mathrm{G}-1\}$,
$U(\text{good},\ t-t') \geq M_\mathrm{S}$
NO
$r_t = R_\mathrm{S}$
YES
$r_t = R_\mathrm{L}$
$U(\text{good},t-1) \geq M_\mathrm{S}$
YES
$r_t = R_\mathrm{N}$
NO
$U(\text{cold},t) \geq M_\mathrm{C}$
YES
$r_t = R_\mathrm{C}$
NO
$r_t = 0$

Fig. 8. The definition of Hard-Penalty-for-excess-Cooling Reward Function (HPC-RF).

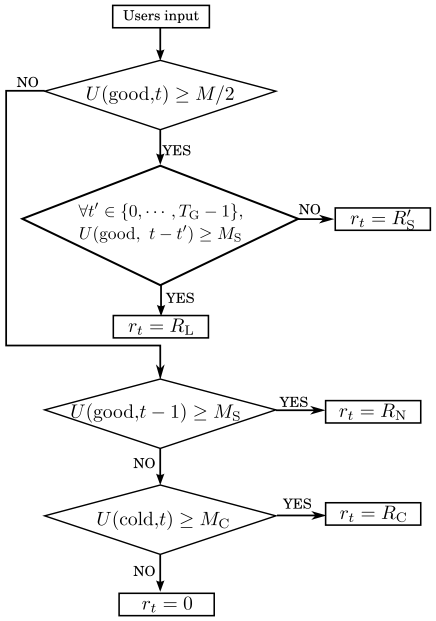

Users input
NO
$U(\text{good},t) \geq M/2$
YES
$\forall t' \in \{0, \cdots, T_\mathrm{G}-1\}$,
$U(\text{good},\ t-t') \geq M_\mathrm{S}$
NO
$r_t = R'_\mathrm{S}$
YES
$r_t = R_\mathrm{L}$
$U(\text{good},t-1) \geq M_\mathrm{S}$
YES
$r_t = R_\mathrm{N}$
NO
$U(\text{cold},t) \geq M_\mathrm{C}$
YES
$r_t = R_\mathrm{C}$
NO
$r_t = 0$

Fig. 9. The definition of Hard-Penalty-for-excess-Cooling-with-Soft-Reward Reward Function (HPCSR-RF).

- $R_\mathrm{S} > 0$: Reward for keeping a good control.
- $R_\mathrm{N} < 0$: Penalty on the degradation on the users' sensations.
- $R_\mathrm{C} < 0$, $\rho \cdot R_\mathrm{L}$, $\rho \cdot R_\mathrm{S}$: Penalty for excess cooling, which means dissipation of energy, where $0.0 < \rho < 1.0$.

The first type is Excess-Cooling-Unaware Reward Function (ECU-RF). ECU-RF does not care about any excess cooling, that is, it does not use $R_\mathrm{C} < 0$, $\rho \cdot R_\mathrm{L}$ nor $\rho \cdot R_\mathrm{S}$. Fig.6 shows how ECU-RF determines the reward r_t, where $U(s,t)$ is the number of users whose sensations are $s \in \mathcal{S}$ at time t, and $M_\mathrm{S} = \lceil \frac{2}{3} N \rceil$ is the minimum number of users to be satisfied.

The second type is Soft-Penalty-for-excess-Cooling Reward Function (SPC-RF). SPC-RF reduces the reward with the reduction rate ρ when the AC control is excess cooling. Fig.7 shows how SPC-RF determines the reward r_t, where M_C is the number of users that are not allowed to feel *cold*.

The third type is Hard-Penalty-for-excess-Cooling Reward Function (HPC-RF). HPC-RF gives a negative reward R_C when the AC control is excess cooling. Fig.8 shows how HPC-RF determines the reward r_t.

The above three types have been proposed in [8]. The fourth type, which is the new proposal of this paper, is presented in the next subsection.

D. Reward Function for Large Number of Users

In this paper, we propose the fourth type of RF as a new method. The RF is referred to as HPC-with-Soft-Reward Reward Function (HPCSR-RF), which is a revised version of HPC-RF. The aim is to improve the scalability against

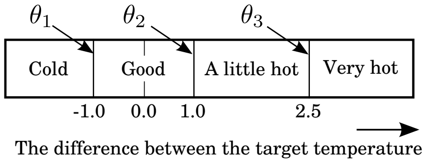


Fig. 10. Sensory scale of virtual user.

TABLE II
SIMULATION CONDITIONS.

Item	Used values
User's dist. from AC d	1, 2, 3
Target temperature	26.0 ±0.5 °C
Initial temperature $T(0,d)$	Same as outer air temperature
Outer air temperature $T_{\rm o}$	30∼35 °C
θ_1	-1.0 ±0.3 °C
θ_2	1.0 ±0.3 °C
θ_3	2.2 ±0.3 °C
Number of users to be satisfied $M_{\rm S}$	$\lceil \frac{2}{3}N \rceil$
Maximum number of epochs $L_{\max}$	500
Duration of satisfaction state needed for reaching goal T_G	10 time steps

the number of users. As shown later in the simulation results presented in section V, SI-L with HPC-RF for a large number of users such as 24 does not achieve as good convergence as for a smaller number of users such as 12 and 3. Although HPC-RF provides a small positive reward such as $R_{\rm S}$ only when $U({\rm good},t) \geq M_{\rm S}$, HPCSR-RF provides a small positive reward too when $U({\rm good},t) < M_{\rm S}$. The small reward of HPCSR-RF, $R'_{\rm S}$, is a function of $U({\rm good},t)$ as follows:

$$R'_{\rm S} = \begin{cases} R_{\rm S} & ;\ U({\rm good},t) \geq M_{\rm S} \\ R_{\rm S} \cdot \eta^{U({\rm good},t)-M_{\rm S}}; & M_{\rm S} > U({\rm good},t) \geq M/2 \\ 0 & ;\ U({\rm good},t) < M/2. \end{cases} \tag{15}$$

Fig.8 shows how HPCSR-RF determines the reward r_t.

V. Numerical Simulation

A. Simulation Setting

The virtual users used in the simulation have sensory scales as shown in Fig.10, where thresholds θ_1, θ_2 and θ_3 represent the boundaries between *cold* and *good*, between *good* and *a little hot* and between *a little hot* and *very hot*.

The simulation conditions are summarized in Table II. The equal number of virtual users is arranged at each distance $d \in \{1, 2, 3\}$. At each epoch, set randomly the initial room temperature $T(0,d)$, the outer air temperature $T_{\rm o}$, and the target temperature and the thresholds θ_1, θ_2 and θ_3 for each user.

We evaluate the five types of methods as shown in Table III. In addition to the number of users $N = 12$ considered in [8], the simulations are performed for the numbers of users $N = 3$ and 24. Note that, (w/o, SPC) is performed only for $N = 3$ because the number of states exponentially increases with N. The used parameters for the evaluated methods are summarized in Table IV. These parameters are determined by preliminary simulations. How to determine will be described in detail in the next subsection.

TABLE III
EVALUATED METHODS.

Name	Sensation Integration				Reward Function			
	w/o	MV	A	L	ECU	SPC	HPC	HPCSR
(w/o, SPC)	○					○		
(MV, ECU)		○			○			
(A, HPC)			○				○	
(L, HPC)				○			○	
(L, HPCSR)				○				○

B. Simulation Result

We evaluate five types of methods (w/o, SPC), (MV, ECU), (A, HPC), (L, HPC) and (L, HPCSR) in terms of the time needed for reaching the goal and the averaging amount of cold air needed for reaching the goal. The simulation results are shown in Figs. 11 and 12.

Fig.11 shows the time needed for reaching the goal, which requires to keep at least $M_{\rm S}$ users' satisfaction for consecutive $T_{\rm G}$ time steps. In the results, a smaller time means that the users may enter their sensation fewer times. Since this directly relates to the users' comfortableness, this factor should be the first priority in the optimization of the used parameters. Therefore, the used parameters shown in Table IV are the ones providing the almost best performance in terms of the averaging time for the last 200 epochs.

From Fig.11, we can observe that, for any N, (L, HPCSR) achieves the almost minimum of the convergence value. Especially, for $N = 24$, (L, HPCSR) much outperforms (L, HPC), which is the second best. This tendency implies that (L, HPCSR) is good at the scalability to the number of users.

The decreasing rate in Fig.11 is another important factor to be taken into account. Note that the decreasing rate means how much the time needed for reaching the goal drops at each epoch. We can observe that, for any N, (MV, ECU) exhibits the almost largest decreasing rate in early stage. However, the convergence values are much larger than the ones of (L, HPC) and (L, HPCSR). For $N = 3$, (A, HPC) achieves the largest decreasing rate. However, for $N = 12$ and 24, its decreasing rate is the smallest. This implies that (A, HPC) requires more time as N increases. For $N = 3$, (L, HPCSR) exhibits the smallest decreasing rate. However, (L, HPCSR) becomes better with N and for $N = 24$ it is the best among all the methods but (MV, ECU). This also implies that (L, HPCSR) is good at the scalability to the number of users.

Fig. 12 shows the averaging amount of cold air needed for reaching the goal at each epoch l. Since a smaller amount of cold air means more energy efficient, a smaller amount is preferable. The averaging values are calculated for the last $T_{\rm G}$ time steps[1].

From Fig. 12, we can observe that, for $N = 3$, (w/o, SPC) achieves the best performance. However, (w/o, SPC) is applicable only for the small number of users and it requires a large number of user inputs.

From here, we focus on the methods (MV, ECU), (A, HPC), (L, HPC) and (L, HPCSR). For every N, every

[1] In [8], the averaging amount of cold air is calculated for all the time steps spent at each epoch. The number of the spent time steps varies greatly depending on the methods. In general, the averaging amount decreases as the number of spent time steps increases. We regard that a redundant control is performed at early time steps in the epoch spending many time steps, and we calculate the averaging values for the last $T_{\rm G}$ time steps in this paper.

TABLE IV
PARAMETERS USED FOR EACH METHOD.

(a) For $N = 3$.

Item	(w/o,SPC)	(MV, ECU)	(A, HPC)	(L, HPC)	(L, HPCSR)
Discount rate γ	0.7	0.7	0.9	0.6	0.9
Learning rate α	0.2	0.08	0.4	0.1	0.05
Max. temp. in B.S. T_{b0}	60	3.0	4.0	6.0	5.0
Min. temp. in B.S. T_{b1}	50	1.0	0.2	0.5	0.5
Reduction rate ρ	0.3	–			
M_{S}	2				
M_{C}	1	–	1		
R_{L}	400	30	40	40	40
R_{S}	30	3	3	3	3
R_{N}	-400	-40	-40	-40	-40
R_{C}	–		-1	-1	-1
η	–	–	–	–	10

(b) For $N = 12$.

Item	(w/o, SPC)	(MV, ECU)	(A, HPC)	(L, HPC)	(L, HPCSR)
Discount rate γ	–	0.7	0.9	0.8	0.7
Learning rate α	–	0.09	0.6	0.3	0.2
Max. temp. in B.S. T_{b0}	–	3.0	5.0	4.0	4.0
Min. temp. in B.S. T_{b1}	–	1.0	0.1	0.1	0.1
M_{S}	–	8			
M_{C}	–	4			
R_{L}	–	30	40	40	40
R_{S}	–	4	3	5	3
R_{N}	–	-45	-40	-40	-40
R_{C}	–	–	-1	-10	-10
η	–	–	–	–	10

(c) For $N = 24$.

Item	(w/o, SPC)	(MV, ECU)	(A, HPC)	(L, HPC)	(L, HPCSR)
Discount rate γ	–	0.7	0.9	0.9	0.7
Learning rate α	–	0.1	0.6	0.3	0.05
Max. temp. in B.S. T_{b0}	–	3.0	5.0	5.0	7.0
Min. temp. in B.S. T_{b1}	–	1.0	0.1	0.1	0.1
M_{S}	–	12			
M_{C}	–	7			
R_{L}	–	30	40	40	40
R_{S}	–	4	4	3	3
R_{N}	–	-45	-40	-40	-40
R_{C}	–	–	-1	-10	-1
η	–	–	–	–	10

methods converge to approximately the value 0.2, which may be the optimal value when achieving the minimum value in terms of the number of time steps needed for reaching the goal. The difference among the methods exists in the convergence speed of the cooling amount. Unlike the case of the time steps, since a lower amount is better and the amount increases with the epoch l, a slower convergence is preferable.

According to the result, for every N, (MV, ECU) is almost the worst. This result is affected by the Excess-Cooling-Unaware Reward Function (ECU-RF)[2].

(A, HPC) is the worst for $N = 3$, but it is the best for $N = 12$ and 24. It is considered that this relates to the convergence speed in the time needed for reaching the goal. The convergence speed of (A, HPC) is the fastest for $N = 3$, but it is the slowest for $N = 12$ and 24.

(L, HPC) is the best for $N = 3$, but it is not good for $N = 12$ and 24. On the other hand, (L, HPCSR) keeps the second best position for any N. Further, as N increases, its performance becomes closer to the best one of (A, HPC). This implies that (L, HPCSR) is good at the scalability of the number of users in terms of the energy efficiency.

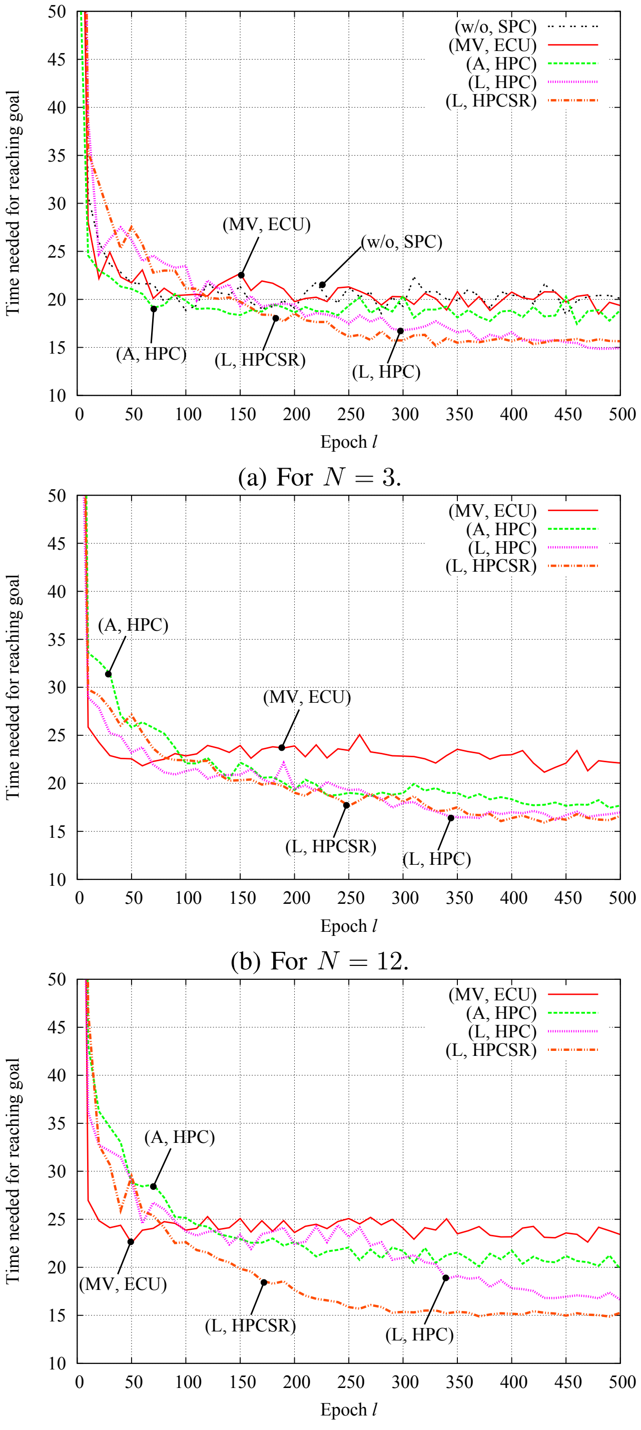
VI. CONCLUSION

In this paper, we proposed the air conditioner (AC) control system based on users' sensations. The purpose is to realize the control system that improves the low-performance AC in terms of energy efficiency and comfortableness performance. The system trains the control policy by using Q-learning with entered users' sensations. In order to cope with the large number of users, we proposed three types of integration methods of users' sensations. Especially, the proposed method based on likelihood was the most effective in terms of the time required for the control. Further, in order to reduce the energy consumption in cooling, we proposed four types of reward functions, which penalize the policy performing excess cooling. One of the four reward functions was newly proposed in this paper, in order to improve the scalability to the number of users. We have demonstrated the effectiveness of the proposed methods in terms of the control speed and the energy efficiency. Especially, it has been shown that the new proposal in this paper, (L, HPCSR), is good at the scalability of the number of users in terms of both the control speed and the energy efficiency.

[2] In [8], we cannot observe the effect of ECU. This is because the used parameter was not appropriate.

REFERENCES

[1] K. Sato, M. Samejima, M. Akiyoshi and N. Komada, "A Scheduling Method of Air Conditioner Operation using Workers Daily Action



(a) For $N = 3$.

(b) For $N = 12$.

(c) For $N = 24$.

Fig. 11. Time needed for reaching goal at each epoch l.

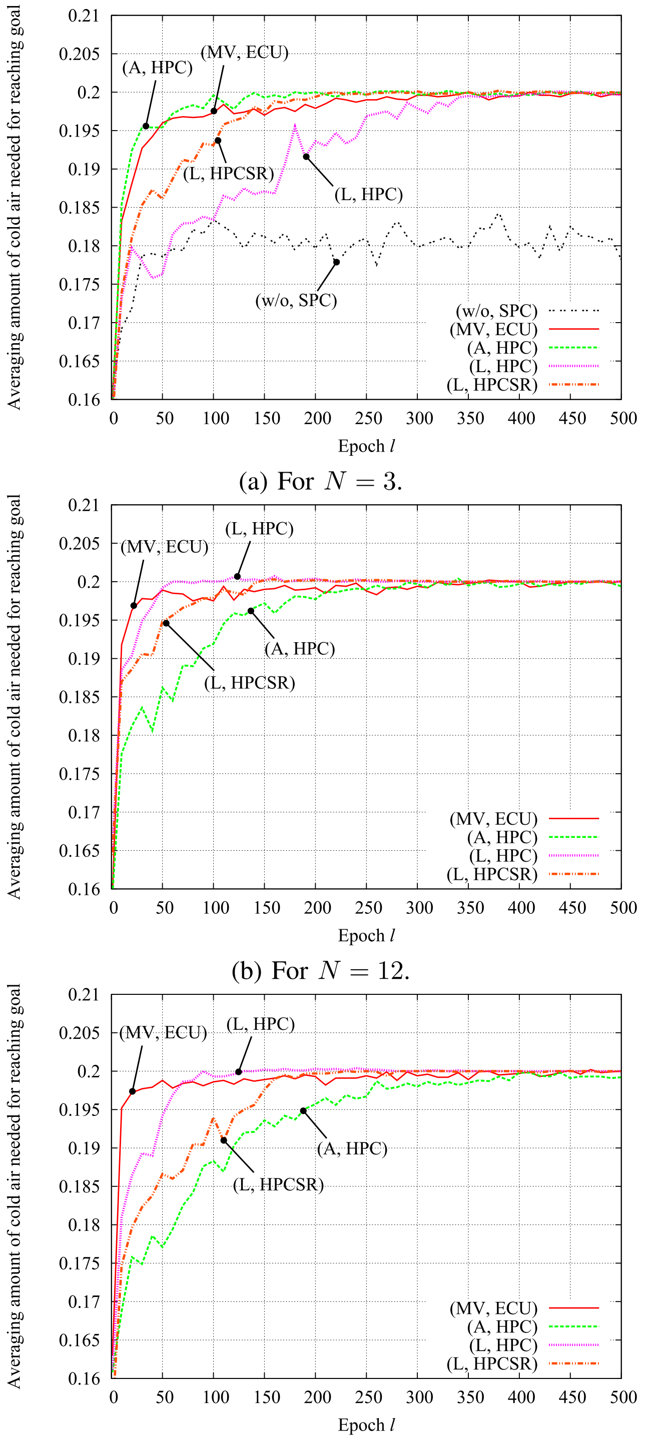
Plan towards Energy Saving and Comfort at Office," *2012 IEEE 17th Conference on Emerging Technologies and Factory Automation*, pp. 1-6, 2012.

[2] M. Bal, "Industrial applications of collaborative Wireless Sensor Networks: A survey," *2014 IEEE 23rd International Symposium on Industrial Electronics*, pp. 1463-1468, 2014.

[3] C. Tunca, H. Alemdar, H. Ertan, O.D. Incel and C. Ersoy, "Multimodal Wireless Sensor Network-Based Ambient Assisted Living in Real Homes with Multiple Residents," *Sensors*, Vol. 14, No. 6, pp. 9692-9719, 2014.

[4] J. Si, A. Barto, W. Powell and D. Wunsch, "Robust Reinforcement Learning for Heating, Ventilation, and Air Conditioning Control of Buildings," *Handbook of Learning and Approximate Dynamic Programming*, pp.517-534, Wiley-IEEE Press, 2004.

[5] T. Hiroyasu, A. Nakamura, M. Yoshimi, M. Miki and H. Yokouchi,



(a) For $N = 3$.

(b) For $N = 12$.

(c) For $N = 24$.

Fig. 12. Averaging amount of cold air needed for reaching goal at each epoch l.

"Lighting Control System using an Actor-Critic type Learning Algorithm," *2010 Second World Congress on Nature and Biologically Inspired Computing*, pp. 140-145, 2010.

[6] C.J.C.H. Watkins, "Learning from Delayed Rewards," Ph D Thesis, University of Cambridge, England, 1989.

[7] J.J. Grefenstette, "Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms," *Machine Learning*, Vol. 3, pp. 225-245, 1988.

[8] Y. Yamaguchi, N. Shigei and H. Miyajima, "Air Conditioning Control System Learning Sensory Scale Based on Reinforcement Learning," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2015, IMECS 2015, 18-20 March, 2015, Hong Kong, pp. 1-6, 2015.