Springer

Journal of Mechanical Science and Technology 23 (2009) 311~323

Journal of
Mechanical
Science and
Technology

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-0924-5

Tunnel ventilation control via an actor-critic algorithm employing nonparametric policy gradients †

Baeksuk Chu[1], Daehie Hong[1] and Jooyoung Park[2,*]

[1]*Division of Mechanical Engineering, Korea University, Seoul, 136-701, Korea*
[2]*Department of Control and Instrumentation Engineering, Korea University,*
Chungnam, 339-700, Korea

(Manuscript Received March 27, 2008; Revised August 12, 2008; Accepted September 22, 2008)

Abstract

The appropriate operation of a tunnel ventilation system provides drivers passing through the tunnel with comfortable and safe driving conditions. Tunnel ventilation involves maintaining CO pollutant concentration and VI (visibility index) under an adequate level with operating highly energy-consuming facilities such as jet-fans. Therefore, it is significant to have an efficient operating algorithm in aspects of a safe driving environment as well as saving energy. In this research, a reinforcement learning (RL) method based on the actor-critic architecture and nonparametric policy gradients is applied as the control algorithm. The two objectives listed above, maintaining an adequate level of pollutants and minimizing power consumption, are included into a reward formulation that is a performance index to be maximized in the RL methodology. In this paper, a nonparametric approach is adopted as a promising route to perform a rigorous gradient search in a function space of policies to improve the efficacy of the actor module. Extensive simulation studies performed with real data collected from an existing tunnel system confirm that with the suggested algorithm, the control purposes were well accomplished and improved when compared to a previously developed RL-based control algorithm.

Keywords: Actor-critic architecture; Nonparametric methods; Policy search; Reinforcement learning (RL); Tunnel ventilation control

1. Introduction

Recently, as the number of vehicles passing through tunnels has increased and the length of newly constructed tunnels has been extended, tunnel ventilation systems have drawn a great deal of interest [1,2]. All internal combustion engines produce exhaust gases containing noxious compounds and smoke such as CO, HC, NOx, dust etc. The toxic substances in the vehicular tunnels may cause fatal harm to the human body, and the low VI (visibility index) induced by smoke may considerably reduce drivers' safety due to poor visibility and even cause traffic accidents. Therefore, the amount of these substances should not exceed acceptable levels. A tunnel ventilation system provides drivers with a comfortable and safe driving environment by generating a sufficient amount of airflow and diluting the concentrations of noxious and dangerous contaminants to acceptable levels. In order to achieve this purpose, a tunnel ventilation system operates mechanical equipment such as jet-fans, blowers and dust collectors which consume large amount of energy. Therefore, it is desired to have an efficient operating algorithm for the tunnel ventilation in the aspects of saving energy as well as safe and comfortable driving environments.

The pollutants in a tunnel are exhausted from passing vehicles as moving sources. Moreover, their transient behavior is characterized by a time delay.

† This paper was recommended for publication in revised form by Associate Editor Hong Hee Yoo

*Corresponding author. Tel.: +82 11 9003 1810, Fax.: +82 41 865 1820
E-mail address: parkj@korea.ac.kr

Due to the complex and nonlinear system, it is difficult to control tunnel ventilation systems with conventional control methods. One of popular control methods for such systems is fuzzy logic control, and there have been many studies for tunnel ventilation control using fuzzy logic [3-5]. However, the tunnel ventilation control using the fuzzy logic has a few problems for building a rule database and determining appropriate membership functions. To overcome such problems, the reinforcement learning (RL) method was alternatively employed by us in [6]. This study basically follows the RL methodology [6] as the control scheme but with an improved learning algorithm. The RL method is a goal-directed learning of a mapping from situations to actions without relying on exemplary supervision or complete models of the environment. The goal of RL is to maximize a reward or reinforcement signal which is an evaluative feedback from the environment. In the process of constructing a reward of the tunnel ventilation system, maintaining pollutant concentration level under an allowable limit is the most important purpose. Energy consumption is also a significant factor and included in the reward formulation. Consequently, the controller used in this study is designed to optimally satisfy both control objectives through the learning process of RL.

RL has been an active research area in machine learning, control engineering, etc. [7-11]. Among many categories of RL, this study is based on the actor-critic algorithm which is sometimes called an adaptive-heuristic-critic (AHC) learning architecture [9]. In this class of learning structure, the controller is divided into two components: the critic (evaluation) module and the actor (control) module. Both modules have their own learning processes, respectively. In this research, the actor adjustment is determined by the 'nonparametric policy gradient' method which performs a gradient-based policy search in the feature space associated with the Gaussian kernel function [12]. Note that the policy search in a reproducing kernel Hilbert space gives a rigorous extension of the conventional policy search techniques to non-parametric settings [13]. The critic module is adjusted by the 'recursive least-squares (RLS)' based estimation algorithm in order to improve the efficiency of the use of data [14]. While most of previous studies about RL concentrated on narrow application areas such as controlling an inverted pendulum or a few robotics problems, this research shows that RL can be also implemented in various real-world systems.

Table 1. Specifications of Dunnae tunnel.

Tunnel	Dunnae
Length	3,300 m
Width	9.2 m
Height	7.2 m
Lane	2
Cross-sectional area	65.65 m^2
Ventilation	Jet-fan type

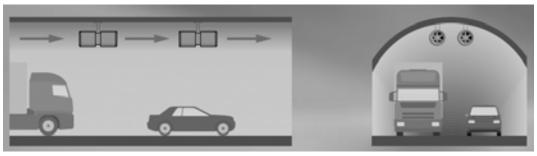

Fig. 1. Schematic diagram of Dunnae tunnel with jet-fans.

This paper is organized as follows. In Section 2, the target tunnel ventilation system is briefly introduced. In Section 3, the basic concepts of the RL and an actor-critic algorithm based on nonparametric policy gradient and RLS estimation scheme are described for tunnel ventilation control. Then, in Section 4, the results of simulations studies performed with real data collected from the target tunnel system are shown and the performance of the suggested controller is compared with a previously developed RL-based algorithm. It is confirmed that the suggested controller shows higher performance in terms of both maintaining pollutant concentration level under an allowable limit and saving energy consumption compared with the previously developed RL-based algorithm. Finally, the last section contains concluding remarks.

2. Tunnel ventilation system

The Dunnae Tunnel located on Youngdong highway in Korea was selected as the target system for this study. Fig. 1 and Table 1 show a schematic diagram and detailed specifications of the tunnel, respectively [6]. To observe the pollutant levels, CO and VI sensors were installed inside the tunnel at an appropriate interval. The traffic counter located at the tunnel entrance records the number of cars entering the tunnel. In order to ventilate the pollutants, a total of 32 jet-fans was installed on the ceiling.

The distribution of the pollutants inside the tunnel is usually expressed as a one-dimensional diffusion-advection equation [2, 4, 5],

$$\frac{\partial c}{\partial t}=\frac{\partial}{\partial x}\left(k\frac{\partial c}{\partial x}\right)-V_w\frac{\partial c}{\partial x}+q \tag{1}$$

where c is the pollutant concentration, x is the distance from the entrance of the tunnel, V_w is the wind velocity and k is the diffusion coefficient. The first term on the right-hand side explains the diffusion of the pollutants and the second term does the advection by wind. The pollutant source q from vehicles passing through the tunnel increases the pollutant level inside the tunnel, which is the only source to determine the tunnel pollutant distribution. This information is obtained from the real data collected from the target tunnel system for simulations. However, because the advection and source terms generally dominate the pollutant distribution, the diffusion term is often ignored. Then, the one-dimensional advection equation can be rewritten as

$$\frac{\partial c}{\partial t}=-V_w\frac{\partial c}{\partial x}+q \tag{2}$$

To estimate the change in pollutant distribution, it is necessary to identify the wind velocity inside the tunnel. It can be calculated by the force balance equation, which is expressed as

$$\begin{aligned}&\rho AL\frac{dV_w}{dt}=\sum F\\&\sum F=F_t+F_j+F_r+F_n\end{aligned} \tag{3}$$

where ρ is the density of air, A is the cross-sectional area of the tunnel, and L is the longitudinal length of the tunnel. $\sum F$ is the summation of the forces that affect the wind in its flow velocity inside the tunnel [15-17], which consist of following four elements:

- F_t : the traffic ventilation force by passing vehicles
- F_j : the equipment ventilation force by jet-fan operation
- F_r : the combination of the wall friction resistance and fluent loss at the entrance and exit
- F_n : the wind resistance by the natural wind outside the tunnel

Eqs. (2) and (3) are used for simulations evaluating the suggested controller and comparing with conventional algorithms. F_t is the traffic ventilation force by piston effect of vehicles passing through the tunnel. This traffic ventilation force is caused by drag force of vehicles which is composed of form drag configured by pressure-drop at the front of vehicles and pressure-recovery at the end of vehicles, and frictional drag induced by flow on the surface of vehicles. Since the piston effect by form drag dominates the traffic ventilation force, we considered only the piston effect by form drag for simulations, which is presented by

$$F_t=\sum_{k=1}^{N_t}\frac{\rho}{2}C_{d_k}A_{v_k}(V_k-V_w)\left|V_k-V_w\right| \tag{4}$$

where N_t is the total number of vehicles in the tunnel, C_{d_k} is the drag coefficient depending on vehicle type, A_{v_k} is the frontal area of vehicle, and V_k is the speed of vehicle. The number, speed, and type of vehicle are measured by the traffic counter and speedometer equipped at the entrance of the tunnel. Table 2 shows the frontal area and the drag coefficient depending on vehicle type and cross-sectional area of the tunnel.

The equipment ventilation force by jet-fan operation, F_j, is formulated by

$$F_j=\eta N_j\rho A_j\left|V_j\right|(V_j-V_w) \tag{5}$$

where η is the pressure-rise coefficient of jet-fan, A_j is the cross-sectional area of jet-fan, N_j is the number of jet-fans currently running, and V_j is the wind velocity discharged from jet-fan. The thrust force by jet-fan cannot be completely transferred to the ventilation force due to influences of wind velocity inside the tunnel, setup interval between equipment, gap between jet-fan and ceiling, and so on. Therefore, the pressure-rise efficiency is usually known as 80 to 95% [18]. In this study, the number of running jet- fans is the control variable manipulated by the controller. To achieve the control purposes of the target system, an RL-based intelligent controller which optimally adjusts the number of running jet-fans, will be designed in the following chapters.

Table 2. Frontal area and drag coefficient depending on vehicle type and cross-sectional area of the tunnel.

Cross-sectional area (m^2)	143		98		75		58		42	
Vehicle type	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small
Frontal area (m^2), A_v	7.11	2.31	7.11	2.31	7.11	2.31	7.11	2.31	7.11	2.31
Drag coefficient, C_d	0.74	0.48	0.92	0.53	1.09	0.59	1.31	0.66	1.68	0.78

Friction resistance, F_r, is modeled in terms of the wall friction resistance and fluent loss at the entrance and exit like the following:

$$\begin{aligned} F_r &= -f\frac{\rho}{2}\frac{L}{D}AV_w|V_w| - \zeta_i\frac{\rho}{2}AV_w|V_w| \\ &\quad -\zeta_e\frac{\rho}{2}AV_w|V_w| \end{aligned} \tag{6}$$

where f is the friction resistance coefficient of the tunnel wall, D is the hydraulic diameter of the tunnel, and ζ_i and ζ_e are the friction loss coefficients at the entrance and exit, respectively. The wall fiction resistance acts in the opposite direction of the flow and induces the increase of the thrust force of the tunnel ventilation equipment. Since this resistance force is caused by the wall friction inside the tunnel, it can be represented as a head loss equation due to the friction in the pipe or duct system. The friction resistance coefficient is generally assumed as 0.025 from empirical formula [19]. And the fluent loss at the entrance and exit is calculated by the frictional loss due to flow separation at the entrance and exit. The friction loss coefficient at the exit is chosen as 1 under turbulence flow condition, and the friction loss coefficient at the entrance is assumed as 0.6 in the case of round-type entrance [19].

The change in the pressure at the entrance and exit of the tunnel can be considered by the wind resistance by the natural wind outside the tunnel, F_n, which was studied by Blendermann [19]. It is represented as following equation,

$$F_n = C_n\frac{\rho}{2}AV_n^2\cos\psi\cos|\psi| \tag{7}$$

where C_n is the geometric compensation coefficient according to the shape of the entrance and exit, which is chosen as 0.25 [19], V_n is the velocity of the natural wind, and ψ is the incidence angle of the external wind heading toward the entrance or exit of the tunnel, which is assumed as 0 degree.

3. An actor-critic method employing nonparametric policy gradients

In this section, the RL algorithm based on the actor-critic architecture and nonparametric policy gradients is derived to solve the tunnel ventilation control problem. The derived algorithm draws control efforts from the actor distribution, and the adjustment is based on the strategy of performing a gradient-based search in the feature space associated with the Gaussian kernel function, which leads to an actor distribution update in a nonparametric setting.

3.1 Preliminaries

The actor-critic model includes two principal components, the critic (evaluation) module and the actor (control) module [7, 9]. The actor is used to generate optimal control actions according to a certain policy. The critic is used to evaluate the policy represented by the actor and to provide the actor with evaluation information. The critic and actor modules gradually converge toward optimal performance with their own learning processes. In this research, the actor adjustment is determined by nonparametric policy gradients, and the critic adjustment is decided by an RLS-based estimation algorithm.

Like other RL methodologies, the actor-critic algorithm is distinguished from different kinds of computational approaches by emphasizing the direct interaction with its environment, without relying on exemplary supervision or complete models of the environment. The critic receives the state vector and the external reinforcement signal, the reward which is an evaluative feedback, from the environment as inputs, and transforms them into the evaluation information for actor's policy improvement. Using the information from the critic and the environment, the actor outputs the control actions that tend to increase the long-run sum of the reward and gradually updates itself for more optimal performance.

General RL problems [7] can be represented via states $s \in S$, actions $a \in A$, rewards $r \in R$, and time steps $t \in \{0,1,2,\cdots\}$, in which a learning agent interacts with an environment. The objective of the learning agent is to pursue a policy that can maximize the discounted sum of rewards,

$$J(\pi) \triangleq E\left\{\sum_{i=0}^{\infty} \gamma^i r_i \mid s_0, \pi\right\}, \tag{8}$$

where $\gamma \in (0,1)$ is the discount rate, r_i is the immediate reward observed after the state transition from state s_i to s_{i+1}, s_0 is a designated start state, and π denotes the policy from which actions are chosen. Note that the case with the start state not fixed on a designated place but distributed across the state space can be easily handled by employing a probability distribution function for the start state. The action generating policies can be deterministic or stochastic, and when it is stochastic as in this paper, the policy is generally described by a conditional probability:

$$\pi(a \mid s) \triangleq p\{a_t = a \mid s_t = s\}. \tag{9}$$

Note that by introducing the state value function

$$V^{\pi}(s) \triangleq E\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t = s, \pi\right\} \tag{10}$$

and the state-action value function

$$Q^{\pi}(s,a) \triangleq E\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t = s, a_t = a, \pi\right\}, \tag{11}$$

one can rewrite the objective function in the following form [10]:

$$\begin{aligned} J(\pi) &= V^{\pi}(s_0) \\ &= \int_A \pi(a \mid s_0) Q^{\pi}(s_0, a) da \\ &= \int_S d^{\pi}(s) \int_A \pi(a \mid s) r(s,a) da ds, \end{aligned} \tag{12}$$

where $d^{\pi}(s)$ and $r(s,a)$ are a discounted state distribution and the expected reward, respectively.

3.2 Critic trained by RLS-TD(λ)

As mentioned before, the essence of the actor-critic methods is in using separate parameterized families for the actor part which is represented by the policy distribution $\pi_\theta(a \mid s)$, and the critic part which is represented by value functions. For the parameterized families for the critic part, this paper considers the function of the form

$$\tilde{V}_v(s) \triangleq \phi^T(s) v, \tag{13}$$

which approximates the state value function $V^{\pi_\theta}(s)$ for action-generating policy π_θ. Also, for the training of the critic part, we use the strategy of RLS-TD(λ) [14], derivation of which is described below for readers' convenience. From the Bellman equations [7, 10]

$$\begin{aligned} Q^{\pi_\theta}(s,a) &= r(s,a) + \gamma \int_S p(s' \mid s,a) V^{\pi_\theta}(s') ds', \\ V^{\pi_\theta}(s) &= \int_A \pi(a \mid s) Q^{\pi_\theta}(s,a) da, \end{aligned} \tag{14}$$

one can see that through a sampled trajectory, $V^{\pi_{\theta_i}}(s_i)$ can be approximated by $r_i + \gamma V^{\pi_{\theta_i}}(s_{i+1})$; thus $r_i + \gamma \tilde{V}_v(s_{i+1})$ is a valid estimate for the $V^{\pi_{\theta_i}}(s_i)$. Also from the usual strategy using the eligibility trace [7], one can see that in order for the approximator $\tilde{V}_v(s)$ to be useful in the t-th time step, it is desirable to minimize the following:

$$\begin{aligned} \Psi_t(v) &\triangleq \left\| \sum_{i=0}^{t} z_i (\tilde{V}_v(s_i) - (r_i + \gamma \tilde{V}_v(s_{i+1}))) \right\|^2 \\ &= \left\| \sum_{i=0}^{t} z_i (\phi^T(s_i) - \gamma \phi^T(s_{i+1})) v - \sum_{i=0}^{t} z_i r_i \right\|^2, \end{aligned} \tag{15}$$

where z_i is the eligibility trace vector defined via

$$\begin{aligned} z_i &= \gamma \lambda z_{i-1} + \phi(s_i), \; i = 1,2,\cdots, \\ z_0 &= \phi(s_0), \end{aligned} \tag{16}$$

and $\lambda \in [0,1]$ is the trace-decay parameter. Note that minimizing Eq. (15) is simply a least-squares problem utilizing the entire history of agent-environment interactions up to the t-th time step. When there is a need to put more emphasis on recent observations, the use of the so-called forgetting factor $\beta \in (0,1)$ is desirable. In this case, the following needs to be used instead of Eq. (15).

$$\tilde{\Psi}_t(v) \triangleq \left\| A_t v - b_t \right\|^2, \tag{17}$$

where

$$A_t \,\Box\, \sum_{i=0}^{t} \beta^{t-i} z_i (\phi^T(s_i) - \gamma\phi^T(s_{i+1})), \tag{18}$$

$$b_t \,\Box\, \sum_{i=0}^{t} \beta^{t-i} z_i r_i. \tag{19}$$

Note that for $t \geq 1$, the above A_t and b_t can be written in the following recursive form:

$$A_t = \beta A_{t-1} + z_t(\phi^T(s_t) - \gamma\phi^T(s_{t+1})), \tag{20}$$

$$b_t = \beta b_{t-1} + z_t r_t. \tag{21}$$

Also, note that when A_t is invertible, the optimal solution to the problem of minimizing Eq. (17) is obviously

$$v_t = A_t^{-1} b_t. \tag{22}$$

However, A_t is usually not invertible until a sufficient number of samples have been included in its summation. A common strategy used in the recursive least-squares method for ensuring the invertibility of A_t is to use δI for its initialization [14]. Employing the strategy leads to the use of

$$A_0 = \delta I + \phi(s_0)(\phi^T(s_0) - \gamma\phi^T(s_1)), \tag{23}$$

where δ is a positive number, instead of

$$\begin{aligned} A_0 &= z_0(\phi^T(s_0) - \gamma\phi^T(s_1)) \\ &= \phi(s_0)(\phi^T(s_0) - \gamma\phi^T(s_1)). \end{aligned} \tag{24}$$

Now, by applying the matrix inversion formula

$$(A + XY)^{-1} = A^{-1} - A^{-1}X(I + YA^{-1}X)^{-1}YA^{-1}, \tag{25}$$

to Eqs. (20)-(23), one can obtain recursive update rules for the solution v_t minimizing Eq. (17). More specifically, let

$$\begin{aligned} &A_0 = \delta I + \phi(s_0)(\phi^T(s_0) - \gamma\phi^T(s_1)), \\ &A_t \,\Box\, \beta A_{t-1} + z_t(\phi^T(s_t) - \gamma\phi^T(s_{t+1})) \text{ for } t \geq 1, \\ &P_t \,\Box\, A_t^{-1} \text{ for } t \geq 0, \end{aligned} \tag{26}$$

and $K_t \,\Box\, P_t z_t$ for $t \geq 0$. Then with the update rules

$$\begin{aligned} z_t &= \gamma\lambda z_{t-1} + \phi(s_t), \\ P_t &= \frac{1}{\beta}\left(P_{t-1} - \frac{P_{t-1} z_t (\phi^T(s_t) - \gamma\phi^T(s_{t+1})) P_{t-1}}{\beta + (\phi^T(s_t) - \gamma\phi^T(s_{t+1})) P_{t-1} z_t} \right), \\ K_t &= \frac{P_{t-1} z_t}{\beta + (\phi^T(s_t) - \gamma\phi^T(s_{t+1})) P_{t-1} z_t}, \end{aligned} \tag{27}$$

the critic parameter vector v_t minimizing Eq. (17) can be obtained by

$$v_t = v_{t-1} + K_t(r_t - (\phi^T(s_t) - \gamma\phi^T(s_{t+1}))v_{t-1}. \tag{28}$$

Note that the resultant state approximator $\tilde{V}_{v_t}(s) = \phi^T(s)v_t$ plays an important role in the update process for the actor part. In the following, we derive a way to incorporate the use of nonparametric policy gradients for updating the actor part.

3.3 Actor trained via nonparametric policy gradients

The main role of the actor is to generate actions via a parameterized family. At each state $s \in S$, an action $a \in A$ is drawn in accordance with the conditional distribution $\pi_\theta(a \mid s)$, where θ is the parameter vector characterizing the distribution. Thus, the objective we seek to maximize can be written as follows:

$$J(\pi) = J(\theta) = \int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a \mid s) r(s,a) da ds. \tag{29}$$

One of the convenient strategies for seeking the best distribution parameter θ is to utilize the direction of $\nabla_\theta J(\theta)$, which is often called the policy gradient. According to the famous policy gradient theorem [8,10], the policy gradient can be written as follows[1]:

$$\begin{aligned} &\nabla_\theta J(\theta) \\ &= \nabla_\theta \left(\int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a \mid s) r(s,a) da ds \right) \\ &= \int_S d^{\pi_\theta}(s) \int_A \nabla_\theta \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) da ds \\ &= \int_S d^{\pi_\theta}(s) \int_A \nabla_\theta \pi_\theta(a \mid s) \\ &\quad \cdot (Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)) da ds \\ &= \int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) \\ &\quad \cdot (Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)) da ds. \end{aligned} \tag{30}$$

[1] Note that the equality

$$\begin{aligned} &\int_S d^{\pi_\theta}(s) \int_A \nabla_\theta \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a) da ds \\ &= \int_S d^{\pi_\theta}(s) \int_A \nabla_\theta \pi_\theta(a \mid s)(Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)) da ds \end{aligned}$$

holds true because $\int_A \nabla_\theta \pi(s,a) da = 0$ for $\forall s \in S$. Also note that in this paper, we assume that the policy distribution π_θ is such that the gradient $\nabla_\theta \log \pi_\theta(a \mid s)$ is well-defined.

From the Bellman equation [7, 10]

$$Q^{\pi_\theta}(s,a) = r(s,a) + \gamma \int_S p(s' \mid s,a) V^{\pi_\theta}(s') ds', \qquad (31)$$

We see that through a sampled trajectory, $Q^{\pi_{\theta_k}}(s_k, a_k)$ can be approximated by $r_t + \gamma V^{\pi_{\theta_k}}(s_{k+1})$; thus $r_t + \gamma \tilde{V}_{v_k}(s_{k+1})$ and $\tilde{V}_{v_k}(s_k)$ are valid estimates for $Q^{\pi_{\theta_k}}(s_k, a_k)$ and $V^{\pi_{\theta_k}}(s_k)$, respectively. Hence, these approximation-steps via the sampled trajectory yield the following estimate:

$$\begin{aligned}
&[\nabla_\theta J(\theta)]_{\theta=\theta_t} \\
&= \int_S d^{\pi_\theta}(s) \int_A \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) \\
&\quad \cdot (Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s))\Big|_{\theta=\theta_t} dads \\
&\approx (Q^{\pi_{\theta_t}}(s_t, a_t) - V^{\pi_{\theta_t}}(s_t))[\nabla_\theta \log \pi_\theta(a_t \mid s_t)]_{\theta=\theta_t} \qquad (32) \\
&\approx (r_t + \gamma \tilde{V}_{v_t}(s_{t+1}) - \tilde{V}_{v_t}(s_t))[\nabla_\theta \log \pi_\theta(a_t \mid s_t)]_{\theta=\theta_t} \\
&= (r_t + \gamma \phi^T(s_{t+1}) v_t - \phi^T(s_t) v_t) \\
&\quad \cdot [\nabla_\theta \log \pi_\theta(a_t \mid s_t)]_{\theta=\theta_t}
\end{aligned}$$

Therefore, one can update the actor parameter θ via the following simple gradient-based rule:

$$\begin{aligned}
\theta_{t+1} &\leftarrow \theta_t + \alpha [\nabla_\theta J(\theta)]_{\theta=\theta_t} \\
&\approx \theta_t + \alpha (\widetilde{\mathrm{TD}}_t) [\nabla_\theta \log \pi_\theta(a_t \mid s_t)]_{\theta=\theta_t},
\end{aligned} \qquad (33)$$

in which $\alpha > 0$ is the learning rate, and $\widetilde{\mathrm{TD}}_t$ is the temporal difference computed by

$$\widetilde{\mathrm{TD}}_t \ \square \ r_t + \gamma \tilde{V}_{v_t}(s_{t+1}) - \tilde{V}_{v_t}(s_t). \qquad (34)$$

In this paper, we consider the Gaussian actor policy whose distribution is set as

$$\pi_\theta(a \mid s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{-(a - \theta^T \psi(s))^2}{2\sigma^2} \right\}. \qquad (35)$$

In order to make rich functional expressions possible with $\theta^T \psi(s)$ in Eq. (35), this paper considers the case that $\psi : S \to F$ is the feature map associated with the Gaussian Mercer kernel $k : S \times S \to R$ (For details on the Mercer kernels, see *e.g.*, [12]). Hence, in the corresponding feature space F , the following 'kernel trick' holds true [12]:

$$< \psi(x), \psi(z) >= k(x,z) = \exp(-\|x - z\|^2 / 2\sigma_0^2) \qquad (36)$$

From the chain rule, one can compute the following gradient:

$$\nabla_\theta \log \pi_\theta(a_t \mid s_t) = (a_t - \theta^T \psi(s_t)) \psi(s_t) / \sigma^2 \qquad (37)$$

Hence, the update rule Eq. (33) can now be written as follows:

$$\begin{aligned}
\theta_{t+1} &\leftarrow \theta_t + \alpha [\nabla_\theta J(\theta)]_{\theta=\theta_t} \\
&\approx \theta_t + \alpha (\widetilde{\mathrm{TD}}_t)(a_t - \theta^T \psi(s_t)) \psi(s_t) / \sigma^2
\end{aligned} \qquad (38)$$

According to this update rule, the actor parameter θ_t can be represented in the following form:

$$\theta_t = \sum_{i=1}^{t-1} c_i \psi(s_i), \qquad (39)$$

where $c_i \ \square \ \alpha(\widetilde{\mathrm{TD}}_i)(a_i - \theta_i^T \psi(s_i)) / \sigma^2$ is the coefficient introduced for simpler notation. Note that Eq. (39) corresponds to the so-called 'representer theorem' widely used in kernel methods [12]. Now by plugging Eq. (39) into Eq. (37) and then applying the kernel trick of Eq. (36) to the result, one can obtain the following nonparametric policy gradient:

$$\begin{aligned}
&\nabla_\theta \log \pi_\theta(a_t \mid s_t) \\
&= \left(a_t - \sum_{i=1}^{t-1} c_i k(s_i, s_t) \right) \psi(s_t) / \sigma^2
\end{aligned} \qquad (40)$$

Also, the actor distribution at time t becomes

$$\begin{aligned}
&\pi_\theta(a_t \mid s_t) \\
&= \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\left(a_t - \sum_{i=1}^{t-1} c_i k(s_i, s_t) \right)^2 / 2\sigma^2 \right\}.
\end{aligned} \qquad (41)$$

Here the mean of the Gaussian, $\sum_{i=1}^{t-1} c_i k(s_i, s_t)$, is nonparametric.

Considering the function form of the above actor distribution, there are things in its present form we need to worry about - growing sample sizes. To overcome this difficulty with a more parsimonious representation, we adopt the so-called ALD (approximate linear dependency) analysis [20]. The sparsification procedure using the ALD analysis can be summarized as follows [20]: First, we assume that after having observed samples $s_0, \cdots, s_{t-1}$, we have collected a dictionary consisting of a subset of the samples $D_{t-1}=\{\tilde{s}_i\}_{i=1}^{m_{t-1}}\subset\{s_i\}_{i=1}^{t-1}$. Entries of the dictionary are required of the property that $\{\psi(\tilde{s}_i)\}_{i=1}^{m_{t-1}}$ are linearly independent. When a new sample s_t is given, we test whether $\psi(s_t)$ is approximately linearly dependent on the previous dictionary vectors by

$$\delta_t \,\square\, \min_{\{a_0,\cdots,a_{m_{t-1}}\}}\left\|\sum_{i=0}^{m_{t-1}} a_i\psi(\tilde{s}_i)-\psi(s_t)\right\|^2 \le \upsilon, \qquad (42)$$

where $\upsilon > 0$ is a user-defined accuracy parameter. The optimization problem in Eq. (42) can be easily transformed into a convex quadratic program by use of the kernel trick of Eq. (36), thus solving it is straightforward. Also, if the ALD condition of Eq. (42) is met by the new sample s_t, $\psi(s_t)$ can be safely approximated as follows:

$$\psi(s_t)\approx\sum_{i=0}^{m_{t-1}} a_i^*\psi(\tilde{s}_i), \qquad (43)$$

where

$$\{a_0^*,\cdots,a_{m_{t-1}}^*\} \qquad (44)$$
$$\square\, \arg\min_{\{a_0,\cdots,a_{m_{t-1}}\}}\left\|\sum_{i=0}^{m_{t-1}} a_i\psi(\tilde{s}_i)-\psi(s_t)\right\|^2 .$$

In the otherwise case (*i.e.*, when it turns out that $\psi(s_t)$ is not approximately linearly dependent on the feature vectors of the samples in dictionary D_{t-1}), we add it to the dictionary (*i.e.*, $D_t \leftarrow D_{t-1}\cup\{s_t\}$). In this way, all samples up to time t can be approximated as linear combinations of the vectors in D_t. Consequently, the actor parameter θ_t and distribution $\pi_{\theta_t}(a_t\mid s_t)$ can be safely approximated as follows, and in the process of updating the actor parameter, we can use the following sparse representation for Eq. (38):

$$\theta_t\approx\sum_{i=0}^{m_{t-1}}\tilde{c}_i\psi(\tilde{s}_i) \qquad (45)$$

$$\begin{aligned}
&\pi_\theta(a_t\mid s_t)\\
&\approx\frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\left(a_t-\sum_{i=0}^{m_{t-1}}\tilde{c}_i k(\tilde{s}_i,s_t)\right)^2/2\sigma^2\right\}\\
&=\frac{1}{\sigma\sqrt{2\pi}} \qquad (46)\\
&\cdot\exp\left\{-\frac{\left(a_t-\sum_{i=0}^{m_{t-1}}\tilde{c}_i\exp(-\|\tilde{s}_i-s_t\|^2/2\sigma_0^2)\right)^2}{2\sigma^2}\right\}
\end{aligned}$$

More details on the steps for sparsification via ALD analysis are well described in [20].

3.4 Algorithm

The algorithm considered in this paper can be summed up as repeating two tasks: An agent-environment interaction task in which the agent interacts with its environment with an action generated by the current policy and observes the consequence of the interaction, and a task for the value estimation and action improvement in which the agent optimizes its policy by updating the actor and critic parameters on the basis of the non-parametric policy gradient and the recursive least-squares. More precisely, the sequence of the considered algorithm is as follows:

```
Given:
- $k(x,z)=\exp(-\|x-z\|^2/2\sigma_0^2)$, Gaussian kernel function, in use for the actor distribution
- Basis functions $\phi(s)\,\square\,[\phi_1(s)\cdots\phi_K(s)]^T$ in use for $\tilde{V}_v(s)\,\square\,\phi(s)^T v$, which approximates the state value function
- Forgetting factor $\beta\in(0,1)$
- Discount rate $\gamma\in(0,1)$
- Trace-decay parameter $\lambda\in[0,1]$
- Constant $\delta>0$
- Accuracy parameter $\upsilon$ for determining the level of sparsity in kernel expansion

Initialize the state and learning parameters. Also, set the dictionary to be a null set.

for $t:=0,1,2,\cdots$ do
    (1) According to the current state $s_t$, compute $\pi_{\theta_t}(a_t\mid s_t)$. Then, draw a control action $a_t$ from the distribution $\pi_{\theta_t}(\cdot\mid s_t)$.
    (2) Take the action $a_t$, and observe the reward $r_t$ and the next state $s_{t+1}$.
    (3) Use the RLS-TD($\lambda$) rules Eqs. (27) and (28) to update $v_t$ of the critic.
    (4) Compute the temporal difference of Eq. (34).
    (5) Check the ALD condition by computing $\delta_t$ of Eq. (42). If $\delta_t<\upsilon$, $\psi(s_t)$ is approximated as in Eq. (43). If not, $s_t$ is added to the dictionary.
    (6) Use Eqs. (38), (45) and (46) to update the actor distribution.
end
```

4. Simulations and experimental results

A large number of previous studies about RL-based algorithms have been confined to theoretical applications such as the inverted pendulum control or a few robotics problems. However, this paper explores practical implementation of the developed RL-based method on a real system: a tunnel ventilation system. The proposed control algorithm is verified with computer simulations. Simulation data were gathered from a real tunnel system, Dunnae Tunnel located in Youngdong highway in Korea. The states measured by the sensors consist of CO pollutant levels, VI, and pollutant emission rate by passing vehicles. It is noted that only the CO level and pollutant emission rate is considered in the control design and adding a VI level to the control algorithm is quite straightforward.

To solve the continuous state space problem in RL, a linear function approximator is used for the state value function estimator in the critic. It is designed as a linear combination of three basis components parameterized by a weight vector like the following:

$$\tilde{V}_v(s) \Box \phi^T(s)v \tag{47}$$

where $\phi^T(s) = [\phi_1(s) \quad \phi_2(s) \quad 1]$ is used as the basis vector. The first component of the basis $\phi_1(s)$ is the normalized difference of the CO sensor feedback from an allowable reference CO pollutant level, 25 ppm in this study. The second basis $\phi_2(s)$ is the normalized difference between the average reference emission rate and currently observed emission rate. The third component is a bias term. The control output of the proposed algorithm is the relative number of running jet-fans to the nominal number of which the jet-fans are operated under the condition of nominal pollutant level. The total number of jet-fans which can be driven is 32 and the nominal number is chosen as 15. In the actor module, a normal distribution is employed as the density $\pi_\theta(a \mid s)$ that governs the control output selection. The actual action a is chosen by exploring a range around the mean point, which is determined by the kernel-based functional expression $\theta^T\psi(s)$ with a density function expressed as

$$\pi_\theta(a \mid s) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{-(a-\theta^T\psi(s))^2}{2\sigma^2} \right\} \tag{48}$$

where the variance σ is constant. The selected control output corresponding to the relative number of running jet-fans has the critical role of determining the wind velocity in the force balance equation, Eq. (3). With the wind velocity and the measured emission rate, the distribution of pollutants is identified in the governing equation, Eq. (2).

As mentioned in section 3, the purpose of the learning agent is to obtain a policy that can maximize the discounted sum of rewards. Therefore, the reward formulation is a main criterion for the RL process and an important connection between the control algorithm and the system. The reward reflects the objective to be achieved by the controller and a penalty for violating a constraint of the system. In this study, the reward has been constructed by combining the pollutant reduction term as the objective with the energy consumption term as the constraint. In Eq. (49), the pollutant level over an allowable limit and the energy consumption proportional to the number of running jet-fans are combined with a weighting factor, K.

$$\text{reward} = \begin{cases} -\left\{(CO_{current} - CO_{ref}) + K \cdot E_{JF}\right\} \\ \qquad\qquad\qquad ,\text{if } CO_{current} > CO_{ref} \\ -E_{JF} \qquad\quad ,\text{if } CO_{current} < CO_{ref} \end{cases} \tag{49}$$

where CO_{ref} is the allowable reference CO pollutant level, 25ppm, $CO_{current}$ is the current CO sensor feedback, and E_{JF} is the energy consumed by the operation of jet-fans. In RL methodology, reward usually does not have any unit or dimension but is given as a real number. In Eq. (49), K is just a simple weighting factor between the two control criteria in reward formulation; thus it has no unit.

The proposed approach was applied to the target system with the following parameters:

- Initial value function parameter vector $v_0 = [0 \quad 0 \quad 0]^T$
- Learning rate $\alpha = 0.5$
- Forgetting factor $\beta = 0.99$
- Discount rate $\gamma = 0.5$
- Trace-decay parameter $\lambda = 0.5$
- Constant $\delta = 20$
- Accuracy parameter $\upsilon = 0.000004$
- Width of the Gaussian kernel $\sigma_0 = 1.0$
- Standard deviation of the actor distribution $\sigma = 1.5$

- Reward weighting factor $K = 0.06$

While a time step is 1 min, simulations are implemented for 5000 time steps which are equivalent to10 replications of real data for 500 samples. Fig. 2 shows the peak value of CO inside the tunnel for 5000 time steps about the 'uncontrolled case'. Fig. 3 describes the 3-D plot of the pollutant distribution for the last 50 time steps. In this study, the 'uncontrolled' case is defined as when only a nominal number of jet-fans, which is chosen as 15 among the whole 32 jet-fans, is constantly being operated. Therefore, the pollutant emission by passing vehicles is the only input source to the system. In order to obtain the pollutant emission rate, the traffic volume information from the real tunnel is used. Then, the CO pollutant distribution inside the tunnel is determined from the governing equations, Eqs. (2) and (3), making use of the real traffic volume information and various parameters identifying the tunnel. In this case, any control input except the operation of the nominal number of jet-fans is not conducted to ventilate the tunnel. As such, it is shown that the maximum CO pollutant level considerably exceeds 25ppm, the control objective.

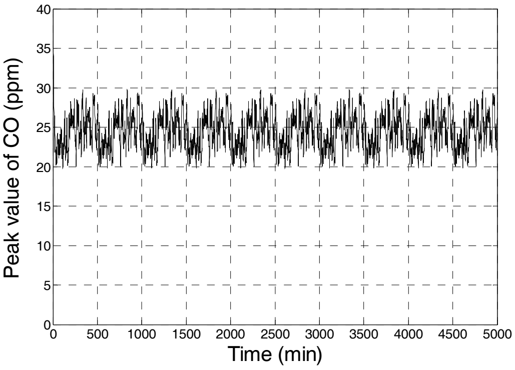

Fig. 2. Peak value of CO for the whole 5000 time steps about the 'uncontrolled case'.

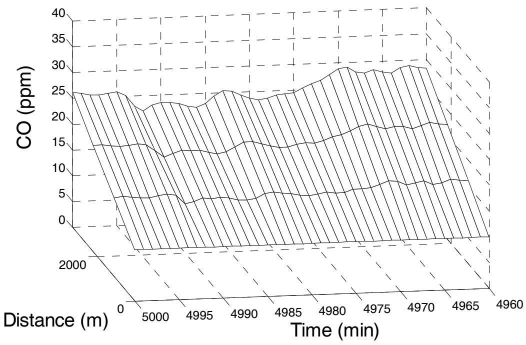

Fig. 3. 3-D plot of the pollutant distribution for the last 50 time steps about the 'uncontrolled case' (CO vs. time and longitudinal distance along tunnel).

Using the same actual traffic data as in the case of the uncontrolled system, simulations for the 'controlled' system were performed. If control inputs based on the RL algorithm are added to the system, the performance of the systems in terms of the reduction of the CO pollutant level and energy consumption is significantly improved. Fig. 4 describes the learning process of the proposed RL-based controller through a sample case. As the learning progresses, the controller increases the number of running jet-fans, such that the CO level is maintained under the allowable limit of 25ppm. On the other hand, if the CO pollutant is maintained well below the allowable limit and excessive energy is consumed by running unnecessary overworking jet-fans, the controller decreases the number of jet-fans and saves the energy consumption. These two facts explain that the RL-based controller appropriately follows the control objectives expressed by the reward formulation for this system. After a sufficient time is spent for learning, as shown in Fig. 5, the CO pollutant level along time axis stays near the allowable limit and the energy consumption becomes very efficient, which is explained later with a table for comparison. In this sample case, the time it takes to learn the new system characteristics is approximately observed as 2500 steps (2500 min).

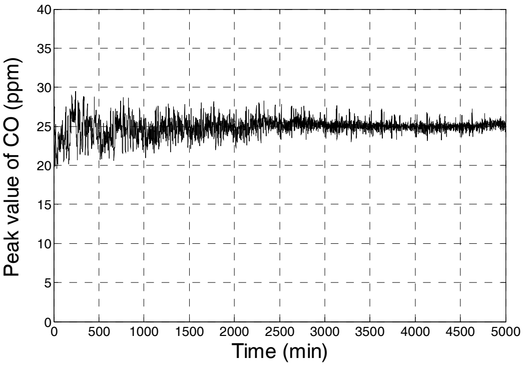

Fig. 4. Peak value of CO for the whole 5000 time steps about the 'controlled case by the proposed RL-based controller'.

Table 3. Mean, standard deviation, maximum/minimum value of peak CO level and consumed energy during the last 500 time steps averaged with 10 episodic tasks.

Case	CO level (ppm)				Energy (kWh)
	CO_{mean}	CO_{std}	CO_{max}	CO_{min}	
Uncontrolled (constant operation of nominal number of jet-fans)	24.48	2.23	29.71	19.78	3750
Controlled with the proposed RL-based controller	25.05	0.72	27.30	22.59	3396
Controlled with a previously developed RL-based controller	24.34	0.95	27.42	21.87	3617

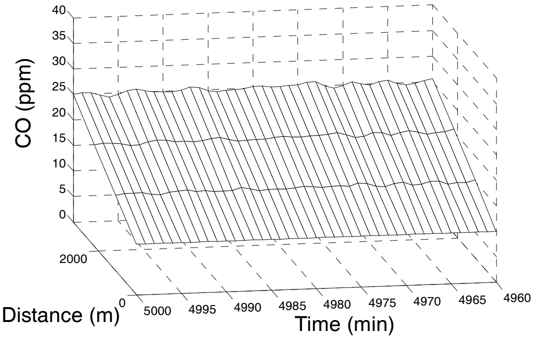

Fig. 5. 3-D plot of the pollutant distribution for the last 50 time steps about the 'controlled case by the proposed RL-based controller' (CO vs. time and longitudinal distance along tunnel).

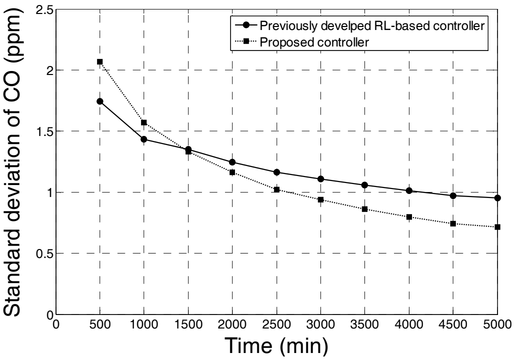

Fig. 6. Standard deviations averaged with 10 episodic tasks for the proposed controller and a previously developed RL-based controller.

In order to analyze the performance of the tunnel ventilation system quantitatively, the standard deviation for the recent 500 samples of peak CO values is calculated every 500 steps during the learning process. Fig. 6 represents the result of the standard deviation analysis averaged with 10 episodic tasks for the proposed controller and a previously developed RL-based controller [6]. The controllers which are trained to achieve the objectives explained above should have low standard deviations. Both results show the decreasing trend of the standard deviations along the time axis. While the previously developed controller shows lower standard deviations during the early steps, the suggested controller finally approaches to lower standard deviations as the learning progresses. A more rigorous formulation of the proposed action selection policy employing nonparametric settings seems to need more steps to be optimized at first, however in the end obtains a superior performance compared to the conventional policy search technique.

The performance of the proposed control method is evaluated with respect to the reduction of CO pollutant level and energy consumption. Table 3 shows the mean value, standard deviation, maximum/ minimum value of peak CO level and consumed energy during the last 500 time steps averaged with 10 episodic tasks. Among three cases, the uncontrolled case corresponds to Figs. 2 and 3, while the controlled case based on the proposed RL method does to Figs. 4 and 5. The third case is about the previously developed RL-based controller.

Three cases have similar mean values of the CO level. However, the maximum CO levels for the controlled cases are lower than that of the uncontrolled case. In addition, the energy consumption is also lower with the controlled cases. These results show that the RL-based control achieves the control objectives of pollutant reduction and low energy consumption. Moreover, when compared to the previously developed RL-based controller, the suggested controller has lower standard deviations as well as lower energy consumption, which means it accomplished an improved performance.

5. Concluding remarks

In order to control a tunnel ventilation system efficiently, a reinforcement learning method based on an actor-critic architecture and nonparametric policy gradients was used. The recursive least-squares (RLS) method was employed for the learning process so as to improve the control performance of the system. The control objectives included maintaining the pollutant concentration level under an appropriate limit and minimizing the control effort. By importing the objectives into the reward formulation of the RL method, a tunnel ventilation controller was designed to produce an optimal control input. The proposed controller was verified through various simulations compared with a previously developed RL-based controller. It was confirmed that the RL-based method enables high performance of the developed system in terms of managing an appropriate pollutant concentration level and saving energy consumption, and the suggested controller accomplishes higher performance than the previous one.

Acknowledgment

This work was supported by a Korea University Grant.

References

[1] A. Bring, T.-G. Malmstrom and C. A. Boman, Simulation and measurement of road tunnel ventilation, *Tunnelling and Underground Space Technology*, 12 (3) (1997) 417-424.

[2] L. Ferkl and G. Meinsma, Finding optimal ventilation control for highway tunnels, *Tunnelling and Underground Space Technology*, 22 (2007) 222-229.

[3] B. Chu, D. Kim, D. Hong, J. Park, J. T. Chung and T. -H. Kim, GA-based fuzzy controller design for tunnel ventilation systems, *Automation in Construction*, 17 (2008) 130–136.

[4] M. Funabashi, I. Aoki, M. Yahiro and H. Inoue, A fuzzy model based control scheme and its application to a road tunnel ventilation system, *Proc. of IECON '91 International Conference on Industrial Electrics, Control and Instrumentation*, 2 (1991) 1596-1601.

[5] P. H. Chen, J. H. Lai and C. T. Lin, Application of fuzzy control to road tunnel ventilation system, *Fuzzy Sets and Systems*, 100 (1998) 9-28.

[6] B. Chu, D. Kim, D. Hong, J. Park, J. T. Chung and T.-H. Kim, Tunnel ventilation control using reinforcement learning methodology, *JSME International Series C*, 47 (3) (2004) 939-945.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, (1998).

[8] R. S. Sutton, D. McAllester, S. Singh and Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, *Advances in Neural Information Processing Systems*, 12 (1999) 1057-1063.

[9] L. P. Kaelbling, M. L. Littman and A. W. Moore, Reinforcement learning: a survey, *Journal of Artificial Intelligence Research*, 4 (1996) 237-285.

[10] J. Peters, S. Vijayakumar and D. Schaal, Reinforcement learning for humanoid robotics, *Proc. of the 3rd IEEE-RAS International Conference on Humanoid Robots*, Karlsruhe, Germany, (2003).

[11] J. Park, J. Kim and D. Kang, An RLS-based natural actor-critic algorithm for locomotion of a two-linked robot arm, *Lecture Notes in Artificial Intelligence*, 3801 (2005) 65-72.

[12] B. Scholkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, (2002).

[13] J. A. Bagnell and J. Schneider, Policy search in reproducing kernel Hilbert space, *Technical Report CMU-RI-TR-03-45*, Robotics Institute, Carnegie Mellon University, (2003).

[14] X. Xu, H. G. He and D. Hu, Efficient reinforcement learning using recursive least-squares methods, *Journal of Artificial Intelligence Research*, 16 (2002) 259-292.

[15] Metropolitan Expressway Public Corporation, The report of ventilation system of tunnel, Tokyo, (1993).

[16] L. Kurka, L. Ferkl, O. Sladek and J. Porizek, Simulation of traffic, ventilation and exhaust in a complex road tunnel, *Proc. of IFAC*, Prague, (2005).

[17] H. -M. Jang and F. Chen, A novel approach to the transient ventilation of road tunnels, *Journal of Wind Engineering and Industrial Aerodynamics*, 86 (2000) 15-36.

[18] A. D. Martegani, G. Pavesi and C. Barbetta, The influence of separation, inclination and swirl on single and coupled jet fans installation efficiency, *Proc. of the 9th International Symposium on the Aerodynamics and Ventilation of Vehicle Tunnels*, (1997) 43-55.

[19] W. Blendermann, On a probabilistic approach to the influence of wind on the longitudinal ventilation of road tunnels, *Proc. of the 2nd International Symposium on the Aerodynamics and Ventilation of Vehicle Tunnels*, (1976) B1-1-B1-24.

[20] Y. Engel, S. Mannor and R. Meir, The kernel recursive least-squares algorithm, *IEEE Transactions on Signal Processing*, 52 (2004) 2275-2285.

Baeksuk Chu received his B.S. degree in 1999, M.S. degree in 2001, and Ph.D. in 2006, respectively in Mechanical Engineering from Korea University. Dr. Chu is currently a Research Professor at the Division of Mechanical Engineering at Korea University in Seoul, Korea. Dr. Chu's research interests are in the area of robotics, control engineering, and reinforcement learning.

Daehie Hong received his B.S. degree in 1985 and M.S. degree in 1987, respectively in Mechanical Engineering from Korea University. He then went on to receive his Ph.D. degree from UC Davis in 1994. Dr. Hong is currently a Professor at the Division of Mechanical Engineering at Korea University in Seoul, Korea. He is currently serving as an Editor of the International Journal of Precision Engineering and Manufacturing. Dr. Hong's research interests are in the area of mechatronics and field robotics.

Jooyoung Park received his B.S. degree in Electrical Engineering from Seoul National University in 1983, and his Ph.D. degree in Electrical and Computer Engi-neering from the University of Texas at Austin in 1992. He joined Korea University in 1993, where he is currently a Professor in the Department of Control and Instrumentation Engineering. Dr. Park's recent research interests are in the area of rein-forcement learning and kernel methods.