

Development of a Modified Random Signal-based Learning using Simulated Annealing

Chang-Wook Han^{1,*}, Yeunghak Lee²

Abstract

This paper describes the application of a simulated annealing to a random signal-based learning. The simulated annealing is used to generate the reinforcement signal which is used in the random signal-based learning. Random signal-based learning is similar to the reinforcement learning of neural network. It is poor at hill-climbing, whereas simulated annealing has an ability of probabilistic hill-climbing. Therefore, hybridizing a random signal-based learning with the simulated annealing can produce better performance than before. The validity of the proposed algorithm is confirmed by applying it to two different examples. One is finding the minimum of the nonlinear function. And the other is the optimization of fuzzy control rules using inverted pendulum.

Key Words: Fuzzy Controller, Neural Network, Optimization, Simulated Annealing.

I. INTRODUCTION

The strong nonlinear behavior and uncertainties of many practical systems and the unavailability of quantitative data regarding the input-output relations make an analytic approach to the behavior analysis of the system even more difficult. Hence, many engineers and scientists have focused their attentions on the computational intelligence-based approaches such as fuzzy logic, neural network, genetic algorithms and etc. In spite of many practical successes, the process for fine tuning of fuzzy control rules in order to get the desired performance is a time-consuming and tedious task even though linguistic control rules are extracted from experts' knowledge. To overcome these difficulties, the self-organizing fuzzy controller was proposed by Mamdani[1] and extended by others.

The simulated annealing technique was introduced by Kirkpatrick et. al.[2] in 1983. The foundations for this algorithm are given by Matropolis et. al. [3]. The simulated annealing algorithm is based on an analogy with the annealing of solids. Downhill moves are always

accepted, whereas uphill moves are accepted with the probability that is a function of temperature.

Random signal-based learning [4], introduced by authors of this paper, is similar to a reinforcement learning of neural network using random signal. The learning object is learned by the value which is generated through the activation function (bipolar step function), which input is random signal, and then compare the performance index with the prior step to generate the reinforcement signal.

This paper describes the application of simulated annealing to a random signal-based learning. The simulated annealing is used to generate the reinforcement signal which is used in the random signal-based learning. Random signal-based learning is poor at hill-climbing, whereas the simulated annealing has an ability of probabilistic hill-climbing. Therefore, hybridizing a random signal-based learning with simulated annealing can produce better performance than before. The validity of the proposed algorithm is confirmed by applying it to two different examples. One is finding the minimum of the nonlinear function. And the other is the optimization of fuzzy control rules using inverted pendulum. The key of this example is to adjust both the width and the center of

Manuscript received February 3, 2015; Revised March 20, 2015; Accepted April 4, 2015. (ID No. JMIS-2015-0005)

Corresponding Author(*): Chang-Wook Han, 176 Eomgwangno, Busanjin-gu, Busan, Korea, Tel : +82-51-890-1665, cwhan@deu.ac.kr

¹Department of Electrical Engineering, Dong-Eui University, Busan, Korea, cwhan@deu.ac.kr

²Avionics Electronic Engineering, Kyungwoon University, Gumi, Korea, annaturu@ikw.ac.kr

membership functions so that the tuned rule-based fuzzy controller can generate the desired performance.

II. RANDOM SIGNAL-BASED LEARNING

The learning of neural networks is the process which its synapse strength is changed according to the prescribed rules to store any information in it. Hebb [5] presented the learning rules as follows: When two neurons A and B are interconnected with each other, if the neuron A is activated, the other neuron B is also activated. As a result, the synapse strength between A and B is represented as

$$m_{BA}^{new} = m_{BA}^{old} + \eta y_A y_B \quad (1)$$

where m is the synapse strength, η is a learning coefficient and y is the output of a system. By the way, synaptic equilibrium occurs in steady state when the synaptic strength m stops moving, that is,

$$\dot{m} = 0 \quad (2)$$

For intentional learning, we intentionally add Gaussian white noise process $n(t)$ to eq. (2). Then the synaptic strength m reaches stochastic equilibrium when only the random noise vector n changes m , that is, $\dot{m}(t) = n(t)$. Then, the zero means noise property of eq. (2) and the stochastic equilibrium condition $\dot{m}(t) = n(t)$ implies the average equilibrium condition

$$E[\dot{m}(t)] = E[n(t)] = 0 \quad \text{for all } t \quad (3)$$

reaches equilibrium.

The learning law takes the following form:

$$m(t+1) = m(t) + \eta r(t) f(n(t) - \theta) \quad (4)$$

where η : learning rate

f : activation function

n : discrete random process with values between 0 and 1

θ : bias (=0.5)

$r(t)$: reinforcement signal

and the reinforcement signal $r(t)$ is defined as follows:

$$r(t) = u(J(t) - J(t-1)) \quad (5)$$

$$\text{where, } u(x) = \begin{cases} 1, & \text{if } x \leq 0 \\ 0, & \text{otherwise} \end{cases}$$

In this learning law, synapses learn when only a performance index is decreased after the learning. In other words, $r(t)$ of eq. (4) will be set by 1. Otherwise, the learning will be rejected, that is, $r(t)$ of eq. (4) will be set by 0. The activation function f is a bipolar sigmoid function as following form:

$$f(x) = 2 / (1 + e^{-\lambda x}) - 1 \quad (6)$$

III. SIMULATED ANNEALING

The simulated annealing is a method that has attracted significant attention for large-scale optimization problems that have many local minima and make reaching the global minimum difficulty.

The idea of the simulated annealing comes from the physical annealing process done on metals and other substances. In metallurgical annealing, a metal body is heated to near its melting point and then slowly cooled back down to room temperature. This process will cause the global energy function of the metal to reach on the absolute minimum value eventually. If the temperature is dropped too quickly, the energy of the metallic lattice will be much higher than this minimum because of the existence of frozen lattice dislocations that would otherwise eventually disappear because of thermal agitation.

Analogous to this physical behavior, the simulated annealing allows a system to change its state to a higher energy state occasionally such that it has a chance to jump out of the local minima and seek the global minimum. The function to be minimized i.e., the performance index, is analogous to the energy of metal, and the control parameter, called temperature, is analogous to the temperature of metal. Downhill moves are always accepted, whereas uphill moves are accepted with the probability that is a function of temperature. The acceptance probability is usually given by the Boltzmann distribution,

$$p = \exp(-\Delta E / T) \quad (7)$$

where ΔE is the change in the performance index and T is the temperature. If ΔE is negative then the new solution is always accepted. If ΔE is positive then the new solution could be accepted with the probability in eq. (7). The possibility of accepting the uphill movement permits escaping from the local minima.

After the decision of acceptance of a new solution, the temperature is adjusted by cooling schedule, and the process is repeated until some convergence conditions are reached. Because of its speed, exponential cooling

schedule is used in this research. The temperature at any stage during the optimization may be expressed as follows:

$$T_k = \alpha^k T_0 \quad (8)$$

where T_0 is initial temperature that is large enough, α is cooling rate and k is time index.

IV. PROPOSED ALGORITHM

In this paper, we propose the application of simulated annealing to a random signal-based learning, that is, we use the simulated annealing to generate the reinforcement signal $r(t)$ in eq. (4). In eq. (7), ΔE is defined as follows:

$$\Delta E = J(t) - J(t-1) \quad (9)$$

where $J(t)$ is performance index.

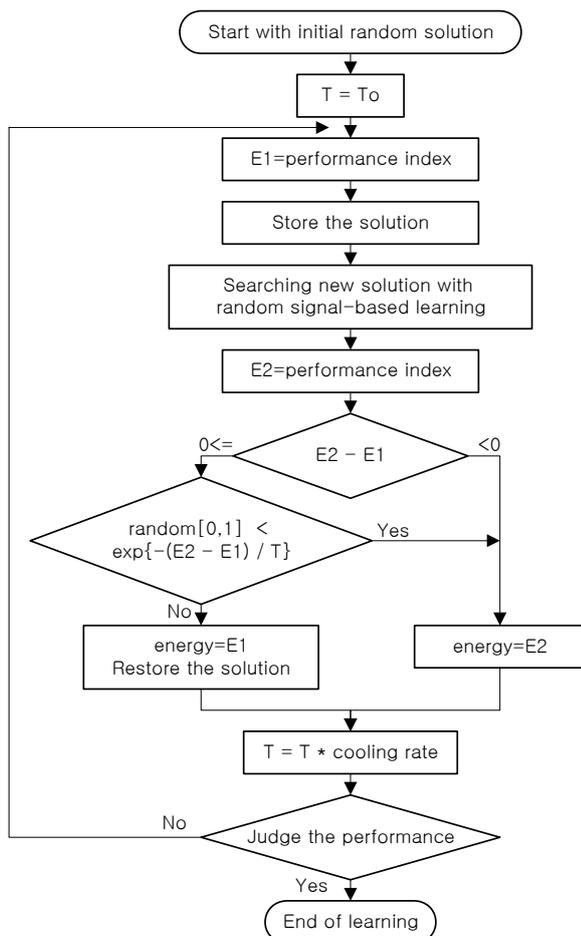


Fig. 1. Flow chart of the proposed algorithm.

In eq. (9), if ΔE is negative then the new solution is always accepted. If ΔE is positive then the new solution

could be accepted when only the acceptance probability p in eq.(7) is greater than the random number between 0 and 1.

Figure 1 shows the proposed algorithm. The proposed algorithm can be described as follows:

- Step 1. Start with initial random solution: Initial solution is chosen at random.
- Step 2. $T=To$: Put initial temperature To into temperature, T .
- Step 3. $E1$ =performance index: Put the performance index, which is evaluated using current solution, into $E1$.
- Step 4. Store the solution: Store the current solution in arbitrary memory location.
- Step 5. Searching new solution with random signal-based learning: The solution is learned by random signal-based learning.
- Step 6. $E2$ =performance index: Put the performance index, which is evaluated using new solution, into $E2$.
- Step 7. $E2-E1$: If $E2$ is less than $E1$ then put $E2$ into energy and goto Step 9. Else goto next Step.
- Step 8. $\text{random}[0,1] < \exp[-(E2-E1)/T]$: If the acceptance probability is greater than random number between 0 and 1 then put $E2$ into energy. Else put $E1$ into energy and restore the solution before learning in Step 4.
- Step 9. $T=T*\text{cooling rate}$: Multiply T by cooling rate.
- Step 10. Judge the performance: If the stopping criterion is satisfied then end algorithm. Else goto Step 3 and repeat until the stopping criterion is satisfied.

V. SIMULATION RESULTS

The validity of the proposed algorithm is confirmed by applying it to two different examples.

One is finding the minimum of the nonlinear function $F(x)$ in eq. (10)[6].

$$F(x) = (x + 0.9)(x + 0.7)(x + 0.2) \quad (10)$$

$$(x - 0.4)(x - 0.7)(x - 0.9) + 0.04$$

where $-1 < x < 1$.

Initial temperature T_0 is 200, α is 0.98, learning rate η is 0.01 and the performance index of this example is $F(x)$ itself because the purpose is to find x which minimize $F(x)$.

Figure 2 shows x and nonlinear function $F(x)$. $F(x)$ has two local minima as well as the global minimum in the specified region of x . We know by calculation that the true solution is about 0.07715.

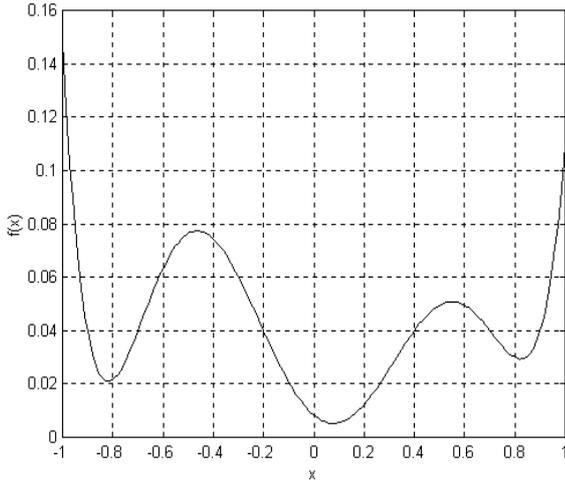
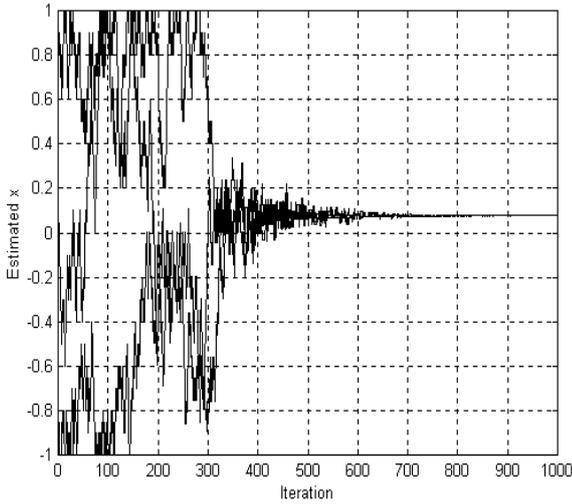

 Fig. 2. x and nonlinear function $F(x)$

 Fig. 3. Estimated x in each iteration

Figure 2 shows x and nonlinear function $F(x)$. $F(x)$ has two local minima as well as the global minimum in the specified region of x . We know by calculation that the true solution is about 0.07715.

Figure 3 shows the estimated x in each iteration for different initial conditions (-0.8, 0.1, 0.9) and eventually converge on the global minimum ($x=0.0771$).

Another example is control of the inverted pendulum using a fuzzy controller. A mathematical model of the inverted pendulum system can be derived as follows:

$$\begin{aligned} (M+m)\ddot{x} - ml(\sin\theta)\dot{\theta}^2 + ml(\cos\theta)\ddot{\theta} &= u \\ m\ddot{x}\cos\theta + ml\ddot{\theta} &= mg\sin\theta \end{aligned} \quad (11)$$

where mass of cart M is 3Kg, length of pole l is 0.6m and mass of pole m is 0.1Kg. Initial temperature T_0 is 0.05, α is 0.98, learning rate η is 0.01 and the

performance index of this example is like this:

$$J = \sum_{j=0}^k [e_j^2 + \dot{e}_j^2] \quad (12)$$

where e is error, \dot{e} is derivative of error and k is the number of input-output pair. The performance index is evaluated by control of the inverted pendulum in constant time.

The number of linguistic variable is 5 and we control only angle of the inverted pendulum. Antecedent parts are fixed and only consequent parts of the fuzzy control rules are considered to learn for our simplicity. That is, the key of this example is to adjust both the width and the center of consequent part membership functions so that the tuned rule-based fuzzy controller can generate the desired performance. The number of centers and widths are 25 respectively because the number of linguistic variable is 5.

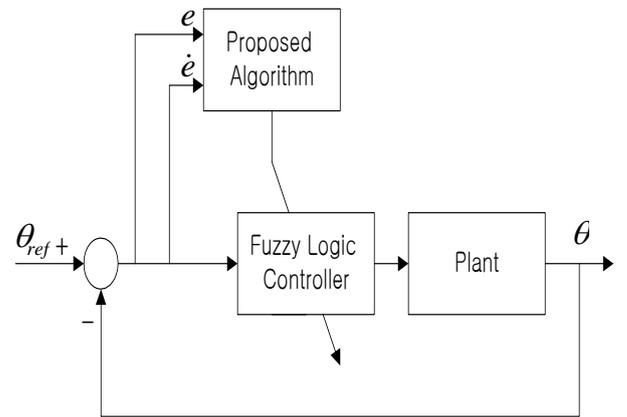


Fig. 4. The structure of self-learning fuzzy controller using proposed algorithm

To show the effectiveness of the proposed algorithms, the simulations of genetic algorithms and the random signal-based learning are achieved respectively. In genetic algorithms, population size is 10, maximum generation is 100, crossover probability is 0.6, mutation probability is 0.013 and all conditions are the same as the proposed algorithm. Also, the parameters and conditions of the random signal-based learning are the same as the proposed algorithm.

Figure 4 shows the structure of self-learning fuzzy controller using the proposed algorithm.

Table 1, table 2, and table 3 show final centers and widths learned by the proposed algorithm, the random signal-based learning and the genetic algorithms respectively.

Table 1. Final centers and widths trained by proposed algorithm.

$e \backslash \dot{e}$	NB	NM	ZE	PM	PB
NB	1.04(0.84)	0.86(1.02)	0.22(1.18)	0.24(1.28)	0.22(0.78)
NM	1.16(0.90)	0.46(1.14)	0.08(1.08)	-0.26(0.78)	-0.68(0.78)
ZE	0.98(0.96)	0.36(1.30)	0.00(1.00)	-0.46(0.76)	-1.20(0.84)
PM	0.54(1.20)	0.10(0.98)	-0.20(0.36)	-0.58(0.54)	-0.90 (1.14)
PB	-0.34(1.04)	-0.40(0.94)	-0.28(1.18)	-0.60(0.82)	-1.12(0.98)

Table 2. Final centers and widths learned by random signal-based learning

$e \backslash \dot{e}$	NB	NM	ZE	PM	PB
NB	0.96(1.18)	1.02(0.76)	0.32(0.96)	0.44(1.02)	0.32(1.18)
NM	0.98(1.14)	0.42(0.96)	0.54(1.00)	-0.06(1.08)	-0.42(1.08)
ZE	0.90(1.08)	0.34(1.12)	0.00(1.06)	-0.46(1.00)	-0.94(0.96)
PM	0.46(1.02)	0.08(1.04)	-0.38(0.96)	-0.46(0.92)	-0.92 (0.96)
PB	-0.10(1.02)	-0.52(0.84)	-0.42(0.86)	-0.98(0.88)	-0.86(1.02)

Table 3. Final centers and widths learned by genetic algorithms

$e \backslash \dot{e}$	NB	NM	ZE	PM	PB
NB	1.00(0.49)	0.89(0.29)	0.87(0.29)	0.44(1.65)	0.01(2.23)
NM	0.60(1.07)	0.71(0.87)	0.36(1.45)	0.01(3.00)	-0.92(0.68)
ZE	0.97(2.81)	0.33(2.42)	-0.01(1.45)	-0.41(0.49)	-0.65(1.07)
PM	0.31(1.65)	0.12(1.07)	-0.25(0.87)	-0.73(1.45)	-0.84 (2.81)
PB	-0.12(2.81)	-0.23(0.10)	-0.97(3.00)	-0.76(0.29)	-0.76(0.87)

Figure 5 shows the controlled output of the inverted pendulum obtained by using the fuzzy control rules after the learning, i.e., table 1, table2 and table 3. The control parameters are the same as during the learning. The proposed algorithm and the random signal-based learning are learned until the performance index is not changed. The genetic algorithm is learned during 100 generations. The variation around 10sec is disturbance. The result shows that the control performance of the proposed algorithm is greatly enhanced in convergence time and the disturbance than the genetic algorithm and the random signal-based learning. Especially, comparing the computational time of the proposed algorithm with that of the genetic algorithms, the proposed algorithm is shorter than the genetic algorithms.

Figure 6 shows the controlled output of the inverted pendulum obtained by using the fuzzy control rules after the learning. In this case, the control parameters are different from the learning, that is, $m=0.2\text{Kg}$, $l=1.2\text{m}$. And also, the variation around 10sec is disturbance. From the result, we can see that the proposed algorithm is more robust in parameter change than other both cases.

Figure 7 shows energy profile during the learning. In this figure, energy moves uphill and downhill according to the acceptance probability p in eq. (9) and finally reaches 0 energy state, i. e., to near the global minimum.

Fig. 8, Fig.9 and Fig. 10 show control input space of the proposed algorithm, the random signal-based learning and the genetic algorithm respectively. In Fig. 8, we expect that the settling time is very short because the gradient of surface around 0 varies smoothly. In Fig. 9, we expect that the chattering phenomenon occurs because it has sharp convex and concave. And in Fig. 10, we also expect that settling time is very long because it has the region that the gradient of surface does not vary.

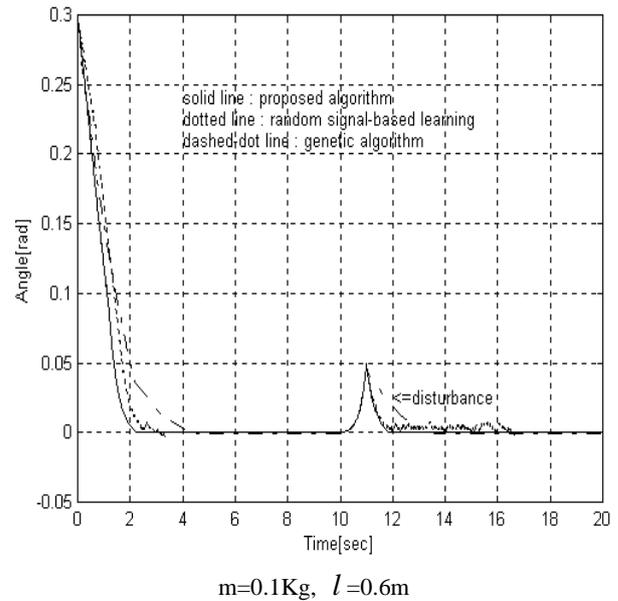


Fig. 5. Simulation results using the learned fuzzy rule table

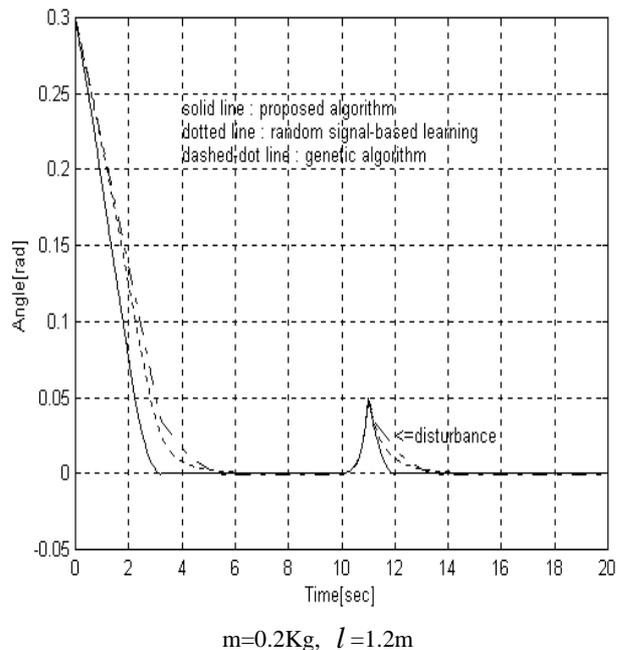


Fig. 6. Simulation results using the learned fuzzy rule table(change parameters)

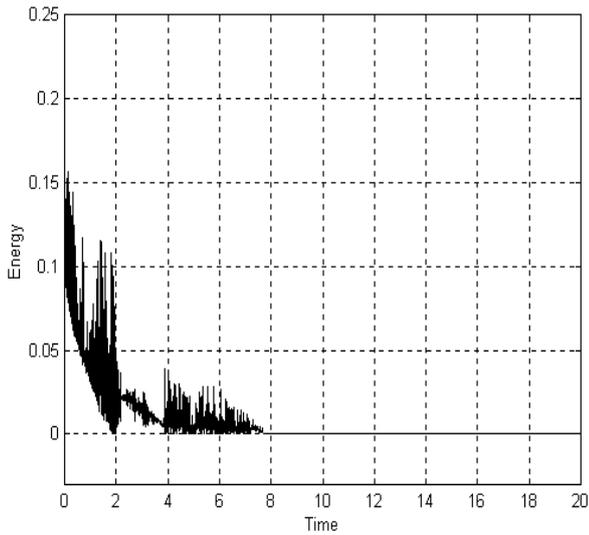


Fig. 7. Energy profile during the learning

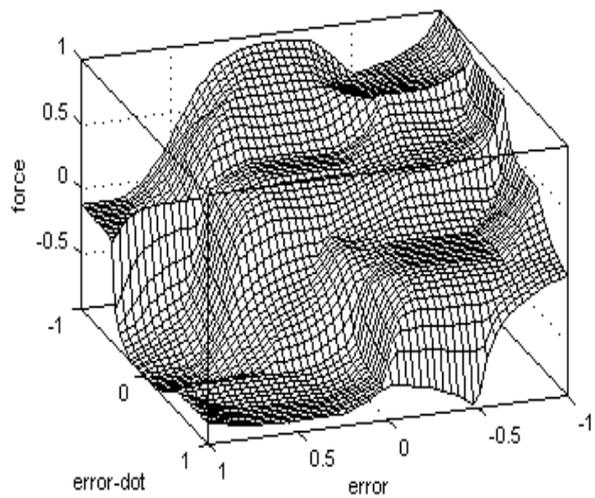


Fig. 10. Control input space of the genetic algorithms

VI. CONCLUSIONS

This paper was described the application of the simulated annealing to a random signal-based learning. The simulated annealing was used to generate the reinforcement signal which was used in the random signal-based learning. The validity of the proposed algorithm was confirmed by using two different examples. From the results, we expect that the proposed algorithm can be applied another optimization and control problems.

REFERENCES

- [1]T. J. Procyk and E. H. Mamdani, "A Linguistic Self-Organizing Process Controller," *Automatica*, vol. 15, no. 1, pp. 15-30, January 1979.
- [2]S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [3]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.* 21, pp. 1087-1092, 1953.
- [4]S. H. Jeong, C. W. Han, J. I. Park and S. H. Kwon, "A Study on Learning Scheme of Self-Learning Rule-Based Fuzzy Controller using Random Variable Sequence," *Proc. of the American Control Conference*, vol. 3, pp. 1862-1863, 1998.
- [5]D. O. Hebb, "*The Organization of Behavior*," John Wiley, New York, NY, 1949.
- [6]Il-kwon Jeong and Ju-jang Lee, "Adaptive simulated annealing genetic algorithm for control applications," *International Journal of Systems Science*, vol. 27, no. 2,

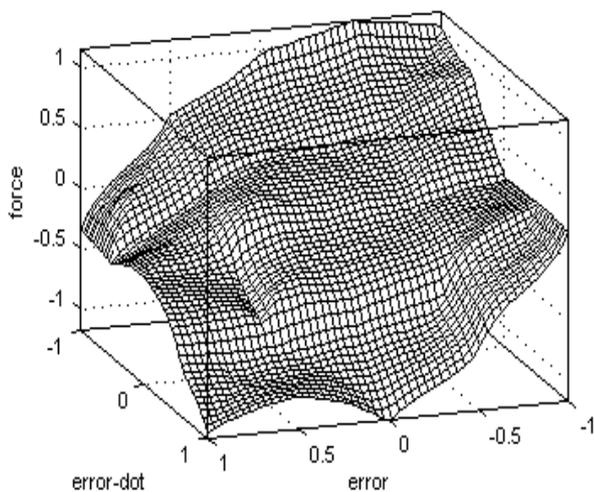


Fig. 8. Control input space of the proposed algorithm

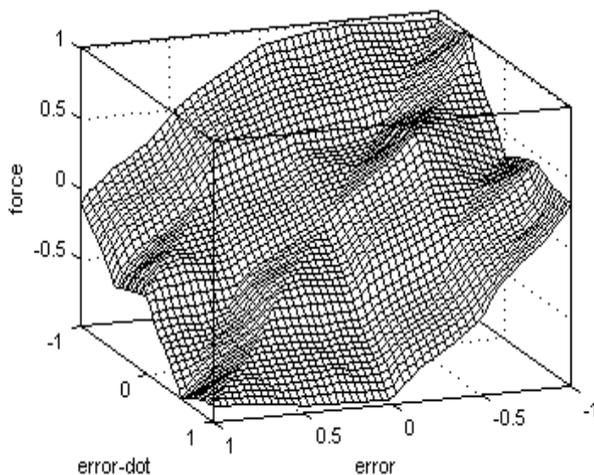


Fig. 9. Control input space of the random signal-based learning

pp. 241-253, February 1996.

- [7] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 129-139, May 1995.
- [8] Chin-Teng Lin and C. S. George Lee, "*Neural fuzzy systems*," Prentice-Hall, 1996.
- [9] D. E. Goldberg, "*Genetic Algorithms in Search, Optimization and Machine Learning*," Addison-Wesley, 1989.

Authors



Chang-Wook Han received the B.S., M.S., and Ph.D. degrees in Electronic Engineering from Yeungnam University in 1994, 1996, and 2002, respectively. He joined the faculty of the Department of Electrical Engineering, Dong-Eui University in 2008, where he has been an associate professor. His research interests include fuzzy-neural networks, intelligent

control, computational intelligence and its applications



Yeunghak Lee received his Ph.D degree from Yeungnam University, Korea, in 2003. He had one year experience at University of Cardiff as postdoc research fellow. Currently, he is a professor of department of avionic electronic engineering at Kyungwoon University. And he is contributing himself as a management editor for the Journal of

Multimedia and Information Systems. His research interests include pattern recognition, embedded system and computer vision.

This is blank Page