Optimal Design of a 2-Layer Fuzzy Controller using the Schema Co-Evolutionary Algorithm

Chang-Hyun Park and Kwee-Bo Sim*

Abstract: Nowadays, versatile robots are developed around the world. Novel algorithms are needed for controlling such robots. A 2-Layer fuzzy controller can deal with many inputs as well as many outputs, and its overall structure is much simpler than that of a general fuzzy controller. The main problem encountered in fuzzy control is the design of the fuzzy controller. In this paper, the fuzzy controller is designed by the schema co-evolutionary algorithm. This algorithm can quickly and easily find a global solution. Therefore, the schema co-evolutionary algorithm is used to design a 2-layer fuzzy controller in this study. We apply it to a mobile robot and verify the efficacy of the 2-layer fuzzy controller and the schema co-evolutionary algorithm through the experiments.

Keywords: Fuzzy control, genetic algorithms, robotics, 2-layer fuzzy controller, schema co-evolutionary algorithm.

1. INTRODUCTION

Fuzzy systems have been used in robot behavior control for many years and much attention has been devoted to their improvement. Fuzzy systems use a method of approximate reasoning, which allows them to make decisions based on vague and incomplete information in a manner which is similar to the way human beings operate. The reasons for use of the fuzzy system to control robots are as follows. Firstly, it provides robust control of the robot in a vague environment and represents clearly the structure of the controller. Secondly, it is reliable, because the degree of controllability is high and the sensitivity to noise or variations in the parameters is low. Finally, it is easy to apply to real situations [1]. Nowadays, robots have become extremely complex with numerous requirements. However, the total number of fuzzy rules and adjustable system parameters increases exponentially with the number of input variables in standard fuzzy reasoning processes [2]. This imposes a heavy burden on the system, from the view point of its control speed and cost. Because of this, classical fuzzy systems cannot perform well the fuzzy reasoning. Therefore, the 2-Layer fuzzy controller (2LFC) was proposed, in order to solve this problem [1]. The 2LFC can provide a good solution to this problem because it is capable of dealing simultaneously with numerous inputs as well as numerous outputs.

The main problem in fuzzy control involves the design of the fuzzy knowledge base. Various approaches to this problem have been proposed, including trial and error, the Takagi-Sugeno-Kang method, dynamic programming, gradient programming, genetic algorithms and reinforcement learning [3]. In particular, evolutionary computation has received considerable attention in recent years. In this paper, we use the schema co-evolutionary algorithm (SCEA) for the design of the fuzzy controller [4]. The SCEA works better than the SGA in the case of complex and difficult problems. In contrast to traditional single population-based evolutionary algorithms, the SCEA in which two populations constantly interact and cooperate with each other, is able to solve these problems more reliably. Also it has a much better chance than the SGA of finding global optima, because the parasite-population searches the schema space.

In this study, we design 2LFC using the SCEA, and we apply it to robot behavior control. Section 2 describes the process and structure of the 2LFC and compares them with those of multilevel fuzzy reasoning systems. In Section 3, we introduce the SCEA and describe the role of the SCEA in the design of the 2LFC. In Section 4, we verify the efficacy of the SCEA through the simulation for fuzzy modeling, design the 2LFC by the SCEA, and apply the designed 2LFC to a simple application, which is a

Manuscript received November 30, 2004; revised June 28, 2005; accepted August 1, 2005. Recommended by Editorial Board member Eun Tai Kim under the direction of Editor Jin Bae Park. This research was supported by the project of Developing SIC (Super Intelligent Chip) and its Applications under the program of Next generation technologies in 2000 of Ministry of Commerce, Industry and Energy.

Chang-Hyun Park and Kwee-Bo Sim are with the School of Electrical and Electronics Engineering, Chung-Ang University 221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea (e-mails: 3rr0r@wm.cau.ac.kr, kbsim@cau.ac.kr)
* Corresponding author.

mobile robot control. Finally, in Section 5, we present our concluding remarks.

2. 2-LAYER FUZZY CONTROLLER (2LFC)

2.1. Multilevel fuzzy reasoning systems and 2LFC

In the general fuzzy reasoning process, the total number of fuzzy rules increases exponentially as the number of input variables increases. Also, the total number of adjustable system parameters increases exponentially as the number of input variables increases. These two perspectives are said to constitute the dimensionality problem [5]. A general fuzzy controller has difficulty controlling a robot with numerous inputs and numerous behaviors. In order to solve these problems, multilevel fuzzy relational systems and multi-layer fuzzy logic controllers were proposed [2,5,6].

Chung and Duan classified hierarchical fuzzy systems into three multistage structures, namely, incremental, aggregated and cascaded [2]. Among them, the aggregated one is shown in Fig. 1. For it, inputs are only allowed to pass to the first reasoning stage consisting of a number of independent sub-stages. The outputs from the first stage form the inputs to the successive stage and such an arrangement can be extended for more stages. In fact, the rationale behind aggregated structure is similar to that of the classifier fusion design or mixtures of experts pursued rigorously in recent years.

Similarly, the 2LFC can provide a robust control and requires only a limited number of rules in order to handle many input variables [1]. Fig. 2 shows the structure of the 2LFC, in which there are four sub-controllers in the first layer and two combined controllers in the second layer. In the first layer, the various inputs are classified into four types and inputted to the sub-controllers. The sub-fuzzy controllers in the first layer perform fuzzy reasoning independently using the proper fuzzy reasoning system. Then, the second layer uses the outputs of the sub-controllers contained in the first layer as the inputs to the combined controllers. At this time, the combined controllers select the required outputs from the first layer as the final inputs with different types. The combined fuzzy controllers in the second layer perform combined fuzzy reasoning. These combined controllers are inclusively used for the purpose of producing the final outputs. Through this process, the 2LFC can perform fuzzy reasoning with numerous inputs and produce numerous outputs.

As shown in Fig. 2, 2LFC is similar to the aggregated structure of a multilevel fuzzy relational system. Also, it is similar to the shape of a neural network. However, 2LFC has three differences.

Firstly, 2LFC has a specialized structure which differs from the aggregated structure of a multilevel fuzzy system. The 2LFC limits the fuzzy reasoning to 2 levels, i.e. 2-layer reasoning. Fuzzy control has been successfully applied to many industrial plants that are mostly nonlinear systems. Generally speaking, in most industrial applications, the stability of control is not fully guaranteed and the reliability of a control system is considered to be more important than the stability [7]. In case of a multilevel fuzzy system, because consecutive nonlinear systems make a longer nonlinear system, it is hard to be aware of the structure of the whole system. So, it is difficult to rely on the system. On the other hand, by limiting the fuzzy reasoning to 2 levels, 2LFC can present the structure better than the multilevel system. Therefore, 2LFC can guarantee the reliability of a whole system.

Second, the 2LFC has more than two stages in the final level to which the final output is connected. Nowadays, the objectives to be controlled are very complex and varied. Therefore, the systems designed to meet these objectives have not only a number and variety of inputs, but also a number and variety of outputs. The multilevel fuzzy system can deal with numerous inputs but only a limited number of outputs. Because the 2LFC has more than two stages in the final level, it can deal with numerous inputs as well as

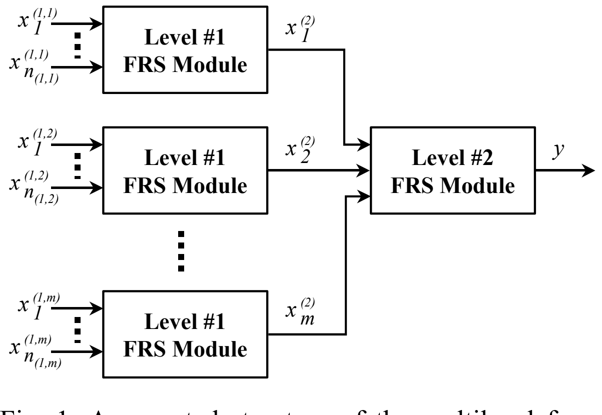


Fig. 1. Aggregated structure of the multilevel fuzzy relational systems.

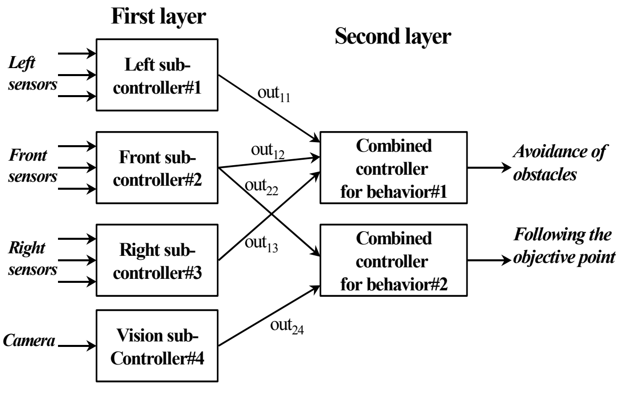


Fig. 2. Structure of the 2LFC; this example has 4 sub-controllers and 2 combined controllers.

numerous outputs. It is similar to the layered structure of an artificial neural network.

Finally, the 2LFC has a fuzzy controller in each stage module; this module is similar to a node in a neural network. Also, the 2LFC has no weights. In this case, the 2LFC bears no similarity to a neural network.

2.2. 2LFC for robot behavior control

Fig. 3 shows a simple application using the 2LFC. The robot that was implemented in this study has a number of ultrasonic sensors and a vision camera. Ultrasonic sensors are used to measure the distance to an impending obstacle and the vision camera is used to perceive the color of the objective. The use of range finder with good performance can make good result. By the way, for emphasizing the advantage of the fuzzy system, this paper considered to have harsh ultrasonic sensors.

The robot has 3 sensors in front, and 3 on each side. Total of 9 ultrasonic sensors are used for measurement of the distance. These sensors have the same type, but there are too many to control. Therefore, the sensors were divided into 3 input groups. In Fig 3, inputs of the sub-controller are 3 sensors on the left, 3 in front, and 3 on the right. The sensors of the vision camera differ from the ultrasonic sensors. Therefore, the input of the sub-controller #4 is a vision camera. out_{ij} represents the relation between the output from the j_{th} sub-controller and the input to the i_{th} combined controller. Fig 3 shows the robot used in the experiment. Data from 9 ultrasonic sensors are directly inputted to the sub-controller#1, #2, and #3. The image of vision camera is inputted to the sub-controller#4, which is the color detector, which is usually to the robot soccer game. Also, we applied 1-dimensional histogram processing to be more effective. Output of the combined controller is the angle of three wheels of robot. Since the robot has synchro-driving wheels such as Fig 3(b), the angle of wheels controls the robot.

The basic function of the robot is to avoid obstacles. In the system of Fig. 2, combined controller#1 is responsible for this behavior. One piece of information required for the avoidance of obstacles is the moving direction of the robot. This requires the outputs of the distance sub-controllers and the inputs of the combined controller#1 which are out_{11}, out_{12}, and out_{13}. Also, we define the moving of the robot to an objective point as an advanced behavior. So, combined controller#2 generates this behavior and therefore needs the position of the objective point. It needs the information which will enable the robot to attain the objective point, that is, the direction and the distance to the objective point, and the inputs of the combined controller#2 which are out_{22} and out_{24} . The front sub-controller generates the distance from the robot to the objective point and the vision sub-controller generates the direction toward the objective point.

Each module has a different fuzzy reasoning system. In this application, a zero-order Sugeno fuzzy model can be used as the fuzzy inference engine [8]. The reason is that the output from the controller is a constant term, which is the direction of the movement for the robot, and a defuzzier is therefore not needed. In this paper, the user indicates which combined controller is activated. Two outputs will be automatically selected or mixed in the future work. Also the number of modules and the connection of them are intuitively determined by a human designer according to the number and the type of inputs and desired outputs. It is easy to solve this problem, because the fuzzy rule in each module is considered to be more important than their connection or the number of modules. In this application, the vision sub-controller#4 has 1 input and the combined controller#2 has 2 inputs. The reason why the other controllers have always 3 inputs is that the number of the same type inputs is 9 and the number of the sub controllers for this is 3. As another example, if the number of inputs is 5, two sub controllers have 2 and 3 inputs respectively, and combined controller#1 has 2 inputs.

3. DESIGN OF 2LFC USING SCEA

3.1. Process of the SCEA

In order to find a good fuzzy controller, in this paper, we use the SCEA. The fundamental process of the SCEA was proposed in [4]. Like the other co-evolutionary algorithms, the SCEA has two different, but cooperatively working populations: a host-population and a parasite-population. The former is made up of candidates of the solution and works in approximately the same way as a conventional genetic algorithm. The latter is a set of schemata, which is used to find useful schemata called "Building Blocks" [9,10]. Fig. 3 shows an overview of the SCEA [4].

The SGA has four major steps for one generation, which are evaluation, selection, crossover, and mutation. The SCEA, however, has an additional step, parasitizing, which is performed before the selection process. Once all of the strings in the host-population are evaluated, some of them are randomly selected for each schema of the parasite-population and then parasitized by the corresponding schema. This allows the strings newly generated during the parasitizing process to be evaluated and their fitness improvement between the original string and the parasitized one to be measured. The parasitized string having the largest improvement value replaces the corresponding string in each schema. Using the amount of improvement,

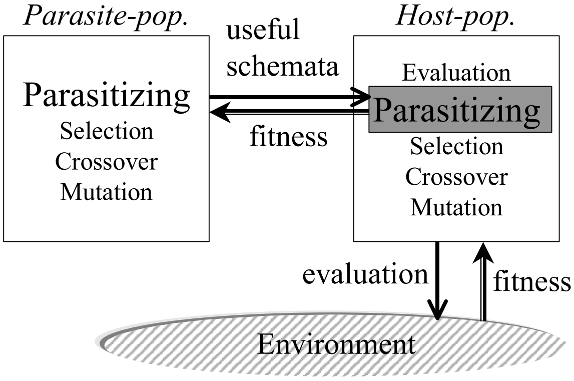


Fig. 3. Overview of the SCEA.

the fitness of each schema in the parasite-population can be evaluated. Therefore, the fitness of the each schema in the parasite-population provides a measure of the usefulness of the schema.

The SCEA applies the same process to the parasite-population after fitness assignment as the SGA does. The fitness of an individual in the parasite-population is calculated by the parasitizing process. Thus, treating the parasite-population takes four steps, i.e. parasitizing, selection, crossover and mutation. The parasitizing process was explained in detail in [4].

3.2. Parameters of 2LFC

In order to use the SCEA to design the 2LFC, we simplify the behavior of the robot. In Fig. 2, only the avoidance of obstacles is dealt with. Thus, the 2LFC consists of sub-controller#1, #2 and #3, and the combined controller for behavior#1. Following paragraph explains one of each fuzzy controller.

Generally, fuzzy reasoning methods are quite varied. They are classified into three types: the direct method, indirect method and hybrid method. We use the zero-order Sugeno fuzzy model of fuzzy reasoning [8]. The advantage of this model is that it includes a defuzzifier in the inference engine. Also, the main feature of this method is that the parameter of consequent is given by a constant term. The value of inference is obtained from following (1). In this expression, λ_i represents the fitness that is obtained from the i_{th} fuzzy rule, c_i represents the constant value of consequent in the i_{th} fuzzy rule and Z represents the final value of the output of the fuzzy inference.

$$Z = \frac{\sum_{i=0}^{n}(\lambda_i \times c_i)}{\sum_{i=0}^{n}\lambda_i} \tag{1}$$

Making use of the above expression, the final value is obtained directly from the output of the fuzzy inference engine.

Each fuzzy controller has 27 fuzzy rules in this application, because the number of inputs is 3 and the number of membership function in the antecedent is 3 in each fuzzy controller ($27 = 3^3$). In general, the number of fuzzy rules is determined according to the number of inputs (=the number of linguistic variables in the antecedent) and the number of linguistic term. For example, assume the fuzzy rule has a following form.

if X_1 is A_1 and X_2 is A_2 and X_3 is A_3, then Y is c

where $X = (X_1, X_2, X_3)$, which are the inputs of the controller, and Y are linguistic variables and (A_1, A_2, A_3) and c their respective linguistic values. Especially (A_1, A_2, A_3) are general membership functions and c is a fuzzy singleton in the zero-order Sugeno fuzzy model. The membership functions and the fuzzy singleton, *c*, which are used in this application are depicted in Fig. 4. Thus, the membership function A_i has three linguistic terms such as small, medium, and large. The number of fuzzy rules is determined by the inputs power of the linguistic terms ($\#Rules = \#Linguistic\ Values^{\#Inputs}$). Therefore, since each fuzzy controller in our application has 3 inputs and 3 linguistic terms(small, medium, and large), the number of fuzzy rules is 27.

The basic rule for the behavior of a mobile robot has the form:

if *Front Distance* is *Large* and *Left Distance* is *Small* and *Right Distance* is *Large* then *Steering* is *NL*....

A fuzzy controller with such membership functions, as in the case of the one shown in Fig. 4(a), needs eight points: *m1~m8*. If these points are determined, the three membership functions of a trapezoidal form can be determined, as shown in Fig. 4(a). Also, a fuzzy controller with such a constant, such as the one shown in Fig. 4(b), needs 7 points: *NL*, *NM*, *NS*, *ZE*, *PS*, *PM*, and *PL*. It should be mentioned that *ZE* represents a zero offset and *ZE* is fixed to the zero value. Therefore, if six points are determined, then the fuzzy singletons for the consequents of the fuzzy rules are determined.

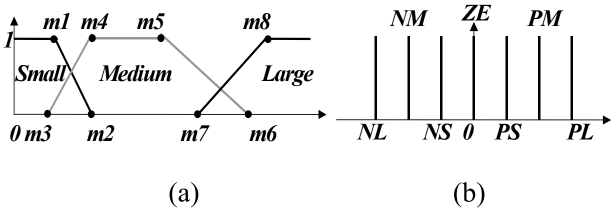


Fig. 4. Linguistic variables in the surface structure (a) membership functions to be calibrated (b) fuzzy singletons to be calibrated.

3.3. Applying the SCEA to the design of 2LFC

The individual in evolutionary algorithm represents not a fuzzy rule but the parameters of membership function and fuzzy singletons, as shown in Fig. 4. The surface structure of fuzzy rules is initially defined by a human designer. The deep structure, where the shape of membership functions and the value of fuzzy singletons are designed, is determined by a fuzzy modeling method that is the SCEA in our paper [3].

Therefore, the individual of the host-population is composed of 14 parameters which represent the shape of membership functions and the value of fuzzy singletons. The number of parameters in the membership functions of the antecedents is 8 and that of the consequents is 6. One parameter needs 6 bits in the form of a string. So, a fuzzy rule needs a total of 84 bits, where this refers to the length of the required string. Fig. 5 shows such a string.

If the above parameters are determined, the deep structure of the fuzzy rules can be determined. Finally, the entire fuzzy controller is designed automatically. With the string in Fig. 5 as the host-population, the SCEA is performed. The main controller in the 2LFC is the combined controller. The reason for this is that it performs a combinative inference. So, the SCEA is performed according to the following steps.

Step 1: In the 2LFC, the combined controller is more important than the sub-controllers. First of all, the SCEA designs the combined controller with 3 inputs.

Step 2: After adding a left sub-controller with 3 inputs, the SCEA designs this sub-controller.

Step 3: After adding a right sub-controller with 3 inputs, this sub-controller is designed.

Step 4: After adding a front sub-controller with 3 inputs, this sub-controller is designed by the SCEA.

The evaluation part of the SCEA process differs according to the problem being investigated. The SCEA evaluates the individuals using the result of the robot behavior. The robot moves according to an angle of the wheel (the output of fuzzy system, which is revised by the SCEA). The evaluation for the mobile robot control is described below. It is supposed that the mobile robot has a manipulator to perform a certain task. The minimum length of the manipulator, excluding the radius of the mobile robot, is 5cm. Thus, the mobile robot needs a minimum distance of 5cm between itself and an object. The robot has to position itself more than 5cm from the wall, in order to protect itself and its manipulator. This is reflected in the evaluation. Also, the robot must not collide with any obstacles; otherwise its fitness will be assigned a value of zero. If the robot successfully moves to the objective point, it is assigned high fitness. Among several such robots, the one that arrives the most rapidly at the objective point is assigned the highest fitness. The fitness function that reflects the above conditions has the following form (2).

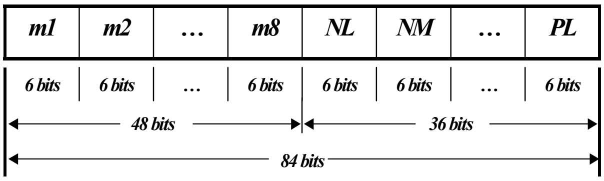


Fig. 5. The string used in the host-population.

$$f = \left(\frac{pos}{11}\right) \times \left(\frac{1 - near}{50}\right) \times \left(\frac{15}{time}\right) \tag{2}$$

The distance that the robot moves is divided into 11 sectors (*pos0~pos11*). *pos0* represents the starting point and *pos11* represents the objective point. If the robot collides with anything, it is assigned a fitness value of zero. The closer the robot moves to the wall, the lower the fitness. The variable, *near*, represents the closeness of the robot to the wall, with the number of correspondence to a distance of 5cm from the wall and the maximum number of closeness being 50. The time variable represents the elapsed time corresponding to the movement of the robot from *pos0* to *pos11*, and *15* is the minimum elapsed time in the case of this simulation. Consequently, the highest fitness value is given to the robot that moves the most rapidly to the objective point without experiencing any collisions and without coming 5cm of any obstacles.

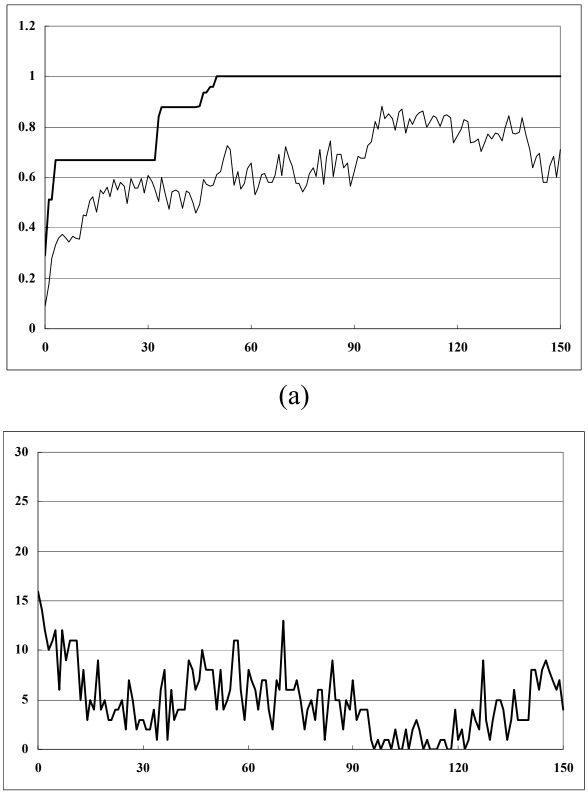
To summarize, the SCEA evolves the population that has many individuals. The change of the individuals represents the change of 14 parameters. As the result, the deep structure of fuzzy rules is changed and then the changed output Z of fuzzy controller is produced with the sensor inputs. The result of the robot behavior is reflected to the fitness function and the individual gets the fitness in evaluation process of the SCEA.

4. EXPERIMENTS AND RESULTS

4.1. Design of the 2LFC by the SCEA

Fig. 6(a) shows the fitness change of the host-population obtained with the SCEA and Fig. 6(b) shows the number of schema that the parasitizing process provides to the host-population. While the fitness of the host-population is low, many schemata are provided by the parasitizing process, as shown in Fig. 6. Although the fitness of the host-population is high, some schemata are provided.

Based on the results obtained from the SCEA, Fig. 7 shows some of the best solutions that are decoded from the individual with the best fitness. Membership functions with a cross shape are generated because we leave completely free our parameters by the SCEA. They are directly decoded to the fuzzy membership



(a)

(b)

Fig. 6. The result of the SCEA - fitness change.(a) The upper line represents the best fitness and the lower line represents the average fitness of the host-population. (b)The schema provided by the parasitizing process.

functions by general computer processing. The mobile robot moves using the 2LFC with these fuzzy rules. The simulation of the mobile robot's behavior is successfully performed using these fuzzy rules. The robot moves rapidly to the objective point without colliding with any obstacles and without coming closer than 5cm to any of the obstacles. The elapsed time from the starting point to the objective point was 5,156msec.

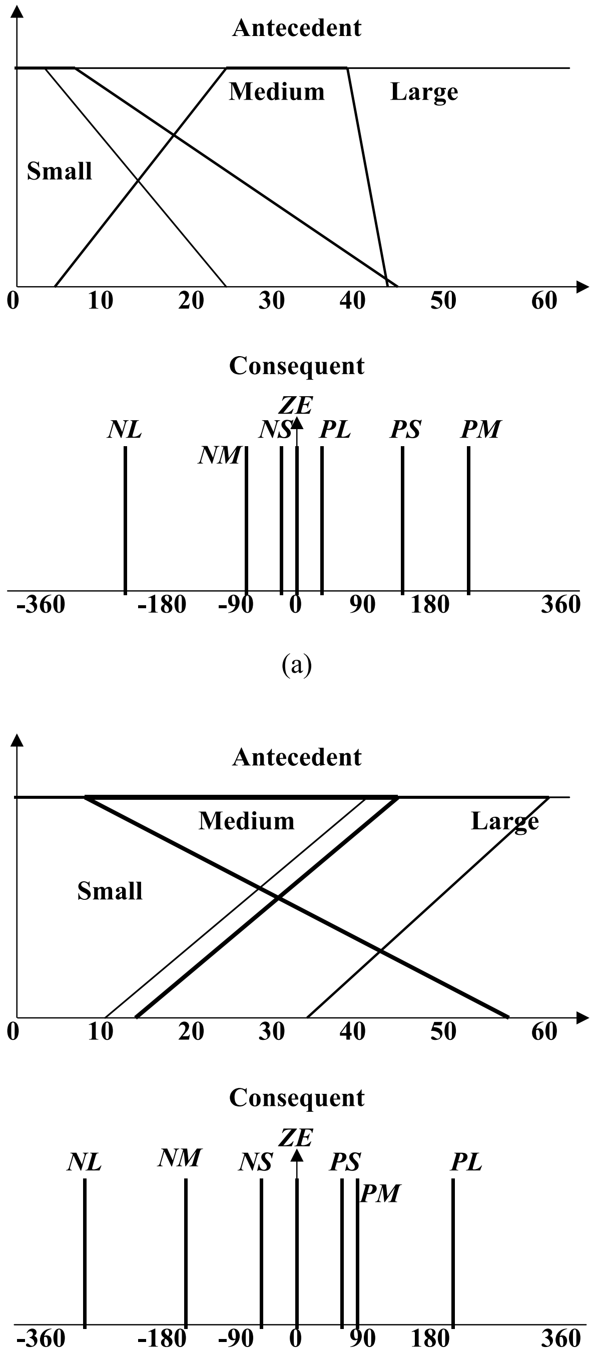
4.2. Experiments on the mobile robot

After applying the 2LFC, that was designed by means of the above simulation, to the mobile robot, we performed experiments to test the avoidance of obstacles. Fig. 8 shows the experiment involving the avoidance of obstacles. As shown in these pictures, the robot moved in such a way as to avoid the obstacles, while following the wall. This corresponds to the result from the optimized controller determined by the SCEA.

Also, the robot with the color detector successfully followed another robot. Fig. 9 shows pictures of the experiment in which one robot followed another robot. The robot in front moves using a 2LFC and the robot in the rear follows using a 2LFC and color detector.

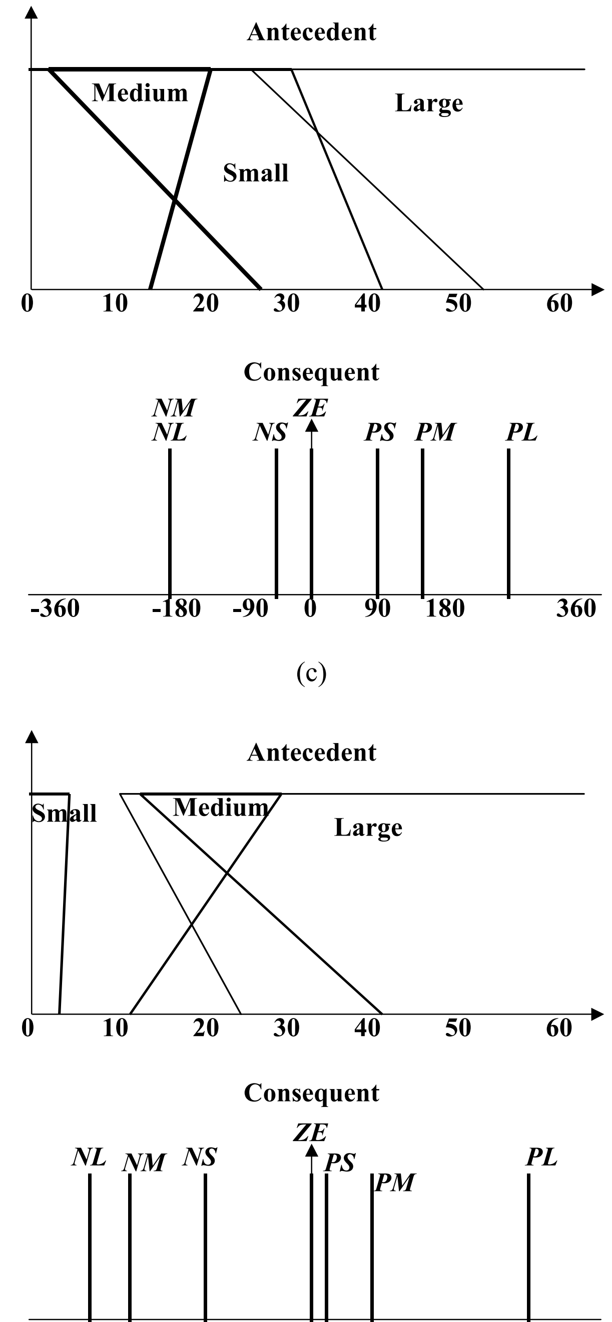
4.3. Experiments on the mobile robot

We compared with SGA to verify the efficacy of the SCEA. We use a string with 84-bits length. The population size of SGA is set for 100. The host and parasite-population sizes of the SCEA are set for 40 and 30 respectively and the same probability of crossover and mutation are used. The sampling size n is set for 2. Therefore the total evaluation number per each generation is 40+(30*2),that is the same number of the population size of SGA. The ratio of the host-population to the parasite-population and the sampling size are heuristically decided.



(a)

(b)



(c)

(d)

Fig. 7. The results for the SCEA - some of the best solutions:membership functions and fuzzy singletons (a) combined controller (b) left controller (c) right controller (d) front controller.

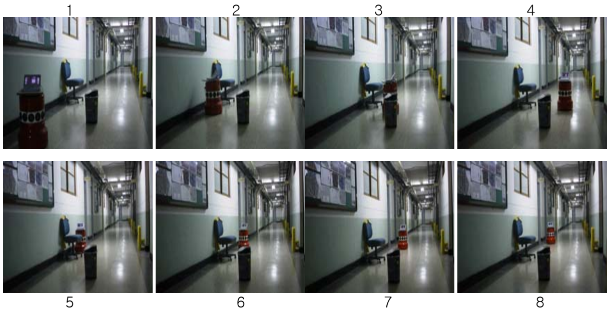


Fig. 8. The experiment involving the avoidance of obstacles.

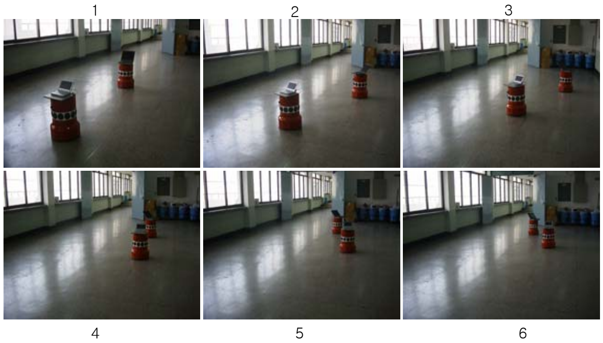


Fig. 9. The experiment on robot following.

Table 1. The number and average generation of the best results (BR) in 100 runs.

Pc	Pm	The number of BR		The avg. generation of BR	
		SGA	SCEA	SGA	SCEA
0.5	0.001	52	87	23.60	32.82
	0.01	89	99	26.88	24.93
	0.1	92	96	46.74	47.49
0.7	0.001	53	93	31.23	31.17
	0.01	91	100	19.82	23.31
	0.1	82	100	57.74	49.27

For the initial parasite-population we generate strings randomly by the same rate of 0, 1 and *. Also each character has the same probability to be mutated into each other. Since the behavior of genetic algorithm is stochastic, its performance usually varies from run to run. Therefore we replicated 100 runs on the design of the combined controller for each combination of the following parameter settings: p_c = 0.5, 0.7 and p_m = 0.001, 0.01, 0.1. Here, p_c and p_m represent the crossover probability and mutation probability of population in SGA and host-population in SCEA, respectively. p_c and p_m of the parasite-population have fixed 0.05 and 0.1 respectively. Also elite-preserving strategy is used. Each search was run to 150 generations in Table 1. The best result represents that the fitness is over 0.95.

These results show that the SCEA can find more than SGA does in the same condition. The algorithm with the high p_m found more. Especially, the SCEA with the 0.01 p_m found almost all the best result in

100 runs. There are two reasons why the average generation (23.60 or 19.82) of the best results in SGA is smaller than one (32.82 or 23.31) in the SCEA. One is that SGA could find best results less than the SCEA does. Another reason is that unless SGA can find initially the global optima, it falls into the local optima. That is, SGA is apt to fall into the local optima. So, whether it can find initially the global optima or not. But, the SCEA can more rapidly find the global optima and has the high probability for finding the global optima as knowing in the Table 1.

5. CONCLUSIONS

In this paper, we introduced the 2-layer fuzzy controller, whose purpose was to control the mobile robot, and we verified the efficacy of the 2LFC through simulations and experiments. Based on the results, it was concluded that the 2LFC can deal with various kinds of inputs (distance and vision in this study), as well as various kinds of output (avoidance of obstacles and robot following). In other words, the 2LFC can provide a robust control using only a small number of fuzzy rules. For the design of the 2LFC, the schema co-evolutionary algorithm was used. The SCEA works better than the SGA when faced with complex and difficult problems. We verified the efficacy of the SCEA, in terms of the design of the fuzzy controller, by comparing it with the SGA. Finally, we performed an experiment in which the robot with the 2LFC, which was designed by the SCEA, moved with various behaviors.

In this study, we demonstrated the performance of the 2LFC only through a simulation and some experiments. Therefore, the future work is the mathematical analysis of the 2LFC and the application to various sectors. After that, we hope that the 2LFC and the SCEA will be utilized in various, complex and difficult problems.

REFERENCES

[1] K. S. Byun, K. S. Heo, C. H. Park, D. W. Lee, and K. B. Sim, "Internet based tele-operation for an autonomous mobile robot with a 2-layer fuzzy controller," *Proc. of International Conference Mechatronics and Information Technology*, pp. 626-631, 2003.

[2] J. C. Duan and F. L. Chung, "Multilevel fuzzy relational systems: Structure and identification," *Soft Computing 6*, Springer-Verlag, pp. 71-86, 2002.

[3] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48-56, 1994.

[4] K.-B. Sim and D.-W. Lee, "Schema co-evolutionary algorithm (SCEA)," *IEICE Trans. Information and System*, vol. E87-D, no. 2, pp. 416-425, 2004.

[5] F. L. Chung and J. C. Duan, "On multistage fuzzy neural network modeling," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 2, pp. 125-142, 2000.

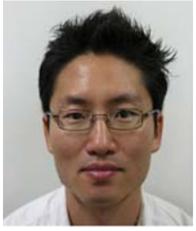
[6] K. Y. Tu, T. T. Lee, and W. J. Wang, "Design of a multi-layer fuzzy logic controller for multi-input multi-output systems," *Fuzzy Sets and Systems*, vol. 3, no. 2, pp. 199-214, 2000.

[7] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 2, pp. 201-224, 1999.

[8] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall International, 1997.

[9] Z. Michalewicz, *Genetic Algorithms + Data Structures Evolution Programs*, 3rd edition, Springer-Verlag, 1995.

[10] M. Mitchell, *An Introduction to Genetic Algorithm*, Bradford Book, MIT Press, 1996.



Chang-Hyun Park received the B.S. and M.S. degrees in Department of Electronic Engineering from Chung-Ang University, Seoul, Korea, in 2001 and 2003 respectively and He is currently pursuing the Ph.D. at the same university.

Kwee-Bo Sim received the B.S. and M.S. degrees in Department of Electronic Engineering from Chung-Ang University, Seoul, Korea, in 1984 and 1986 respectively, and Ph.D. degree in Department of Electronic Engineering from the University of Tokyo, Japan, in 1990. Since 1991, he has been a faculty member of the School of Electrical and Electronic Engineering at the Chung-Ang University, where he is currently a Professor. His research interests include Artificial Life, Neuro-Fuzzy and Soft Computing, Evolutionary Computation, Learning and Adaptation Algorithm, Autonomous Decentralized System, Intelligent Control and Robotic System, Artificial Immune System, Evolvable Hardware, and Artificial Brain, etc. He is a member of IEEE, SICE, RSJ, KITE, KIEE, ICASE, and KFIS.